

A LOW-COMPLEXITY SEARCH ALGORITHM FOR SPEECH CODERS WITH SPARSE EXCITATION

Miguel Arjona Ramírez and Max Gerken

Abstract - A low-complexity joint position and amplitude search algorithm is proposed for deterministic sparse codebooks to be used in code-excited linear predictive coders. The conventional multipulse search is presented as an extension of a general two-stage sequential search, leading to a classification of multistage sequential searches according to the extent that orthogonality constraints are applied throughout the search stages. After its classification, 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 approximately one third in comparison to the focused search within the G.723.1 codec. In particular, considering an extensive speech database, a floating-point version of the joint search method reduces down to one third the number of comparisons per subframe relative to the focused search. Moreover, the joint search performs about one thirtieth as many comparisons as the position-exhaustive search. Further, the complexity of a fixed-point implementation of the joint search is below one quarter that of the focused search and stands below those of the G.729A and IS-641-A coders. Listening tests indicate an equivalence in perceived quality.

Keywords: Speech coding, CELP, ACELP, algebraic multipulse, multistage searches, sparse excitation.

Resumo - Propõe-se o algoritmo de busca conjunta de posição e amplitude em dicionários determinísticos esparsos para uso em codificadores preditivos lineares excitados por código. A busca multipulso convencional é apresentada como uma extensão de uma busca seqüencial genérica de dois estágios, levando a uma classificação das buscas seqüenciais multistágio segundo a intensidade da aplicação das restrições de ortogonalidade durante os estágios da busca. Após sua classificação, deriva-se uma implementação eficiente da busca conjunta que incorpora filtragem regressiva do vetor-alvo residual e precômputo dos elementos de autocorrelação, ocasionando uma redução de complexidade de aproximadamente um terço em comparação com a busca focalizada no codec G.723.1. Em particular, usando-se uma base de dados extensa, o método de busca conjunta reduz a um terço o número de comparações por sub-bloco em relação à busca focalizada. Além disso, a busca conjunta executa aproximadamente um trigésimo do número de comparações

The authors are with the Electronic Systems Engineering and Telecommunications and Control Engineering Departments at Escola Politécnica, Universidade de São Paulo, São Paulo, SP, respectively. E-mails: miguel@lps.usp.br, mgk@lcs.poli.usp.br
Review coordinated by Max H.M. Costa (Ad Hoc Editor) and Rui Seara (Area Editor). Manuscript received 01/2000; revised 03/2001; accepted 08/2001.

da busca exaustiva em posições. Finalmente, a complexidade de uma implementação em ponto fixo da busca conjunta é menor do que um quarto da complexidade da busca focalizada e situa-se abaixo daquelas dos codificadores G.729A e IS-641-A. Em testes de audição verifica-se uma equivalência na qualidade percebida.

Palavras-chave: Codificação de voz, CELP, ACELP, multipulsos algébricos, algoritmos de busca multistágios, excitação esparsa.

1. INTRODUCTION

The original CELP speech coding model, introduced by [1, 2, 3], has ever since undergone a series of changes designed to reduce its complexity and improve the quality of its reconstructed speech signal. At first, transform domains were explored for the fixed random codebook, resulting in efficiency gains under some constraints as a trade-off between speed and memory occupation of the transformed codebook [4].

An important development was the design of sparse random codebooks and attendant search algorithms that could bring about both a quality improvement and a reduction in complexity over random Gaussian codebooks at rates around 8 kbit/s [5]. For example, sparse random codebooks were used to bring CELP coding down to the 4.8 kbit/s rate as long as a moderate decrease in quality could be allowed for [6].

Moreover, structured sparse codebooks also gave rise to a toll quality speech coder at 8 kbit/s [7, 8], which is a conjugate-structure algebraic (CS-ACELP) coder. In this case, the codebook has a multipulse structure [9] that is coupled with an efficient, suboptimal CELP search algorithm. By the way, both the multipulse and the algebraic CELP (ACELP) [10] structures had been previously used at higher bit rates and complexities. Other developments can be found in [15, 16] that present recent reviews and advances in multipulse coders.

It should be stressed that the search algorithm associated to the structured sparse codebook plays an important role in the achievement of efficiency and may by itself make the difference for a given application as in the case of a multimedia speech coder for digital simultaneous voice and data (DSVD) [11, 12]. Therefore, the availability of reduced complexity search algorithms could lead to new applications for the ACELP coder.

Additionally, one should observe that structured sparse codebooks may be searched in multistage schemes in much the same way that the usual adaptive and fixed codebooks in a CELP coder may be searched sequentially in a suboptimal two-stage procedure. Such procedures are important to keep

the overall complexity within bounds at a small penalty in performance over an unfeasible joint optimal search.

Furthermore, it is worth noting that 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 searches and algorithms is essential to the development of coders with greater quality/complexity ratios [13].

Motivated by all this reasoning, this work presents a low-complexity search algorithm for algebraic multipulse codebooks that has been implemented within the reference G.723.1 [14] and G.729 [7] coders as examples of target speech coders, but may be applied to other coders with deterministic sparse codebooks.

This paper has two main goals. The first is the description of multistage codebook searches under a classification that encompasses the basic idea of CELP two-stage searches and optimizing orthogonalizations, all the way through multipulse searches and leaving room for new efficient algebraic multipulse codebook searches.

The second and foremost goal is the presentation of a novel joint position and amplitude search (JPAS) algorithm of algebraic multipulse codebooks which builds upon well-known multipulse search procedures. This joint search algorithm is introduced in a conceptual formulation involving dynamic autocorrelation elements in such a way that the main notion of a chain of two-dimensional optimizations can be properly appreciated. In the sequel the joint search algorithm is elaborated to the point that an efficient implementation is derived. Thereby a foundation is laid for the development of low-complexity multistage search algorithms of deterministic sparse codebooks.

In the following section, notation is established for the basic CELP codebook search. Then, sequential searches in two codebooks are classified in three categories according to their joint degree of optimality. Next, the standard multipulse search is presented as an instance of a multistage sequential search and algebraic codebooks are introduced. Then, the proposed joint position and amplitude search is introduced as a multistage search using dynamic autocorrelations and its complexity is analyzed. Finally, a low-complexity algorithm is derived for the joint search as a function of pre-computed autocorrelation elements, and its complexity and performance are measured and compared with standard algorithms.

