Proc. Speech Tech'88 (New York, NY), pp. 328-331, April 1988

HIGH QUALITY 16 KB/S SPEECH CODING FOR NETWORK APPLICATIONS

Peter Kabal

Electrical Engineering
McGill University
Montreal, Quebec H3A 2A7

INRS-Télécommunications
Université du Québec
Verdun, Quebec H3E 1H6

Abstract

The integration of speech coders into a common carrier network raises important issues for coder design. These include speech quality, coding delay, coder complexity, and robustness to speaker variations and channel errors. This paper discusses new directions in speech coding which address these constraints to allow for toll quality coding of speech at rates down to 16 kb/s. Such a coder uses backward adaptation to limit coder delay and uses an embedded stochastically populated code tree to achieve high quality. These concepts point the way to a reduction by a factor of two in bit rate from the present 32 kb/s coding standard.

1. Introduction

With the widespread use of digital networks, digital coding for speech has come to play a bigger role in common carrier networks. New digital speech coding techniques can be used to further increase the bandwidth efficiency without sacrificing speech quality.

In network applications, coding delay is an important issue. If the digital speech coder is embedded in a network that retains conventional analog telephone interfaces, then echo from the imperfect coupling between the four-wire service and the two-wire subscriber loop will be present. The one-way delay consists of the coding delay, and the propagation delay. It is well known that tolerance to talker echo decreases as the echo delay increases [1]. With long delays, preventative measures such as echo cancellation or suppression need to be applied. In present day networks, these techniques are reserved for long delay toll connections. The cost of echo prevention can be on par with the cost of speech coding. A more desirable scenario is the use of speech coders with coding delays that are small enough that no special echo prevention measures need be invoked for connections that would not need them in the absence of speech coding. Thus, in this paper, the emphasis will be on keeping the coding delay below a few ms.

Simple speech coding techniques are limited in the quality of the reproduced speech as the bit rate is reduced. The CCITT has standardized a 32 kb/s coder for use in common carrier applications. This represents a 2:1 reduction over conventional 64 kb/s log-PCM coding. Although the 32 kb/s CCITT scheme achieves very low coding delays, the straightforward migration of techniques which are applicable at 32 kb/s to lower rates runs into problems with rapidly deteriorating speech quality at rates below 24 kb/s.

In this paper, techniques which will allow for good quality at 16 kb/s will be discussed. While many coders have been proposed for rates near 16 kb/s, a combination of high quality and low delay, features desirable for widespread applicability of the technology, has been heretofore lacking. This paper will discuss the techniques that have been brought together to achieve the 16 kb/s target. More details are available in [2]. Almost certainly any high quality, low delay coder at 16 kb/s will use a similar combination of techniques.

2. Coder Types

Speech coders are often categorized into two classes — waveform coders and analysis/synthesis coders. These two classes tend to have have non-overlapping bit rates, with analysis/synthesis coders having rates below 5 kb/s and waveform coders having rates above 10 kb/s. In fact there are some hybrid systems which have some characteristics of both classes and have intermediate rates (4–16 kb/s).

2.1 Analysis/Synthesis Coders

Linear predictive coding (LPC) is the best known technique in the class of analysis/synthesis coders. These coders model the incoming speech and then transmit the model parameters to the decoder where the speech is resynthesized. The model parameters for LPC allow for a compact representation of the speech. However, the synthetic speech quality and lack of robustness to speaker variation make these systems unacceptable for widespread use.

2.2 Waveform Coders

Waveform coders attempt to reproduce the input waveform at the decoder. At the low complexity end of the scale, logarithmically companded PCM can be used to code speech. Each sample is coded independently, normally with 8-bit precision. To reduce the coding rate below the 64 kb/s rate for log-PCM, more sophisticated waveform coders try to remove redundancy of the speech waveform. Such coders will be the focus of this paper.

3. Adaptive Predictors in Waveform Coders

Waveform redundancy is introduced by the filtering action of the vocal tract and due to the pitch periodicity of the vocal tract excitation. A typical waveform coder removes the predictable part of the speech signal by inverse filtering the speech by an estimate of the vocal tract filter. The residual signal is then coded and transmitted. At the decoder the reconstructed residual signal is used to excite a vocal tract filter to produce the output speech signal. The analysis filter and the synthesis filters are inverses of one another.

