Fast CELP coding based on algebraic codes.

J-P. Adoul, P. Mabilleau, M. Delprat and S. Morissette

Communication Research Center
University of Sherbrooke
Sherbrooke, P.Q., CANADA, J1K 2R1

Abstract

Code-Excited Linear Prediction (CELP) produces high quality synthetic speech at low bit rate. However the basic scheme leads to huge computational loads. The paper describes a related scheme, which allows real time implementation on current DSP chips. The very efficient search procedure in the codebook is achieved by means of a new technique called "backward filtering" and the use of algebraic codes. RSB performances are reported for a variety of conditions.

Introduction

Significant progress has recently been made in low bit rate coding of speech, which suggests that high quality synthetic speech may be produced at 4800 bps and lower rates.

In that field, Code Excited Linear Prediction (CELP) is a very promising technique for narrow band coding of speech. However the huge amount of computations involved appeared as quite a challenge when it was first introduced [1]. The comparatively high quality at low bit rate is achieved through an analysis-by-synthesis procedure using both short-term and long-term prediction as shown in figure 1.

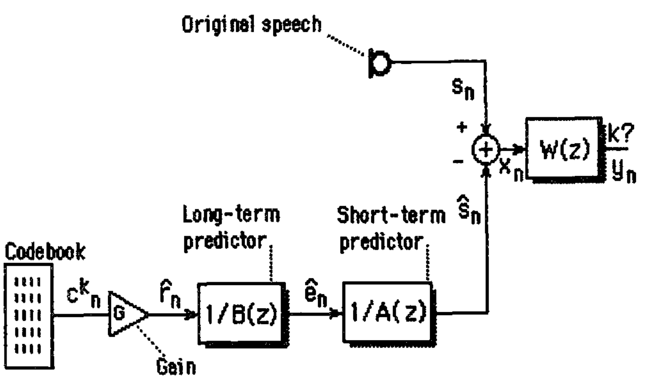

Figure 1: original proposal

The analysis procedure consists of finding the innovation sequence in the codebook which is optimum with respects to a subjective error criterion. Each codeword (or sequence) c_k is scaled by an optimal gain factor G_k and processed through the inverse filters $1/B(z)$ (pitch predictor) and $1/A(z)$ (linear-prediction inverse filter). The difference x_n between the original and synthetic signals (s_n and $\hat{s}_n$) is processed through the perceptual weighting filter $W(z)$ and the "best" sequence is then chosen to minimize the energy of the perceptual error signal y_n.

As mentioned by Trancoso and Atal [2], the huge amount of computations mainly comes from the search procedure for finding the optimum innovation sequence, and particularly from the filtering of all the sequences through both long-term and short-term predictors. Thus efforts have been concentrated on this problem.

We shall discuss in turn four proposals which will enable a dramatic reduction in computation. The four proposals are:

a) A modification of the basic structure.

b) The concept of "backward filtering" together with the use of an LPC-filter codebook.

c) The use of short innovation sequences with a tree coding approach.

d) The use of algebraic structures for the innovation codebook.

Transformation of the basic scheme

A simple and useful perceptual weighting filter is expressed as $W(z) = A(z)/A(z/\gamma)$ (1)
where γ is the perceptual weighting coefficient (chosen around 0.8) and $A(z)$ is the linear prediction filter: $A(z) = \Sigma_i a_i z^{-i}$. The filter $W(z)$ can be moved both in the upper branch and in the lower branch, as shown in figure 2.

In the upper branch the original signal is processed through the analysis filter $A(z)$, yielding a residual signal e_n from which the pitch parameters are derived. Then this residual signal is processed through the inverse filter $1/A(z/\gamma)$. In the lower branch the

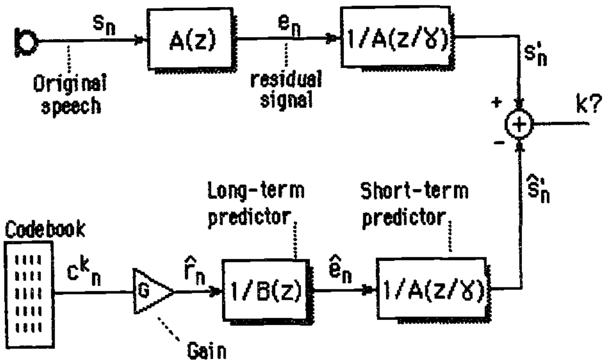

Figure 2: the perceptual weighting filter is moved both in the upper and in the lower branch.

inverse filter $1/A(z/\gamma)$ now replaces $1/A(z)$.

