A new fast algebraic fixed codebook search algorithm in CELP speech coding

Václav Eksler\textsuperscript{1,2}, Redwan Salami\textsuperscript{2}, Milan Jelínek\textsuperscript{1,2}

\textsuperscript{1}Speech and Audio Coding Research Group, University of Sherbrooke, Quebec, Canada
\textsuperscript{2}VoiceAge Corporation, Montreal, Quebec, Canada

Vaclav.Eksler@Usherbrooke.ca, rewdans@voiceage.com, Milan.Jelinek@Usherbrooke.ca

Abstract

This paper introduces a very fast search algorithm of algebraic fixed codebook in CELP-based speech codecs. The proposed method searches codebook pulses sequentially, and recomputes the fixed codebook gain, the so-called backward filtered target vector, and a certain reference signal after each new pulse is determined. This results in a significant complexity reduction compared to existing methods while preserving the same speech quality. The presented algorithm is used in the new embedded speech and audio codec (G.EV-VBR) being currently standardized by the ITU-T.

Index Terms: speech coding, CELP, algebraic codebook

1. Introduction

Most of the current speech codecs are based on the code excited linear prediction (CELP) technology. In CELP, the speech signal is synthesized by filtering an excitation signal through an all-pole synthesis filter. The excitation is typically composed of two parts - a first stage contribution signal is selected from an adaptive codebook and a second stage contribution signal is selected from a fixed codebook. Generally speaking, the adaptive codebook excitation models the periodic part of the excitation and the fixed codebook excitation, also called innovative excitation, is added to model the evolution of the speech signal. In order to guarantee high quality, a very large fixed codebook is usually needed for modeling the innovative excitation.

The fixed codebook (FCB) can have many structures. The most common is the algebraic structure \cite{1} which allows for the use of very large codebooks without the need for codebook storage. The algebraic fixed codebook consists of a set of different combinations of pulses with amplitudes +1 and -1. The codebook index, or codeword, is defined by pulse positions and signs. Thus, no look-up tables are needed, since the FCB excitation vector at the decoder is constructed using the codebook index information only. The number of pulses in a codeword, and thus the codebook size, is then limited by the available bit budget.

Due to the algebraic structure, the codebook can be effectively searched by fast non-exhaustive algorithms using only some combinations of pulses (some codewords). Although many different algorithms of algebraic FCB search were introduced in recent years, the FCB search is still one of the most complexity consuming parts of CELP codecs. In this paper we introduce a new algorithm of a very low complexity algebraic codebook search and explain its implementation in the ITU-T G.EV-VBR speech and audio codec. The performance of the algorithm is compared with existing methods.

The paper is organized as follows. A principle of the algebraic codebook search with some examples of existing algorithms is described in Section 2. The new algorithm is introduced in Section 3. Its implementation in G.EV-VBR codec and performance evaluation are given in Section 4 before concluding the paper in Section 5.

2. Algebraic codebook search

The FCB is searched by minimizing the error between a target signal \(x(n)\) and the scaled filtered innovative excitation \cite{2}

\[
E = \min_k \left\{ \sum_{n=0}^{N-1} \left[ x(n) - g_c \cdot y_k(n) \right]^2 \right\},
\]

where \(g_c\) is the fixed codebook gain, and \(y_k(n)\) is the filtered fixed codebook vector (further denoted as filtered fixed codevector) with \(n\) being the sample index in the subframe of length \(N\). The signal \(y_k(n)\) is the fixed codevector \(c_k(n)\) at the codebook index \(k\) convolved with \(h(n)\), where \(h(n)\) is the impulse response of the weighted synthesis filter \(H(z)\). The target signal \(x(n)\) is the weighted input signal after subtracting the zero input response of the weighted synthesis filter and the scaled filtered adaptive codebook excitation.