3.1 Formant Filter

. In a conventional view of voiced speech production, the vocal folds excite the vocal tract. From a filtering point of view the vocal tract has resonances (formants) which are importance for speech perception. From a signal processing viewpoint, the formant synthesis filter can be modelled as an all-pole filter of order 4–12 (order 8 will be used for the coder). The inverse filter removes the effects of the vocal tract resonances to produce a lower energy signal (the formant residual) which is more readily coded than the original input speech.

3.2 Pitch Filter

In many speech coders for medium rates, only a formant filter is used. The residual signal contains pitch spikes superimposed on a background random-like signal. This is consistent with the interpretation that the formant predictor removes the effect of vocal tract filtering to leave just the excitation signal, which for voiced speech consists of the glottal pulses. The problem is that the high peak-to-average ratio for the formant predicted residual is difficult to code at low bit rates.

Since in steady state voiced regions, the pitch pulses in the residual are similar in shape, an inverse pitch filter can be used to predict the shape of a pitch pulse based on previous pitch pulses. With a pitch predictor included in the system, the peak level of the pitch pulses is reduced. The pitch predictor is specified by a combination of the pitch period and the coefficient values themselves. Both the pitch estimate, and the coefficient values are made adaptive [3].

At the synthesis end, the coded excitation signal is input to the pitch synthesis filter which inserts pitch pulses. The resultant signal is then used to excite the conventional formant synthesis filter. Fig. 1 shows a schematic diagram of a coder using both a formant and a pitch filter, $F(z)$ and $P(z)$ respectively. In addition, it shows a noise shaping filter $N(z)$ which will be introduced later.

3.3 Forward versus Backward Adaptation

The filter parameters (formant and pitch filter coefficients) can be adapted either in a forward or backward fashion. The former involves transmission of the filter parameters as side information. The latter involves

implementing a local decoder at the transmitter and then adapting the filter parameters from the locally reconstructed speech. Each method has advantages and disadvantages and is appropriate under different conditions.

Forward adaptation can generate a filter based on the uncoded (clean) input speech, while backward adaptation necessarily adapts the filter based on the reconstructed signal. This reconstructed signal has embedded coding noise which can hamper the adaptation process, even when the perceived effect of the coding noise is minimal.

Forward adaptation necessarily involves the transmission of side information. The filter information is updated on a block by block basis, allowing for the transmission of the filter parameters only once per block. Clearly the update rate must be commensurate with the rate of change of the vocal tract. Typical values update range from 40–200 filter updates per second. In addition, the input speech is usually buffered and the "best" filter parameters can be determined for the whole block of samples. Indeed, this buffering is a major source of delay in many coding systems.

Backward adaptation uses a locally reconstructed speech signal both at the coder and decoder. No explicit side information need be transmitted. The drawback involves the fact that the filter is always updated from old data — the predictor is "stale". Some compensation is available since without a side information penalty, the predictor can be updated more often than with forward adaptation.

The analysis stage for determining the filter coefficients involves averaging of the input speech information, in the form of implicit or explicit estimates of the average correlation values. A time window is serves to select a region of speech which is then averaged for the analysis. There is a tradeoff between smoothness of the estimates and ability to track rapid changes.

3.3.1 Backward Adaptive Formant Filter

The requirement of low coding delay suggests the use of a backward adaptive formant filter. Furthermore, in order to keep a high prediction gain, a lattice based adaptation algorithm which is updated sample-by-sample is used [4]. In terms of its fidelity criterion, this filter is optimal at each time instant. The adaptation window is chosen to accentuate the most recent samples in order to lessen the deleterious effects of a stale predictor.

3.3.2 Backward Adaptive Pitch Filter

No precedent for a backward adaptive pitch filter exists in the literature. Indeed, conventional approaches lead to a poor prediction gain. Experiments show that the pitch filter is in some sense too sharply tuned to a particular window of samples. Applying this window to the future samples is not entirely appropriate, yet is required by the backward adaptation. This is the problem of a stale predictor again. One solution which increases the prediction gain is to "soften" the pitch filter, making the filter more robust to signal variations. This is accomplished by (conceptually) adding white noise to the signal for the purposes of determining the coefficients.