2. BASIC CELP SEARCH

For a given target vector \mathbf{u} , a standard CELP search algorithm tries each codevector \mathbf{c}_i , $i = 1, 2, \dots, M$, in its codebook producing, after scaling and filtering, reconstructed vectors

$$\tilde{\mathbf{u}}_i = \eta_i \mathbf{H} \mathbf{c}_i, \quad (1)$$

where η_i are suitable gain factors and matrix \mathbf{H} , the impulse response matrix, is given by

$$\mathbf{H}(i, j) = h(i - j), \quad (2)$$

where $h(n) = 0$ for $n < 0$ and $h(n)$ for $n = 0, 1, \dots, L - 1$

is the impulse response of the weighted synthesis filter truncated to the subframe length L .

The reconstruction error vectors

$$\boldsymbol{\varepsilon}_i = \mathbf{u} - \tilde{\mathbf{u}}_i \quad (3)$$

are used for the selection criterion of the codevector by its index as

$$\xi = \underset{i=1,2,\dots,M}{\operatorname{argmin}} \{ \|\boldsymbol{\varepsilon}_i\|^2 \}. \quad (4)$$

An alternative search criterion may be derived by observing Figure 1. By minimizing $\|\boldsymbol{\varepsilon}_i\|^2$, the squared norm of the reconstructed target vector $\tilde{\mathbf{u}}_i$ is maximized over the indices i in the codebook, i. e., the index of the selected codevector can be found as

$$\xi = \underset{i=1,2,\dots,M}{\operatorname{argmax}} \{ \|\tilde{\mathbf{u}}_i\|^2 \}. \quad (5)$$

Writing out the squared norm of $\tilde{\mathbf{u}}_i$ as a function of the weighted codevector $\mathbf{q}_i = \mathbf{H} \mathbf{c}_i$, we have

$$\|\tilde{\mathbf{u}}_i\|^2 = \frac{C_i^2}{\sigma_i^2}, \quad (6)$$

where

$$\sigma_i^2 = \mathbf{q}_i^T \mathbf{q}_i \quad (7)$$

is the squared norm or energy of the weighted codevector and

$$C_i = \mathbf{u}^T \mathbf{q}_i \quad (8)$$

is the zero-lag crosscorrelation, or simply correlation, between the weighted codevector and the target vector.

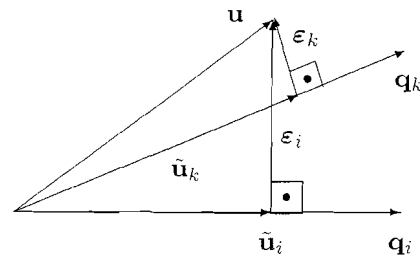


Figure 1. Geometrical representation of the codebook search in a CELP coder (index k is a better choice than index i).

The important point about the correlation is that it may be efficiently determined by a single inner product as

$$C_i = \mathbf{t}^T \mathbf{c}_i, \quad (9)$$

where

$$\mathbf{t} = \mathbf{H}^T \mathbf{u} \quad (10)$$

is the backward-filtered target vector [17, 18]. However, really efficient computations of the squared norm given by Equation (6) will ultimately depend on structuring or simplifying the calculation of the energy, expressed by Equation (7). When the codevectors are sparse, it is very convenient to express their weighted energies as

$$\sigma_i^2 = \mathbf{c}_i^T \mathbf{H}^T \mathbf{H} \mathbf{c}_i = \mathbf{c}_i^T \boldsymbol{\Phi} \mathbf{c}_i, \quad (11)$$

where

$$\Phi = \mathbf{H}^T \mathbf{H} \quad (12)$$

is the autocorrelation matrix of the impulse response matrix.

A CELP coder usually makes use of two codebooks at least, namely, an adaptive codebook and a fixed codebook, which are suboptimally searched as a two-stage codebook. Adaptive codevectors are lagged vectors extracted from the past composite excitation sequence and fixed codevectors are innovations. Usually, the adaptive codebook is searched first and then the fixed codebook is searched for a match of the residual target vector

$$\mathbf{u}_f = \mathbf{u} - \tilde{\mathbf{u}}_a, \quad (13)$$

where $\tilde{\mathbf{u}}_a$ is the reconstruction of the target vector provided by the adaptive codebook.

3. SEARCHING TWO CODEBOOKS

Speech coders provided with two codebooks require a much higher complexity search than those provided with a single codebook if the aim is an optimal search [3].

In an optimal search of two codebooks every pair of codevectors in their Cartesian product must be searched, causing the total number of levels in the compound codebook to be the product of the levels in the single individual codebooks. In practice, an exhaustive search over such a codebook is prohibitively expensive. Therefore, sequential searches are usually used instead. Most often, corrections are effected in order to get closer to the optimal solution.

In the following, three search procedures in two codebooks are examined:

- Plain sequential searches;
- Sequential searches with final joint gain calculation;
- Orthogonal sequential searches.

The final joint gain calculation is an enhancement of plain sequential searches that lends itself to implementation with little additional complexity as it must be carried out just once per subframe. On the other hand, in the case of nonstructured fixed codebooks, orthogonal sequential searches have a higher complexity since they require one orthogonalization per codevector searched. However, when fixed codebooks are structured as in the VSELP coder [19], whose codevectors are defined as sums of basis vectors, the number of required orthogonalizations is reduced. For instance, in the VSELP coder only 14 basis vectors should be filtered instead of the whole set of 256 codevectors resulting from the sequential composition of the two fixed codebooks.

The next three subsections will deal with each one of the three sequential search procedures in turn. Without loss of generality, it is assumed that the first codebook to be searched is an adaptive codebook and that the second one is a fixed codebook. Actually, the framework built in these sections will be applied in later sections to the case when both codebooks are stages of a multistage fixed codebook.

3.1 PLAIN SEQUENTIAL SEARCHES

A plain sequential search consists of two operationally independent searches, where each one spans its own codebook once.

The first search uses target vector \mathbf{u} shown in Figure 2 and delivers the reconstructed target vector

$$\tilde{\mathbf{u}}_a = \eta_a \mathbf{q}_a. \quad (14)$$

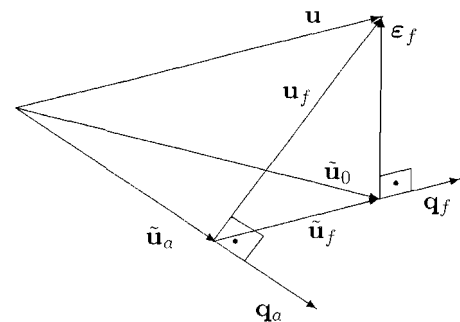


Figure 2. Geometrical representation of sequential search for target vector \mathbf{u} and residual target vector \mathbf{u}_f .

The second search takes as its target vector the residual error after the first search, which is given by Equation (13).

