A Sequential Search Algorithm with Signal-Selected Pulse Amplitudes

Lucas M. J. Barbosa and Luís G. Meloni
Universidade Estadual de Campinas, Campinas SP, Brazil

Abstract— **With the objective of reducing computational complexity in algebraic code-excited linear predictive (ACELP) coders, this paper describes the use of a low-complexity sequential search algorithm with signal-selected pulse amplitudes. The described algorithm was inserted in both the ETSI GSM-AMR and ITU-T G.729 codecs, and when compared to some standard algorithms it showed to be quite efficient while causing only a slight degradation in the voice quality.**

I. INTRODUCTION

IT is well known that code-excited linear predictive (CELP) coders can achieve good voice quality at very low bit rates [1]. The impressive performance of CELP codecs however relies on the execution of complex and time-consuming computations that, in many cases, prevents coder implementations from achieving the desired performance. It is therefore desirable to pursue new ways of performing coding tasks more efficiently.

Analyzing the different modules of a CELP coder, it becomes clear that the fixed codebook search plays an important role in the coder computational complexity [2]. Although CELP coder standards already specify sub-optimal solutions for the fixed codebook search problem, the standard algorithms are still responsible for great part of the CELP coder complexity.

Since any simplification in the search procedure generally leads to degradation in speech quality, it is an interesting task to find forms of minimizing this degradation while reducing the computation complexity as much as possible. However, any simplification in the fixed codebook search must be carried out carefully since the innovation sequence has a great influence in the reconstructed signal, despite its apparent residual contribution [7].

Algebraic Code-Excited Linear Predictive (ACELP) coders make use of sparse excitation codebooks, where the search task is reduced to the search of the best combination of signs and positions to the pulses. However, even in algebraic codebooks an optimal solution for the fixed codebook search problem would be unpractical. Moreover, it would be inefficient since current CELP coders with sub-optimal algorithms can achieve voice quality quite as good the achieved with an exhaustive search procedure.

ACELP coders have been adopted in many speech codecs, such as the ITU-T G.729 [3] and the GSM Adaptive Multi-Rate (GSM-AMR) [6] coders. Also, several search schemes have been proposed to reduce the complexity of the fixed codebook search in such coders [4,8,9].

In this work the usage of a simple sequential search with signal-selected pulse amplitudes is described. The algorithm has been tested in both the ITU-T G.729 and the ETSI GSM Adaptive Multi-Rate (GSM-AMR) reference codecs. Although the main focus was on the complexity reduction (what has been indeed achieved), the algorithm maintains a good voice quality in a PSQM analysis while compared to others low complexity algorithms.

In next section we present a brief review of the ACELP search. The sequential search algorithm with signal-selected pulse amplitudes is described in Section III, while section IV treats of the insertion of the algorithm in the GSM Adaptive Multi-Rate and in the G.729 is described. There the algorithm is analyzed in its complexity and performance in comparison to some standard algorithms.

II. ACELP CODEBOOK SEARCH

The innovations search task in ACELP coders corresponds to the search of the best combination of signs and positions for the pulses. The number of pulses in ACELP usual coders varies from 2 (e. g. GSM-AMR at the lower rates), to 10 (e. g. GSM Enhanced Full Rate (EFR) speech transcoder [5]).

The general CELP search is performed with the objective of minimizing the mean-squared error between a target signal and a reconstructed signal. Mathematically, to choose the optimal codevector i corresponds to minimize the error E_i given by

$$E_i = \|\mathbf{u} - \eta_i \mathbf{H}\mathbf{c}_i\|^2 = \|\mathbf{u} - \eta_i \mathbf{q}_i\|^2 \quad (1)$$

where $\mathbf{u}$ is the target vector, $\mathbf{c}_i$ is the codevector, $\mathbf{q}_i$ represents the filtered codebook innovation, and η_i is a gain factor. All vectors are of size K, where K is the target signal length. Also in (1), $\mathbf{H}$ is a $K \times K$ filtering matrix called the impulse response matrix. Actually, $\mathbf{H}$ is a lower triangular Toepliz matrix given by $H(i,j) = h(i-j)$, where $h(n)$ is the impulse response of the weighted synthesis filter.