Pitch prediction in a backward adaptive context can be helpful in coding steady-state sounds accurately. However, the benefits in transition regions are less marked. The overall benefit is a subtle increase in speech quality. Indeed, with the computational complexity associated with pitch prediction, the cost/benefit ratio is not entirely favorable. However, this conclusion will probably change in the near future.

4. Residual Coding

The signal amplitude changes can be attributed to a combination of speaker level changes, but also to changes in the effectiveness of the predictors. When the vocal tract and/or the pitch is changing, the predictors are not as effective at removing redundancies and result in a residual with a higher amplitude. A basic adaptive scaling strategy can be quite useful in reducing the dynamic range of the signals that are actually quantized.

4.1 Adaptive Scaling

Conventional coding of the residual signal has employed a relatively simple adaptive quantizer, for sample-by-sample coding. The adaptive quantizer employs strategies to change the quantizer scaling. Desirable characteristics of the scaling are that it have a relatively fast attack and somewhat slower decay. A simple but effective approach is the Jayant quantizer adjustment scheme [5]. In this scheme, each output codeword has associated with it a multiplier. The codewords for the outer levels of the quantizer have multipliers which are larger than unity, while the inner levels (those nearest zero) have multipliers which are smaller than unity. In this way, the quantizer outputs are used as a cue to change the step sizes or equivalently the input scaling. The dynamics can be controlled by adjusting the relative sizes of the multipliers that are larger than unity to those which are smaller than unity. With an appropriate choice for the multipliers, this type of adaptation can mimic an exponential window average of the output speech energy.

4.2 Delayed Decision Coding

Even after adaptive scaling, the peak-to-average ratio of the signal to be coded can still be large. Entropy coding can be effective in dealing with signals with large peak-to-average ratios. In entropy coding, the relatively rare peak values are coded with code words that are longer than the short code words used for the low amplitude portion of the residual. The disadvantage of this approach is the buffering that is needed to interface the variable length codewords to the constant rate channel. The buffering itself may entail delays of 100 ms and more.

The alternative considered here is a delayed-decision coder, which can be viewed as a tree-structured coder. In this type of scheme, one imagines a coder in which all the possible quantizer outputs are arranged in a code tree which branches with increasing time. The nodes in the tree are labelled with the output values. Conventional sample-by-sample coding can also be put into this format. However, in that case the node values are very regular.

If delayed decisions are now allowed, overall improvements can be obtained. The improvement is due to the fact that a locally suboptimal decision may give a better result in the long run. For instance, if two samples further on in the signal, the signal amplitude increases dramatically, it might be advantageous to start the signal increase a little early so that at the time of the signal increase, the step size of the quantizer is appropriate to that level.

4.3 Stochastically Populated Tree Code

The code tree that has been described so far is a deterministically populated tree. Further improvement can be elicited by using a so-called stochastically populated code tree. One can imagine choosing a random signal value for each node in the tree. A reasonable approach would be to have the distribution of the values be the same as that measured for the signal to be coded. For instance a Laplacian (two-sided exponential) is a good fit to speech signals after prediction. Conceptually at least, both the coder and decoder can store identical copies of the code tree. Such a random tree allows for a more diverse set of sample paths through the tree. Since the value of a node only depends on its position in the tree, and the position in the tree depends only on past transmitted codewords, no explicit side information need be sent to allow the decoder to track the coder (at least in the absence of transmission errors).