The searches follow the process described in Section 2 and, at the end of the two searches, target vector \mathbf{u} may be reconstructed by applying the composite excitation

$$\mathbf{e}_0 = \eta_a \mathbf{c}_a + \eta_f \mathbf{c}_f \quad (15)$$

to the weighted synthesis filter to get the reconstruction of the target vector

$$\tilde{\mathbf{u}}_0 = \eta_a \mathbf{q}_a + \eta_f \mathbf{q}_f, \quad (16)$$

where the adaptive codevector gain is

$$\eta_a = \frac{\mathbf{u}^T \mathbf{q}_a}{\mathbf{q}_a^T \mathbf{q}_a} \quad (17)$$

and the fixed codevector gain is

$$\eta_f = \frac{\mathbf{u}_f^T \mathbf{q}_f}{\mathbf{q}_f^T \mathbf{q}_f}. \quad (18)$$

3.2 SEQUENTIAL SEARCHES WITH FINAL JOINT GAIN CALCULATION

The following orthogonality relations are sufficient to establish the orthogonality of the final error vector \mathbf{e}_f to the plane $[\mathbf{q}_a \ \mathbf{q}_f]$ (see Figure 2)

$$\mathbf{e}_f \perp \mathbf{q}_f \iff \mathbf{e}_f^T \mathbf{q}_f = 0, \quad (19)$$

$$\mathbf{e}_f \perp \mathbf{q}_a \iff \mathbf{e}_f^T \mathbf{q}_a = 0. \quad (20)$$

Upon reconsideration of the procedure of plain sequential searches seen in Section 3.1, the reconstruction represented by Equation (16) can be improved if, instead of taking the gain factors η_a and η_f from Equations (17) and (18), respectively, their values are set free for a while by using expression (16). Thereby, the target vector is decomposed as

$$\mathbf{u} = \eta_a \mathbf{q}_a + \eta_f \mathbf{q}_f + \mathbf{e}_f \quad (21)$$

as depicted in Figure 2.

Taking inner products of both sides of Equation (21) by \mathbf{q}_a and \mathbf{q}_f in turn, we get

$$\mathbf{u}^T \mathbf{q}_a = \eta_a (\mathbf{q}_a^T \mathbf{q}_a) + \eta_f (\mathbf{q}_f^T \mathbf{q}_a) + (\boldsymbol{\varepsilon}_f^T \mathbf{q}_a) \quad (22)$$

$$\mathbf{u}^T \mathbf{q}_f = \eta_a (\mathbf{q}_a^T \mathbf{q}_f) + \eta_f (\mathbf{q}_f^T \mathbf{q}_f) + (\boldsymbol{\varepsilon}_f^T \mathbf{q}_f), \quad (23)$$

respectively.

Using orthogonality conditions (19) and (20), Equations (22) and (23) reduce to the following system of linear equations

$$\begin{bmatrix} \mathbf{q}_a^T \mathbf{q}_a & \mathbf{q}_a^T \mathbf{q}_f \\ \mathbf{q}_a^T \mathbf{q}_f & \mathbf{q}_f^T \mathbf{q}_f \end{bmatrix} \begin{bmatrix} \eta_a \\ \eta_f \end{bmatrix} = \begin{bmatrix} \mathbf{u}^T \mathbf{q}_a \\ \mathbf{u}^T \mathbf{q}_f \end{bmatrix}, \quad (24)$$

which may be solved for the final gains.

3.3 ORTHOGONAL SEARCHES

The procedure of sequential searches with final joint gain calculation presented in the previous section just constrains the error vector to be orthogonal to plane $[\mathbf{q}_a \ \mathbf{q}_f]$ a posteriori. Sequential searches can be drawn a little closer to the joint optimal search if the second search is carried out by enforcing orthogonality throughout. This can be done as follows.

Initially, the adaptive search with target vector \mathbf{u} is carried out as in Section 3.2, yielding filtered codevector \mathbf{q}_a and initial gain η_a . Next, the orthogonal search of the fixed codebook is performed through the following set of operations (see also Figure 3):

- each codevector $\mathbf{c}_{f,i}$, $i = 1, 2, \dots, M_f$, is filtered, resulting in the corresponding weighted codevectors

$$\mathbf{q}_{f,i} = \mathbf{H} \mathbf{c}_{f,i}; \quad (25)$$

- plane $[\mathbf{q}_a \ \mathbf{q}_{f,i}]$ is represented by an orthogonal basis by determining the component \mathbf{w}_i of $\mathbf{q}_{f,i}$ which is orthogonal to \mathbf{q}_a as

$$\mathbf{w}_i = \mathbf{q}_{f,i} - \frac{\mathbf{q}_a^T \mathbf{q}_{f,i}}{\mathbf{q}_a^T \mathbf{q}_a} \mathbf{q}_a; \quad (26)$$

- the projection of target vector \mathbf{u} upon orthogonal vector \mathbf{w}_i produces vector

$$\mathbf{t}_i = \frac{\mathbf{w}_i^T \mathbf{u}}{\mathbf{w}_i^T \mathbf{w}_i} \mathbf{w}_i. \quad (27)$$

whose squared norm is given by

$$\|\mathbf{t}_i\|^2 = \frac{(\mathbf{w}_i^T \mathbf{u})^2}{\mathbf{w}_i^T \mathbf{w}_i}; \quad (28)$$

- codevector $\mathbf{c}_{f,\xi}$ is selected such that

$$\xi = \operatorname{argmax}_{i=1,2,\dots,M_f} \{\|\mathbf{t}_i\|^2\}; \quad (29)$$

- joint computation of the final gain factors η_a and η_f by solving the System of Equations (24) with $\mathbf{q}_f = \mathbf{H} \mathbf{c}_{f,\xi}$, so that the total composite excitation \mathbf{e} is obtained as

$$\mathbf{e} = \eta_a \mathbf{c}_a + \eta_f \mathbf{c}_{f,\xi}. \quad (30)$$

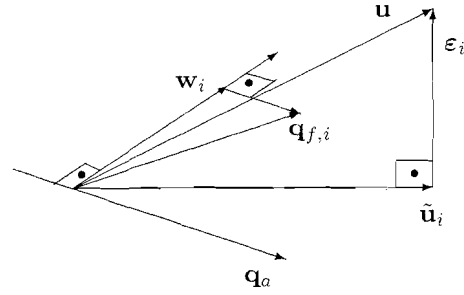


Figure 3. Geometrical representation of vectors involved in an orthogonal sequential search.