Minimizing \(E\) from (1) results in the optimum fixed codebook gain

\[
g_c = \frac{\sum_{n=0}^{N-1} x(n)y_k(n)}{\sum_{n=0}^{N-1} (y_k(n))^2}.
\]

It can be shown that the fixed codebook search is performed by maximizing the term

\[
\exists_k = \frac{\left( \sum_{n=0}^{N-1} x(n)y_k(n) \right)^2}{\sum_{n=0}^{N-1} (y_k(n))^2}.
\]

In case of algebraic fixed codebook, the searched fixed codevector \(c_k(n)\) containing \(M\) unity pulses with signs \(s_j\) and positions \(m_j\) is given by

\[
c_k(n) = \sum_{j=0}^{M-1} s_j \delta(n-m_j),
\]

where \(s_j = \pm 1\), \(\delta(n) = 1\) for \(n = 0\) and \(\delta(n) = 0\) for \(n \neq 0\). The filtered fixed codevector can be then expressed in the form

\[
y_k(n) = c_k(n) * h(n) = \sum_{j=0}^{M-1} s_j h(n-m_j).
\]
The algebraic codebook search in (3) can be now described using matrix notation as a maximization of the following criterion (all vectors are supposed column vectors and \(T\) denotes transpose)

\[
\mathcal{Z}_k = \frac{(x^T y_k)^2}{y_k^T y_k} = \frac{(x^T Hc_k)^2}{c_k^T H^T H c_k} = \frac{(d^T e_k)^2}{c_k^T \Phi c_k}, \tag{6}
\]

where \(H\) is the lower triangular Toeplitz convolution matrix with diagonal \(h(0)\) and lower diagonals \(h(1), \ldots, h(N-1)\). Vector \(d = H^T x\) is the correlation between \(x(n)\) and \(h(n)\), also known as the backward filtered target vector (since it can also be computed using time-reversed filtering of \(x(n)\) through the weighted synthesis filter)

\[
d(n) = \sum_{i=0}^{N-1} x(i) h(i-n) \tag{7}
\]

and matrix \(\Phi = H^T H\) is the matrix of correlations of \(h(n)\). Both \(d\) and \(\Phi\) are usually computed prior to the codebook search. If the algebraic codebook contains only a few nonzero pulses, the computation of the maximization criterion (6) for all possible indexes \(k\) is very fast using non-exhaustive search algorithms. Examples are the nested-loop search [2] and the global pulse replacement [3]. Another method widely used in speech coding standards is the so-called depth-first tree search [4] which is based on determination of few pulse positions at a time in several subsets.

A simple search was used in G.723.1 codec at 6.3 kbps [5]. The excitation consists of several signed pulses in a frame with a fixed gain for all pulses. The pulses are sequentially searched by updating the signal \(d(n)\) and placing each new pulse at the absolute maximum of \(d(n)\). The search is repeated for several gain values, but the gain is assumed constant for each iteration.

### 3. Proposed algorithm

The proposed algorithm is built on the principle of [5]. It uses a sequential search of the pulses by maximizing a criterion based on a certain reference signal. The optimum fixed codebook gain, filtered target vector and reference signal are then recomputed after each new pulse is determined.

Similar to [6] let us define a reference signal \(b(n)\) as a weighted sum of the signal \(d(n)\) from (7) and an excitation signal \(r(n)\) that it would yield zero the minimization error in (1). The signal \(r(n)\) is obtained by filtering the target signal \(x(n)\) through the inverse of the weighted synthesis filter \(H(z)\) with zero states. The reference signal \(b(n)\) is expressed as

\[
b(n) = \frac{r_d}{E_r} r(n) + \beta d(n), \tag{8}
\]

where \(E_d = d^T d\) is the energy of the signal \(d(n)\), and \(E_r = r^T r\) is the energy of the excitation signal \(r(n)\). The value of the constant scaling factor \(\beta\) is closer to zero for large number of pulses and increases with smaller number of pulses.

Further, the pulse signs are pre-determined using the signal-selected pulse amplitude approach used for example in AMR-WB codec [6]. In this approach, the sign of a pulse at specific position is set a priori equal to the sign of the reference signal \(b(n)\) at that position. That is, we define the vector \(z(n)\) containing the signs of the reference signal \(b(n)\).