The pitch predictor is chosen to be a single tap predictor : $B(z) = 1 - b\,z^{-T}$ (2)

where b is the gain and T is abusively called the "pitch period". The expression of the output signal $\hat{e}_n$ of the pitch predictor $1/B(z)$ is then derived from (2):

$$\hat{e}_n = \hat{r}_n + b\,\hat{e}_{n-T} \quad (3)$$

with $\hat{r}_n = G_k\, c_n^k \quad , n = 0, \ldots, N-1$ (4)

where N is the block size (length of the codewords).

During the search procedure, the signal $\hat{e}_{n-T}$ is known and does not depend on the codeword currently tested. Thus the pitch predictor $1/B(z)$ can be removed from the lower branch if the signal $b\hat{e}_{n-T}$ is subtracted from the residual signal in the upper branch [2]. Using (3), the signal $\hat{e}_{n-T}$ is obtained by processing the delayed signal $\hat{r}_{n-T}$ through the pitch predictor $1/B(z)$; and $\hat{r}_{n-T}$ is computed from already known codewords, chosen for preceding blocks, provided that the pitch period T is restricted to values greater than the block size N.

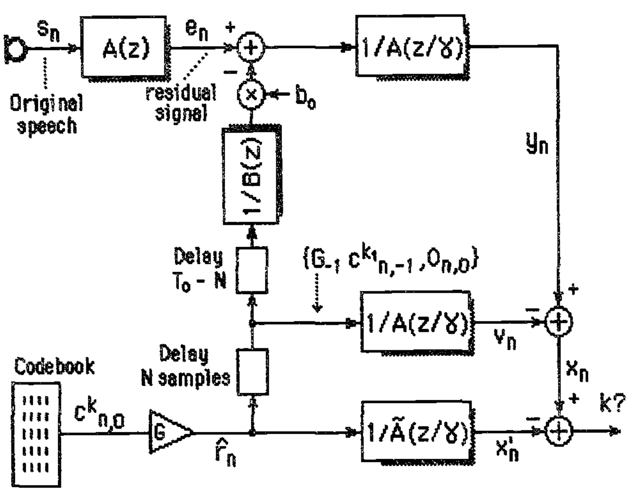

Figure 3: the long-term predictor and the "memory" of the short-term predictor are removed from the lower branch

Now, let us consider the output signal $\hat{s}'_n$ of the inverse filter $1/A(z/\gamma)$:

$$\hat{s}'_n = \hat{e}_n - \sum_{i=1}^{M} a_i \gamma^i \hat{s}'_{n-i} = \hat{x}'_n + v_n \qquad \text{with}$$

$$\hat{x}'_n = \hat{e}_n - \sum_{i=1}^{L} a_i \gamma^i \hat{x}'_{n-i} \quad , \; L = \mathrm{Min}(n-1, M) \qquad \text{and}$$

$$v_n = -\sum_{i=L+1}^{M} a_i \gamma^i \hat{s}'_{n-i} - \sum_{i=1}^{L} a_i \gamma^i v_{n-i}$$

where n=0 is the beginning of the block. The signal $\hat{x}'_n$ represents the output of the inverse memoryless filter $1/\tilde{A}(z/\gamma)$, while v_n is the output of $1/A(z/\gamma)$ without any excitation. The signal v_n only depends on the codewords chosen for preceding blocks, so it can be removed from the lower branch and subtracted from the upper one [2].

