A MULTISTAGE SEARCH OF ALGEBRAIC CELP CODEBOOKS

Miguel Arjona Ramírez and Max Gerken

Electronics Eng. Dept. - Escola Politécnica
University of São Paulo
miguel@lps.usp.br and mgk@lcs.poli.usp.br

ABSTRACT

A joint amplitude and position search procedure is proposed for searching algebraic multipulse codebooks. It is implemented within the reference G.723.1 codec as an example. This joint search method is shown to reduce down to one third the number of comparisons per subframe relative to the focused search over an extensive speech database. An efficient implementation of the joint search is derived which incorporates backward filtering of the residual target vector and precomputation of autocorrelation elements, bringing about a reduction in complexity of one third in comparison to the focused search. The joint search performs about one thirtieth as many comparisons as the full position search.

1. INTRODUCTION

Since its inception the original CELP speech coding model has undergone a series of changes designed to reduce its complexity and improve the quality of its reconstructed speech signal [1]. A milestone in this development was the introduction of ACELP coders using codebooks with a multipulse structure [2] coupled with efficient, suboptimal search algorithms.

Efficient search procedures are necessary to keep the complexity within bounds playing an important role in the achievement of overall efficiency and may by themselves make the difference for a given application as in the case of a multimedia speech coder for digital simultaneous voice and data (DSVD) [3]. Therefore, one may expect that the availability of reduced complexity search algorithms could lead to new applications for the ACELP coder.

Furthermore, in a CELP coder the fixed codevector has a great influence on reconstructed speech quality despite its apparently residual contribution. Consequently, the improvement of fixed algebraic codebook search algorithms is essential to the development of coders with better quality/complexity ratios.

Motivated by this reasoning, this work presents an efficient search algorithm for algebraic multipulse codebooks that has been implemented within the reference G.723.1 codec [4] as an example, but is not limited to it.

This work was partially supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under Grant no. 93/0181-0 and by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under Grant no. 300521/92-8.

2. BASIC ACELP CODEBOOK SEARCH

For a given target vector u, a standard CELP search algorithm tries each codevector c_i, $i = 1, 2, \ldots, M$, in its codebook producing scaled filtered reconstruction vectors

$$\tilde{u}_i = \eta_i H c_i = \eta_i q_i, \tag{1}$$

where η_i are suitable gain factors while $q_i = Hc_i$ are filtered codevectors. The impulse response matrix H is given by $H(i,j) = h(i-j)$, where $h(n) = 0$ for $n < 0$ and $h(n)$ for $n = 0, 1, \ldots, L$ is the impulse response of the weighted synthesis filter truncated to the subframe length L. In Eq. (1) $\tilde{u}_i$ may be considered as projections of target vector u over the filtered codevectors q_i. Therefore, the best codevector is obtained selecting the corresponding minimum norm error vector between all reconstruction error vectors $\varepsilon_i = u - \tilde{u}_i$ or the corresponding maximum length projection between all projections $\tilde{u}_i$. Considering the latter selection criterion, there are some remarks that are of interest. Writing out the squared norm as a function of the weighted codevector q_i, we have

$$\|\tilde{u}_i\|^2 = C_i^2/(q_i^T q_i) \tag{2}$$

whose numerator is the square of $C_i = u^T q_i$, which is the zero-lag crosscorrelation, or simply correlation, between the weighted codevector and the target vector. Further, the correlation may be efficiently determined by a single inner product as

$$C_i = t^T c_i, \tag{3}$$

where $t = H^T u$ is the backward-filtered residual target vector.

However, really efficient computations of the squared norm (Eq. (2)) will ultimately depend on structuring or simplifying the calculation of the squared norm or energy of the weighted codevector in the denominator of Eq. (2). When the codevectors are sparse, it is very convenient to express their weighted energies as

$$q_i^T q_i = c_i^T H^T H c_i = c_i^T \Phi c_i, \tag{4}$$

where $\Phi = H^T H$ is the autocorrelation matrix of the impulse response matrix.

Multipulse excitations as used in ACELP codebooks are sparse signals (vectors) whose nonzero samples are isolated from one another. They may be described in general as

$$e = G \sum_{k=0}^{M-1} \alpha_k I(:, m_k),$$

where M is the number of pulses, α_k and m_k are pulse amplitudes and positions, respectively, whereas G is the overall excitation signal gain. $I(:, m_k)$ indicates the column m_k of the $L \times L$ identity matrix I, where L is the subframe length. Usually $M = 4$ and $L = 60$, values that will be used in the following.

