A 6.7 KBPS VSELP SPEECH CODER IMPLEMENTATION ON TMS320C54X BASED ON FAST CODEBOOK SEARCH TECHNIQUE

K. N. Dharani Kumar¹, K. M. M. Prabhu² and P. C. W. Sommen³

¹ Department of Electrical Engineering, Indian Institute of Technology, Madras - 600036, India (dharani@ee.iitm.ernet.in)
² Department of Electrical Engineering, P. O. Box 513, Bldg. Eh 3, 26, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. On leave from the Indian Institute of Technology, Madras-600036, India (prabhu_kmm@hotmail.com)
³ Department of Electrical Engineering, P. O. Box 513, Bldg. Eh 3, 34, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (p.c.w.sommen@tue.nl)

SUMMARY: In this paper, a 6.7-kbps Vector Sum Excited Linear Prediction (VSELP) coder with less computational complexity is presented. A very efficient VSELP codebook with 9 basis vectors and a heuristic K-selection method (to reduce the search space and complexity), is constructed to get the stochastic codebook vector. The coder is implemented on a TMS320C541 processor.

1. Introduction

The Code Excited Linear Prediction (CELP) [1] coders have proved to be more efficient in the design of low bit-rate speech coders during the last decade. The major drawback in the CELP structure is that the complexity of the coder is very high. To overcome this problem, the VSELP is quite a noticeable technique. Toll quality speech with reduced complexity can be achieved using VSELP coding and also the encoded speech is robust to channel errors. In this paper we present the implementation of a 6.7-kbps VSELP speech coder on TMS320C541 DSP processor. The K-selection method, which searches the stochastic codebook fast, is used to reduce the complexity. This paper is organized as follows: Section 2 gives a brief description of how a VSELP encoder works with the details of the VSELP codebook structure. Section 3 explains the process of getting 2⁹ codevectors in the stochastic codebook. Section 4 explains the training process and optimization of basis vectors. In Section 5 we introduce the K-selection method. Sections 6 and 7 provide the details of DSP implementation and the performance of the coder with fast search method. Section 8 provides the conclusions.

2. VSELP Encoder

VSELP coders use the analysis-by-synthesis [2] method to select the excitation codevector from given codebooks. It uses one stochastic codebook for randomness and one adaptive codebook for extracting periodicity of the excitation signal. A bandwidth expanded LPC all-pole filter is used as the synthesis filter to shape the spectral envelope of the excitation signal. A frame consists of four subframes. The filter coefficients are estimated in every frame and updated once a sub-frame through interpolation. The codebook gains and indices are selected in every sub-frame to minimize the total weighted error.

The stochastic codebook of 6.7-kbps VSELP coder is constructed using one VSELP codebook. The 6.7-kbps speech coder uses a VSELP excitation codebook [3] consisting of 2⁹ codevectors, where M is the number of basis vectors vₘ, [m = 1 to 9]. The space candidate vector uᵢ(ₙ), (i = 0, 1, …., 511) in the stochastic codebook is constructed using a set of M (M = 9) basis vectors, as

\[ uᵢ(ₙ) = \sum_{m=1}^{M} \thetaₘᵢ vₘ(ₙ), \text{for } 0 ≤ i ≤ 2^M - 1, 0 ≤ n ≤ NSF, \]

where NSF is the sub-frame length equal to 40. The value of the binary variable \( \thetaₘᵢ \), a (+1, -1) for \( 0 ≤ i ≤ 2^M - 1 \) and \( 1 ≤ m ≤ M \) is chosen as follows: \{\( \thetaₘᵢ \)\} defines a \( 2^M - 1 \times M \) matrix of binary elements. Each row of this matrix is assigned a 9-bit index, \( i \), (between 0 and \( 2^M - 1 \)). If the binary code has a zero in \( m^{th} \) location, then \( \thetaₘᵢ = -1 \), else \( \thetaₘᵢ = +1. \) Instead of using binary coded values, the VSELP coder implements a Gray coded version of this.

