Code Excited Linear Prediction Coding of Speech at 4.8 kb/s

P. Kabal

INRS-Télécommunications
3 Place du Commerce
Ile des Soeurs, Qué.
CANADA H3E 1H6

July 1987

Rapport technique de l'INRS-Télécommunications no. 87-36

Code Excited Linear Prediction Coding of Speech at 4.8 kb/s

Abstract

This report describes a software implementation of an algorithm for digital coding of speech at low bit rates. The reconstruction of the speech signal is accomplished by exciting a cascade of a formant synthesis filter and a pitch synthesis filter with an excitation waveform. The excitation waveform is selected from a dictionary of waveforms using a frequency weighted mean-square error criterion. At transmission rates in the neighborhood of 5 kb/s, this scheme produces speech with better quality than any other known scheme.

Code Excited Linear Prediction Coding of Speech at 4.8 kb/s

1. Introduction

This report discusses a study of a new type of speech coding algorithm that in many ways fills a void in the capabilities of present generation speech coders. Previously studied waveform coding schemes tend to have a knee in the speech quality / bit rate curve such that for rates substantially below 10 kb/s, the quality of the reproduced speech falls off rapidly. At rates below 3 kb/s, vocoders which produce synthetic quality speech are the only alternative. The Code Excited Linear Prediction (CELP) coding scheme studied here tends to fill in the gap between waveform coders and vocoders at rates around 5 kb/s.

Rates around 5 kb/s are of practical importance. While bandwidth is less of an issue in new digital services based on fibre optic transmission, a large class of applications still are heavily bandwidth limited. Two examples of applications for 5kb/s coders are for secure voice transmission over existing analog facilities, and in wide-area voice communications.

The secure voice terminal application combines a low bit rate speech coder, a digital encryption device, and a data modem. Practical considerations limit the reliable full-duplex capabilities of the switched telephone network to rates around 5 kb/s. This provides the impetus for developing speech coders that produce good quality speech at this data rate. The second application involves the use of radio systems with relatively wide coverage. Examples are cellular systems for both mobile and fixed applications, and wide coverage satellite systems for mobile and fixed voice applications. In both cases, bandwidth is a scarse resource. Due to the use of a wide coverage medium, digital encryption is also desirable. For these applications, a range of bit rates is possible. However, at rates near 5 kb/s, digital systems again become competitive in bandwidth with analog systems, while maintaining the benefits of a digital implementation.

A study of coding at low bit rates us timely from the viewpoint of application driven needs, but also in light of advances in semiconductor technology in the form of single chip digital signal processors, which allow for the real-time implementation of complex algorithms. The complexity of the schemes being considered is at the high end of the scale for speech coders. In spite of this, the hardware complexity for the schemes studies can be measured in terms of one or at most a small number of the next generation of the DSP chips.

The work described in this report has been aimed at producing good quality speech at bit rates around 4800 b/s. At these rates efficient coding of the side information becomes important. Previous studies of CELP have concentrated on coding the waveform information and synthesizing

the speech with uncoded side information. The waveform information rate is in fact the smaller part of the data flow that has to be sustained by our coder. The work on CELP has been supported by both the Communications Research Center and the Natural Sciences and Engineering Research Council (NSERC). The work for the former agency has concentrated on side information coding, particularly for the synthesis filter parameters. The work for NSERC has been concentrated on efficient waveform coding and synthesis.

This report will provide an overview of the CELP coder. To this end we describe the CELP coder implementation and the design philosophy for the CELP algorithm that is used.

2. Background

Linear predictive coders (LPC) use an excitation waveform to drive a synthesis filter to produce reconstructed speech at low bit rates. In conventional LPC, this excitation waveform is either a pulse train for voiced speech or a noise waveform for unvoiced speech. The process of determining the form of the excitation involves a voiced/unvoiced decision as well as finding the pitch period for the pulse train. In practice, both of these operations are difficult to perform accurately. This rigid classification also ignores the possibility of mixed forms of excitation and more general excitation patterns. As a result, LPC is not robust to atypical speakers and suffers in noisy acoustic environments.

More recently, new techniques have been developed which allow for a more general excitation waveform. In multi-pulse coding, the excitation waveform is modelled as pulses which can take on arbitrary amplitudes and can be located in arbitrary positions. The excitation waveform is obtained by optimizing the positions and amplitudes of a fixed number of pulses to minimize an objective measure of performance [1][2]. This method can be thought of as an extension of traditional LPC methods with a more elaborate derivation of a general excitation pattern.

The success of multi-pulse coding is in large part due to the objective measure used to optimize the parameters of the excitation waveform. The objective measure is a frequency weighted mean-square error criterion. The frequency weighting is derived from the bandwidth expanded form of the synthesis filter used in conventional LPC. This frequency weighting reflects the properties of human auditory perception reasonably accurately, specifically the masking of coding noise in the formant regions of the speech spectrum.

