Speech and Audio Coding for Wireless and Network Applications, pp. 3-9, Kluwer Academic, 1993

HIGH QUALITY LOW-DELAY SPEECH CODING AT 12 KB/S

J. Grass [1], P. Kabal [1,2], M. Foodeei [2] and P. Mermelstein [1,2,3]

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

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

[3] BNR
16 Place du Commerce
Verdun, Quebec
Canada H3E 1H6

INTRODUCTION

For low-delay speech coders, the research challenge is to obtain higher compression rates while maintaining very high speech quality and meeting stringent low delay requirements. Such coders have applications in telephone networks, mobile radio, and increasingly for in-building wireless telephony.

A low-delay CELP algorithm operating at 16 kb/s has been proposed for CCITT standardization [1, 2, 3, 4]. An alternate coding structure operating at the same rate is based on an ML-Tree algorithm [5]. Both algorithms offer near-network quality with coding delays below 2 ms at 16 kb/s. In this work, we modify these basic coder structures to operate at the reduced rate of 12 kb/s while retaining high speech quality.

In the low-delay coders considered here, the following common features may be identified.

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

Delayed-decision coding, as implemented in codebook (CELP), tree, and trellis coding, can efficiently represent the residual signal. This is done by postponing the decision as to which quantized residual 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 obtained by systematically examining the performance resulting from the use of each sequence. The sequence with the lowest perceptually weighted error (original signal sequence 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 entry is transmitted to the decoder. In addition, adaptive gain scaling of the excitation signal is used since it improves the excitation representation by reducing the dynamic range of the excitation set. At the encoder, the error

signal is passed through a perceptual weighting filter prior to the error minimization. At the decoder, an optional postfiltering stage can be added to further improve perceptual quality.

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. The overall coder bit-rate is obtained by multipling the sampling frequency f_s by the number of bits/sample ($I = f_s \times R$).

For block-based coding, if a coder sequence (R bits/sample) of length N and a codebook size of J are used, the following relation holds.

$$R = \frac{1}{N} \log_2 J = \frac{k}{N} \quad (J = 2^k). \tag{1}$$

Fractional coding rates are easily obtained by selecting the proper codebook size J and codevector dimension N.

An alternative to block-based coding is a sliding window code for the excitation. In tree and trellis coding, different sequences have several common elements and individual sequences form a path in the tree or trellis. Tree structures [6, 7] are considered here. 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 the *branching factor*. If β symbols per node are used, the encoding rate R in bits per symbol is given by

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

Fractional rates can be achieved either by selecting a β value greater than one (*multi-symbols/node*) or by using the concept of a *multi-tree*. In the latter alternative, the branching factor of the tree at different depths changes along the paths (see [8, 9] for more detail).

LOW-DELAY BLOCK-BASED CODING

The low-delay CELP algorithm originally designed for 16 kb/s [2], was modified to operate at 12 kb/s. The bit-rate of the block-based coder is determined by the sampling rate multiplied by the codebook size (number of bits) and divided by the vector length used in the codebook (Eqn. 1). The sampling rate was kept fixed at 8 kHz. A number of different combinations of the parameters were examined. The best of these combinations was found to be a 9 bit codebook and a 6-sample vector size (which corresponds to an encoding delay of 0.75 ms). The codebook design uses a full search approach rather than partitioning into shape/gain sub-codebooks. The codebook was retrained for the lower bit-rate.

The modified coder operating at 12 kb/s maintains good quality for female talkers but the quality degrades somewhat for male speakers. This difference can be attributed to the ability of the 50th order predictor (autocorrelation with analysis updated every 24 samples) to capture some aspects of pitch for

female talkers but not for male talkers. Higher order predictors were studied by Foodeei and Kabal [9, 10]. High order (up to 80) covariance analysis allows for the capture of pitch redundancies associated with male talkers. Furthermore, the Cumani algorithm provides a numerically stable algorithm for determining the coefficients of the high-order filter [11].

