Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (Toronto, ON), pp. 25-28, May 1991

Low-Delay CELP and Tree Coders: Comparison and Performance Improvements

Majid Foodeei[1] and Peter Kabal[1,2]

[1] Electrical Engineering
McGill University
Montreal, Quebec, H3A 2A7

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

Abstract

There has been a recent interest in network-quality speech coders with low-delay at 16 kb/s for CCITT standardization There is reason to believe that coders with rates below 16 kb/s will be able to meet the same quality standards The challenge is to develop high-performance mutually-compatible components for the target coder A stochastic tree coder based on the (M,L) search algorithm suggested by Iyengar and Kabal and a low-delay CELP proposed by Chen are considered First, the individual components (predictors, gain adaptation, excitation coding) of the two coders are analyzed Second, the performance of the two types of coders is compared The two coders have comparable performance at 16 kb/s under clean channel conditions. Finally, methods to improve the performance of the coders, particularly with a view of bringing the bit rate to below 16 kb/s are studied Suggestions to improve the performance include an improved high-order predictor (applicable to both coders), and training of the excitation dictionary as well as a better gain adaptation strategy for the tree coder

1. Introduction

A summary of the CCITT standardization specification for 16 kb/s low-delay coders is shown in Table 1 In future, similar kinds of requirements (e g low-delay and robustness to channel errors) can be expected for coding at bit rates below 16 kb/s

Parameter	CCITT Requirement	Objective
Coding Delay	≤ 5 ms	≤ 2 ms
$P_e = 0$ $P_e = 10^{-3}$ $P_e = 10^{-2}$	Distortion < 4 qdu Not worst than G.721 Not worst than G.721	
Tandeming for speech	3 Asynchronous Tandems with distortion < 14 qdu	Synchronous Tandems
Transmit Signaling Tones	DTMF	
Transmit Music		No annoying effects
Operate at lower rates		Graceful degradation
Complexity		As low as possible

Table 1 CCITT Standardization - characteristics for low-delay 16 kb/s coders [1]

The delayed-decision tree coder based on the (M,L) algorithm of [2] (LD-tree) and the Low-Delay Code Excited Linear Predictive coder of [3,4] (LD-CELP) may both be considered as potential candidate coders for low-delay network-quality applications Performance quality equivalent to 7 bits/sample log-PCM with delays less than 2 ms under clear channel conditions is achieved by the two coders. Satisfactory performance quality, under noisy channel conditions, is also reported for the LD-CELP The performance of the two coders, however has not been compared under the same conditions. In order to make the conditions for the comparison fairer, the pitch prediction filter in LD-tree is eliminated. The original LD-tree coder is a delayed-decision tree coder based on a generalized ADPCM. In an attempt to better unify the two coders, an equivalent interpretation for the LD-tree is given in Fig 1 The LD-CELP block diagram is shown in Fig. 2 The following common features may be identified.

- parameter selection using analysis-by-synthesis,
- high performance predictors for redundancy removal,
- gain scaling unit and the gain adaptation,
- perceptual weighting (noise-shaping), and
- excitation sequence dictionary or codebook with delayed search

Delayed-decision coding, as implemented in codebook (CELP), tree, and trellis coding, can efficiently represent the excitation signal This is done by postponing the decision as to which excitation signal is to be selected In an analysis-by-synthesis approach, the search for the optimum excitation dictionary or codebook entry at the encoder is effectively done by systematically trying each sequence. The sequence with the lowest perceptually weighted error (original signal to reconstructed signal) is selected To generate the reconstructed signal, the encoder uses a replica of the decoder The index corresponding to the selected sequence is transmitted to the decoder Adaptive gain scaling of the excitation signal improves the excitation representation by reducing the dynamic range of the excitation set The excitation signal is multiplied by the adaptive gain factor and then passed through the synthesis filter to generate the reconstructed signal The error signal is passed through a perceptual weighting filter prior to the error minimization. Note that in Fig. 1 (LD-tree), the flow of the speech sample processing is on a sample-by-sample basis while in the Fig 2 (LD-CELP), the flow is on a vector-by-vector basis

Postfiltering which improves the performance of conventional CELP or the original LD-tree proposed in [2] is not used. Postfiltering causes severe distortion to accumulate during tandem coding In addition, postfiltering is not appropriate for non-speech signals such as modems signals