Multi-pulse coding can produce good quality speech at rates between 4.8 kb/s and 16 kb/s. At the top end of this range, the quality is near toll quality, while at the bottom end, the quality is communication quality. Specifically the quality is rated to be equivalent to 6.5 bit log-companded PCM at 16 kb/s and 5 bit log-PCM at 4.8 kb/s [2]. The quality at 5 kb/s falls a little short of the quality needed for wide-spread dissemination of the technology. The total computational load (multiply/add rate) for multi-pulse coding is less than 1,000,000 operations per seconds, well within

the capabilities of single chip signal processors now available. Pitch loops have been added to multi-pulse coders to improve their performance [3]. However, the pitch analysis adds significantly to the computational load.

The data rates around 4.8 kb/s are of major interest in speech coding. This is the highest standard rate at which data can be transmitted using relatively simple hardware in a bandwidth compatible with the bandwidth available in an analog voice channel. Higher rates require channel equalization and even channel selection for reliable operation. As such, rates near 5 kb/s allow the coexistence of digitally encoded voice and analog voice using the same facilities. The digital voice service might even be part of an integrated voice/data service.

Atal and Schroeder [4] have described work on a coder using a small dictionary of randomly generated Gaussian excitation waveforms. The best waveform, in the sense of minimizing the frequency weighted mean-square error, is chosen to resynthesize the speech. A major difference between this scheme and conventional multi-pulse coding is the incorporation of a pitch synthesis filter. This additional component obviates the need for the excitation waveform to contain pitch pulse information. It is this type of scheme which is described here, with rates near 5 kb/s being the target. The coding scheme to be described is termed Code Excited Linear Prediction (CELP). The following section motivates the algorithm from an analysis/synthesis point of view.

3. Code Excited Linear Predictive Coding

In CELP, each trial waveform is synthesized by passing it through a two part cascade synthesis filter. The first part, termed the pitch synthesis filter, inserts pitch periodicities into the reconstructed speech. The second filter is the formant synthesis filter which introduces a frequency shaping related to the formant resonances produced by the human vocal tract. Both filters are all-pole structures, using an FIR filter in a feedback configuration. The synthesis stage of an CELP coder is shown in Fig. 1.

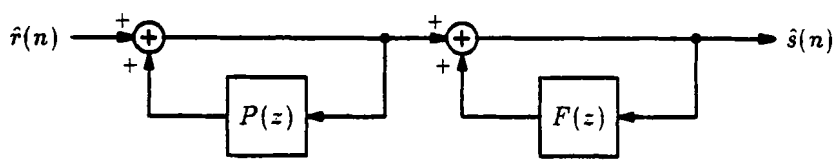


Fig. 1 Synthesis stage for an CELP coder

In CELP, the excitation waveform $\hat{r}(n)$ is chosen from a dictionary of waveforms. Conceptually, each waveform in the dictionary is passed through the synthesis filters to determine which waveform "best" matches the input speech. The optimality criterion is based on the same type of frequency

weighted mean-square error criterion used in multi-pulse coding [1]. The index of the "best" waveform used is transmitted to the decoder. In addition, both the formant and pitch filters are updated periodically. The parameters of these filters are sent to the decoder as side information to allow it to form the appropriate synthesis filters.

The CELP coder does not directly need an analysis stage. Ideally the synthesis filters would be optimized for each trial waveform. The formulation of an optimal (in a mean-square sense) formant synthesis filter leads to a highly non-linear set of equations which is not amenable to solution. However, the formant filter can be implemented as the inverse of a filter determined by an analysis step. While a filter determined in this manner is not strictly optimal, the method (described in the next section) is computationally feasible.

The pitch synthesis filter can also be determined from an analysis step. Indeed, early forms of the coder utilized this form of analysis. However, under certain constraints, the pitch synthesis filter can be chosen to optimize the reconstructed speech signal. Then a pitch analysis step is not needed in the final configuration, although a suboptimal pitch filter can be determined from an analysis step.

4. Analysis Stage

Conventionally the formant synthesis filter is an all-pole structure. This structure is consistent with a vocal tract model and has been shown to be able to produce good quality speech. The usual approach is to derive the corresponding synthesis filter by analyzing the input speech. The inverse filter (an all-zero prediction error filter) is determined by finding the filter which minimizes the mean-square prediction error (see Fig. 2). The formant predictor removes sample-to-sample correlations. An additional pitch predictor can be employed to remove correlations at longer time lags. This latter filter removes periodicities due to the pitch excited nature of speech. The use of formant and pitch prediction filters has been studied in more detail in [5]. While a pitch analysis is avoided in the final CELP configuration by optimizing the pitch synthesis filter directly, it is useful from a tutorial point of view to introduce both analysis filters at this time.