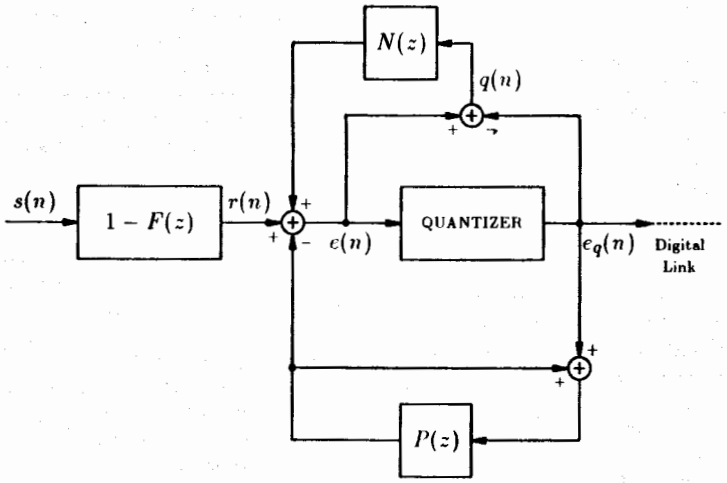
The exponential explosion in the storage for the code tree is unacceptable. However, since a fixed decision delay will be imposed, the number of nodes in contention at any given time is manageable. Furthermore, the randomness can be mimicked by having a dictionary of node values which is initially populated with random numbers. In operation, the dictionary is addressed by the last k transmitted samples, where k is chosen large enough to get the desired apparent randomness (total of N_D possible node values).

Even for moderate delays, the number of paths in contention can be extremely large. The (M, L) algorithm restricts the number of paths in contention at any given time to M [6][7]. A judicious choice of M can allow most of the benefits of delayed decision coding to be realized with a substantial decrease in complexity over a full search scheme.

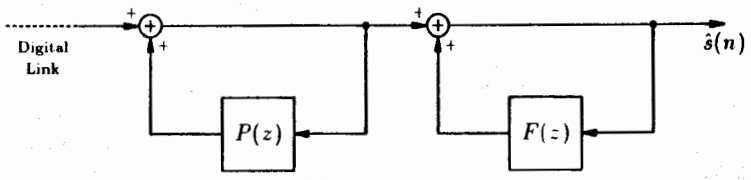
The improvement due to delayed decision coding for such a stochastic tree code is shown in Fig. 2 for a representative sentence. A similar figure for a deterministic code would show a larger SNR for small values of delay but saturating more quickly at a level below that for the stochastic tree. For the same decision delay of 8 samples, the speech quality of a stochastically populated tree is significantly better than that for a deterministic tree.

5. Noise Shaping

It is well known that high amplitude formant regions can mask lower amplitude signals at the same frequency. These effects can be utilized to an advantage by shaping the spectrum of the coding noise using filter $N(z)$ as shown in Fig. 1. Conventional ADPCM coders produce a coding noise spectrum that tends to be flat with frequency. The noise spectrum can be shaped to allow more coding noise in the formant regions where it is masked by the high amplitude formant signals [8]. This allows a corresponding decrease in the coding noise in those regions between formants. The overall perceived effect can be that of reduced distortion. The noise shaping must be chosen carefully, since too much noise in the formant regions leads to poorly defined and varying formant frequencies. Also it is clear that the



(a) Encoder



(b) Decoder

Fig. 1 Predictive coder structure

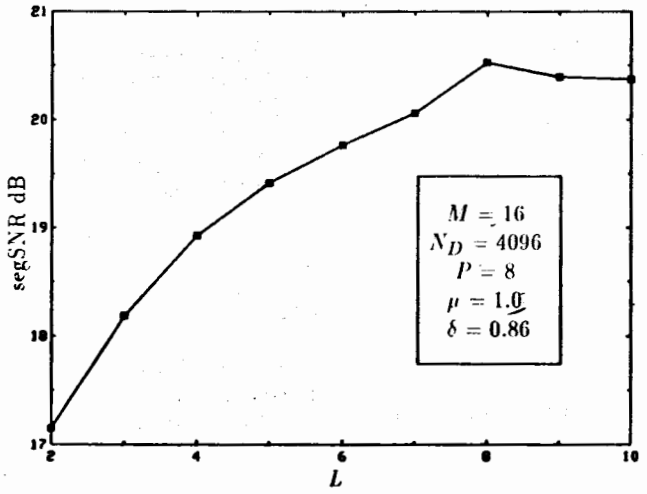


Fig. 2 Segmental SNR versus decision delay for a stochastic tree code

noise shaping must follow the changes in formant frequencies. Fortunately, this information is implicit in the formant filter and hence need not be transmitted explicitly. To accomplish noise shaping $N(z)$ can be chosen to be a bandwidth expanded verion of $F(z)$.

