Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (San Diego, CA), pp. 384-387, March 1984

EFFICIENT COMPUTATION AND ENCODING OF THE MULTIPULSE EXCITATION FOR LPC

M. Berouti[1], *H. Garten*[1], *P. Kabal*[2,3], *P. Mermelstein*[1,2,3]

[1]*Bell-Northern Research*
Verdun, Quebec H3E 1H6

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

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

Abstract

This paper discusses the analysis techniques used to derive the excitation waveform for multipulse coding of speech. A computationally efficient formulation is derived for both covariance and correlation type analyses. These methods differ in the way block edges are treated. Several methods for pulse amplitude and position determination are given, ranging from a purely sequential one to one which reoptimizes pulse amplitudes at each step. It is shown that the reoptimization scheme has a nested structure that allows a reduction in the computations. An efficient method for pulse position coding is given. This method can essentially achieve the entropy limit for randomly placed pulses. Experimental results are given for typical configurations including computational requirements and speech quality assessments.

1. Introduction

In this paper we discuss our work on high quality speech coding at bit-rates in the range 4.8 to 16 kb/s. The main objective of our work is the cost-effective real-time hardware realization of a speech codec using general purpose digital signal processing integrated circuits (DSP chips). The algorithm chosen for evaluation is the multipulse speech coding approach introduced by Atal and Remde [1]. The approach was originally viewed as a technique to provide an added improvement to the tradional LPC vocoder. The improvement is accomplished by better modelling of the excitation waveform used for synthesis. Ideally, the excitation waveform should by the linear prediction residual. In LPC, the excitation used is a train of periodic pulses separated by the pitch period for voiced sounds, or a noise waveform for unvoiced sounds. To improve upon this model, the multipulse approach models the ideal excitation by non-zero samples (pulses) separated by realtively long sequences of zero-values samples. The multipulse model is complete when the sampling instants (temporal positions) and amplitudes of the pulses are known in a given interval of time. The pulse amplitudes and positions are chosen such that the reconstructed speech signal satisfies a fidelity criterion relative to the original input speech.

We present a computationally efficient approach for the derivation of the multipulse excitation together with techniques to optimize the pulse amplitudes. We also described in detail an efficient means of encoding the pulse positions. Computational complexity and speech quality assessments are also given.

2. Multipulse Excited Speech Coding

The multipulse analysis generates the linear prediction residual signal following conventional practice for adaptive predictive coding schemes. The speech is divided into frames of samples, with a new set of analysis (and synthesis) filter coefficients being produced for each frame. The purpose of the multipulse analysis is to replace or model the residual signal by a sequence of pulses. The pulses are used as input to the LPC all-pole synthesis filter $H(z)$. The difference between the reconstructed output speech and the original input is to be minimized. The objective measure of performance used is the weighted mean-squared error. As suggested by Atal and Remde [1], the weighting is accomplished by means of the filter:

$$W(z) = \frac{H(\gamma z)}{H(z)} \tag{1}$$

where γ is a bandwidth expansion factor such that the weighting allows the SNR to be lower in the formant regions since the noise in such regions is better masked by the speech signal.

Multipulse analysis, viewed as a residual modelling process, can be represented as shown in Fig. 1, where the weighting and synthesis filters have been combined to produce a bandwidth expanded synthesis filter, $H(\gamma z)$. The causal impulse response of this filter will be denoted $\{h_n\}$. The residual passing through this filter produces the "desired" signal,

$$d_n = \sum_{k=-\infty}^{\infty} r_k h_{n-k}\ , \tag{2}$$

where $\{r_n\}$ is the residual. Note that the modified synthesis filter is an artifice used only for the determination of the multipulse excitation.