With some mathematical manipulation it can be shown that the optimum codevector is the one that maximizes the expression:

$$\tau_i = \frac{C_i^2}{\sigma_{\mathbf{q}_i}^2} = \frac{(\mathbf{u}^T\mathbf{H}\mathbf{c}_i)^2}{\mathbf{c}_i^T\mathbf{H}^T\mathbf{H}\mathbf{c}_i} = \frac{(\mathbf{d}^T\mathbf{c}_i)^2}{\mathbf{c}_i^T\mathbf{\Phi}\mathbf{c}_i} \quad (2)$$

where $\mathbf{d} = \mathbf{H}^T\mathbf{u}$ represents the correlation between the impulse response and the target vector, and $\mathbf{\Phi} = \mathbf{H}^T\mathbf{H}$ is called the covariance matrix of the impulse response.

Since the covariance matrix $\mathbf{\Phi}$ and the correlation $\mathbf{d}$ are the same for every codevector i, they should be calculated only once, and prior to the codebook search. Then, based on equation (2), an optimal search algorithm should calculate the factor τ_i for each codevector $\mathbf{c}_i$ in the codebook. The complexity of the fixed codebook search problem therefore relies in the number of

Lucas M. J. Barbosa and Luís G. Meloni are with the Departamento de Comunicações, Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, SP, Brazil, 13083-970, Phone: +55 19 3788-3818, Fax +55 19 3289-1395.Emails:{lucasbar, meloni}@decom.fee.unicamp.br

possible codevectors $\mathbf{c}_i$. At this point, the innovation sparseness of algebraic multipulse codebooks plays an important role in the simplification of equation (2). In algebraic multipulse codebooks the codevectors $\mathbf{c}_i$ only have a certain number of values (pulses) different from zero. So, the terms C_i and $\sigma^2_{q_i}$ may be simplified to:

$$C_i = \sum_{k=0}^{M-1} \alpha_k d(m_k) \tag{3}$$

$$\sigma^2_{q_i} = \sum_{k=0}^{M-1} \phi(m_k, m_k) + 2 \sum_{k=0}^{M-2} \sum_{k \neq \ell = 0}^{M-1} \alpha_k \alpha_\ell \phi(m_k, m_\ell) \tag{4}$$

where M represents the number of pulses and m_k their positions in the codevector $\mathbf{c}_i$. The values α_k denote the pulses amplitudes and are restricted to be +1 or -1.

As can be easily seen, equations (3) and (4) may be rewritten as

$$\begin{aligned} C_i &= \alpha_0 d(m_0) + \alpha_1 d(m_1) + \alpha_2 d(m_2) + \cdots && (5)\\ \sigma^2_{q_i} &= \phi(m_0, m_0) \\ &+ \phi(m_1, m_1) + 2\alpha_0\alpha_1\phi(m_0, m_1) && (6)\\ &+ \phi(m_2, m_2) + 2\alpha_0\alpha_2\phi(m_0, m_2) + 2\alpha_1\alpha_2\phi(m_1, m_2) \\ &+ \cdots \end{aligned}$$

The analysis of (5) and (6) makes clear that C_i and $\sigma^2_{q_i}$ may be more efficiently calculated through a nested loop scheme, in which each loop corresponds to a pulse position m_i, and the contribution of the pulse is added in the loop.

III. Sequential Search Algorithm with Signal Selected Pulse Amplitudes

The maximization of τ_i, as given in (2), involves the calculations of C_i and $\sigma^2_{q_i}$ for each combination of pulse positions and amplitudes (signs). In many coders [3-6] a simplification in the evaluation of τ_i is used, in which a sign is assigned to each position of the codevector $\mathbf{c}_i$ prior to the search, what reduces the ACELP search to finding the best positions for the pulses. Considering the pulse signs to be pre-selected by a vector $\mathbf{b}$, equations (3) and (4) become:

$$C_i = \sum_{k=0}^{M-1} d'(m_k) \tag{7}$$

$$\sigma^2_{q_i} = \sum_{k=0}^{M-1} \phi'(m_k, m_k) + 2 \sum_{k=0}^{M-2} \sum_{k \neq \ell = 0}^{M-1} \phi'(m_k, m_\ell) \tag{8}$$

where $d'(i) = d(i)\,\mathrm{sign}(b(i))$. Here, $\mathbf{\Phi}'(i,j)$ are the terms of $\mathbf{\Phi}'$, the covariance matrix of the impulse response taking the signs in consideration., and they are given by $\mathbf{\Phi}(i,j)\,\mathrm{sign}(b(i))\,\mathrm{sign}(b(j))$