Assuming a sampling rate of 8 kHz, the low-delay requirement for network applications limits the encoder delay to 5–8 samples (0 625–1 0 ms) The back-to-back delay for an encoder/decoder is usually 2–3 times the encoder delay This meets the objective of 2 ms

2. Coding of the excitation signal

Consider block coding of the excitation. To obtain a 16 kb/s coding rate, $R = 2$ bits/sample is used. If a R-ary coder sequence of length N and R bits/sample are used, the codebook size is

$$J = (2^R)^N \qquad (1)$$

In the case of the LD-CELP coder, $R = 2$ bits/sample and $N = 5$ samples/vector are used The codebook size is $J = 1024$, 10 bits per vector Fractional coding rates which are needed for rates below 16 kb/s, are easily obtained by varying the codebook size J and codevector dimension N (Eq. 1).

Consider a sliding window code for the excitation In tree coding, different sequences have several common elements and individual sequences form a path in the tree A consistent assignment of branch number is used throughout the tree which results in a unique *path map* for each path sequence. The path information for the best path is transmitted to the decoder The number of branches b, per node is called branching factor If β symbols per node are used, the encoding rate R in bits per symbol is given by

$$R = \frac{1}{\beta}\log_2 b = \frac{k}{\beta} \quad (b = 2^k) \qquad (2)$$

Choosing $\beta = 1$ sample per node and the branching factor $b = 4$, results in a rate $R = 2$ bits/sample (Eq 2) Fractional rates can be achieved using the concept of a *multi-tree* as suggested in [5] The idea is that the branching factor of the tree at different depths changes along the paths

LD-tree stochastic tree

The stochastic tree speech coders perform better than deterministic ones A distinction can be made between the innovation code tree which represents the excitation signal and the reconstruction code tree which represents the reconstructed output signal. The nodes of the innovation tree are populated from a Laplacian random number dictionary of size 2^k (in this study 2^k=4096). The reconstructed code tree is obtained by multiplying each of the innovation tree node values by a gain factor and then passing it through a synthesis filter.

Reduced search complexity is achieved using the (M,L) algorithm, where M denotes the maximum number of paths kept in contention and L is the number of samples in each of these paths or the decision delay length. This study uses $M = 16$ and $L = 8$ which results in encoding delay of 8 samples. The end-to-end delay is comparable to the one obtained in LD-CELP. The branching factor of the tree is 4, which means the tree coder produces 2 bits/sample At time instant n, each of the M (maximum) paths in contention are extended. The error accumulated for each of the $4M$ extended paths is calculated. The path with the lowest accumulated error is selected. The two bit branch code of the root of this path L samples back is the only information transmitted to the receiver at time n (indicated by $c(n-L)$ in Fig. 1). Only valid paths that stem from this root are kept (maximum M).

LD-CELP codebook

The LD-CELP, like the conventional CELP, searches the codebook for the best matching codevector (each vector is 5 samples long) using analysis by synthesis and by minimizing the perceptually weighted error (Fig. 2) As indicated by the thick lines Fig. 2, the signal processing is on a block-by-block basis. This block characteristic of the LD-CELP versus the sliding window characteristic of the LD-tree is the main conceptual difference between the two coders The computations for LD-CELP search are synchronized with the blocks, while for the LD-tree search the computations are distributed in time.

Separation of the zero-state response response and the zero-input response can be used to reduce the computation load during the search for the best matching codevector.

A product vector quantizer (Gain/Shape VQ) is used to reduce the search complexity LD-CELP uses a 7-bit shape codebook and a 3-bit gain codebook Note that the gain component is a form of side information that is transmitted. The shape/gain codebook is trained [6]. For individually optimized gain/shape codebooks, the initial shape codevectors were chosen from numbers with Gaussian probability distribution In a closed loop gain-adaptive training algorithm used, the distortion-versus-iteration does not necessarily decrease monotonically The codebook with the lowest distortion after a preset number of iterations is saved

The shape and gain codevector indices are pseudo-Gray coded A single bit error in a received codevector tends to be close to the transmitted one. Pseudo-Gray coding results in a significant improvement in a noisy channel environment [4].

3. Redundancy removal

LD-tree lattice predictor

The redundancy removal component in the LD-tree coder includes a formant predictor $F(z)$ ($p = 8$ in the original LD-tree) which acts on the short-term redundancies in the input speech, $F(z) = \sum_{i=1}^{p} a_i z^{-i}$. The adaptation of the predictor is done in a backward fashion using the adaptive Lattice algorithm. The reflection coefficients are converted to the direct form before use in the prediction filter. The pitch filter in the original LD-tree configuration has been removed, since it performs poorly in the presence of channel errors (incorrect lag values at the decoder cause error propagation).