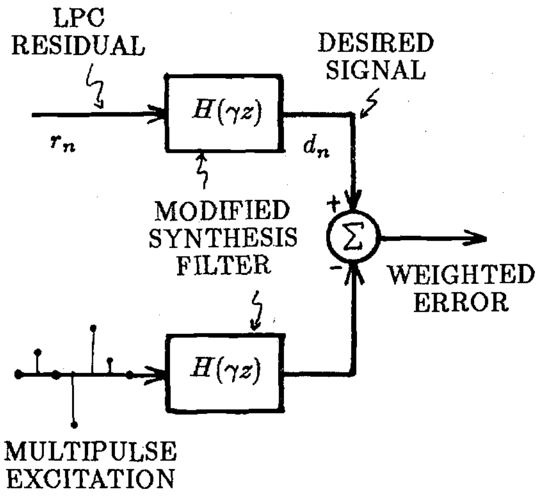

Fig. 1 Residual modelling

2.1 Sequential Pulse Placement

The analysis frame is further broken into blocks of N samples for the purpose of determining the multipulse excitation. N_p pulses are placed in each block.

Consider first placing a single pulse at position m. The resulting weighted mean-squared error for the block is

$$\overline{e^2} = \sum_n (d_n - A_m h_{n-m})^2 \,. \tag{3}$$

The limits of this and later sums are left unspecified for the moment so as to accommodate both a covariance and an autocorrelation type of analysis. The optimal pulse amplitude is obtained by differentiating this expression with respect to A_m to give

$$\hat{A}_m = \frac{\alpha_m}{\phi_{mm}} \,, \tag{4}$$

where α_m is a vector of cross-correlation terms,

$$\alpha_m = \sum_n d_n h_{n-m} \,, \tag{5}$$

and ϕ_{ij} is a matrix of correlation terms,

$$\phi_{ij} = \sum_n h_{n-i} h_{n-j} \,. \tag{6}$$

A naive approach would be to try exhaustively each possible location m within N, compute the amplitude from (4) and evaluate the error from (3). Instead we develop a computationally efficient pulse determination procedure by substituting the optimal value for a pulse at the m^{th} location into the expression for the mean-squared error which gives an expression that depends only on the position of the pulse:

$$\overline{e^2} = \sum_n d_n^2 - \frac{\alpha_m^2}{\phi_{mm}} \,. \tag{7}$$

To minimize the error given in (7), it can be seen that the best position for a single pulse is that value of m which maximizes α_m^2/ϕ_{mm}.

The effect of the pulse that has just been placed can be removed from $\{d_n\}$ to give a new sequence $\{d'_n\}$ to be used to determine the optimal position for the next pulse,

$$d'_n = d_n - \hat{A}_{\hat{m}} h_{n-\hat{m}} \,. \tag{8}$$

It is more convenient to substitute the new value of d'_n directly into (5) to give an update formula for the cross-correlation:

$$\alpha'_m = \alpha_m - \hat{A}_{\hat{m}} \phi_{\hat{m}m} \,. \tag{9}$$

The cross-correlation maximization approach given above is computationally more efficient than the exhaustive search by at least an order of magnitude.

2.2 Block Edge Effects

If the limits of the mean-squared error sum span N samples, we get a covariance type of analysis in which no assumptions are made about the signal outside the block. This method discounts the effect of that part of the impulse response that spills outside the block. In this case, ϕ_{ij} given in (6) is a matrix of covariance terms.

The autocorrelation form of analysis is obtained by allowing the limits of the error sum to extend from $-\infty$ to $+\infty$. The residual $\{r_n\}$ is assumed windowed such that it is zero outside the block. This method of analysis takes into account that part of the impulse response that spills outside the block. The autocorrelation terms in (6) form a Töplitz matrix, requiring only the first row of the matrix of values $\phi_{i}j$ to be determined. The autocorrelation formulation also simplifies the search for the best pulse position: ϕ_{mm} does not depend on m and (7) is minimized by placing a pulse in the position which maximizes $|\alpha_m|$.