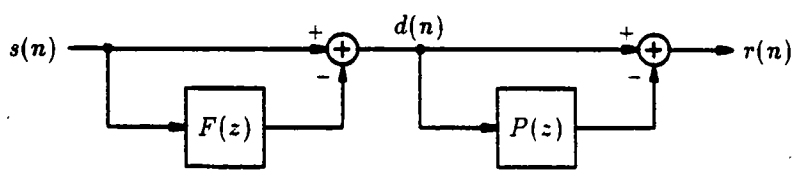


Fig. 2 Analysis stage

The residual $r(n)$ after both analysis filters is very noise-like in appearance. If the resulting residual signal is normalized, the resulting statistics are very much like those for a white Gaussian source. This ideal residual can be used as an input excitation waveform to the corresponding synthesis filters to reproduce the input speech exactly. The synthesis filters in this case are the inverses of the filters used in the analysis stage. Of course, the ideal residual is not available at the receiver. The actual excitation will be selected from the dictionary of waveforms.

4.1 Formant prediction

The first stage of forming the residual signal is the removal of sample-to-sample correlations. This stage of prediction will be termed *formant* prediction since the corresponding synthesis filter frequency response has an envelope which displays resonances corresponding to the formants. Several different methods to calculate the predictor coefficients may be used.

The predictor filter uses a linear combination of past input values to form a predicted value. The predictor coefficients are chosen to minimize the mean-square value of the prediction residual. The prediction residual signal is the difference between the input signal and the predicted value,

$$d(n) = s(n) - \sum_{k=1}^{N_f} a_k s(n-k) \ , \tag{1}$$

where the $\{ s(n) \}$ is the input signal and the predictor coefficients are the N_f values $\{ a_k \}$. The block of predictor output samples under consideration has N samples. This is the interval over which the coefficients are applied. The interval of data used to generate the coefficients may be different that this block of samples. We term this the analysis interval. The formant predictor residual energy for the analysis interval is

$$\overline{e_d^2} = \sum_{n=-n_1}^{n_2} d^2(n) \ . \tag{2}$$

It is this quantity which will be minimized by the appropriate choice of the predictor coefficients. The exact limits on the sum determine the type of analysis used.

A number of different analysis methods can be used to minimize the mean-square error. The covariance approach minimizes the above sum directly over a block of N samples and hence is optimal in the mean-square sense for each block of data. However, the resulting prediction error filter may be non-minimum phase, which means that the corresponding synthesis filter may be unstable. In the autocorrelation method, the input signal is windowed and the error is minimized over an infinite interval. With a time-limited window, only a finite number of terms in the error sum are actually non-zero. The autocorrelation approach leads to a set of equations which is more efficiently solved and which gives a minimum phase solution and hence a stable synthesis filter. Both of these methods perform similarly for sufficiently long block sizes. The differences can be attributed to differing initial

conditions. For the sequel, the covariance approach is outlined. We return to the autocorrelation method to interpret the error criterion in the frequency domain.

For a covariance analysis, the residual energy is minimized over a finite analysis frame. Assuming that the samples for the current block of samples have indices ranging from 0 to $N-1$, we choose $n_1 = 0$ and $n_2 = N-1$, making the analysis frame coincide with the current block of samples. In this case, the optimal (in the sense of minimizing the residual energy) coefficients are applied to generate the residual.

Covariance analysis leads to the following set of equations to be solved for the formant predictor coefficients,

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \cdots & \phi(1,N_f) \\ \phi(2,1) & \phi(2,2) & \cdots & \phi(2,N_f) \\ \vdots & & & \vdots \\ \phi(N_f,1) & \phi(N_f,2) & \cdots & \phi(N_f,N_f) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N_f} \end{bmatrix} = \begin{bmatrix} \phi(0,1) \\ \phi(0,2) \\ \vdots \\ \phi(0,N_f) \end{bmatrix} . \tag{3}$$

The correlation terms in this equation are defined by

$$\phi(i,j) = \sum_{n=0}^{N-1} s(n-i)s(n-j) . \tag{4}$$

Each correlation term has exactly N terms. Data values outside the interval $[0, N-1]$ are needed, specifically, values back to $s(-N_f)$ are used in the computation of the correlation terms.

The positive definite set of equations above can be solved using a Cholesky decomposition. Recursive relationships can be used to compute the correlation terms needed in the matrices. The correlation $\phi(i,j)$ can be computed as

$$\phi(i,j) = \phi(i-1,j-1) + s(N-i)s(N-j) - s(-i)s(-j) . \tag{5}$$

This relationship shows that only the top row of the matrix need be computed in full from the defining formula for the correlations. The elements proceeding down the diagonals of the matrix can be computed recursively from these values. In addition, since the matrix of values is symmetric, only about half of the elements need be calculated.