In our algorithm we further employ the autocorrelation method [7]. The autocorrelation method is an alternative approach for error minimization in the FCB search procedure in which the matrix of correlations \(\Phi\) from (6) with elements

\[
\phi(i,j) = \sum_{n=0}^{N-1} h(n-i)h(n-j), \quad i, j = 0, \ldots, N-1, \tag{9}
\]

is reduced to a Toeplitz form by modifying the summation limits in (9) so that \(\phi(i,j) = \alpha(i-j)\), where

\[
\alpha(i) = \sum_{n=0}^{N-1} h(n)h(n-i), \quad i = 0, \ldots, N-1. \tag{10}
\]

It can be shown that for an algebraic codebook with \(M\) pulses the criterion (3) to be maximized can be written using substitution of (4) and (10) into the last statement of criterion (6) as

\[
\mathcal{Z}_k = \left(\sum_{i=0}^{M-1} s_i d(m_i)\right)^2 \tag{11}
\]

and similarly the fixed codebook gain from (2) can be expressed as

\[
g_c = \frac{\sum_{i=0}^{M-1} s_i d(m_i)}{\alpha(0)} \tag{12}
\]

When sequential pulse search is used, the term in (11) reduces to

\[
\mathcal{Z}_k = \frac{(d'(m_j))^2}{\alpha(0)} \tag{13}
\]

which is maximized by finding the index of the maximum absolute value of \(d'(n)\) that corresponds to the position of the searched pulse \(m_j\). The signal \(d'(n)\) in (13) is supposed to be the signal \(d(n)\) updated before each new pulse is sequentially searched. Further, it has been verified experimentally that in case of sequential search a higher signal-to-noise ratio (SNR) is achieved by replacing the signal \(d(n)\) by the reference signal \(b(n)\). Then the maximization term at (13) changes to

\[
\mathcal{Z}_k = \frac{(b'(m_j))^2}{\alpha(0)}. \tag{14}
\]

Now, the proposed algebraic FCB search can be summarized in the following steps:

I. Compute the signal \(d(n)\), the correlation vector \(\alpha(n)\), the reference signal \(b(n)\), and the sign vector \(z(n)\).

II. Find the first pulse: From (14), the first pulse (\(M = 1\)) is found as the index of maximum absolute value of the reference signal \(b(n)\). If we use the sign vector \(z(n)\), this can be expressed as

\[
m_0 = \text{index}\left[\max(z(n) \cdot b(n))\right], \tag{15}
\]

\[
s_{z_0} = z_0(m_0). \tag{16}
\]

III. Find following pulses: The other pulses are searched sequentially for \(j = 1\) to \(M-1\). The search of each new pulse starts with computing the fixed codebook gain \(g_c\) and the update of the reference signal \(b(n)\). The fixed codebook gain from (12) for the previously found pulses (pulses \(m_0, \ldots, m_{j-1}\)) is given by
The value of the scaling factor $\beta$ as used in (8) and (22) can be adaptively changed with the number of already found pulses. The idea is to increase its value as more pulses are found. This way we emphasize the contribution of the found pulses from the original target signal $x(n)$. Using (5) this can be written as

$$x'(n) = x(n) - g^{(j-1)} \sum_{j=0}^{N-1} s_j h(n - m_j)$$

(20)

and substituting $x'(n)$ from (20) in (7) and using (10) we get an update of the signal $d(n)$, i.e.

$$d'(n) = d(n) - g^{(j-1)} \sum_{j=0}^{N-1} s_j \alpha(n - m_j).$$

(21)

Now the reference signal $b(n)$ is updated as

$$b'(i) = \frac{E_d}{E_r} r(n) + \beta d'(n)$$

(22)

and the new pulse $m_i$ is found similarly to (15) and (16) as

$$m_j = \text{index}[\max(z_b(n), b'(n))].$$

(23)

$$s_j = z_b(m_j).$$

(24)

IV. Compute fixed codevectors and filtered fixed codevectors using (4) and (5), respectively.

The above procedure can also be applied to algebraic codebooks with interleaved single-pulse permutation (ISPP) design in which the subframe is divided into several tracks of interleaved positions [2]. In this approach, pulses are searched one by one, one track at a time, in several iterations. In each iteration the search starts with a different track.