The new scheme, with the long-term predictor and the "memory" of the short-term predictor removed from the lower branch, is represented in figure 3. The two filterings per codeword are here reduced to a single memoryless filtering, with a significant cut in the computational load, and this structure enables further modifications.

The "Backward Filtering" principle and VQ of LP-filters

The search procedure now consists of minimizing the error signal energy E over the current block, and figure 3 leads to the expression

$$E = \Sigma_n (x_n - G g_n)^2 \quad ; \text{sum over } n = 0, 1, \ldots, N-1 \quad (5)$$

where g_n is the response of $1/\tilde{A}(z/\gamma)$ to the currently tested codeword and G is the related gain factor. The minimization is achieved in two steps:

1) Find G, setting $\partial E/\partial G = 0$.

From (5) it comes:

$$\partial E/\partial G = -2\, \Sigma_n\, x_n g_n + 2G\, \Sigma_n\, g_n^2$$

So $$G = (\Sigma_n\, x_n g_n) \,/\, (\Sigma_n\, g_n^2) \quad (6)$$

Replacing the expression (6) in equation (5) gives

$$E = \Sigma_n\, x_n^2 - ((\Sigma_n\, x_n g_n)^2 \,/\, (\Sigma_n\, g_n^2)) \,.$$

2) Minimize E over the codebook entries.

Since x_n does not depend on the codeword tested, minimizing E is equivalent to maximizing the weighted inner product $P_\psi = P/\alpha_k$ where $\alpha_k^2 = \Sigma_n\, g_n^2$ is the energy of the response of the memoryless filter $1/\tilde{A}(z/\gamma)$ to the currently tested codeword k, and P is the inner product between x_n and g_n:

$$P = \Sigma_n\, x_n g_n \quad (7)$$

Thus the search procedure consists of finding the codeword k which maximizes P_ψ over the codebook, as represented in figure 4a.

Since g_n is the response of the memoryless inverse filter $1/\tilde{A}(z/\gamma)$ to the currently tested codeword $c = (c_n)$, it can be written as the convolution product between c and the impulse response f_n of $1/\tilde{A}(z/\gamma)$:

$$g_n = \sum_i c_i f_{n-i} \qquad (8)$$

Using (7) and (8), it comes

$$P = \sum_{n=0}^{N-1} (x_n \sum_{i=0}^{N-1} c_i f_{n-i}) = \sum_{i=0}^{N-1} (c_i \sum_{n=0}^{N-1} x_n f_{n-i})$$

or

$$P = \sum_{i=0}^{N-1} c_i d_i \qquad (9)$$

with

$$d_i = \sum_{n=0}^{N-1} x_n f_{n-i} \qquad (10)$$

Considering the temporally reversed sequences x_n' and d_n' defined by $x_n' = x_{N-n}$ and $d_n' = d_{N-n}$, equation (10) is equivalent to

$$d_n' = \sum_i x_i' f_{n-i} \qquad (11)$$

The sequence d_n' appears in equation (11) as the convolution product between x_n' and the impulse response of $1/\tilde{A}(z/\gamma)$.Thus it can be computed as the response of $1/\tilde{A}(z/\gamma)$ to the sequence x_n', as shown in figure 4b.

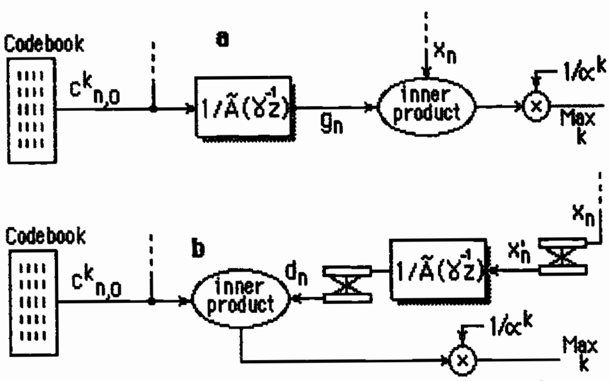

Figure 4: the "backward filtering" principle

The sequence x_n is temporally reversed, inverse filtered through $1/\tilde{A}(z/\gamma)$ and temporally reversed again, giving sequence d_n, which does not depend on the currently tested codeword. This procedure is called the "backward filtering". Now the search procedure consists of maximizing the weighted inner product P_w, computed as the inner product between c_n and d_n, scaled with $1/\alpha_k$.