The covariance procedure just outlined does not guarantee stability of the synthesis filter. The all-pole synthesis filter may have poles outside the unit circle. If the ideal residual (output of the analysis stage) is used to excite the synthesis filters, pole/zero cancellation occurs and the instability will not be observed. However, any coding noise superimposed on the residual passes through only the synthesis filter and will excite any instabilities. This instability will usually be transient. To avoid stability problems, a modified covariance scheme is used to provide a non-optimal, but stable formant synthesis filter [6][7]. This scheme uses the same coefficient matrix and vector as does the normal covariance scheme.

The residual minimizing criterion can be shown to result in a synthesis filter which matches the formant envelope of the original speech. The fit is such that formant peaks are better matched than

the valleys. This fit is appropriate for human speech perception, as it has been shown that these features are important for good quality.

The residual matching property can be expressed in the frequency domain for an autocorrelation analysis. The expression for the error is given by [8]

$$\overline{e_d^2} = \frac{g^2}{2\pi} \int_{-\pi}^{\pi} \frac{|S(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega \ , \tag{6}$$

where $S(e^{j\omega})$ is the Fourier transform of windowed input sequence, $H(e^{j\omega})$ is the Fourier transform of the synthesis filter, and g is an appropriate constant that will be chosen to normalize the output value. The synthesis filter can be written in z-transform notation as

$$\begin{aligned} H(z) &= \frac{1}{1 - F(z)} \\ &= \frac{1}{1 - \sum_{k=1}^{N_f} a_k z^{-k}} \ , \end{aligned} \tag{7}$$

The act of minimizing the prediction error also minimizes the value of the frequency domain integral in the above equation. This integral can be interpreted as measuring the fit between $|H(e^{j\omega})|$ and $|S(e^{j\omega})|$. The contribution will be largest and the fit will be enhanced in those regions of the spectrum in which $|S(e^{j\omega})|$ is large, i.e. at the formant peaks.

4.2 Pitch filtering

The residual signal from a formant analysis filter still shows pitch periodicity. The purpose of the second stage in the analysis procedure is to remove pitch period redundancy.

Consider a predictor with a group of coefficients corresponding to lags $M, M+1, \ldots, M+N_p-1$. The resulting residual term is

$$r(k) = d(k) - \sum_{k=1}^{N_p} b_k d(n-k-M+1) \ . \tag{8}$$

where the $\{ b_k \}$ are the pitch predictor coefficients. The input signal to the pitch predictor is the output of the formant predictor, $\{ d(n) \}$.

The best pitch lag value can be determined in a number of different ways. In the present analysis configuration, the input signal to the pitch predictor has had near sample correlations removed by the formant prediction error filter. Then a nearly optimal strategy to choose the pitch lag is to find the peak of the sliding window correlation sum [5],

$$\tau(M) = \sum_{m=M}^{M+N_p-1} \frac{\phi(0,m)}{\phi(m,m)} \ , \tag{9}$$

where in this case $\phi(i,j)$ is the correlation function for $d(n)$.

A pitch predictor structure employing more than one coefficient is advantageous if the pitch period is not an integral number of samples long. Coefficients with lags on either side of the optimal position for a single coefficient serve to "interpolate" the pitch period correlation. The convention of using only adjacent lags makes it unnecessary to code and transmit the positions of the lags individually.

The actual procedure used is now described in more detail. The pitch period is determined by searching for the peak in $\tau(M)$. The search range is limited to an interval constrained to lie above the range of lags used by the formant predictor and below a value corresponding to the maximum pitch period that would normally be encountered in speech.

The actual determination of the coefficient values proceeds by solving a set of equations analogous to those used for the formant predictor. Given, a set of N_p lags $\{D_1, D_2, \ldots, D_{N_p}\}$, the equations to be solved to minimize the residual energy with coefficients corresponding to these lags are

$$\begin{bmatrix} \phi(D_1,D_1) & \phi(D_1,D_2) & \cdots & \phi(D_1,D_{N_p}) \\ \phi(D_2,D_1) & \phi(D_2,D_2) & \cdots & \phi(D_2,D_{N_p}) \\ \vdots & & & \vdots \\ \phi(D_{N_p},D_1) & \phi(D_{N_p},D_2) & \cdots & \phi(D_{N_p},D_{N_p}) \end{bmatrix} \begin{bmatrix} b_{D_1} \\ b_{D_2} \\ \vdots \\ b_{D_{N_p}} \end{bmatrix} = \begin{bmatrix} \phi(0,D_1) \\ \phi(0,D_2) \\ \vdots \\ \phi(0,D_{N_p}) \end{bmatrix} . \tag{10}$$

For pitch prediction as described, the lags are adjacent,

$$D_i = M + i - 1 . \tag{11}$$

As in the case of formant predictor formulation, the pitch synthesis filter may be unstable. Stabilization of the filter can be easily implemented following the strategy outlined in [9].