4. MULTIPULSE SEARCHES

Multipulse excitations are sparse signals whose nonzero samples are isolated from one another. They may be described in general as

$$\epsilon(n) = G \sum_{k=0}^{M-1} \alpha_k \delta(n - m_k), \quad n = 0, 1, \dots, L - 1. \quad (31)$$

where M is the number of pulses, $\delta(n)$ is the unit pulse, L is the subframe length, α_k and m_k are pulse amplitudes and positions, respectively, and G is the overall excitation signal gain.

Following the search method of Section 3.1, multipulse codebooks are usually searched in M sequential stages [15, 16]. In this case, at stage j the available subcodevectors are

$$\mathbf{c}_i^{(j)} = \mathbf{I}(:, i), \quad i \in \mathcal{I}^{(j)}, \quad (32)$$

where $\mathbf{I}(:, i)$ represents the i th column of the $L \times L$ identity matrix, and the corresponding filtered subcodevectors are

$$\mathbf{q}_i^{(j)} = \mathbf{H}(:, i), \quad i \in \mathcal{I}^{(j)}, \quad (33)$$

where $\mathbf{H}(:, i)$ stands for the i th column of the impulse response matrix defined by Equation (2) and $\mathcal{I}^{(j)} = \{0, 1, \dots, L - 1\} - \{m_0, m_1, \dots, m_{j-1}\}$. The set $\{m_0, m_1, \dots, m_{j-1}\}$ is the set of all the previously chosen pulse positions.

For searching purposes, it is convenient to rewrite the excitation given by Equation (31) as

$$\epsilon(n) = \sum_{k=0}^{M-1} A_k \delta(n - m_k), \quad (34)$$

where individual pulse amplitudes are independently denoted as $A_k = G \alpha_k$. Using this notation, the reconstructed vector component contributed by stage j is given by

$$\tilde{\mathbf{u}}_f^{(j)} = A_j^{(j)} \mathbf{H}(:, m_j). \quad (35)$$

The weighted reconstruction error vector at stage j is defined as

$$\boldsymbol{\varepsilon}^{(j)} = \mathbf{u}_f^{(j)} - \tilde{\mathbf{u}}_f^{(j)}. \quad (36)$$

where

$$\mathbf{u}_f^{(j)} = \begin{cases} \mathbf{u}_f, & j = 0 \\ \boldsymbol{\varepsilon}^{(j-1)}, & j = 1, 2, \dots, M - 1 \end{cases} \quad (37)$$

stands for the target vector at stage j in an extension to the procedure outlined in Section 3.2.

The multipulse excitation search criterion is the minimization of the squared norm of the error vector, just as described in Section 2, which leads to pulse position

$$m_j = \underset{i \in \mathcal{I}^{(j)}}{\operatorname{argmin}} \left\{ \boldsymbol{\varepsilon}^{(j)T} \boldsymbol{\varepsilon}^{(j)} \right\} \quad (38)$$

and amplitude

$$A_j = \frac{\mathbf{t}_f(m_j)}{\|\mathbf{H}(:, m_j)\|^2} \quad (39)$$

for the j th pulse, where $\mathbf{t}_f(m_j)$ is a sample of the backward-filtered target vector defined by Equation (10). Therefore, in Equation (39) A_j is computed in a similar way to the gain factors determined by Equations (17) and (18).

Finally, at the end of stage $j = M-1$ the whole set of pulse amplitudes is recalculated by a higher-dimensional extension to the procedure developed in Section 3.2. Taking into account Equation (33), Matrix Equation (24) may be translated into

$$\Phi_M \begin{bmatrix} A_0 \\ A_1 \\ \vdots \\ A_{M-1} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_f(m_0) \\ \mathbf{t}_f(m_1) \\ \vdots \\ \mathbf{t}_f(m_{M-1}) \end{bmatrix}. \quad (40)$$

where the system matrix $\Phi_M = [\phi(m_i, m_j)]_{i,j=0}^{M-1}$ on the left-hand side is an $M \times M$ submatrix of the $L \times L$ impulse response autocorrelation matrix given by Equation (12). A previous derivation of Equation (40) from a different standpoint may be found in [20].

5. THE ALGEBRAIC MULTIPULSE CODEBOOK

Algebraic multipulse excitation satisfies the same formation rule as the more general multipulse excitation described in Section 4 with the additional constraint that algebraic multipulses take on positive or negative unit values only, that is, $\alpha_k = \pm 1$. In the following, the number of pulses in Equation (31) will be $M = 4$ and the subframe length will be either $L = 40$ or $L = 60$ samples, respectively for the G.729 and G.723.1 coder, which are values commonly adopted. Extension to other values is straightforward.

Additionally, the algebraic multipulses to be considered [14] 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 set of positions spanned by the even grid is

$$G_e = \{0, 2, \dots, L-2\} \quad (41)$$

and the corresponding collection of positions in the odd grid is

$$G_o = \{1, 3, \dots, L-1\}. \quad (42)$$

The even grid is represented in tabular form in Table 1, where each row is a different phase track.

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

Table 1. Even ACELP position grid for the G.723.1 coder.

Therefore, a pulse position m_k may be decomposed as

$$m_k = \lambda_k S + \theta_k + \gamma_k, \quad (43)$$

where $S = 8$ is the spacing between successive pulses in a row, λ_k is the position of pulse m_k in its row, θ_k is the phase of pulse m_k or its row identifier and γ_k is the pulse position parity or its grid identifier.

As displayed in Table 1, the phases $\theta = 4$ and $\theta = 6$ hold one position each beyond the subframe length. Some search methods consider these virtual positions while others do not (Section 6). If only actual subframe positions are considered, then the number of pulse positions of phases $\theta_j \in \{4, 6\}$ is $N_j = 7$ whereas $N_j = 8$ holds for phases $\theta_j \in \{0, 2\}$.

6. JOINT POSITION AND AMPLITUDE SEARCH

The proposed joint position and amplitude search (JPAS) [21] is described in this section for fixed algebraic multipulse codebooks such as the one presented in Section 5.

The reconstructed residual target vectors are composed of shifted impulse responses as

$$\tilde{\mathbf{u}}_f(n) = G \sum_{k=0}^{M-1} \alpha_k h(n - m_k), \quad n = 0, 1, \dots, L-1 \quad (44)$$

and this composition has motivated the JPAS algorithm which selects one by one the M shifted impulse responses which jointly define a partial direction along which the residual target vector provides its 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 \mathbf{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 algorithm selects position $i = m_0$ which maximizes the squared norm of the projection along the shifted impulse responses

$$\tau_i = \frac{(\mathbf{u}_f^T \mathbf{H}(:, i))^2}{\Phi(i, i)} \quad (45)$$

where \mathbf{H} is the impulse response matrix of the weighted synthesis filter and Φ is its autocorrelation matrix given by Equation (12).