The backward filtering (figure 4b) is strictly equivalent to the former procedure (figure 4a) and consequently the results are unchanged. But instead of one filtering per codeword, only one filtering for all the codewords is required here, which cuts the computational load by a factor approximately equal to the linear prediction order M.

Hierarchical vector quantization is used to quantize the linear prediction filter A(z) for each frame of signal. A hierarchical codebook of 1024 filters is generated from a large training set, using an efficient design algorithm [4]. For an equivalent quality,vector quantization requires a much lower bit rate than scalar quantization. Moreover, the hierarchical (binary tree) structure in the codebook enables fast search procedures during the quantization process.

Apart from these significant advantages, vector quantization of the LPC filters has an even greater interest in a CELP environment. As a matter of fact, the search procedure described above involves the computing of the coefficients $1/\alpha_k$, which only depends on the codeword and on the linear prediction filter. Using vector quantization, the number of possible LPC filters is finite and reasonably small (≤ 1024). Thus all the possible values of $1/\alpha_{k,i}$ can be pre-computed and stored. If the size K of the codebook containing the innovation sequences is not too large this storing will require reasonable amounts of memory.

Algebraic codes, short innovation sequences and tree coding

Smaller codebooks can be used with an equivalent bit rate if the block size N is reduced. This has the advantage of significantly decreasing the computational load, although the coding procedure is less efficient with a small block size. A tree coding structure, for instance with the ML algorithm, can be used to reduce this drawback. During the search procedure the L "best" codewords for the current block are retained as hypothesis for the next block. This increases the amount of computations proportionally with L, but the algorithm is already very efficient for L=2 . An additional delay is introduced, so the pitch period T must be greater than 2N. Another efficient way of taking into account the influence of a codeword on the next block is to compute a longer sequence of signal d_n by "backward filtering". The increase in computation is minimal.

The whole coding scheme, including the ML algorithm, is represented in figure 5.

In stochastic excited LPC the innovation codebook is populated with iid Gaussian samples. In fact, as pointed out in [6], at rates half bit and below sequences of +1 and −1 are just as good.

Algebraic structure from code theory can be used to provide efficient innovation codebooks. In particular

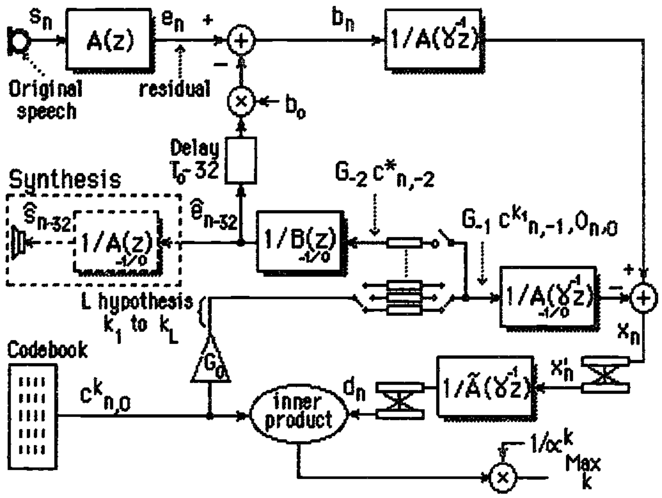

Figure 5: complete coding scheme incorporating backward filtering and tree coding.