As a further constraint, algebraic multipulses take on positive or negative unit values only, that is, $\alpha_k = \pm 1$. Additionally, the algebraic multipulses considered in the following [4] have each pulse position taken from interleaved sequences of equidistant pulses. Each sequence, therefore, has a different phase, which contributes with one and only one position to the excitation signal. Further, even and odd positions are kept separate in two different grids. The even grid is represented in tabular form in Table 1, where each row is a different phase.

Table 1: Even ACELP position grid.

Phase	Positions
$i_0 = 0$	0, 8, 16, 24, 32, 40, 48, 56
$i_1 = 2$	2, 10, 18, 26, 34, 42, 50, 58
$i_2 = 4$	4, 12, 20, 28, 36, 44, 52, (60)
$i_3 = 6$	6, 14, 22, 30, 38, 46, 54, (62)

3. JOINT POSITION AND AMPLITUDE SEARCH

The proposed joint position and amplitude search (JPAS) of algebraic multipulses [5] is described in this section for the fixed codebook of the ACELP coder specified on ITU-T Recommendation G.723.1 [4] at the 5.3 kbit/s rate.

Each innovation in this codebook is made out of $M = 4$ pulses and a new innovation is issued every 7.5 ms. Accordingly, reconstructed residual target vectors are composed of shifted impulse responses as

$$\tilde{u}_f = G \sum_{k=0}^{M-1} \alpha_k H(:, m_k),$$

where $H(:, m_k)$ indicates the column m_k of the impulse matrix H. This composition has motivated the JPAS procedure which selects one by one the M shifted impulse responses which jointly define a partial direction along which the residual target vector provides the greatest projection. It selects one pulse position and amplitude at each one of M iterations.

The search process starts by determining the projections of the residual target vector u_f along the partial projection directions defined by the shifted impulse responses for the first iteration. Positions searched in previous iterations are kept constant while corresponding amplitudes are jointly readjusted, resulting in the determination of the next partial projection direction.

The first iteration in a JPAS procedure selects position $i = m_0$ which maximizes the squared norm of the projection along the shifted impulse responses

$$\tau_i = \left(u_f^T H(:, i)\right)^2 / \Phi(i, i).$$

Let the filtered subcodevector selected in iteration $j-1$ be

$$q_f^{(j-1)} = \sum_{k=0}^{j-1} \alpha_k^{(j-1)} H(:, m_k). \tag{5}$$

Due to the algebraic structure of ACELP codebooks, pulse amplitudes are restricted to ± 1 so that, considering a new pulse at position m_j, only four new filtered subcodevectors are possible: $q_f^{(j)} = \pm(\pm H(:, m_j) + q_f^{(j-1)})$. Therefore, considering a new pulse at position i in iteration j, only two partial projection directions are admissible,

$$P^{(j)}(:, i) = H(:, i) + q_f^{(j-1)}, \tag{6}$$

which is called the primary partial projection direction and

$$S^{(j)}(:, i) = H(:, i) - q_f^{(j-1)}, \tag{7}$$

which defines the secondary partial projection direction[1]. In Equations (6) and (7) $P^{(j)}(:, i)$ and $S^{(j)}(:, i)$ indicate respectively the ith columns of matrices $P^{(j)}$ and $S^{(j)}$, whose columns comprise all the primary and secondary projection directions at iteration j.

In the following, the selected vectors and signs are identified after each iteration of the search process. Details about the search process follow at the end of this section.

Let the selected partial projection direction for iteration j be $f^{(j)} = H(:, m_j) + \sigma^{(j-1)} q_f^{(j-1)}$, where $\sigma^{(j-1)}$ is the sign of the selected projection direction on the plane defined by the jth pulse and the available filtered subcodevector $q_f^{(j-1)} \triangleq H c_f^{(j-1)}$. More precisely, $\sigma^{(j-1)} = 1$ if the primary projection direction is chosen and $\sigma^{(j-1)} = -1$ otherwise. The projection of the residual target vector along the selected partial projection direction $f^{(j)}$ yields the partially reconstructed target vector as $\tilde{u}_f^{(j)} = A^{(j)} f^{(j)}$, where $A^{(j)}$ is a signed gain factor whose sign is $\beta^{(j)} = \mathrm{sign}(A^{(j)})$, while its absolute value $G^{(j)} = |A^{(j)}|$ will become the gain only after the last iteration $j = M-1$. Thus, the weighted subcodevector at the end of the jth iteration is

$$q_f^{(j)} = \beta^{(j)} f^{(j)}. \tag{8}$$

Using this notation it results that the weighted subcodevectors are defined by the chosen shifted impulse responses and by the corresponding signs as $q_f^{(0)} = \beta^{(0)} H(:, m_0)$ for the first iteration and as

$$q_f^{(j)} = \beta^{(j)} H(:, m_j) + \beta^{(j)} \sum_{l=0}^{j-1} \prod_{k=l}^{j-1} \sigma^{(k)} \beta^{(k)} H(:, m_l) \tag{9}$$

for the remaining iterations.