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

$$\mathbf{q}_f^{(j-1)} = \sum_{k=0}^{j-1} \alpha_k^{(j-1)} \mathbf{H}(:, m_k). \quad (46)$$

Due to its algebraic structure, pulse amplitudes in the ACELP codebook are restricted to ± 1 so that, considering a new pulse at position m_j , only four new filtered subcodevectors are possible

$$\mathbf{q}_f^{(j)} = \pm \left(\pm \mathbf{H}(:, m_j) + \mathbf{q}_f^{(j-1)} \right). \quad (47)$$

Therefore, considering a new pulse at position i in iteration j , only two partial projection directions are admissible,

$$\mathbf{P}^{(j)}(:, i) = \mathbf{H}(:, i) + \mathbf{q}_f^{(j-1)}, \quad (48)$$

which is called the primary partial projection direction and

$$\mathbf{S}^{(j)}(:, i) = \mathbf{H}(:, i) - \mathbf{q}_f^{(j-1)}, \quad (49)$$

which defines the secondary partial projection direction¹. In Equations (48) and (49) $\mathbf{P}^{(j)}(:, i)$ and $\mathbf{S}^{(j)}(:, i)$ indicate the i th columns of matrices $\mathbf{P}^{(j)}$ and $\mathbf{S}^{(j)}$, respectively, whose columns comprise all the primary and secondary projection directions at iteration j , respectively.

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

$$\mathbf{f}^{(j)} = \mathbf{H}(:, m_j) + \sigma^{(j-1)} \mathbf{q}_f^{(j-1)}. \quad (50)$$

where

$$\sigma^{(j-1)} = \begin{cases} 1, & \text{if primary projection} \\ -1, & \text{if secondary projection} \end{cases} \quad (51)$$

is the sign of the selected projection direction on the plane defined by the j th pulse and the available filtered subcodevector $\mathbf{q}_f^{(j-1)}$, which is given by

$$\mathbf{q}_f^{(j-1)} \triangleq \mathbf{H} \mathbf{c}_f^{(j-1)}. \quad (52)$$

The projection of the residual target vector along the selected partial projection direction $\mathbf{f}^{(j)}$ yields the partially reconstructed target vector as

$$\tilde{\mathbf{u}}_f^{(j)} = A^{(j)} \mathbf{f}^{(j)}, \quad (53)$$

where $A^{(j)}$ is a signed gain factor whose sign is

$$\beta^{(j)} = \text{sign}(A^{(j)}), \quad (54)$$

while its absolute value

$$G^{(j)} = |A^{(j)}| \quad (55)$$

will become the gain only after the last iteration $j = M - 1$. Thus, the weighted subcodevector at the end of the j th iteration is

$$\mathbf{q}_f^{(j)} = \beta^{(j)} \mathbf{f}^{(j)}. \quad (56)$$

Using this notation it results that the weighted subcodevectors are defined by the chosen shifted impulse responses and by the corresponding signs as

$$\mathbf{q}_f^{(0)} = \beta^{(0)} \mathbf{H}(:, m_0) \quad (57)$$

for the first iteration and as

$$\mathbf{q}_f^{(j)} = \beta^{(j)} \mathbf{H}(:, m_j) + \beta^{(j)} \sum_{l=0}^{j-1} \prod_{k=l}^{j-1} \sigma^{(k)} \beta^{(k)} \mathbf{H}(:, m_l) \quad (58)$$

for the remaining iterations.

Consequently, the selected codevector, $\mathbf{c}_f = \mathbf{c}_f^{(M-1)}$, can be expressed as

$$\mathbf{c}_f = \sum_{j=0}^{M-1} \alpha_j^{(M-1)} \mathbf{I}(:, m_j) \quad (59)$$

and the resulting weighted codevector is

$$\mathbf{q}_f = \sum_{j=0}^{M-1} \alpha_j^{(M-1)} \mathbf{H}(:, m_j), \quad (60)$$

where

$$\alpha_l^{(j)} = \begin{cases} \beta^{(l)} & \text{if } l = j \\ \beta^{(j)} \prod_{k=l}^{j-1} \sigma^{(k)} \beta^{(k)} & \text{otherwise.} \end{cases} \quad (61)$$

Equation (61) describes the resulting pulse signs after each iteration j and can be derived by comparison of Equations (60) and (58).

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 (48) and (49), the primary and secondary partial projection directions for iteration j are collected in matrices $\mathbf{P}^{(j)}$ and $\mathbf{S}^{(j)}$, respectively. Further, for the computation of their corresponding projections below, we will define the primary autocorrelation matrix

$$\mathcal{P}^{(j)} = (\mathbf{P}^{(j)})^T \mathbf{P}^{(j)} \quad (62)$$

and the secondary autocorrelation matrix

$$\mathcal{S}^{(j)} = (\mathbf{S}^{(j)})^T \mathbf{S}^{(j)}. \quad (63)$$

For iterations $j = 1, 2, \dots, M - 1$, the squared norms of the projections along the primary partial directions are computed according to

$$\tau_i^{(j)} = \frac{\left(\mathbf{u}_f^T \mathbf{P}^{(j)}(:, i) \right)^2}{\mathcal{P}^{(j)}(i, i)} \quad (64)$$

and the squared norms of the projections along the secondary partial directions are determined by

$$\nu_i^{(j)} = \frac{\left(\mathbf{u}_f^T \mathbf{S}^{(j)}(:, i) \right)^2}{\mathcal{S}^{(j)}(i, i)}. \quad (65)$$

This procedure closes with the selection of position m_j which satisfies

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

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

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, i.e.,

$$\mathcal{I}^{(0)} = \{0, 1, \dots, L - 1\}. \quad (66)$$

Furthermore, the grid parity is defined as γ_0 by the parity of the pulse at position $m_0 = 8\lambda_0 + \theta_0 + \gamma_0$ selected in the first iteration, whose position has been decomposed according to Equation (43). Accordingly, if $\gamma_0 = 0$, the even grid $G_0 = G_e$ is selected (see Section 5) and otherwise, if $\gamma_0 = 1$, the odd grid $G_0 = G_o$ is selected.