5. Synthesis Stage

The residual produced by the analysis stage can be used to excite a synthesis stage to reconstruct the original signal. Rather than transmitting the residual, an alternate excitation waveform is selected as described in the next section. At the decoder, the excitation waveform drives a pitch synthesis filter in cascade with a formant synthesis filter. The function of the pitch filter is to reinsert periodic components during voiced speech, while the formant filter shapes the overall spectrum to reproduce the formant resonances. The excitation waveform itself fills in missing details. The selection of a waveform from the dictionary can be considered to be a vector quantization problem. The difference between the CELP case and conventional vector quantization is that the error criterion is the time-varying frequency-weighted error.

A new waveform is selected for each block of samples. In addition, the pitch and formant synthesis filters are updated at intervals. For convenience, we will refer to a frame of samples. Each

frame of samples will be subdivided into subframes. It turns out that the formant filter is updated once per frame, while the gain, pitch filter parameters and waveform selection are updated at the subframe level.

The formant synthesis filter coefficients at the coder are updated at the beginning of each frame. This means, of course, that the filters are time varying. For the synthesis filters to reproduce the correct output signal with an ideal residual as excitation requires that the synthesis filters be updated in synchronism with the analysis filters. The implementational form (e.g. direct form or lattice) of the filters must also be matched. For example, if the analysis filters are implemented in direct form, then a direct form synthesis filter is required to keep the states of the filters in step at frame boundaries when the coefficients change. We choose a direct form implementation.

The information required at the synthesis stage consists of information about the excitation waveform and information about the synthesis filters. The excitation waveform is specified in terms of the index of its dictionary entry and a gain factor. The dictionary elements are stored in normalized form and the gain factor is necessary to denormalize these entries.

The pitch synthesis filter is specified in terms of the pitch lag and the pitch coefficient(s). The formant synthesis filter is specified in terms of the formant filter coefficients.

6. Residual Modelling

After formant and pitch prediction, the ideal residual signal is very noiselike. Rather than transmitting the residual, the coder transmits the index to a waveform stored in a dictionary of waveforms. The coder determines the index as that of the stored waveform which "best" matches the residual signal. This approach has been termed analysis-by-synthesis. This match is performed on a block of input residual samples. In this way, the residual may be considered to be transmitted using a vector quantization strategy.

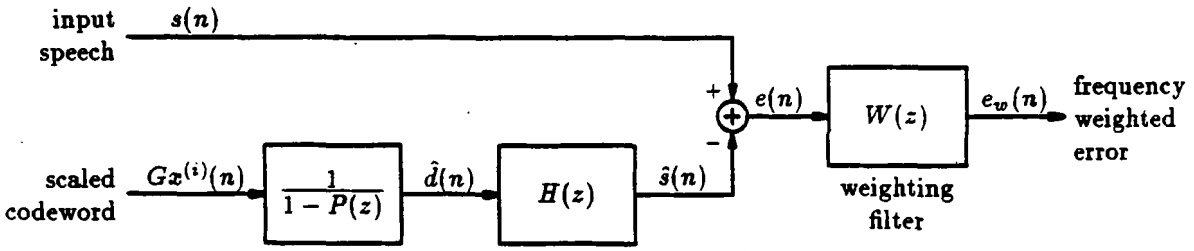
6.1 Frequency weighted error criterion

At low bit rates, it is important to use psycho-acoustical phenomenon as part of the optimality criterion used to match the residual against the stored waveforms. A frequency weighting that has been successfully used in multi-pulse coding is one based on the formant structure of the speech signal. Let the response of the formant synthesis filter be $H(z)$. Fig. 3 shows the model used for the residual analysis part of the system. The synthesis filter used for reconstruction is $H(z)$. In measuring the error, a spectral weighting is used as suggested by Atal and Remde [1]. The spectral weighting filter is introduced to take advantage of the properties of human auditory perception. Since more noise can be tolerated in the formant regions than in the valleys between formants, a

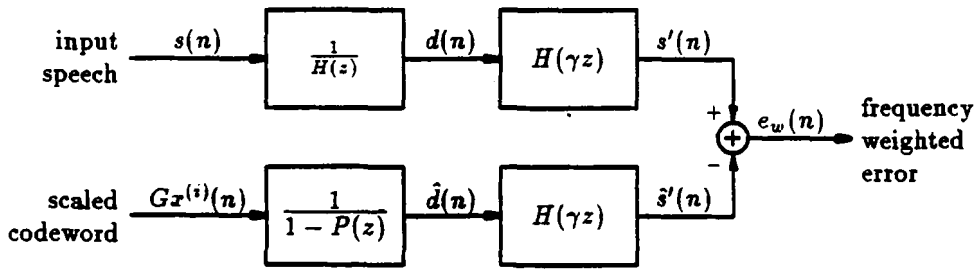
weighting which deemphasizes the formant regions is chosen. The weighting filter is

$$W(z) = \frac{H(\gamma z)}{H(z)} \quad . \tag{12}$$

The $H(\gamma z)$ filter represents a bandwidth expanded version of the original filter $H(z)$. The factor gamma is typically in the range 1/0.75 to 1/0.85. The ratio of these terms has the desired property of deemphasizing the high amplitude formant regions.