2.3 Jointly Optimal Pulse Amplitudes

The pulse amplitude determination part of the procedure can be modified to give the jointly optimal pulse amplitudes. The mean-squared error after having placed n_p pulses at positions m_i is:

$$\overline{e^2} = \sum_n (d_n - \sum_{i=1}^{n_p} A_{m_i} h_{n-m_i})^2 , \tag{10}$$

Differentiating this expression with respect to the amplitudes, A_{m_i}, gives the following set of simultaneous equations to be solved,

$$\begin{bmatrix} \phi_{m_1 m_1} & \phi_{m_1 m_2} & \cdots & \phi_{m_1 m_{n_p}} \\ \phi_{m_2, m_1} & \phi_{m_2, m_2} & \cdots & \phi_{m_2 m_{n_p}} \\ \vdots & \vdots & & \vdots \\ \phi_{m_{n_p} m_1} & \phi_{m_{n_p} m_2} & \cdots & \phi_{m_{n_p} m_{n_p}} \end{bmatrix} \begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \\ \vdots \\ \hat{A}_{n_p} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n_p} \end{bmatrix} . \tag{11}$$

The matrix of correlation terms is formed by selecting rows and columns from the matrix with elements ϕ_{ij}. The above set of equations can be solved using a Cholesky decomposition of the correlation matrix.

Two forms of pulse amplitude reoptimization can be used. After all pulse positions have been determined using the sequential procedure, the pulse amplitudes can be reoptimized using (11) for all N_p pulses. A fully optimal scheme can be employed to reoptimize the pulse amplitudes at each step of the sequential procedure. The computational burden of the matrix solution part of the sequential optimal scheme is not as onerous as might at first appear. The Cholesky decomposition of the correlation matrix expresses the matrix as the product of a lower triangular matrix and its transpose. The results of the decomposition for n_p pulses is computed by adding one row to the triangular matrix determined for $n_p - 1$ pulses without disturbing the previously determined portion. The first stage of back substitution in the solution phase also works in this manner. A new element is computed without disturbing previously determined elements. Only the last stage of back substitution need be done in full as new pulses are added. The major computational load in the sequential optimal scheme is actually due to the complete updating of the cross-correlation (9) necessary at each stage of the computation.

3. Pulse Position Coding

Several methods can be used to code the pulse positions. The most efficient of these is a combinatorial coding scheme. Given N_p pulses, there are "N-choose-Np" $\binom{N}{N_p}$ possible ways to place the pulses in N different locations. The following algorithm generates an index representing one of the $\binom{N}{N_p}$ possible choices.

Consider a binary N-vector of weight N_p. Ones mark the positions of the pulses, while zeros mark the empty positions. Index the vectors from 0 to $\binom{N}{N_p} - 1$, with the vectors arranged in lexicographic order. The coding operation is accomplished by traversing a trellis whose nodes are represented by combinatorial terms. In doing so, the data vector is searched from the most significant position to the least significant position, n running from $N-1$ to 0. Whenever a one is encountered in position n, the index value is increased by $\binom{n-1}{m}$, where m is the number of ones yet to be found. The resulting value is the index to the lexicographic ordering of the pulse position vectors. The complementary problem of determining a data vector containing pulse positions can be solved by comparing the index value with the same combinatorial values to determine if a one or zero should be placed in a given position.

This coding procedure requires N_p additions and N_p references to combinatorial terms. Decoding requires N references to the combinatorial terms, but still only N_p additions. The combinatorial terms can be stored in a table of less than $N_p N$ values. Alternately, several algorithms are available to generate the required terms recursively.

The arithmetic to determine the index and to determine the pulse positions from an index must be done on quantities represented with $\lceil \log_2 \binom{N}{N_p} \rceil$ bits. Modifications to the basic scheme are possible to reduce the precision of the required arithmetic at a small loss in coding efficiency.

4. Pulse Amplitude Coding

The pulse amplitudes can be coded efficiently by normalizing them by the root mean-square pulse amplitude. After scaling, the pulse amplitudes have a bimodal distribution. An optimal minimum mean-squared error quantizer was designed for this distribution. For moderate to low bit rates, 4 bits were allocated for the transmission of the normalized pulse amplitudes. The rms is transmitted using 6 bits.

5. Computational Complexity

In order to evaluate the hardware requirements for a multipulse coder, the number of arithmetic operations ($N_{\mathrm{op}}^{\times}$ for multiplications, N_{op}^{+} for additions and $N_{\mathrm{op}}^{\div}$ for divisions) was determined. The data access and control overhead has not been quantified. This overhead depends on the instruction set of a particular processor and on the particular implementational techniques employed.