A one-pole exponential window is used on the analysis data. The formant prediction filter coefficients are updated in a delayed update configuration. In Fig. 1 this is shown as $\{a_i(n-2L)\}$ or $\{a_i(n-L)\}$ which means that the update algorithm at time instant n uses samples as recent as $2L$ or L samples back. Reference [2] shows that L sample delay update strategy actually results in better prediction gains than zero delay update strategy. As seen in Fig 1, this also results in complexity reduction of the coder since only one update of the LPC coefficients is done for all branches in contention.

LD-CELP high-order predictor

LD-CELP also eliminates the separate pitch synthesis filter However, a backward adaptive high order LPC predictor filter is used to remove both formant and pitch redundancies. LPC analysis using the Auto-correlation method with a Barnwell data window is used for this filter [4]. An order 50 predictor is used since the the prediction gain and the coder SNR saturate at order 20 for male speakers and around order 50 for female speakers

In LD-CELP, bandwidth expansion is used to make the noise less perceivable. In effect the formant peaks are widened in the frequency response by moving the poles away from the unit circle or concentrating the noise in the formant regions. Better robustness to channel errors is attained. The modified LPC predictor is $F(z/\lambda)$ with $\lambda = e^{-2\pi B/8000} = 0.988$ ($B = 15$ Hz) . White noise correlation technique is used to "clamp" the spectral dynamic range to 40 dB and to reduce the problems due to ill-conditioning.

The LD-CELP uses a recursive Barnwell window (to distribute the computation load for implementational considerations) for the backward adaptation of the prediction filter. The update of the coefficients is done every 8th vector (5 ms).

Since the spectral changes in the speech signals are relatively slow varying, a significant computation load reduction is obtained with a minimum loss of performance by updating the coefficients less frequently (e.g. every 20 or 40 samples in LD-CELP).

4. Adaptive gain scaling

To increase the dynamic range of the excitation signal, each excitation value is multiplied by the node gain to yield the excitation sample e_q In LD-tree, the node gain is backward adaptive,

$$G^2(n+1) = \delta^2 G^2(n) + (1-\delta)e_q^2(n) \quad \text{where} \quad (\delta = 0\ 86).$$

The above adaptation is an exponentially averaged variance estimate with δ controlling the effective window length. This adaptation strategy was used because of its simple implementation within a tree search [2]

For LD-CELP, two methods for gain scaling were suggested The Jayant gain adapter generalized for vectors is robust to channel errors The alternate adaptive logarithmic gain predictor gives higher clean channel performance This method uses a relatively sophisticated prediction based on past gain values In both methods, a leakage factor close to unity is used in order to improve the robustness [4]

5. Perceptual weighting of the error

For the LD-tree, the perceptual weighting filter used has the form

$$W(z) = \frac{1 - F(z)}{1 - F(z/\lambda)}, \quad \text{where} \quad F(z/\lambda) = \sum_{i=1}^{p} a_i \lambda^i z^{-i}$$

is the bandwidth expanded version of $F(z)$ ($\lambda = 0.85$ is used). The excitation signal is passed through the perceptual weighting filter before the (M,L) tree search algorithm is applied.

For LD-CELP, the perceptual weighting filter is

$$W(z) = \frac{1 - Q(z/\lambda_1)}{1 - Q(z/\lambda_2)}, \quad 0 < \lambda_2 < \lambda_1 \leq 1 \quad (\lambda_1 = 0\ 9, \lambda_2 = 0\ 4)$$

where

$$Q(z/\lambda_1) = \sum_{i=0}^{p} q_i \lambda_1^i z^{-i} \quad \text{and} \quad Q(z/\lambda_2) = \sum_{i=0}^{p} q_i \lambda_2^i z^{-i}.$$

The weighting filter uses a separate LPC analysis with order 10 rather than the 50th order analysis filter Use of a 50th order weighting filter results in speech artifacts The analysis for computing the q_i's uses the *unquantized* speech