(a) Synthesis model



(b) Alternate synthesis model

Fig. 3 Frequency weighted error calculation

An equivalent model for the purposes of pulse determination is shown in the second part of Fig. 3. The weighting filter has been absorbed into the formant synthesis filter to produce a modified formant synthesis filter, $H'(z) = H(\gamma z)$. The bandwidth expanded filter is also an all-pole filter with new coefficients $a'_k = a_k \gamma^{-k}$. The causal impulse response of this filter will be denoted $\{ h'(k) \}$. The pitch reconstituted residual passing through this filter produces the "desired" signal,

$$s'(n) = \sum_{k=-\infty}^{\infty} d(k) h'(n-k) \; , \tag{13}$$

where $\{ d(n) \}$ is the pitch reconstituted residual. In practice, the filter response is held constant for the duration of the analysis frame. The desired signal is generated as in the conventional synthesis procedure for a predictive coder with appropriate care being taken at frame boundaries when the filter responses change. It should be noted that the modified synthesis filter is an artifice used only to determine an appropriate excitation waveform. The synthesis filter at the decoder remains $H(z)$.

6.2 Selecting the synthesis parameters

Each trial waveform is passed through the synthesis filters to produce a trial weighted reconstructed signal. The weighted squared error is

$$\varepsilon = \sum_{n=0}^{N-1} [s''(n) - \hat{s}'(n)] \,, \tag{14}$$

The value of the squared error is minimized over the choice of the synthesis parameters. The parameters in question are: waveform (waveform index and gain factor) and pitch filter (pitch lag and filter coefficients). The framework for deriving the synthesis parameters is described in a companion report [10]. The results are briefly summarized here.

An optimum scheme searches over all combinations of the pair $\{M, i\}$ representing the pitch lag and waveform index. For each such pair, the gain factor G and the coefficients of the pitch filter are chosen to minimize the squared error expression. The solution for the optimum gain factor and coefficients is linear in these parameters if the pitch lag is at least as large as the subframe size. The computation of these parameters in this way is computationally burdensome, even for a small number of waveforms. Instead a sequential approach is suggested in which the pitch filter is optimized for a zero waveform input. Then given the pitch filter parameters, a search is conducted over waveform indices (optimizing G for each).

The requirement that the pitch lag be at least as large as the subframe size can be bypassed in some special cases. In [10], the case for a single pitch coefficient is considered. It is shown that the optimum value can be found from a cubic equation. However, the preferred solution method involves substituting the quantized values of the pitch coefficient in the expression for the resulting error. The value which gives the smallest error is chosen. The computation associated with this approach is tolerable.

7. Bit Allocation

The target bit rate is 5000 bits/sec. Consider a sampling rate of 8000 samples/sec. The original system proposed by Atal and Schroeder used a dictionary of 1024 waveforms, each of duration 40 samples. This requires a transmission rate of 2000 b/s. Their original system used 16 formant predictors updated every 160 samples (20 ms) and 3 pitch predictors updated every 40 samples (5 ms). Conventional coding of such a large number of coefficients at such a fast update rate would lead to a side information rate of over 7000 b/s.

Our early experimentation used a similar configuration (with unquantized side information). Excellent reproduced speech results. By optimizing the pitch synthesis filter to match the excitation waveform chosen, the number of waveforms can be significantly reduced with little or no loss in

quality. Specifically, a dictionary of 32 waveforms with an optimized pitch filter performs essentially the same as the original system with 1024 waveforms. This reduces the bit rate for waveform coding to 1000 b/s. The bit allocations used are summarized in Table 1.

Parameter	Transmission Rate		
	bits	update	b/s
pitch filter	10	40	2000
formant filter	24	240	800
waveform index	5	40	1000
gain factor	5	40	1000
		total	4800

Table 1 Bit allocations for a 4800 b/s CELP coder

As described in a companion report, much effort was expended to efficiently code the formant synthesis parameters [11]. Vector quantization was considered to be to computationally burdensome for the present application. Our quantization scheme is based on a line spectral frequency (LSF) parameterization of the formant synthesis filter. We can achieve better than a 2:1 bit rate reduction over conventional reflection coefficients quantization, while maintaining a better spectral match after quantization. This LSF coding uses 21 bits to code 10 formant coefficients. The coefficients are computed for frames of 120 samples (15 ms). The coefficients of every second frame are coded. The coding uses a combination of intra- and inter-frame coding. The alternate frames are interpolated from the other frames. A 3-bit code specifies which interpolation value is to be used. The scheme then uses a total of 24 bits every 240 samples. This is a bit rate of 800 b/s. We note that vector quantization of the formant filter parameters seems to possible at 10 bits/10 coefficients. We believe our quantization would produce better results. Crosmer and Barnwell describe a similar formant filter coding scheme based on line spectral frequencies [12]. The main conceptual difference is our use of an adaptive predictor for the frame-to-frame coding of the LSF's, and our use of interpolation for alternate frames.