5.1 Pulse Correlation

Because of the bandwidth expansion factor, the impulse response of the bandwidth expanded filter dies off relatively quickly and can be truncated. For the covariance form of multipulse analysis, computational savings may be realized by calculating the correlation terms defined in (6) recursively,

$$\phi_{i-1,j-1} = \phi_{ij} + h_{N-i}h_{N-j} \,. \qquad (12)$$

For the autocorrelation form of analysis, the correlation terms can be generated by filtering $\{h_{-n}\}$ using the recursive form of the bandwidth expanded synthesis filter.

Two forms can be used to calculate the initial cross-correlation—either a direct computation using (5) or by filtering $\{d_{-n}\}$ using the all-pole form of the bandwidth expanded filter. The cross-correlation terms are updated using (9).

5.2 Pulse Locations and Amplitudes

The search for the best pulse location involves comparisons which are counted as additions. The pulse amplitude determination can take on one of three forms—the sequential approach (no reoptimization), reoptimization after all the pulse positions have been determined or reoptimization as the position of each new pulse is determined. The optimization uses a modified form of the Cholesky decomposition technique which avoids square root operations at the expense of having to store the full coefficient matrix (N_p by N_p).

5.3 Example

Consider a coder operating at 16 kb/s. For this configuration, we use a frame length of 192 samples, divided into blocks of 32 samples. Eight pulses are placed in each block. An eighth order LPC analysis is used to derive the residual.

With the purely sequential approach to pulse amplitude determination, the total operation count per analysis frame is

$$N_{\text{op}}^{\times} = \begin{cases} 11\,474 & \text{(covariance form)} \\ 9\,900 & \text{(autocorrelation form)} \end{cases}$$

$$N_{\text{op}}^{+} = \begin{cases} 11\,318 & \text{(covariance form)} \\ 11\,252 & \text{(autocorrelation form)} \end{cases}$$

$$N_{\text{op}}^{\div} = \begin{cases} 1\,544 & \text{(covariance form)} \\ 56 & \text{(autocorrelation form)} \end{cases}$$

The autocorrelation approach requires significantly less computation, particularly with respect to divisions. Reoptimization of the pulse amplitudes at the end adds about 8% to the operations count, while the full sequential optimization option adds about 35% to the operations count. It should be noted that the residual generation part of the multipulse algorithm, common to all LPC-based schemes, accounts for one third of the computational load given above. For speech sampled at 8 kHz, the coder requires 413 000 multiplies and 469 000 adds per second. This rate is easily sustained in the present generation of digital signal processing chips.

The decoder is considerably simpler in structure than the coder. It merely forms a pulse train to excite the synthesis filter. For speech sampled at 8 kHz, the decoder requires 68 000 multiplies and 74 000 adds per second.

6. Experimental Results

6.1 Covariance vs. Autocorrelation Formulation

When tested with speech inputs, the covariance approach tends to give very slightly better overall SNR values. However, listening tests indicate that differences are small and not consistently in favour of one method or the other. For sine wave inputs, by contrast, the differences are more marked. The covariance method tends to give a significantly better SNR.

6.2 Pulse Amplitude Optimization

The full sequential optimization option gives better SNR values than the purely sequential method. The perceived improvement for practical configurations is less than the increase in SNR would indicate, being small at best. Much of the improvement can be obtained with the single reoptimization stage after all pulse positions have been determined. This strategy avoids gross pulse amplitude errors due to the ignoring of the pulse interactions.

6.3 Subjective Test Results

The coder speech performance at 4.8 kb/s (2 pulses every 4 ms), 9.6 kb/s (4 pulses every 4 ms) and 16 kb/s (8 pulses every 4ms) was compared to μ-law PCM coding of speech. Formal subjective evaluations are being carried out using 10 naive subjects and 4 test sentences. The speech material was recorded under quiet conditions. Each rate is presented to the listeners 24 times. Informal results show that multipulse coding at 4.8, 9.6 and 16 kb/s is equivalent in quality to PCM speech coded with 4 bits, 5 bits and 7 bits per sample respectively. The results from the formal test will be given at the conference.

References

1. B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rate", *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Paris, France, 1982, pp. 614–617.