6. Comparison

The LD-tree and LD-CELP coders as described above were compared Table 2 shows the segmental SNR results for two utterances by a male and a female speaker. These results are typical of other utterances. The segSNR results are close for the two coders. The LD-tree does somewhat better for male utterances and the LD-CELP is slightly better for female utterances The informal subjective tests also agree with the above conclusion. The above comparison is for no channel errors Note that the design of the original version of LD-tree coder did not consider channel errors The coded speech using the LD-tree degrades rapidly under the noisy channel conditions while the LD-CELP withstands P_e of 10^{-3} and 10^{-2} with acceptable levels of quality loss

Coder	CAT		OAK	
	Male	Female	Male	Female
LD-CELP	17.9	19.1	18 4	20 1
LD-tree	19 1	19 1	19.6	19 3

Table 2 Comparison of Coder segSNR (dB)

7. Discussion

The essential difference between the coders is the block versus sliding window excitation coding Without channel error considerations, sliding window coders would seem to be preferable in terms of performance alone There are no block edge effects with sliding window techniques However, channel errors propagate for longer times within the sliding block structure In addition, pseudo-Gray coding to mitigate the effect of errors is possible with block codes Coarse simulation execution time comparisons indicate that the two coders have comparable complexities The tradeoff here would seem to be clean channel performance versus noisy channel performance. Note that there are important applications (e g. undersea optical fibre transmission systems) in which channel errors are not significant. On the other hand, for other applications (mobile radio or in-building wireless), channel error rates can be much more severe than the rates in the CCITT objectives.

Both coders contain structures with similar functions. The various components can be mixed and matched between the coders It must be kept in mind that in a backward adaptive structure, each component must perform well. For instance a good excitation coding results in an accurate reconstructed signal which in turn is used to adapt the predictor. A breakdown in either the excitation coding or the predictor update results in breakdown of the coder.

Under clean channel conditions, even though LD-CELP uses a 50th order predictor compared to the LD-tree 8th order predictor, the overall speech quality is very similar. An advantage accrues to the high-order predictor for female speech when the pitch range falls within 50 samples High-order predictors can be used in either coder.

The perceptual weighting filters used in the LD-tree and LD-CELP coders not only differ in form (as seen earlier) but also are different in the use of *quantized* or *unquantized* speech signal to update their coefficients. The use of high-order synthesis filter in the LD-CELP has forced the coder to use a separate predictor filter for the perceptual weighting filter. Again this strategy can be carried along with high order to the LD-tree coder.

The bandwidth expansion applied to the high-order LPC predictor of LD-CELP can also be applied to the LPC predictor in the LD-tree coder. This can also improve the robustness to channel errors by making the noise less perceivable

8. Improvements

The speech quality of both LD-CELP and LD-tree at 16 kb/s is very high. "Improvements" applied to the coders are not very noticeable. The ultimate goal is that such improvements will allow the high quality coding at lower rates Of course, there may be a substantial computational penalty to be paid for these improvements.

Better predictors

The 50-th order predictor in LD-CELP does capture pitch effects within its lag range. However, the use of the Auto-correlation method does not fully exploit the high lag correlations because of window edge effects. The well-conditioned high-order Cumani Covariance-Lattice method does not have the above window/order problems and produces higher prediction gains [7]. The Cumani algorithm is one of a larger class of algorithms (*Pure Order Recursive Ladder Algorithms*) which are potentially useful for high-order prediction [8].

Results of extensive experiments to obtain high-quality LPC analysis for high-order predictors (as presented in the companion paper [7]) can be used to improve the performance of both coders. For clean speech, the prediction gain of the Cumani Covariance-Lattice method [9]) is several dB higher (approximately 2 dB for female and 3 dB for male utterances) than the Barnwell Auto-correlation method. The reasons for this better performance are discussed in that paper Table 3 compares the Cumani Covariance Lattice method with the Auto-correlation method in LD-CELP (both order 50). The effect of backward adaptation based on the (noisy) reconstructed signal is such that the objective performance of the two methods is not noticeably different However, informal subjective tests indicate that the differences are either absent (both are of very high quality with no noticeable degradations) or there is a slight preference for the Covariance Lattice method (especially for male speakers).