The remaining 3000 b/s is allocated to the gain, the pitch lag, and the pitch coefficient(s). The target bit rate only allows for the use of a single pitch coefficient. Our experience is that fairly rapid update of the pitch predictor is necessary for good quality. The pitch predictor is updated every 5 ms. The allocation then reduces to allocating 15 bits every 5 ms.

The scheme adopted uses 5 bits to code the gain parameter. The gain parameter is coded as sign and differential magnitude. The pitch lag and the pitch coefficient are coded together with 10 bits. The pitch lag takes on one of 73 values and the single pitch coefficient takes on one of 14 values. The coding approach is described in [10]. In that report, the need to allow the pitch lag to

fall below the subframe size is stressed. The configuration chosen allows the pitch to cover the range from 31 samples to 103 samples, corresponding to pitch frequencies of 78–258 Hz.

Recent papers show the allocations other workers have used for CELP coding. Kroon [13] reports an allocation of 1800 b/s for the formant coefficients, and 1500 b/s for the waveform coding. Rose and Barnwell report an alternate scheme with zero allocation for the waveform component [14]. This scheme uses a second pitch synthesis filter to increase the diversity of excitation signals available. In a more recent paper, Rose and Barnwell drop the second pitch filter to reduce the complexity, but at a sacrifice in quality [15]. Our experience has shown that a small number of waveforms (here 32) is preferable to the second pitch filter.

8. Postfiltering for Noise Reduction

Adaptive postfiltering is a technique used to enhance the subjective quality of speech signals contaminated by background coding noise. The postfilter emphasizes the important frequency components of the noisy speech and attenuates the others. The goal is to carry this out without inflicting unacceptable spectral distortion to the original speech signal.

Ramamoorthy and Jayant [16][17] consider a postfilter with the response of the form

$$G(z) = \frac{1 - \sum_{k=1}^{N_p} a_k \beta^k z^{-k}}{1 - \sum_{k=1}^{N_q} b_k \alpha^k z^{-k}} . \tag{15}$$

The log-spectrum of this filter is the difference of the log-spectra of two linear predictors of order N_p and N_q. The predictor coefficients are scaled by the parameters β and α to move the singularities radially in the z-plane. In the case of the coder considered in [16] and [17] the predictor has both poles and zeros. The postfilter used had two poles and six zeros. In the ADPCM application, the filter coefficients are available to the decoder as part of the normal operation of the decoder and the postfiltering can be applied with no bit rate penalty. The effect of the postfiltering is to reduce the effect of the coding noise but at the expensive of some muffling.

A modification of the adaptive postfiltering strategy was used by Chen and Gersho [18]. The coder in this case uses an all-pole synthesizer. The postfilter is defined in terms of the (time-varying) coefficients of of the synthesis filter,

$$G(z) = (1 - \mu z^{-1}) \frac{1 - \sum_{k=1}^{N_f} a_k \beta^k z^{-k}}{1 - \sum_{k=1}^{N_f} a_k \alpha^k z^{-k}} . \tag{16}$$

The principle is that the radially shifted denominator term results in a spectral tilt. The numerator uses the same base coefficients but radially shifts them further. The tilts then tend to cancel. The leading term in the postfilter expression is configured as a simple highpass filter to compensate for any residual tilt. The overall effect of the postfilter is such that noise (and signal) in the spectral valleys is suppressed.

In general, postfiltering causes three kinds of distortion to the speech. First, the spectral energy in the valleys of the spectrum is suppressed. However, the distortions at the spectral valleys have little perceptual impact [19][20]. Second, The high frequency components tend to fall off due to the spectral tilt. The problem of spectral tilt can be made inconsequential with the compensation used by Chen and Gersho. The third distortion is due to the discontinuous change in the spectral energies at block boundaries. The solution is to smooth out the energy contour.