For the following iterations ($j = 1, 2, \dots, M - 1$), only the phases not yet selected are actually searched so that a row in the position grid (Table 1) is eliminated from the search domain after each pulse selection and the set of indices searched is

$$\mathcal{I}^{(j)} = G_0 - \mathcal{G}_0 - \mathcal{G}_1 - \dots - \mathcal{G}_{j-1} \quad (67)$$

with

$$\mathcal{G}_i = \{\theta_i + \gamma_0, \theta_i + \gamma_0 + 8, \dots, \theta_i + \gamma_0 + 8(N_i - 1)\}$$

where N_i stands for the number of pulses in the chosen grid with the same phase as pulse position m_i as pointed out in Section 5.

Summing up, the JPAS algorithm is a kind of orthogonal search (see Section 3.3) where optimizations are carried out in planes or two-dimensional subspaces defined by each shifted impulse response whose phase track 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 as summarized by Equation (47). Further, the position and amplitude searches are conjugated by step 3 of the closing procedure of the algorithm, described just below Equation (65). Therefore, the JPAS algorithm is less suboptimal than the standard multipulse search [15, 16] which determines pulse positions by means of Equation (38) before using Equation (39) to find their amplitudes (see Section 4).

The number of searches in a subframe distributes over six classes identified by the number of subcodevectors searched along each individual search path as listed in Table 2. One should consider that each one of the six search paths in Table 2 may occur as one of four permutations because there are two phase tracks with 8 positions and two phase tracks with 7 positions (see Section 5). Considering equiprobable permutations, one arrives at an average of 150 searches per subframe.

The joint search, as opposed to both the focused and the position-exhaustive searches, does not use elements off

Search path class	Comparison count
(8,8,7,7)	$60+2(22+14+7) = 146$
(8,7,8,7)	$60+2(22+15+7) = 148$
(8,7,7,8)	$60+2(22+15+8) = 150$
(7,8,8,7)	$60+2(23+15+7) = 150$
(7,8,7,8)	$60+2(23+15+8) = 152$
(7,7,8,8)	$60+2(23+16+8) = 154$

Table 2. Breakdown of the number of comparisons per search path for the JPAS algorithm.

the main diagonals of its autocorrelation matrices as shown by the denominators of Equations (64) and (65). 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.

7. EFFICIENT JOINT SEARCH

The correlation elements involved in the joint search (Section 6) are dynamic values in the sense that they are computed along the search path. As shown below, it turns out that the dynamic correlations may be computed as functions of precalculated autocorrelations of the impulse response matrix and samples of the backward-filtered residual target vector.

Using Equations (56), (48), (49) and (10), it is possible to express the correlation in the primary and secondary partial projection directions, $\mathcal{P}^{(j)}(:, i)$ and $\mathcal{S}^{(j)}(:, i)$, used in Equations (64) and (65), as follows

$$C_{p,i}^{(j)} \triangleq \mathbf{u}_f^T \mathbf{P}^{(j)}(:, i) = \mathbf{t}_f(i) + \beta^{(j-1)} C_f^{(j-1)} \quad (68)$$

$$C_{s,i}^{(j)} \triangleq \mathbf{u}_f^T \mathbf{S}^{(j)}(:, i) = \mathbf{t}_f(i) - \beta^{(j-1)} C_f^{(j-1)}, \quad (69)$$

where

$$\begin{aligned} C_f^{(j-1)} &\triangleq \mathbf{u}_f^T \mathbf{f}^{(j-1)} \\ &= \mathbf{t}_f(m_{j-1}) + \sigma^{(j-2)} \beta^{(j-2)} C_f^{(j-2)} \end{aligned} \quad (70)$$

is the correlation chosen in iteration $j - 1$ using the procedure described in Section 6. In Equations (68) to (70) $\mathbf{t}_f(i)$ is the i -th sample of the backward-filtered target vector defined by Equation (10).

Using Equations (12), (48), (49), (46), (62) and (63), 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 Equations (64) and (65), as follows

$$\mathcal{P}^{(j)}(i, i) = \phi(i, i) + 2 \sum_{l=0}^{j-1} \alpha_l^{(j-1)} \phi(i, m_l) + E^{(j-1)} \quad (71)$$

$$\mathcal{S}^{(j)}(i, i) = \phi(i, i) - 2 \sum_{l=0}^{j-1} \alpha_l^{(j-1)} \phi(i, m_l) + E^{(j-1)}, \quad (72)$$

Miguel Arjona Ramírez and Max Gerken
A Low-Complexity Search Algorithm for Speech Coders with Sparse Excitation

Search Algorithm	PC1		PC2		N _{cor}
	ET (%)	ET _f (%)	ET (%)	ET _f (%)	
Focused	17.0	100.0	17.7	100.0	416
Joint	11.5	67.6	11.6	65.5	448

PC1 (PC2): Personal computer with a 133-MHz (100-MHz) Pentium processor under the Windows 95 (Windows NT) operating system.

ET: execution time as a fraction of real time (the duration of the speech signal under coding).

ET_f: execution time of given search algorithm as a fraction of the execution time of the focused search algorithm.

N_{cor}: number of precomputed autocorrelation elements.

Table 3. Measure of complexity of floating-point versions of two ACELP search algorithms over the test partition of the TIMIT database.

where $E^{(j-1)}$ is the squared norm of $\mathbf{q}_f^{(j-1)}$, 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 6. 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 3.

8. COMPLEXITY AND PERFORMANCE MEASUREMENTS

In this section, results of complexity measurements of the joint search are presented and compared first to both the focused search (FOCS) and the position-exhaustive search by means of floating-point implementations. Secondly, the results of extensive complexity measurements and performance tests of a fixed-point version of the JPAS algorithm and equivalent procedures in recent standard speech coders are presented and contrasted. The focused search [22, 8] is considered as implemented in the reference ITU-T 5.3 kbit/s G.723.1 codec [14]. 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 [12]. These sign simplifications are kept up in the position-exhaustive search, which only differs from standard FOCS in that it searches all the 4096 combinations of 4 pulse positions in the grid of the chosen parity.

The average number of comparisons per codevector search is estimated at the end of Section 6 for the dynamic autocor-

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

Table 4. Statistics about number of comparisons per subframe for three 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
Joint	146	149.62	154

Table 5. Statistics about number of computed autocorrelation elements per subframe for three search algorithms over the test partition of the TIMIT database.

relation version of the JPAS algorithm. Also, this number of comparisons was measured over the whole collection of 1680 signals in the test partition of the TIMIT database [23] 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 4. 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 5, the joint search displays a considerable decrease in the number of these elements. Finally, as shown in Table 6, objective performance measurements carried out with the segmental signal-to-noise ratio (SNRSEG) and the perceptual speech quality measure (PSQM) [24] detect 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 when both are implemented in floating point at 5.3 kbit/s.