In addition, recent work in adaptive postfiltering has been applied to give subtle improvements in speech quality, again with no transmission of side information [9].

6. Effects of Transmission Errors

With the system proposed, no explicit transmission of side information takes place. Generation of filter parameters is assumed to proceed identically in the coder and decoder. However with transmission errors, the receiver no longer will be updated in the same manner as the transmitter, leading to the possibility of mistracking, possibly with catastrophic effects. The severity of the errors encountered will depend on the application of the coder. In conventional common carrier applications, backbone routes have error rates that are indeed very small. Then error events are rare and if degradations can be contained to the neighborhood of the error event, adequate performance can be attained. In other applications, errors can be more persistent. Examples are mobile radio applications, with a channel with widely variable conditions. One approach to mitigating the effects of transmission errors is to have all of the adaptation algorithms use finite memory. This will prevent error effects from propagating. In addition, automatic resetting of parameters to known values in silence or near silence can resynchronize a coder and decoder. Containment of the error events will generally lead to slightly inferior speech quality. The extent of the tradeoff depends on the final application.

7. Robustness to Input Speech

Coders often need to operate in hostile acoustic environments. As the bit rate is lowered, coders use more and more speech modelling into account and in some sense become tailored for speech rather than a more general input signal. However, the modelling used in the proposed coder is not so speech specific as to preclude other input signals. Specifically, the input could be speech which is impaired with background noise. The type of algorithm described is surprisingly robust to such input speech degradations.

8. Conclusions and Summary

The speech coder reached by the foregoing arguments is an extension of conventional backward adaptive ADPCM. The predictor consists of two components — a formant predictor and a pitch predictor. To help keep a high prediction gain, the formant predictor uses a lattice based adaptive algorithm that is updated at each sample time. The adaptation window is chosen to accentuate those samples close to the present time to lessen the deleterious effects of a stale predictor. The pitch predictor uses means to make it less tuned to a particular speech segment. In addition, the predictor loop contains a feedback filter which implements noise shaping to help reduce the perceptual effects of such noise.

The coder itself employs a st[illegible] popu[illegible]e code, with an 8 sample coding delay. To reduce the computational complexity, only a subset ($M = 16$ values) of the 4^8 possible paths are extended at any given time.

In a backward adaptive coder, it is the intertwined nature of the performances of the quantizer and predictors which controls the overall performance. A good coder is necessary in order to [illegible] the [illegible]tion n[illegible]se to a low level. The quantization noise itself affects the performance of the predictors. The techniques discussed has improved both aspects of the coder to bring the performance up to very high levels, not [illegible] component in isolation. The resulting speech has been judged to be perceptually of the same quality as 7 bit log-PCM, i.e. toll quality.

Acknowledgments:

Vasu Iyengar carried out the research summarized in this paper. This work was supported in part by a grant from the Natural Sciences and Research Council (Canada).

References

1. Technical Staff, Bell Telephone Laboratories. *Transmission Systems for Communications*, Fifth Edition, Bell Telephone Laboratories, 1982.
2. V. Iyengar and P. Kabal, "A low delay 16 kb/sec speech coder", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, paper 16.S6.6, New York, NY, April 1988.
3. R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding", to appear *IEEE Trans. Acoust., Speech, Signal Processing*.
4. J. I. Makhoul and L. K. Cosell, "Adaptive lattice analysis of speech", *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-29, June 1981.
5. N. S. Jayant, "Adaptive Quantization with a One Word Memory", *Bell Syst. Tech. J.*, Vol. 52, pp. 1119–1144, Sept. 1973.
6. J. B. Anderson, and J. B. Bodie, "Tree encoding of speech", *IEEE Transactions on Information Theory*, vol. IT-21, pp. 379–387, July 1975.
7. H. G. Fehn and P. Noll, "Multipath search coding of stationary signals with applications to speech", *IEEE Trans. on Commun.*, vol. COM-30, pp. 687–701, April 1982.
8. B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria", *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-27, pp. 247–254, June 1979.
9. J.-H. Chen and A. Gersho, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 2185–2188, Dallas, Texas, April 1987.