Analysis Method	Male		Female	
	SNR	segSNR	SNR	segSNR
Barnwell-Auto	19 6	21 0	24 1	22 6
Covariance-Lattice	19 7	20 7	22 9	21 7

Table 3 Objective coder performance comparison of LD-CELP with two analysis techniques (dB)

Improvements to LD-tree

As it is done for the LD-CELP codebooks, training of the innovation dictionary boosts the performance of the LD-tree coder. In an experiment segSNR improvements of about 1 dB were obtained when the stochastic innovation dictionary was trained Due to the initial high quality, this "improvement" did not change the perceived quality

Although the exponentially averaged gain adaptation method of LD-tree is adequate for clean channels, a better gain adapter suited for the stochastic tree coders is required to overcome the malfunctioning of the LD-tree coder under noisy channel conditions Preliminary experiments have verified this effect. Simple remedies were applied to achieve much better robustness, but with a loss of coder performance. Use of more complex gain adaptation strategies similar to the ones used in the LD-CELP could provide error robustness with no loss or even a possible increase in performance with no errors If the gain adaptation updates are delayed (similar to the prediction coefficient update), this method becomes computationally practical.

A hybrid coder, taking the best components from the two archetypes is a good bet to push coding rates below the current 16 kb/s. Consider a coder with a trained stochastic tree, high-order Covariance Lattice predictor, and logarithmic predictive gain adaptation Such a coder would probably only be marginally better at 16 kb/s, but may allow for high quality speech coding at rates of 12-14 kb/s.

References

1. N. S. Jayant, "High-quality coding of telephone speech and wideband audio," *IEEE Communication magazine*, pp 10-20, Jan. 1990.
2. V. Iyengar and P. Kabal, "A low delay 16 kb/s speech coder,"in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, ICASSP-88*, pp. 243-246.
3. AT&T contributions to CCITT Study Group XV and T1Y1 2 (October 1988 - July 1989).
4 J. H Chen, "A Robust low-delay CELP speech coder at 16 kb/s," *IEEE workshop on speech coding for telecommunications*, Vancouver, Canada, Sept. 1989.
5. J D. Gibson and W.-W. Chang, " Fractional rate multi-tree speech coding," in *Proc. IEEE Int. Conf Acoust , Speech, Signal Processing, ICASSP-89*, pp. 53.1.1-53.1.4.

6 J H Chen and A. Gersho, "Gain-adaptive vector quantization with application to speech coding," *IEEE Trans. Commun* , pp. 918-930, Sept. 1987.

7 M Foodeei and P Kabal "Backward Adaptive Prediction High-Order Predictors and Formant-Pitch Configuration," in *Proc IEEE Int Conf Acoust , Speech, Signal Processing*, Toronto, Ont May 1991

8. P. Strobach, "Recursive Triangular Array Ladder Algorithms," *IEEE Trans. on Signal Processing*, vol. SP-39, Jan 1991, pp 122-136.

9 A. Cumani, "On a Covariance Lattice Algorithm for Linear Prediction," in *Proc Int Conf Acoust , Speech, Signal Processing*, Paris, Frace, 1982, pp 651-654