When ISPP is used, the above searched procedure is changed as follows. We suppose here that the number of tracks is equal to the number of searched pulses $M$ (one pulse per track) although the proposed algorithm can be easily extended also for other configurations. The first iteration is initialized in step II by assigning $m_0$ to $T_0$, $m_1$ to $T_1$, …, $m_{N-1}$ to $T_{N-1}$. The procedure is then repeated from step II to step IV by assigning the pulses in the initialization to different tracks and computing equations (20) to (24) for $n \in T_j$ only. The number of iterations is equal to $M$. Finally, the set of pulses selected in the iteration that maximizes the criterion (3) is chosen.

The FCB search using the ISPP design is assumed in the rest of the paper.

3.1. Scaling factor value

The value of the scaling factor $\beta$ as used in (8) and (22) can be adaptively changed with the number of already found pulses. The idea is to increase its value as more pulses are found. This way we emphasize the contribution of the updated signal $d'(n)$ in updated reference signal $b'(n)$ where there is less number of pulses left to be determined.

Some examples of optimum values of scaling factor $\beta$ as used in the implementation mentioned in section 4 are

$$\beta = \begin{cases} 2.00 & \text{for the first pulse search} \\ 2.25 & \text{for the second pulse search} \\ \infty & \text{for the third and forth pulse search} \end{cases}$$

for the 20-bit codebook and

$$\beta = \begin{cases} 4.00 & \text{for the first pulse search} \\ \infty & \text{for the second pulse search} \end{cases}$$

for 12-bit codebook from the Table 1. The value $\beta = \infty$ means that the updated reference signal $b'(n)$ is equal to the updated signal $d'(n)$ and the update (22) need not to be computed. These values of scaling factor $\beta$ were found experimentally and are generally different for other codebook sizes.

3.2. Track order determination

As stated earlier, the search procedure searches pulses sequentially track by track. The order of tracks can be chosen sequentially in accordance with the track number. i.e. for the 4-track codebook e.g. the first iteration searches tracks in the order $T_0 - T_1 - T_2 - T_3$ the second iteration in the order $T_1 - T_2 - T_3 - T_0$ etc. However the sequential order of tracks is not optimal.

In our algorithm we order tracks in accordance with the absolute maximum of the reference signal $b(n)$ per track at the beginning of the search process. This new track order determination helps to find a more accurate estimate of a potential pulse position.

4. Implementation and performance

The presented fast FCB search was developed to reduce complexity of the G.EV-VBR codec being currently standardized by ITU-T [8]. The G.EV-VBR is an embedded codec comprising 5 layers, where the algebraic FCB search is employed in the first two layers. The first layer (L1, 8 kbps) uses a classification-based ACELP technique, and the second layer (L2, additional 4 kbps) uses an algebraic codebook technique to encode the error signal from the first layer. The frame length is 20 ms and is divided into four subframes at the internal sampling rate of 12.8 kHz.

More specifically, the algebraic FCB search in L1 employs 20-bit and 12-bit codebooks. Their usage in different subframes depends on the classification-based coding mode. The FCB search in L2 employs the 20-bit codebook in two subframes and the 12-bit codebook in the other two subframes. The configurations of these codebooks are summarized in Table 1.

| Table 1. Summary of algebraic fixed codebook configurations in G.EV-VBR codec. |
|---|---|---|---|
| codebook size | number of pulses | number of tracks | positions per track |
| 12-bit | 2 | 2 | 32 |
| 20-bit | 4 | 4 | 16 |

The baseline version of G.EV-VBR employed a FCB search known as depth-first tree search [4] and the presented algorithm performance was tested during the consecutive G.EV-VBR codec development. The objective of the new FCB search was to achieve similar synthesized speech quality as with the depth-first tree search method [4], but with
a significant decrease of complexity. The new search was employed both in 12-bit and 20-bit codebooks of the G.EV-VBR codec using scaling factor values β from Section 3.1.