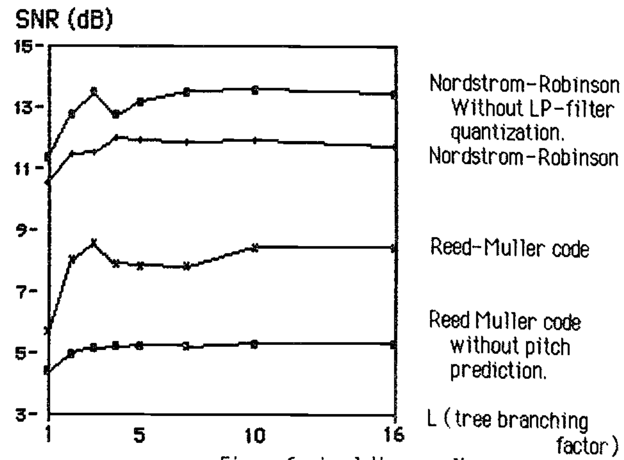

Figure 6: simulation results.

the performances for block size N=16 samples using the Reed-Muller code [16,5,8] (5/16 bits per sample) and the nonlinear Nordstrom-Robinson code (16,8,6) (1/2 bits per sample) are reported below. These codebooks lead to global rates as low as 2400 and 4800 bps respectively.

The great advantage of algebraic codes versus stochastic codebooks is that fast algorithms can be used to compute the inner products during the search procedure. For the above codes, fast algorithms based on the Fast Hadamard Transform (FHT) are detailed in [3]. It provides an additional cut in the computation, which enables the use of higher rate codebooks with a related increase in the synthetic speech quality.

Experimental results

The CELP coding technique, as represented in figure 5, has been tested both with simulations and real time implementation. Extensive simulation work has been done to determine the respective influence of the different parameters and the performances of different codes. Pitch prediction is clearly very efficient, while the degradation caused by the quantization of the LPC filters shows that the LPC-filter codebook must be designed with great care. With the Reed-Muller code the synthetic speech quality is quite fair, and it is very high with the Nordstrom-Robinson code (which requires a higher bit rate). The ML-tree coding algorithm is useful at small branching factor L=2 or 3. Figure 6 gives in a concice fashion the impact of rate (5/16 or 1/2), pitch prediction, LP quantization and the tree branching factor on the Signal to Noise Ratio (SNR) in dB.

With all the transformations described above, combined with the use of algebraic codes, the CELP coding procedure can be implemented on current DSP chips. A real time implementation, using the Reed-Muller code, has been performed on a single TMS 320-10. For a global rate of 4800 bps, the synthetic speech quality is quite fair. Examples of synthetic speech obtained with this method will be played at the conference.

Conclusion

Several transformations of the basic CELP scheme, a new technique called "backward filtering" combined with the vector quantization of the LPC filters and the use of algebraic codes have been presented in this paper. This leads to a very efficient search procedure which enables real time implementation with high synthetic speech quality at 4800 bps.

Our current work focuses on the optimization of the LP codebook, testing new algebraic codes and combinations of codes. Real time implementations, with TMS 320-20 and TMS 320C25 are underway.

Acknowledgements: The authors wish to express their thanks to Claude Laflamme for many valuable suggestions and for the real time implementation work.

References

[1] M.R. Schroeder and B.S. Atal, "Code-excited linear prediction(CELP): high-quality speech at very low bit rates", in Proc. Int. Conf. on Acoustics, Speech and Signal Proc., March 1985.
[2] I.M. Trancoso and B.S. Atal, "Efficient procedures for finding the optimum innovation in stochastic coders", in Proc. Int. Conf. on Acoustics, Speech and Signal Proc., paper nº 44.5, April 1986.
[3] J.P. Adoul, C. Lamblin, "A comparison of some algebraic structures for CELP coding of speech", ICASSP, April 1987.
[4] J.P. Adoul, 'Speech coding algorithms and vector quantization', chapter 3 of "Advanced digital communications and signal processing", K. Feher editor, Prentice Hall, NJ (1987).
[5] J.P. Adoul, C. Lamblin, A. Leguyader, "Baseband speech coding at 2400 bps using Spherical Vector Quantization", ICASSP, April 1984.
[6] M.R. Schroeder, N.J. Sloane, "New permutation codes with Hadamard unscrambling", IEEE Int. Symp. on Inf. Theory, June 1985.