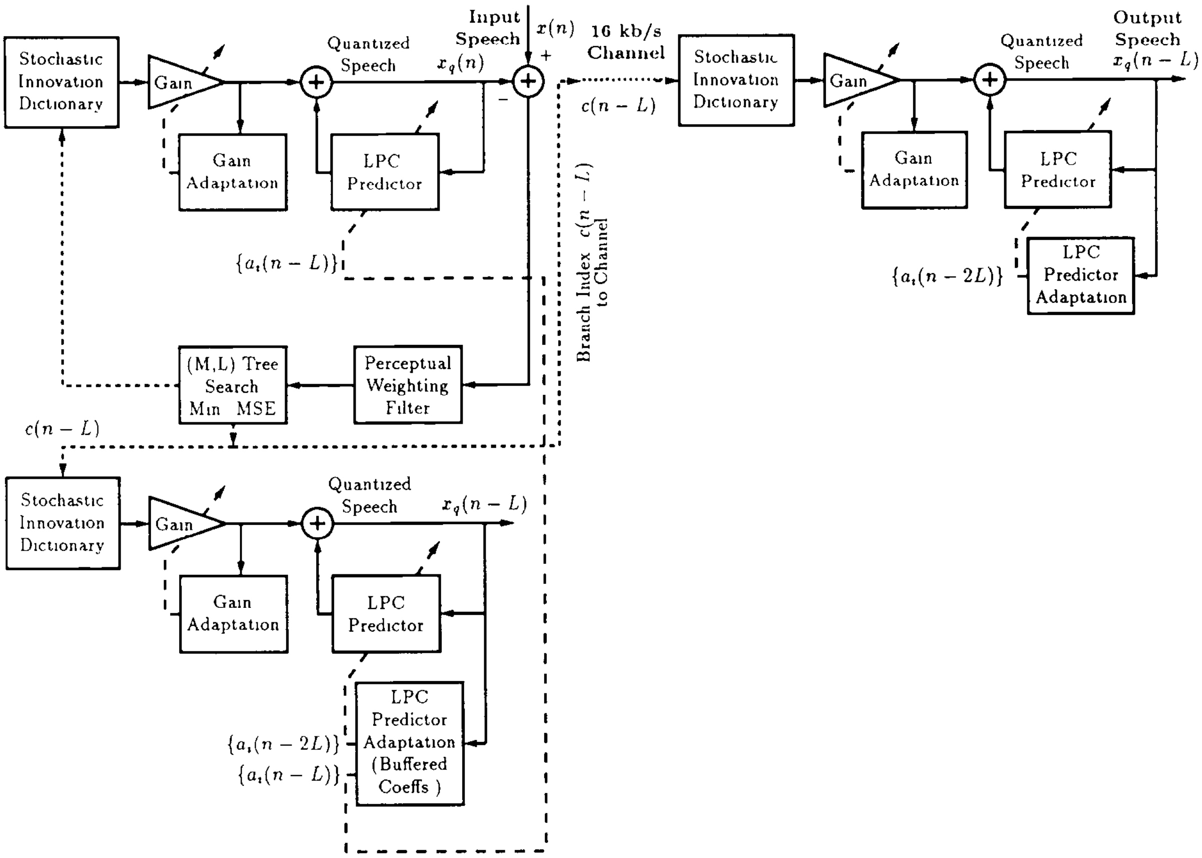
Stochastic Innovation Dictionary
Gain
Gain Adaptation
Quantized Speech $x_q(n)$
Input Speech
$x(n)$
+
−
LPC Predictor
$\{a_i(n-L)\}$
16 kb/s Channel
$c(n-L)$
Branch Index $c(n-L)$ to Channel
(M,L) Tree Search Min MSE
Perceptual Weighting Filter
$c(n-L)$
Stochastic Innovation Dictionary
Gain
Gain Adaptation
Quantized Speech $x_q(n-L)$
LPC Predictor
$\{a_i(n-2L)\}$
$\{a_i(n-L)\}$
LPC Predictor Adaptation (Buffered Coeffs)
Stochastic Innovation Dictionary
Gain
Gain Adaptation
Quantized Speech
Output Speech $x_q(n-L)$
LPC Predictor
$\{a_i(n-2L)\}$
LPC Predictor Adaptation

Fig. 1 LD-tree Encoder and Decoder Block Diagram

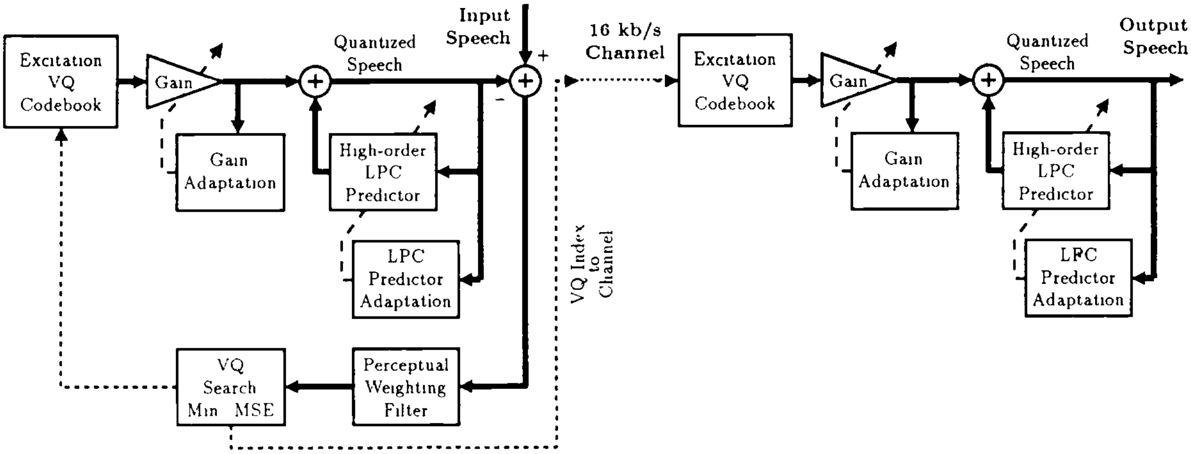

Fig. 2 LD-CELP Encoder and Decoder Block Diagram