Consider a running estimate of the energy of the sequence $\{\hat{x}(n)\}$,

$$\hat{E}_n = \lambda \hat{E}_{n-1} + \hat{x}^2(n) - \lambda^L \hat{x}^2(n-L) \ . \tag{17}$$

The window used to calculate the energy estimate is a truncated exponential window

$$w(n) = \begin{cases} \lambda^n & 0 \le n < L \\ 0 & \text{otherwise} \ . \end{cases} \tag{18}$$

This value is calculated for the input and the output of the postfilter. The square root of the ratio serves as a compensation to smooth out energy changes. A block based smoothing based on a raised cosine window was also tried. It has no advantage over the sample-by-sample scheme adopted and indeed involves an additional delay in the signal path.

The parameters chosen for the postfilter are $\mu = 0.25$, $\beta = 0.6$, $\alpha = 0.8$, $L = 20$, and $\lambda = 0.94$. The postfilter has a subtle but positive effect on the perceived quality of the speech. With the correct choice of parameters, the postfilter never adds distortion, but can in many cases reduce the perceptual impact of the coding noise. Since the improvement comes at no bit rate penalty, the inclusion of postfiltering is an asset. The computational complexity involved in its implementation is moderate compared to the overall CELP coder.

9. Summary and Conclusions

The CELP coder is running as a simulation in Fortran on a VAX 8600. The quality of the resulting speech has been informally ranked about equal to that for 6-bit log-PCM. The computational complexity of CELP is at the high end of the scale for coders.

The formant analysis that needs to be carried out is a small part of the computational load. Work is in progress to assess the computational complexity of the formant coding using LSF's.

Preliminary results show that the LSF computations can indeed be carried out using fixed point arithmetic. The computational load of this part of the scheme seems moderate.

The bulk of the computational load is in the waveform selection and pitch filter optimization module. The major computational load is in the form of filtering and correlation operations. These operations are very efficiently implementable on DSP chips. A single DSP chip (200 ns cycle time) can probably handle this part of the algorithm. In fact, one can be hopeful that the whole coder could be implemented on a single DSP chip, if not the present generation, the next generation (100 ns cycle time).

CELP is a very promising scheme for coding at practically important rates. The implementation discussed in this report produces high quality speech. The quality of the output is as good and possibly better than most competing schemes. The computational complexity is within grasp in present technology.

References

1. B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates", *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Paris, France, pp. 614–617, May 1982.
2. M. Beyrouti, P. Kabal, P. Mermelstein, H. Garten, "Computationally efficient multi-pulse speech coding", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, San Diego, Cal., pp. 10.1.1–10.1.4, March 1984.
3. S. Singhal and B. S. Atal, "Improving performance of multi-pulse LPC coders at low bit rates", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, San Diego, Cal., pp. 1.3.1–1.3.4, March 1984.
4. B. S. Atal and M. R. Schroeder, "Stochastic models for low bit rate coding of speech signals", *Int. Symposium on Inform. Theory (abstract)*, p. 124, Sept. 1983.
5. P. Kabal and R. P. Ramachandran, "Pitch prediction filters in speech coding", submitted to *IEEE Trans. Acoust., Speech, Signal Processing.*
6. 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.
7. B. W. Dickinson, "Autoregressive estimation using residual energy ratios", *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 503–506, July 1978.
8. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.
9. R. P. Ramachandran and P. Kabal, "Stability and performance analysis of pitch filters in speech coders", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 937–946, July 1987.
10. J.-L. Moncet and P. Kabal, "Codeword Selection for CELP Coders", *INRS-Telecommunications Technical Report*, 1987.
11. C. C. Chu and P. Kabal, "Coding of LPC Parameters fo Low Bit Rate Speech Coders", INRS-Telecommunications Technical Report 87-19, March 1987.
12. J. R. Crosmer and T. P. Barnwell, III, "A low bit rate segment vocoder based on line spectrum pairs", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Tampa, Florida, pp. 7.2.1–7.2.4, April 1985.
13. P. Kroon and B. S. Atal, "Quantization procedures for the excitation in CELP coders", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Dallas, Texas, pp. 1649–1652, April 1987.
14. R. C. Rose and T. P. Barnwell III, "The self excited vocoder — an alternate approach to toll quality at 4800 bps", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 453–456, April 1986.
15. R. C. Rose and T. P. Barnwell III, "Quality comparison of low complexity 4800 bps self excited and code excited vocoders", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 1637–1640, April 1987.
16. V. Ramamoorthy and N. S. Jayant, "Enhancement of ADPCM speech by adaptive postfiltering", *AT&T Bell Labs Tech. J.*, pp. 1465–1475, Oct. 1984.
17. N. S. Jayant and V. Ramamoorthy, "Adaptive postfiltering of 16 kb/s-ADPCM speech", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 829–832, April 1986.
18. J. H. Chen and A. Gersho, "Real-time vector APC speech coding at 4800 bps with adaptive postfiltering", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Dallas, Texas, pp. 2185–2188, April 1987.
19. J. L. Flanagan, *Speech Analysis, Synthesis and Perception, Second Edition*, Springer-Verlag, 1972.
20. J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, 1976.