Clearly, the position-exhaustive search is equivalent to the focused search and the focused search seems to perform a little better than the joint search on objective grounds. The practical significance of these results, therefore, will depend upon the results of subjective listening tests which have been performed for the fixed-point versions of coders operating at 8 kbit/s. These tests are described at the end of this section and their results are commented upon.

For the floating-point version of the JPAS algorithm, the measure of complexity used is the execution time expressed as a fraction of real time, which is taken to be the duration of

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

Table 6. Performance of three floating-point ACELP search algorithms over the test partition of the TIMIT database.

Search Algorithm	AOC (WMOPS)	ROC (%)	N _{cor}
Focused	6.31	100.0	616
Joint	1.54	24.4	616
Depth-first	1.89	30.0	616
Enhanced	3.41	54.0	1600

AOC: absolute operational complexity.
ROC: relative operational complexity which assigns unity to the focused search
N_{cor}: number of precomputed autocorrelation elements.

Table 7. Measure of worst-case complexity of fixed-point versions of four ACELP search algorithms over the test partition of the TIMIT database.

the speech signal under coding. Both the joint search and the focused search execute on two personal computers, coding the whole test partition of the TIMIT database. The execution times displayed in Table 3 show that the joint search takes 2/3 as long to execute as the focused search.

Inside the reference G.723.1 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 [25].

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 [12]. Consequently, the impact of a lower-complexity fixed search would be greater in this coder and a fixed-point implementation of the JPAS algorithm has been tested against equivalent standard coders. A more detailed operational account of the JPAS algorithm may be found in [26].

Specifically, the fixed-point implementation of the JPAS algorithm has been inserted in the reference implementation of G.729 coder and is compared to the focused search algorithm as implemented in this reference coder [7] as well as to the depth-first tree search (DFTS) algorithm [27] and the fixed search algorithm of the IS-641-A enhanced full-rate (EFR) voice codec [28]. The DFTS algorithm was taken from the G.729A reference coder and inserted into the reference G.729 coder.

The operational complexities of the fixed-point implementations are measured in worst-case weighted million operations per second (WMOPS) with results given in Table 7. The joint search algorithm actually confirms its lower complexity which stands below 25% that of the focused search.

Furthermore, this complexity ratio achieves to advantage the estimate of 1/3 derived in terms of a comparison count in Section 6 (see also [29]). The depth-first tree search ranks close behind in complexity at a mark of 30% while the enhanced full-rate search stands halfway at 54%.

On the other hand, speech quality is highest and best distributed over the signals in the database for the enhanced full-rate search as measured by the PSQM whereas the best results in terms of SNRSEG are those of the depth-first tree search as seen in Table 8.

Further, subjective quality has been assessed by having 10

Search Algorithm	SNRSEG			PSQM		
	Av.(dB)	N _S ⁽⁺⁾	N _S ⁽⁻⁾	Av.	N _p ⁽⁺⁾	N _p ⁽⁻⁾
Focused	11.35	0	0	1.66	0	0
Joint	11.07	5	1675	1.72	1547	133
Depth-first	11.26	279	1400	1.68	1138	542
Enhanced	11.14	188	1492	1.65	853	826

Av.: Average.

N_S⁽⁺⁾: number of signals whose reconstruction by the given search algorithm has SNRSEG above that of the reconstruction by the focused search algorithm.

N_S⁽⁻⁾: number of signals whose reconstruction by the given search algorithm has SNRSEG below that of the reconstruction by the focused search algorithm.

N_p⁽⁺⁾: number of signals whose reconstruction by the given search algorithm has PSQM above that of the reconstruction by the focused search algorithm.

N_p⁽⁻⁾: number of signals whose reconstruction by the given search algorithm has PSQM below that of the reconstruction by the focused search algorithm.

Table 8. Performance of fixed-point versions of four ACELP search algorithms over the test partition of the TIMIT database.

female and 10 male listeners evaluate five versions of four sentence pairs played through headphones in randomized order for each speaker. Each stimulus consists of two short sentences of about 3 seconds with an intervening 0.5 second silence gap and all the sentences were extracted from the Polidata Speech Database created by the Man-Machine Communication Group (MMC) in 1997 at the University of São Paulo. The same Portuguese text was uttered by two male and two female speakers in an office environment. The format of the test was an absolute category rating (ACR) with a 5-point scale where 5 is excellent and 1 is bad [30] and the results are shown in Table 9. The joint search algorithm is perceived as equivalent to the other search procedures and, surprisingly, the three efficient search methods have performed better than the more extensive focused search. It has been previously pointed out that sparse excitation can have a superior perceptual effect [5] but not associated with reduced search algorithms. Therefore, the degradation detected by the objective measures has gone unperceived, or rather has possibly been compensated for by other perceptual characteristics. Actually, some subjects have reported a more pleasing experience listening to some stimuli despite their lower sharpness.

9. CONCLUSION

The features of deterministic sparse codebooks have been analyzed and exploited to derive the joint position and amplitude search (JPAS) algorithm. A three-tiered classification of multistage searches is presented which encompasses standard CELP two-stage adaptive-fixed codebook searches as well as multistage multipulse searches. Thereafter, the JPAS innovation search has been described and classified as a two-dimensional optimized sequential search. The JPAS

Miguel Arjona Ramírez and Max Gerken
A Low-Complexity Search Algorithm for Speech Coders with Sparse Excitation

Listen.	Speak.	Orig.	FOCS	JPAS	DFTS	EFR
Female	Female	3.50	3.30	3.50	3.40	3.55
Male	Female	4.45	3.55	3.55	3.70	3.90
Female	Male	3.70	3.75	3.90	3.70	3.75
Male	Male	3.60	3.80	3.75	3.95	3.75
Female	All	3.60	3.52	3.70	3.55	3.65
Male	All	4.03	3.68	3.65	3.82	3.82
All	All	3.81	3.60	3.68	3.69	3.74

Table 9. Subjective quality of fixed-point versions of four ACELP search algorithms in ACR tests from recordings taken with environmental office noise.

algorithm has been tested within both the G.723.1 and the G.729 reference coders in floating-point and fixed-point implementations, respectively. For the fixed-point versions, the operational complexity of the joint search is lower than one quarter as many operations as the focused search algorithm without incurring in any perceptual degradation. Further, for the floating-point versions, it takes the joint search just two thirds as long as the focused search to execute. Such reductions in complexity may enable new applications for CELP coders with deterministic sparse codebooks.

REFERENCES

[1] B.S. Atal and M. Schroeder. "Stochastic coding of speech at very low bit rates". in *Proc. Int. Conf. Comm.*, Amsterdam, 1984, pp.1610-1613.