Consequently, the selected codevector, $c_f = c_f^{(M-1)}$, can be expressed as $c_f = \sum_{j=0}^{M-1} \alpha_j^{(M-1)} I(:, m_j)$ and the resulting weighted codevector is

$$q_f = \sum_{j=0}^{M-1} \alpha_j^{(M-1)} H(:, m_j), \tag{10}$$

where

$$\alpha_l^{(j)} = \begin{cases} \beta^{(l)} & \text{if } l = j \\ \beta^{(j)} \prod_{k=l}^{j-1} \sigma^{(k)} \beta^{(k)} & \text{otherwise.} \end{cases} \tag{11}$$

Equation (11) describes the resulting pulse signs after each iteration j and can be derived by comparison of Equations (10) and (9).

Finally, in completion of the description of the JPAS algorithm, the procedure for selection of the projection directions for iteration j will be explained.

As indicated in Equations (6) and (7), the primary and secondary partial projection directions for iteration j are collected in matrices $P^{(j)}$ and $S^{(j)}$, respectively. Further, for the computation of their corresponding projections below, we will define the primary and secondary autocorrelation matrices as

$$\mathcal{P}^{(j)} = (P^{(j)})^T P^{(j)} \quad \text{and} \quad \mathcal{S}^{(j)} = (S^{(j)})^T S^{(j)}. \tag{12}$$

[1] The opposite directions could be chosen as well. However, this choice leads to simpler expressions of the ensuing equations.

For iterations $j = 1, 2, \ldots, M-1$, the squared norms of the projections along the primary and the secondary partial directions, respectively $\tau_i^{(j)}$ and $v_i^{(j)}$, are computed according to

$$\tau_i^{(j)} = \frac{\left(u_f^T P^{(j)}(:,i)\right)^2}{\mathcal{P}^{(j)}(i,i)} \quad \text{and} \quad v_i^{(j)} = \frac{\left(u_f^T S^{(j)}(:,i)\right)^2}{\mathcal{S}^{(j)}(i,i)}. \qquad (13)$$

This procedure closes with the selection of position m_j which satisfies

1. $J = \underset{i \in \mathcal{I}^{(j)}}{\text{argmax}} \left\{\tau_i^{(j)}\right\}, \quad K = \underset{i \in \mathcal{I}^{(j)}}{\text{argmax}} \left\{v_i^{(j)}\right\};$
2. $m_j = \begin{cases} J & \text{if } \tau_J^{(j)} = \max\{\tau_J^{(j)}, v_K^{(j)}\} \\ K & \text{otherwise} \end{cases}$

In addition, the set $\mathcal{I}^{(j)}$ of shift indices i for the search depends on the order j of the iteration. For the first iteration ($j = 0$), every shift index within the subframe range is searched. For the following iterations ($j = 1, 2, 3$), only the phases not yet selected are actually searched so that a row in the position grid (see Table 1) is eliminated from the search domain after each pulse selection. Furthermore, the grid parity is defined by the parity of the pulse selected in the first iteration.

Summing up, the JPAS process is a kind of orthogonal search where optimizations are carried out in planes or two-dimensional subspaces defined by each shifted impulse response whose phase has not yet been selected together with the current filtered subcodevector. It should be noticed that these optimizations do not involve orthogonalizations but rather an exhaustive test of all the admissible new partial projection directions. Therefore, the JPAS procedure is less suboptimal than the standard multipulse search .

The number of searches in a subframe distributes over six classes that may be identified by the number of subcodevectors searched along each individual search path. One should consider that each one of the six search paths may occur as one of four permutations because there are two phase tracks with 8 positions and two phase tracks with 7 positions (see Table 1). Simple reasoning shows that the number of searches lies between 146 and 154. Moreover, considering equiprobable permutations, an average of 150 searches per subframe results.