Using the covariance-lattice predictor in the block-based coder at 12 kb/s instead of the autocorrelation predictor, the quality of the male speech is improved. The covariance-lattice predictor has been shown to increase prediction gain over 2 dB for male speakers [10]. In the 12 kb/s coder, the overall objective performance of the coder in terms of SNR did not change. This may be attributed to the the fact that the adaptation is based on the reconstructed speech. Perceptually however, the covariance-lattice technique provides improvements in the coder for male speakers.

LOW-DELAY TREE CODER

The ML-Tree algorithm was originally used in a configuration with a 3-tap pitch predictor. The adaptive predictor, with dynamic determination of the pitch lag, suffers from error propagation effects. Using an 8th order formant predictor and a simple gain adjustment procedure, the ML-Tree coder at 16 kb/s has speech comparable to that of LD-CELP at the same bit rate [9, 12].

At 16 kb/s, the coding tree has a branching factor of 4 at each sample (2 bits per sample). Our strategy to lower the bit rate is to use combined vector-tree coding (*multi-symbols/node*). The encoding delay is a function of the path length and the number of samples populating each node. The overall bit-rate is given by the sampling rate divided by the number of samples considered at each node and multiplied by the number of bits to represent the branching factor (Eqn. 2). Two configurations were studied, one using 3 bits for the branching factor and 2 samples per node while in the second configuration 6 bits are used for the branching factor and 4 samples per node. The former combination was preferred.

Prediction Filter

The original implementation of the low-delay tree coder uses the generalized predictive coder configuration [5]. In this structure, the reconstruction error is given by $R(z) = Q(z)\frac{1-N_1(z)}{1-F(z)}$. $F(z)$ is the predictor filter, $N_1(z)$ is the noise feedback function and $Q(z)$ is the quantization error. $N_1(z)$ is set equal to $F(z/\mu_1)$. The feedback filter in the this structure provides a method to shape the noise spectrum.

An alternative configuration of the generalized predictive coder structure is that given by Atal and Schroeder [13]. In this closed-loop structure shown in Fig. 1, the perceptual weighting takes the same form as that used in the block-based coder; $W(z) = \frac{1-N_2(z)}{1-N_1(z)}$ where $N_1(z)$ is set equal to $F'(z/\mu_1)$ and $N_2(z)$ to $F'(z/\mu_2)$. The noise feedback filter is no longer directly linked to the prediction filter. The weighting filter can be determined from the clean input speech signal. Furthermore, the prediction filter and perceptual filter need not be of the same order. The noise feedback filters were 10th order filters, adapted

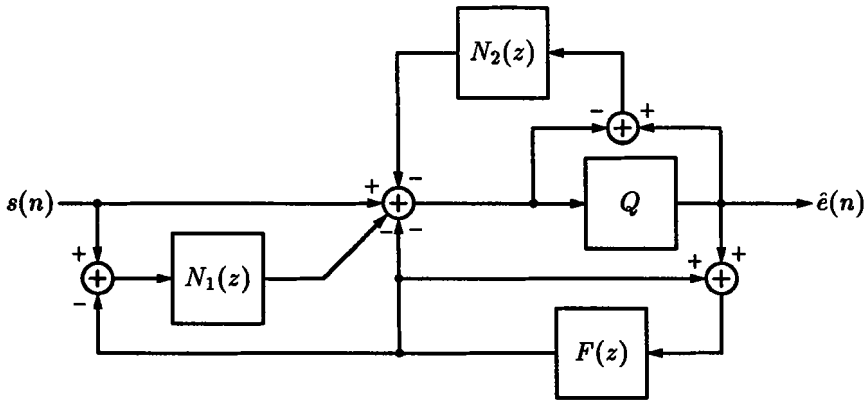

Fig. 1 A closed-loop configuration with generalized noise feedback

from the clean speech (the same as in LD-CELP). For this choice, the resulting speech was significantly better than that for the original configuration of the low-delay tree coder.