In many standard search algorithms, such as the depth-first tree search [4] and the focused search [3], the signs pre-selection is done by the vector $\mathbf{d}$ itself:

$$b(n) = d(n) \tag{9}$$

Therefore C_i is simply given by

$$C_i = \sum_{k=0}^{M-1} |d(m_k)| \tag{10}$$

In other coders, such as the GSM Enhanced Full Rate and the GSM Adaptive Multi-Rate (at the higher rates), the vector $\mathbf{b}$ is composed by the sum of the normalized vector $\mathbf{d}$ and a normalized long-term prediction residual [5-6].

$$b(n) = \frac{\mathrm{res}_{LTP}(n)}{\sqrt{\sum_{i=0}^{M-1} \mathrm{res}^2_{LTP}(i)}} + \frac{d(n)}{\sqrt{\sum_{i=0}^{M-1} d^2(i)}} \tag{11}$$

After having pre-selected the pulse amplitudes, the codebook search consists in finding a combination of pulse positions for which τ_i is maximized, where C_i and $\sigma^2_{q_i}$ are given by (7) and (8).

To avoid the evaluation of τ_i for all possible pulse position combinations, the sequential algorithm searches the pulse positions one after another, simplifying the search procedure, but also introducing some imprecision in the maximization of (2). The algorithm starts choosing the first pulse by maximizing the term $d'^2(m_i)/\Phi'(m_i, m_i)$, $i = 0, 1, \ldots, M$ (this is actually the best position for a 1-pulse codebook). The second pulse is found checking the remaining pulse positions (according to the codebook structure considered) and testing their contributions to C_i and $\sigma^2_{q_i}$. The other pulses are found in the same way.

Prior to the search process, the matrix $\mathbf{\Phi}$ and the vector $\mathbf{d}'$ must be calculated. Note that the computational complexity is lower if the calculation of $\mathbf{\Phi}'$ is done inside the search algorithm, since not all terms of $\mathbf{\Phi}'$ will be used.

In fact, the sequential search algorithm with signal-selected pulse amplitudes is a multistage search procedure somewhat similar to the joint position and amplitude search algorithm [8], but with signs pre-selection.

IV. Computational Complexity and Performance Evaluation

In this section, we present results of performance and complexity measurements of the sequential search algorithm described previously. The algorithm is compared to some standard algorithms, such as the focused and the depth-first tree (DFTS) [4] search. For being also a multistage process, the joint position and amplitude search has been included in the comparisons. Finally, to be taken as a reference point, a position-exhaustive search algorithm has also been implemented. The performance tests were done by the insertion of the sequential search algorithm with signal-selected pulse amplitudes in both, the ITU-T G.729 and the ETSI GSM Adaptive Multi-Rate codecs

The computational complexity of the algorithms involved in the comparisons has been measured by an implementation of the G.729 codec in the Texas Instruments fixed-point TMS320C6202 digital signal processor [10]. Although such DSP supports a high level of parallelism, in the sense of minimizing the influence of hardware specificities in the results, all parallelism optimizations have been disabled during the tests.

A total of 110 speech files from region DR1 of the "Telephone Network Acoustic-Phonetic Continuous Speech Corpus"

NTIMIT [11] has been used for the objective performance measurements. The NTIMIT database consists of signals obtained from actual telephone lines under a variety of line conditions. The files of region DR1 constitute 5min 52s of speech, corresponding to 70230 subframes.

A. Implementation in the ITU-T G.729 codec

The G.729 codec has a 4-pulse algebraic codebook, represented in Table I. The 40 positions in a subframe are divided into 4 tracks, where each track contains one pulse.