The construction of the stochastic codebook vector set is illustrated in Fig. 1. The target signal \( p(n) \) of excitation search procedure is produced from the perceptually weighted filter \( W(z) \) with an input speech signal \( s(n) \). The adaptive codebook consists of 128 elements. Depending upon the Lag \( L \), (20 ≤ L ≤ 147), the adaptive codevector \( bₐ \) is constructed using different sets of 40 elements each from the adaptive codebook. The adaptive and stochastic codebook gains are denoted by \( \beta \) and \( \gamma \), respectively. The VSELP codebook search procedure selects the indices \( L \)
and \( i \) from the adaptive and stochastic codebooks in order to minimize the total weighted error, which is defined as

\[
E_{L,i} = \sum_{n=0}^{\text{NSF}-1} (p(n) - \beta b_{L}^i(n) - \gamma f_i(n))^2.
\]

(2)

where \( b_{L} \) and \( f_i \) are the optimal codevectors chosen from adaptive and stochastic codebooks, respectively. Eqn. (2) shows that we must search for all the combinations of the adaptive and the stochastic codevectors to find out the optimal excitation signal, which is computationally very complex. Therefore, in real-time implementation, the search procedure is executed separately and sequentially, which means that the adaptive codebook is searched for the optimal excitation vector \( b_{L}^i \) by keeping \( \gamma = 0 \) and then the stochastic codebook is searched. To achieve the joint optimization during sequential searching, the weighted stochastic codevectors are orthogonalized to previously weighted excitation signal. After all the codevectors are selected, the 7-bit gains are evaluated simultaneously using LBG algorithm [4]. Fig.1 shows the block diagram of the codebook search procedure in the encoder (for bit allocation of 6.7-kbps VSELP coder refer to Table 1). As discussed earlier, the adaptive codebook search procedure selects the codeword \( L \) that minimizes \( E_{L,i} \), which is defined as

\[
E_{L} = \sum_{n=0}^{\text{NSF}-1} (p(n) - \beta b_{L}^i(n))^2.
\]

(3)

where \( \beta \) is the optimal gain for each weighted codevector \( b_{L}^i \). Instead of minimizing the error \( E_{L,i} \) we can maximize the match score \( (MS_{L,i}) \), defined as

\[
MS_{L,i} = |p|^2 \cos^2 \omega_{\beta b_{L}^i},
\]

where \( \omega_{\beta b_{L}^i} \) is the angle between the two vectors \( p \) and \( b_{L}^i \). Because of the free gain term \( \beta \), the adaptive codebook search procedure can be considered by selecting the codevector \( b_{L}^i \), which maximizes \( \cos^2 \omega_{\beta b_{L}^i} \). After getting the optimal codevector \( b_{L}^i \), every weighted stochastic codevector must be orthogonalized to \( b_{L}^i \). Fortunately, for the VSELP codebook, this task reduces orthogonalizing only \( M \) weighted basis vectors \( q_m \) to \( b_{L}^i \) as shown in Fig. 2. Then \( f_i \), that is \( f_i \) after orthogonalization to \( b_{L}^i \), can be expressed as:

\[
f_i^* = \sum_{m=1}^{M} \theta_{im} q_m^*(n), \quad \text{for } 0 \leq i \leq 2^M - 1, \quad 0 \leq n \leq \text{NSF}.
\]

(4)

Now we can find the optimal excitation codeword \( i \) which minimizes \( E_{L,i} \), given by:

\[
E_i = \sum_{n=0}^{\text{NSF}-1} (p(n) - \gamma f_i^*(n))^2.
\]

(5)

where \( \gamma \) is the optimal gain for \( f_i^* \). As in the case of adaptive codebook search, the optimal codeword \( i \) maximizes \( \cos^2 \omega_{\gamma f_i^*} \), where \( \omega_{\gamma f_i^*} \) is the angle between the two vectors \( p \) and \( f_i^* \) [3] (refer to Fig. 2).

3. Stochastic Codebook Simplification

The stochastic codebook in a CELP coder is generated using randomly generated white Gaussian noise. We try to bring down the size of the stochastic codebook to the required level, which is spread uniformly over an \( n \)-dimensional sphere. To simplify things, we assume that the elements of the codebook vectors are 1, 0 and \(-1\). Since we consider the vector length \( n = 40 \), there are \( 3^{40-1} \) (zero vector) possible vectors in the 40-dimensional space, which is too large, since we have only 9-bits reserved for the codebook index. So we have to bring it down to the required codebook size \( (2^4) \). Now considering the approximation that the speech residuals are Gaussian distributed and are small enough, some vectors in the codebook can be set to zeros. But how many have to be made zero? For this question, there is no theoretical support, but based on experimental results the NSA CELP standard (Federal 1016) [5] team used a 77% zero codebook and have reported a fairly good performance. We also follow this suggestion (we make it 75%, because 77% of 40 is not an integer). So we have 30 zeros out of the total 40 components, and the remaining 10 components are either 1’s or –1’s. After this simplification, the size of the codebook is given by

\[
S_w^n = 2^w \times \binom{n}{w} = \frac{2^w \times n!}{(n-w)! \times w!} = \frac{2^{10} \times 40!}{30! \times 10!}.
\]