For the prediction filter, a configuration using a high order covariance filter and a configuration using a separate pitch filter were compared. The separate pitch filter performed better in terms of reduced pitch spikes in the residual, but subjectively there was little difference.

Gain Adapter

Several gain adaptation schemes were evaluated in the context of the low-delay tree coder. Particular attention was given to the adaptive logarithmic gain update strategy originally used in the 16 kb/s LD-CELP. It was found that the simple gain adaptation scheme proposed by Iyengar [5] achieved SNR results similar to the more complex gain adapters. Perceptually, a slight preference is given to the LD-CELP gain update method.

Dictionary Training

The dictionary for the innovation tree of the coder can be populated in a random fashion [5]. However, improvements as large as 1.5 dB in the performance of the coder at 12 kb/s were achieved by a new training procedure of the dictionary (training speakers and sentences were different than those used for testing).

The training procedure initially uses a randomly populated codebook. In each iteration, the coder is run, accumulating the unquantized prediction errors (residuals) associated with each released node of the tree. Each unquantized residual is assigned to a Voronoi cell corresponding to an entry in the dictionary with smallest distance to this residual. Note that due to the delayed nature of the tree coder, the unquantized residuals must be retained for the length of the delay. Further, the gain value used at each node of the tree must be kept so that the unquantized residual can be appropriately scaled. The centroid of the unquantized residuals in each Voronoi cell is found and used to replace

the associated dictionary entry in the previous iteration. With an updated dictionary, this process is repeated for several iterations.

DISCUSSION

The speech quality for the block-based coder operating at 12 kb/s is remarkably good. The principal difference when compared to 16 kb/s LD-CELP is a modest degradation for some male speakers. In comparing the two coders at 12 kb/s, the low-delay tree coder produces speech that is slightly better perceptually than that of the block-based coder.

We noted a significant improvement in the low-delay tree coder with the change to a generalized perceptual weighting, with the weighting filter determined from the clean speech rather than the reconstructed speech. Further work is warranted to compare the noise feedback as used in the tree coder with the open-loop weighting used in block-based coders. In addition, the use of high order covariance-lattice predictors in tree coders needs further investigation.

REFERENCES

1. AT&T contributions to CCITT Study Group XV and T1Y1.2 (October 1988–July 1989).
2. Detailed description of AT&T's LD-CELP algorithm, contributions to CCITT Study Group XV, Nov. 1989.
3. Draft recommendation G.728 (coding of speech at 16 kb/s using LD-CELP), CCITT Study Group XV, Dec. 1991.
4. J.-H. Chen, "High-quality 16 kb/s speech coding with a one-way delay less than 2 ms", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, (Albuquerque, NM), April 1990, pp. 453–456.
5. V. Iyengar and P. Kabal, "A low-delay 16 kbits/sec speech coder", *IEEE Trans. Signal Processing*, vol. 39, May 1991, pp. 1049–1057.
6. J. B. Anderson and J. B. Bodie, "Tree coding of speech," *IEEE Trans. on Inform. Theory*, vol IT-21, pp. 379–387, July 1975.
7. N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984.
8. J. D. Gibson and W.-W. Chang, "Fractional rate multi-tree speech coding," *IEEE Trans. Commun.*, vol. 39, pp. 963–974, June 1991.
9. M. Foodeei, "Low-delay speech coding at 16 kb/s and below", *Master of Eng. Thesis , Dept. of Elect. Eng. McGill University*, (May 1991).
10. M. Foodeei and P. Kabal, "Backward adaptive prediction: high-order predictors and formant-pitch configuration", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, (Toronto, Canada), May 1991, pp. 2405–2408.
11. A. Cumani, "On a covariance lattice algorithm for linear prediction", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, (Paris, France), 1982, pp. 651–654.
12. M. Foodeei and P. Kabal, "Low-delay CELP and Tree coders: comparisons and performance improvements", *Proc. Int. Conf. on Acoust. Speech, Signal Processing*, (Toronto, Canada), May 1991, pp. 25–28.
13. 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, June 1979, pp. 247–254.