TABLE I

POSSIBLE POSITIONS OF INDIVIDUAL PULSES IN THE G.729 CODEC

Track	Pulse	Positions							
1	i_0	0	5	10	15	20	25	30	35
2	i_1	1	6	11	16	21	26	31	36
3	i_2	2	7	12	17	22	27	32	37
4	i_3	3	8	13	18	23	28	33	38
		4	9	14	19	24	29	34	39

In the G.729 implementation the algorithm has been compared with the focused, the exhaustive, the joint position and amplitude, and with the depth-first tree search algorithms. In the sequential search the amplitudes selection was done according to (9). The focused search used was the native G.729 codec search algorithm, whereas the depth-first tree search was brought from the ITU-T G.729A codec, and the joint position and amplitude search was implemented as described in [8]. The position-exhaustive search implemented is not an optimal process, since it also executes a signal selection of the pulse amplitudes. In fact, it differs from the standard focused search just in the fact that it searches all combinations of pulse positions in the codebook.

TABLE II

PERFORMANCE OF THE SEARCH ALGORITHMS OVER THE NTIMIT DATABASE (REGION DR1)

Search	PSQM
Focused	1.988
Position-exhaustive	1.973
Joint	2.044
Sequential	2.068
Depth-first	2.009

Table II shows the objective measurements obtained from a PSQM analysis over the 110 speech files in the NTIMIT DR1 region. The perceptual subjective quality measurements indicate that the sequential search presents a slight degradation in the voice quality in comparison to the other search procedures. However, the implementation of the G.729 codec in the Texas Instruments fixed-point TMS32C6202 digital signal processor showed that the degradation in voice quality is followed by a considerable reduction in the computational complexity. Table III shows the number of processor cycles spent by each of the five tested algorithms. The results in Table III include the processing time spent in the amplitudes pre-selection. The sequential search consumes less than half of the processor load of the depth-first tree search, and less than one tenth of the focused search included in the G.729 codec. Also, the sequential search presents a reduction in complexity of about 15% while compared with the joint-position and amplitude search.

TABLE III

COMPLEXITY MEASUREMENT OBTAINED FROM THE FIXED-POINT IMPLEMENTATION OF THE SEARCH ALGORITHMS

Search	Number of Cycles($\times 10^3$)		
	Minimum	Mean	Maximum
Focused	63.3	144.7	186.5
Position-exhaustive	768.1	768.3	768.5
Joint	14.5	16.9	21.0
Sequential	12.4	14.5	17.4
Depth-first	35.2	35.6	36.1

B. Implementation in the GSM Adaptive Multi-Rate codec

The GSM-AMR codec is a Multi-Rate ACELP (MR-ACELP) coder that works in eight different bit rates: 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2 or 12.2kbit/s. Depending on the bit rate, the 40 positions in a subframe are divided in a different number of tracks. The GSM AMR codec makes use of six different algebraic codebook structures, each having its own number of pulses, as described in Table IV. At 7.95 and 7.40kbit/s the codec uses the same codebook structure and search procedure. The same happens at the bit rates 5.15 and 4.75kbit/s. Further information on the GSM-AMR codebook structures can be found in [6].

TABLE IV

NUMBER OF PULSES IN THE ALGEBRAIC CODEBOOK STRUCTURES OF THE GSM ADAPTIVE MULTI-RATE CODEC

Bit rate (kbit/s)	Number of pulses
12.20	10
10.20	8
7.95	4
7.40	
6.70	3
5.90	2
5.15	2
4.75	

The sequential search algorithm was implemented for each bit rate of the GSM Adaptive Multi-Rate codec, with the sign selection done according to (11). The slight differences in the implementations were only carried out with the objective of adjusting the search procedure to the different codebook structures. The algorithm was compared with each of the GSM-AMR codec reference search algorithms, as well as with the joint position and amplitude search (implemented as described in [8]).