The joint search, as opposed to both the focused and the position-exhaustive searches, does not use any elements off the main diagonals of its autocorrelation matrices as shown by the denominators of Eq. (13). The diagonal elements are taken from the autocorrelation matrix Φ of the impulse response of the weighted synthesis filter during the first iteration. For the remaining iterations, they come from the autocorrelation matrices $\mathcal{P}^{(j)}$ and $\mathcal{S}^{(j)}$ of the primary and secondary projection directions, respectively. The next section describes how these elements can be efficiently computed.

4. EFFICIENT JOINT SEARCH

The autocorrelation elements involved in the joint search are dynamic values in the sense that they are computed along the search path. They are the denominators of Eq. (13), whose numerators are correlations. As shown below, it turns out that both the dynamic autocorrelations and correlations may be computed as functions of precalculated autocorrelations of the impulse response matrix and samples of the backward-filtered residual target vector, respectively.

Using Eq. (6), (7) and (8), it is possible to express the correlation in the primary and secondary partial projection directions, $P^{(j)}(:,i)$ and $S^{(j)}(:,i)$, used in Eq. (13), as follows

$$\begin{aligned} C_{p,i}^{(j)} \triangleq u_f^T P^{(j)}(:,i) &= t_f(i) + \beta^{(j-1)} C_{f,m_{j-1}}^{(j-1)} \\ C_{s,i}^{(j)} \triangleq u_f^T S^{(j)}(:,i) &= t_f(i) - \beta^{(j-1)} C_{f,m_{j-1}}^{(j-1)} \end{aligned}$$

where $C_{f,m_{j-1}}^{(j-1)} \triangleq u_f^T q_f^{(j-1)} = t_f(m_{j-1}) + \sigma^{(j-2)} \beta^{(j-2)} C_{f,m_{j-2}}^{(j-2)}$ is the correlation chosen in iteration $j-1$ using the procedure described in Section 3.

Using Eq. (6), (7), (5) and (12), it is possible to express the dynamic autocorrelations of the primary and secondary partial projection directions, $\mathcal{P}^{(j)}(:,i)$ and $\mathcal{S}^{(j)}(:,i)$, used in Eq. (13), as follows

$$\begin{aligned} \mathcal{P}^{(j)}(i,i) &= \phi(i,i) + 2\sum_{l=0}^{j-1} \alpha_l^{(j-1)} \phi(i,m_l) + E^{(j-1)} \\ \mathcal{S}^{(j)}(i,i) &= \phi(i,i) - 2\sum_{l=0}^{j-1} \alpha_l^{(j-1)} \phi(i,m_l) + E^{(j-1)} \end{aligned}$$

where $E^{(j-1)}$ is the squared norm of $q_f^{(j-1)}$, the previous weighted subcodevector.

As the joint search does compute the autocorrelation elements for both the odd and even grids, an additional number of precomputed autocorrelation elements would be necessary besides those used for the focused search. At first glance, it would seem that the number of autocorrelation elements would double, resulting in a total of 832 elements. But only the number of main diagonal autocorrelation elements doubles, as they are needed for determining the grid parity based on the pulse position selected in the first iteration as pointed out in Section 3. The remaining off-diagonal autocorrelation elements necessary for the following iterations may be computed just after the decision about the grid parity is made, and they must extend just over the concerned all-even or all-odd lag pairs. Therefore, only 32 diagonal autocorrelation elements are necessary in addition, making up for a total of 448 elements as shown in Table 5.

5. COMPLEXITY AND PERFORMANCE MEASUREMENTS

In this section, results of complexity measurements of the joint search will be presented and compared to both the focused search and the position-exhaustive search. The focused search [7] will be considered as implemented in the reference ITU-T 5.3 kbit/s G.723.1 codec [4]. This implementation includes some suboptimal simplifications in the operations determining the signs of the pulses of the chosen codevector, referred to as the signal-selected pulse amplitude approach [3]. These sign simplifications are kept up in the position-exhaustive search, which only differs from the standard focused search in that it searches all the 4096 combinations of 4 pulse positions in the grid of the chosen parity.

At the end of Section 3, the average number of comparisons per codevector search is estimated for the dynamic autocorrelation version of the JPAS procedure. This number of comparisons was also measured over the whole collection of 1680 signals in the test partition of the TIMIT

Table 2: Statistics about number of comparisons per subframe for 3 search algorithms over the test partition of the TIMIT database.

Search	Minimum	Mean	Maximum
Focused	72	458.48	2040
Position-exhaustive	4096	4096.00	4096
JPAS	146	149.62	154

Table 3: Statistics about number of autocorrelation elements per subframe for 3 search algorithms over the test partition of the TIMIT database.