(6)

where \( n \) is the dimension and \( w \) is the weight. From Eqn. (6) we know that the codebook is still too large. We know that the human ears are, by nature, relatively insensitive to phase shifts in the speech waveform. Therefore, we try to fix up those 10 (1’s and –1’s) non-zero spikes into 10 fixed positions to further reduce the codebook size. Intuitively, it is reasonable to place the non-zero elements with index \( 4n \), i.e., \( X000X000X \ldots \ldots \), where each \( X \) is either 1 or \(-1\). After fixing up the 10 non-zero spikes, the codebook size drastically reduces to \( 2^{10} \), which is quite close to \( 2^9 \). Now we have a 40-dimensional vector with 10 spikes distributed uniformly, and to further reduce the codebook we can flip these spikes up and down in order to have good shaping ability of the noise like speech residual satisfying the Gaussian conditions. Then we impose a restriction: we allow only even numbers of \(-1\)’s out of these 10 non-zero elements. Thus, we have 10 1’s (1 combination), 8 1’s, 2 – 1’s (45 combinations), 6 1’s, 4 – 1’s (210 combinations), 4 1’s, 6 – 1’s (210 combinations), 2 1’s, 8 – 1’s (45 combinations), and 10 –1’s (1 combination). Totally we can have 512 combinations, i.e., \( 2^9 \), which is the required stochastic codebook size. The above discussion cannot be
are optimal gains, and indices $\sum$ is the learning rate, $\gamma$ is the perceptually weighted input speech and $\gamma t$ is the value of update time, $\omega = \gamma - \beta - \gamma f_i(n)$. Instead of minimizing Eqn. (10), we can maximize the match score $MS_i$ (since the free gain term $\gamma$ is multiplied), as given in Eqn. (11).

$$MS_i = \frac{\langle p_{i}, f_i \rangle}{\|f_i\|^2} = |p|^2 \cos^2 \omega \theta_{pf_i}.$$  

The angle between $p$ and $f_i$ should be minimized to maximize Eqn. (11). From the above equation, $|p|^2$ is constant during the codebook search procedure. Since $f_i$ is a linear combination of $q_m$, we can find a vector whose direction is as close as possible to that of a target vector by adding or subtracting $M$ given basis vectors. One of the heuristic techniques that can be adapted to do this work is to set the combination coefficient $\theta_{im}$ to the sign of $\cos \omega \theta_{pf_i}$ for $1 \leq m \leq M$, but there is a possibility that the quality of speech will be degraded too much. This restricts the heuristic decision to fix some condition so that this method determines $\theta_{im}$ only if $q_m$ satisfies the condition below. We notice that $\theta_{im}$ always equals the sign of $\cos \omega \theta_{pf_i}$ if the resemblance is big enough. Besides, the resemblance can be measured by $SP_m$ as follows:

$$SP_m = \|H_m\|^2 \cos^2 \omega \theta_{pf_m}.$$  

Using Eqn. (12) we can restrict the heuristic decision over the combination coefficients $\theta_{im}$, only if $SP_m$ is greater than a threshold. Threshold is normally set to the expected error rate by using statistics of above comparison test. If the heuristic method is used to decide $K$ largest $SP_m$ values among $M$ combination coefficients, the other $M-K$ coefficients can be decided using the original search procedure [10]. Fig. 5 shows the 6.7-kbps VSELP decoder diagram that is very similar to that of the encoder except that there is no closed loop search procedure and adaptive pre and post filters are added in the signal flow [3].
6. DSP Implementation