Table V shows the objective measurements obtained from a PSQM analysis over the 110 speech samples in the NTIMIT DR1 region. The PSQM difference between the search procedures included in the GSM-AMR codec and the sequential search increases with the bit rate, varying from 0.04 to 0.12. At the lower bit rates the performance of the sequential search with signal-selected pulse amplitudes is very similar to the joint position and amplitude search. However, at the bit rates of 12.2 and 10.2kbit/s the sequential search presents a gain of 0.14 and

TABLE V
PERFORMANCE OF THE SEARCH ALGORITHMS OVER THE NTIMIT DATABASE (REGION DR1)

GSM-AMR Bit rate (kbit/s)	**PSQM**		
	GSM-AMR Standard Search	**Joint Search**	**Sequential Search**
12.20(EFR)	1.481	1.747	1.602
10.20	1.549	1.730	1.656
7.95	2.009	2.064	2.074
7.40 (DAMPS-EFR)	1.998	2.059	2.066
6.70	2.126	2.182	2.184
5.90	2.276	2.321	2.321
5.15	2.497	2.525	2.524
4.75	2.531	2.578	2.572

At 12,2 kbits the GSM Adaptative Multi-Rate codec corresponds to the GSM Enhanced Full Rate (GSM-EFR) codec, and at 7,40 kbits/s it corresponds to the Digital Advanced Mobile Phone System Enhanced Full Rate (DAMPS-EFR) codec.

0.7 respectively in the PSQM analysis in comparison to the joint search.

Although the pulse amplitudes were selected by the vector $\mathbf{b}$, calculated as in (11), no further complexity has been introduced in the coder, since the long-term prediction residual res_{LTP} is already calculated within the GSM-AMR codec. In fact, at the higher rates (12.2 and 10kbit/s), the GSM-AMR search algorithms use this residuum for the signs pre-selection.

V. CONCLUSIONS

The sequential search algorithm with signal-selected pulse amplitudes revealed to be a very low complexity search procedure that introduces only a small degradation in the voice quality when compared to standard search algorithms. It has been tested in both the ITU-T G.729 and the ETSI GSM-AMR codecs and achieved a performance quite as good as the joint position and amplitude search.

VI. ACKNOWLEDGEMENTS

The authors would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) for its support, and Miguel A. Ramírez for his great help with the joint position and amplitude search implementation in the G.729 codec.

REFERENCES

[1] M. R. Schroeder, and B. S. Atal, *Code-excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates*, Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Tampa, 1985, vol. 2, pp. 437-440.

[2] I. M. Trancoso, and B. S. Atal, *Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders*, in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Tokyo, 1986, vol. 4, pp. 2375-2378

[3] ITU-T, *Recommendation G.729, Coding of Speech at 8kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)*, March 1996.

[4] ITU-T, *Recommendation G.729 - Annex A: Reduced Complexity 8 kbit/s CS-ACELP Speech Codec*, March 1996.

[5] ETSI, *Digital Cellular Telecommunications System (Phase 2); Enhanced Full Rate (EFR) Speech Transcoding*, (GSM 06.60 version 6.0.0) ETSI EN 300 726, 1997.

[6] ETSI, *Digital Cellular Telecommunications System (Phase 2+); Adaptive Multi-Rate (AMR) Speech Transcoding*, (GSM 06.90 version 7.2.1) ETSI EN 301 704, 1998.

[7] M. A. Ramirez, *Busca de Inovações e Tratamento de Transitórios em Codificadores de Voz CELP*, Doctoral dissertation, Universidade de São Paulo, São Paulo, Brazil, 1997.

[8] M. A. Ramirez, *Joint Position and Amplitude Search of Algebraic Multipulses*, IEEE Transactions on Speech and Audio Processing, vol. 8, No 5, September 2000.

[9] N. K. Ha, *A Fast Search Method of Algebraic Codebook by Reordering the Search Sequence*, Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol.1, pp. 21-24, 1999.

[10] Texas Instruments Incorporated, *TMS320C6000 CPU and Instruction Set Reference Guide*, October 2000.

[11] C. Jankowski, A. Kalyanswamy, S. Basson and J. Spitz, *NTIMIT: A Phonetically Balanced, Continuous Speech, Telephone Bandwidth Speech Database*, Proc. ICASSP, Albuquerque, Apr. 1990.