Search	Minimum	Mean	Maximum
Focused	416	416.00	416
Position-exhaustive	416	416.00	416
JPAS	146	149.62	154

database [6] for a total of 1h 26min 27s of speech and 688,244 subframes. The measurements were also performed for the reference focused search and for the position-exhaustive search. The results are shown in Table 2. Overall, the average number of searches per subframe for the joint search algorithm is one third as many as that of the focused search algorithm and slightly less than 4% the number of position-exhaustive searches.

Further, the number of autocorrelation elements demanded per subframe was averaged for the three search algorithms. As shown in Table 3, the joint search displays a considerable decrease in the number of these elements.

Finally, objective performance measurements were carried out with the segmental signal-to-noise ratio (SNRSEG) and the perceptual speech quality measure (PSQM) [8] as shown in Table 4, indicating a small decrease of 0.25 dB in SNRSEG and an added distortion of approximately 0.06 units of PSQM incurred by the joint search as compared to the focused search.

For the precomputed autocorrelation version of the joint search, the measure of efficiency used is the execution time taken as a fraction of real time, which is assumed to be the duration of the speech signal under coding. Both the JPAS procedure and the focused search execute on two personal computers for the test partition of the TIMIT database. The execution times displayed in Table 5 show that the joint search takes 2/3 as long to execute as the focused search. Inside the reference ACELP coder the focused fixed search algorithm takes up over 17% of processing time while the fixed search share falls below 12% when the joint search algorithm is used instead.

It must be pointed out that for other CELP coders the complexity share of the fixed search is higher as, for instance, for the G.729 CS-ACELP reference coder, wherein the fixed search takes up about 40% of the whole complexity [3]. Work is under way in connection with the application of the proposed joint search algorithm to a wider selection of coders in the ACELP class and shall be reported on in the future.

6. CONCLUSION

The features of algebraic multipulse codebooks have been analyzed and exploited to derive the joint amplitude and position search (JPAS). This innovation search algorithm has been tested within the G.723.1 reference codec implementation, reducing the number of searches down to one third as many as the focused search and taking just two thirds as long as the focused search to execute, at the cost of a small additional signal degradation. Such a reduction in complexity may enable new applications for the ACELP coder.

Table 4: Performance of the three ACELP searches over the test partition of the TIMIT database.

Search	SNRSEG (dB)	PSQM
Focused	9.44	2.08
Position-exhaustive	9.45	2.07
JPAS	9.19	2.14

Table 5: Complexity of two ACELP searches over the test partition of the TIMIT database.

Search process	Execution time on PC1 (PC2) in % of real time	Precomputed autocorrelations
Focused	17.0% (17.7%)	416
JPAS	11.5% (11.6%)	448

PC1: Personal computer with a 133-MHz Pentium processor under the Windows 95 operating system

PC2: Personal computer with a 100-MHz Pentium processor under the Windows NT operating system

7. REFERENCES

[1] W. B. Kleijn et al., “Fast methods for the CELP speech coding algorithm”, *IEEE Tr. ASSP*, v.38, n.8, pp.1330-1342, Aug. 1990.

[2] B. S. Atal and J. R. Remde, “A new model of LPC excitation for producing natural-sounding speech at low bit rates”, in *Proc. IEEE ICASSP*, Paris, 1982, vol. 1, pp.614-617.

[3] R. Salami, C. Laflamme, B. Bessette, and J.-P. Adoul, “ITU-T G.729 Annex A: Reduced complexity 8 kb/s CS-ACELP codec for digital simultaneous voice and data”, *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 56-63, Sept. 1997.

[4] ITU-T, *Recommendation G.723.1 Dual rate speech coder for multimedia applications transmitting at 5.3 and 6.3 kbit/s,* ITU-T, Geneva, March 19, 1996.

[5] M. A. Ramírez, *Busca de Inovações e Tratamento de Transitórios em Codificadores de Voz CELP*, Dr. thesis, University of São Paulo, São Paulo, Brazil, 1997.

[6] W. M. Fisher et al., “The DARPA speech recognition research database: specifications and status”, *Proc. DARPA Speech Recognition Workshop*, Rep. No. SAIC-86/1546, Palo Alto, Feb. 1986.

[7] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, “Design and description of CS-ACELP: A toll quality 8 kb/s speech coder”, *IEEE Trans. Speech Audio Processing*, vol. 6, no. 2, pp. 116-130, Mar. 1998.

[8] ITU-T, *Recommendation P.861 Objective quality measurement of telephone band (300-3400 Hz) speech codecs,* ITU-T, Geneva, August 30, 1996.