Today all the voice coding algorithms are implemented on DSP Processors. We discuss the problems encountered in the implementation of the coder on a DSP chip. Even though TMS320C541 is a fixed-point DSP processor, the 40-bit arithmetic logic unit (ALU), two 40-bit accumulators A and B and a barrel shifter are very useful while doing DSP operations such as convolution etc. without losing the precision. The input to the VSELP encoder is a 16 bit PCM, represented in 1.15 format. In real time, 160 samples corresponding to 20ms duration of signal are taken per frame. Every frame is divided into 4 sub-frames, each consisting of 40 samples. The fixed-point representation formats are changed from time to time to take care of the overflow and underflow problems. The input speech signal is passed through a perceptually weighted filter $W(z)$. While accumulating the partial sums, the intermediate stages are represented in 5.11 format to avoid the overflows in data. Then the filtered data is converted back into 1.15 format. While calculating LPC coefficients, the reflection coefficients $\{k_i\}$ are represented in 1.15 format because they are always less than unity. In the codebook search procedure, in order to make the coder computationally effective, we have to bring down the number of instructions per codebook search to an optimum level, without degrading the performance. The MAC instruction of the TMS320C541 takes advantage of writing efficient assembly code for search algorithms with optimized speed. [11]. At each stage of the encoding process, different parameters have to be sent to the decoder. These encoder parameters are stored as words at the intermediate stages. After getting all the parameters, each encoded parameter is converted into bits and grouped to form a bit stream. There are 23 parameter words, which accounts to 133 bits after grouping. Therefore, for every 160 samples (1280 bits) input to the encoder, the encoder output is 133 bits, which is transmitted to the receiver. At the decoder these 133 bits are received as 23 parameter values. These parameters are regrouped to form the excitation signal $e(x(n))$ with proper gains. The required speech $s(n)$ is synthesized using this excitation.

7. Performance of the 6.7-kbps VSELP Coder

The performance of the 6.7-kbps coder is evaluated using WSegSNR measure. During optimization of the basis vectors using steepest descent algorithm, the basis vectors are updated once per every 30 frames. Initially the WSegSNR is 13.65 dB. The value increases to 14.41 dB after 18 iterations (refer Fig. 3). Fig. 6 shows the performance of the coder using fast method, and we can make out that the decrease in WSegSNR is only 0.075 dB when we fix 4 basis vectors heuristically. The total computations required for the stochastic codebook search using $K$-combination coefficients can be calculated as:

$$S_k = 44.2 \times \left\{1 - \left(\frac{M-k}{M-1}\right)\right\} \%,$$

44.2% of the computations are needed to compute match score in every sub-frame (also refer to Tables 2 and 3).

8. Conclusions

A 6.7-kbps VSELP coder is implemented on TMS320C541 DSP processor. It is found that the coder performance does not decrease even though we fix 4 basis vectors heuristically. By fixing 4 basis vectors the computational complexity decreases by 40%. It has been shown that the SNR is enhanced via an iterative training process of the basis vectors in a closed-loop fashion.

9. References


11. Reference manuals of TMS320C54x DSP Processor.

**Table 1: Bit Allocation for 6.7-kbps VSELP Coder**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bits / Sub-frame</th>
<th>Bits / Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC coefficients</td>
<td>-</td>
<td>36</td>
</tr>
<tr>
<td>Frame energy</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>VSELP Code book</td>
<td>9</td>
<td>36</td>
</tr>
<tr>
<td>Adaptive Code book</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>GS-P0-P1</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>Total</td>
<td>23</td>
<td>133</td>
</tr>
</tbody>
</table>

**Table 2: Memory Required for Different Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Memory Occupied (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming Window</td>
<td>170</td>
</tr>
<tr>
<td>Adaptive Codebook</td>
<td>128</td>
</tr>
<tr>
<td>Gain Codebook</td>
<td>256</td>
</tr>
<tr>
<td>Codevectors</td>
<td>360</td>
</tr>
<tr>
<td>Squared gains</td>
<td>128</td>
</tr>
<tr>
<td>Total</td>
<td>1042</td>
</tr>
</tbody>
</table>

**Table 3: Computational Complexity of 6.7-kbps VSELP Vocoder**

<table>
<thead>
<tr>
<th>Application</th>
<th>Gray-code (MIPS)</th>
<th>K-Selection (MIPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>17</td>
<td>13.5</td>
</tr>
<tr>
<td>Synthesis</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Fig. 1. VSELP Encoder

Fig. 2. Stochastic Codebook Search

Fig. 3. Basis Vector Optimization

Fig. 4. Distribution Basis Vectors (Gaussian)

Fig. 5. VSELP Decoder

Fig. 6. Performance of the 6.7-kbps Coder using K-Selection