[2] M. Copperi and D. Sereno, "Improved LPC excitation based on pattern classification and perceptual criteria", in *Proc. 7th Int. Conf. Patt. Rec.*, Montreal, 1984, pp. 860-862.

[3] M. R. Schroeder and B. S. Atal. "Code-excited linear prediction (CELP): High quality speech at very low bit rates", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, 1985, vol. 2, pp. 437-440.

[4] I. M. Trancoso, 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.

[5] W. B. Kleijn, D. J. Krasinski and R. H. Ketchum, "Fast methods for the CELP speech coding algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 8, pp. 1330-1342, Aug. 1990.

[6] J. P. Campbell Jr., T. E. Tremain and V. C. Welch, "The DoD 4.8 kbps standard (Proposed Federal Standard 1016)", in B. S. Atal, V. Cuperman and A. Gersho, Ed., *Advances in Speech Coding*, Dordrecht: Kluwer Academic Publishers, 1991, pp. 121-133.

[7] ITU-T *Recommendation G.729 Coding of speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*. ITU-T, Geneva, March 19, 1996.

[8] 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.

[9] 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 Int. Conf. Acoust. Speech, Signal Processing*, Paris, 1982, vol. 1, pp. 614-617.

[10] C. Laflamme, R. Salami and J.-P. Adoul, "9.6 kbit/s ACELP

coding of wideband speech", in B. S. Atal, V. Cuperman and A. Gersho, Ed., *Speech and audio coding for wireless and network applications*, pp. 147-152, Dordrecht: Kluwer Academic Publishers, 1993.

[11] R. Salami, C. Laflamme, B. Bessette and J.-P. Adoul, "Description of ITU-T Recommendation G.729 Annex A: Reduced complexity 8 kbit/s CS-ACELP codec", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, 1997, vol. 2, pp. 775-778.

[12] 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.

[13] T. Honkanen, J. Vainio, K. Järvinen and P. Haavisto, "Enhanced full rate codec for IS-136 digital cellular system", in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Processing*, Munich, 1997, vol. 2, pp. 731-734.

[14] 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.

[15] C. García-Mateo and D. Docampo-Amoedo, "Modeling Techniques for Speech Coding: A Selected Survey", in A. R. Figueiras-Vidal, Ed., *Digital Signal Processing in Telecommunications*, London: Springer-Verlag, 1996, pp. 1-43.

[16] L. M. da Silva and A. Alcaim, "Dicionário Estocástico Multipulso para Codificadores CELP", *Revista da Sociedade Brasileira de Telecomunicações*, vol. 13, no. 2, pp. 83-91, Dec. 1998.

[17] J.-P. Adoul, P. Mabilieu, M. Delprat and S. Morrisette, "Fast CELP coding based on algebraic codes", in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, Dallas, 1987, pp. 1957-1960.

[18] G. Davidson, M. Yong and A. Gersho, "Real-time vector excitation coding of speech at 4800 bps", in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Processing*, Dallas, 1987, pp. 2189-2192.

[19] I.A. Gerson and M.A. Jasiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8 kbps", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Albuquerque, 1990, vol. 1, pp. 461-464.

[20] M. Berouti, H. Garten, P. Kabal and P. Mermelstein, "Efficient computation and encoding of the multipulse excitation for LPC", in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Processing*, San Diego, 1984, pp. 10.1.1-10.1.4.

[21] M. Arjona Ramírez, *Busca de Inovações e Tratamento de Transitórios em Codificadores de Voz CELP*, Doctoral dissertation, Universidade de São Paulo, São Paulo, Brazil, 1997.

[22] C. Laflamme, J.-P. Adoul, R. Salami, S. Morisette and P. Mabilieu, "16 kbps wideband speech coding technique based on algebraic CELP", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Toronto, 1991, vol. 1, pp. 13-16.

[23] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The DARPA speech recognition research database: specifications and status", *Proc. DARPA Speech Recognition Workshop*, Rep. No. SAIC-86/1546, Palo Alto, Feb. 1986.

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

[25] M. Arjona Ramírez and M. Gerken, "A multistage search of algebraic CELP codebooks", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Phoenix, 1999, vol. 1, pp. 17-20.

[26] M. Arjona Ramírez and M. Gerken. "Joint position and amplitude search of algebraic multipulses", *IEEE Trans. Speech Audio Processing*, vol. 8, no. 5, pp. 633-637, Sept. 2000.

[27] ITU-T *Recommendation G.729 - Annex A: Reduced complex-*

- ity 8 kbit/s using CS-ACELP speech codec. ITU-T, Geneva, Nov. 8, 1996.
- [28] TIA/EIA TR45, *IS-641-A TDMA Cellular/PCS - Radio Interface Enhanced Full-Rate Voice Codec, Revision A*, 17 Sep., 1997.
- [29] M. Arjona Ramírez and M. Gerken, "Efficient algebraic multipulse search", in *SBT/IEEE Int. Telecommunications Symposium*, São Paulo, 1998, vol. 1, pp. 231-236.
- [30] P. Kroon, "Evaluation of speech coders", in W. B. Kleijn and K. K. Paliwal, Ed., *Speech Coding and Synthesis*, Amsterdam, Elsevier Science, 1995, pp. 467-494.

Miguel Arjona Ramírez received the E.E. degree from Instituto Tecnológico de Aeronáutica (ITA), Brazil, in 1980 and the M.S.E.E. and Dr.E.E. degrees from University of São Paulo, Brazil, in 1992 and 1997, respectively. In 1981, while studying at Philips International Institute, he developed coding algorithms for a formant speech synthesizer at Philips Electronic Components and Materials Laboratories, The Netherlands. In 1982, he joined Itautec Informática S.A., São Paulo, Brazil, as a Full Development Engineer, eventually becoming an Engineering Development Group Leader for Interactive Voice Response Systems (IVRs) in 1988. He is currently Assistant Professor at Escola Politécnica, University of São Paulo, where he has conducted research on predictive speech coders. His research interests include signal compression, speech coding and recognition, and audio coding with applications to circuit and packet telephony.

Max Gerken received the engineer's (EE/1979), BS (Math/1981) and MS (EE/1984) degrees from University of São Paulo, Brazil. In 1990 he received the Dr.-Ing. degree from the Erlangen-Nürnberg University, Germany. He is currently Associate Professor at the Department of Telecommunications and Control Engineering, Escola Politécnica, University of São Paulo, Brazil. His main research interests are in the areas of digital and adaptive signal processing.