In the evaluation tests described in this paper a database of 128 simple clean speech sentences at nominal level comprising both male and female North American English speakers was used as a speech material. The length of this database was about 456 s. The performance of the algorithm within G.EV-VBR codec was evaluated in layers where the algebraic fixed codebook search is used, i.e., L1 and L2. The results are summarized in Table 2. The efficiency of the new algorithm was evaluated using a segmental SNR measurement on the perceptually weighted speech signal. The complexity is given in WMOPS (Weighted Million Operations Per Second) for the worst case in the fixed point implementation.

As can be seen from the Table 2, the presented algorithm reduces computational requirements significantly, but for a cost of a little segmental SNR decrease compared to the depth-first tree FCB search. Therefore it was decided to use the proposed algorithm only in the second layer (L2) where the SNR drop is insignificant. To confirm that there is no perceptual degradation we conducted two blind comparative A-B listening tests, one for the clean speech condition, and one for the car noise condition. Both tests used G.EV-VBR codec at 12 kbps. The total number of 11 listeners listened to 12 different speech samples from the mentioned database. Results shown in Figure 1 show no statistical difference between the proposed FCB search and the depth-first tree FCB search.

Table 2: The presented algorithm performance and worst case complexity for G.EV-VBR codec at L2 (12 kbps).

<table>
<thead>
<tr>
<th>FCB search</th>
<th>segmental SNR [dB]</th>
<th>20-bit FCB [WMOPS]</th>
<th>12-bit FCB [WMOPS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>depth-first tree FCB</td>
<td>8.992</td>
<td>13.458</td>
<td>6.474</td>
</tr>
<tr>
<td>new FCB in L1 and L2</td>
<td>8.760</td>
<td>3.922</td>
<td>1.228</td>
</tr>
<tr>
<td>new FCB in L2 only</td>
<td>8.950</td>
<td>9.697</td>
<td>4.164</td>
</tr>
</tbody>
</table>

As can be seen from the Table 2, the presented algorithm reduces computational requirements significantly, but for a cost of a little segmental SNR decrease compared to the depth-first tree FCB search. Therefore it was decided to use the proposed algorithm only in the second layer (L2) where the SNR drop is insignificant. To confirm that there is no perceptual degradation we conducted two blind comparative A-B listening tests, one for the clean speech condition, and one for the car noise condition. Both tests used G.EV-VBR codec at 12 kbps. The total number of 11 listeners listened to 12 different speech samples from the mentioned database. Results shown in Figure 1 show no statistical difference between the proposed FCB search and the depth-first tree FCB search.

The presented algorithm was also implemented and tested in the G.729.1 codec [9] at 8 kbps where the original global-pulse replacement FCB search [3] was substituted by the presented new algorithm. In this codec the scaling factor was set $\beta = 6$ for the first pulse search, $\beta = 8$ for the second pulse search and $\beta = \infty$ for the last two pulses search. The G.729.1 codec processes signals in 4 subframes of 40 samples each using ISPP. The pulse position of the pulses $m_1$, $m_2$, and $m_3$ in each subframe is encoded with 3 bits each, while position of $m_3$ is encoded with 4 bits. In addition each pulse sign is encoded with 1 bit which gives a total of 17 bits for the fixed codebook per subframe. From the results summarized in Table 3 it can be seen that the new algorithm is more efficient and less complex than the global-pulse replacement FCB search.

Table 3: The performance and the worst case complexity of the presented algorithm implemented in G.729.1 codec at 8 kbps.

<table>
<thead>
<tr>
<th>FCB search</th>
<th>SNR [dB]</th>
<th>[WMOPS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>global-pulse replacement FCB</td>
<td>10.157</td>
<td>2.556</td>
</tr>
<tr>
<td>new FCB</td>
<td>10.240</td>
<td>2.011</td>
</tr>
</tbody>
</table>

5. Conclusion

The new fast algebraic fixed codebook search introduced in this paper uses sequential pulse search and gain recompute after each new pulse is determined. This scenario helps to significantly reduce the complexity of the algebraic FCB search compared to the existing methods. The algorithm is used for FCB search in the second layer of the new ITU-T G.EV-VBR speech and audio codec being standardized as Recommendation G.718. Although a little SNR decrease was observed between the presented algorithm and the depth-first tree algorithm, it was proven by listening tests that this decrease is inaudible.

6. References


