Complexity Scalability Design in the Internet Low Bit Rate Codec (iLBC) for Speech Coding

Fu-Kun CHEN[†a], Member and Kuo-Bao KUO†, Nonmember

SUMMARY Differing from the long-term prediction used in the modern speech codec, the standard of the internet low bit rate codec (iLBC) independently encodes the residual of the linear predictive coding (LPC) frame by frame. In this paper, a complexity scalability design is proposed for the coding of the dynamic codebook search in the iLBC speech codec. In addition, a trade-off between the computational complexity and the speech quality can be achieved by dynamically setting the parameter of the proposed approach. Simulation results show that the computational complexity can be effectively reduced with imperceptible degradation of the speech quality.

key words: speech coding, speech compression, dynamic codebook, iLBC, complexity scalability

1. Introduction

The speech quality of the analysis-by-synthesis type speech coders [1] depends on the coding of linear predictive coding (LPC) residual signals. Using the adaptive and stochastic codebooks, the code excited linear prediction (CELP) type speech coders [1] can efficiently encode the LPC residual to achieve a good speech quality at low rate but with high computational complexity. For example, ITU-T G.729 [2], [3], G.723.1 [4], GSM Enhanced Full Rate (EFR) speech transcoder [5] and 3GPP AMR codec [6] are typical examples of algebra-CELP (ACELP). The adaptive codebook uses the pass-coded LPC residual to greatly reduce the bit rate of speech communications. However, this data-dependent structure of coding frame results in a degradation of speech quality when the packet loss occurs.

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with Internet. The IETF, supported by the Internet Society (ISOC) recommends the internet Low Bit Rate Codec (iLBC) [7], [8] for speech communications over packet switched network. The iLBC treats each coding frame independently from all others, making it ideal for packet communications. It can improve the speech quality for the situation of packet loss. The commercial software “Skype” [9] has adopted this standard, and it has become a successful product. The iLBC codec is proposed to be considered for different mobile devices. Based on the feature-extracting concept [11], the novel and efficient design of complexity scalability structure for the iLBC speech codec is proposed.

In this paper, Sect. 2 briefly introduces the iLBC speech codec by using the mathematic representation instead of the format as IETF standard. This introduction can help researchers and engineers to understand the iLBC codec. In Sect. 3, the complexity scalability design is proposed for the coding of the dynamic codebook search in iLBC speech codec. In addition, an efficient approach is proposed to effectively reduce the computational complexity for the iLBC codec. In Sect. 4, the complexity and the performance are analyzed for the search procedure of the dynamic codebook. The experimental results will verify the claims. Finally, the conclusions are addressed in Sect. 5.

2. Internet Low Bit Rate Codec (iLBC)

Similar to ITU and MPEG standard, IETF-iLBC speech codec firstly performs the LPC analysis. For interpolation and quantization, the line spectral frequency (LSF) is used for the coding of LPC parameter. Subsequently, the LPC residual is computed by using the quantized and interpolated LPC analysis filter. The LPC filter models the vocal tract and the LPC residual signal models the excitation for
vocal tract. The LPC residual in each coding frame is segmented into 4 and 6 sub-blocks for 20 ms and 30 ms modes, respectively. The two consecutive sub-blocks of the residual processing the maximal weighted energy are then identified. 

For reducing the bit rate, only 57/58 (57 or 58) samples from these two sub-blocks are coded by the scale quantization, the differential pulse code modulation (DPCM). Thus, the samples coded by DPCM is regarded as the start-state and is selected from the first 57/58 samples or the last 57/58 ones of these two sub-blocks for the 20/30 ms (20 ms or 30 ms) modes. The reconstructed DPCM signal will be the elementary codebook to encode the remaining samples. A dynamic codebook encoding procedure is based on the adaptive codebook built from the decoded LPC excitation samples which are the encoded part of sub-blocks. Afterwards, the remaining 23/22 samples in the two sub-blocks containing the start state and the rest of the sub-blocks are sequentially encoded. Thus, the encoding target signal is either the 23/22-sample remaining section or the 40-sample sub-blocks.

The codebooks in iLBC speech codec are the base-codebook, expanded-codebook, the augmented base-codebook and the augmented expanded-codebook. All the codebooks are based on the base-codebook, for example, the expanded-codebook is obtained by filtering the base codebook to form a new one with half delay. In cases of encoding entire sub-blocks, the base and the expanded codebooks are augmented to enrich the codebooks. Thus, the augmented codebooks include the augmented base-codebook and the augmented expanded-codebook. The augmentation vectors in augmented codebooks are constructed in linear combinations between samples corresponding to sample delays ranging from 20 to 39. These codebooks are used in 3 stages in a successive refinement approach, and the resulting three code vector gains are encoded with 5, 4, and 3-bit scalar quantization, respectively.

To find the codevector $c_{i(0)}$ and the gain $g^{(m)}$ for the target signal $x^{(m)}$ in the second stage, the coding criteria is

$$
\text{Min}_i \left[ w^T \left( x^{(m)} - g^{(m)} c_{i(0)} \right) \right]^2
$$

$$
= \text{Min}_i \left[ x^{(m)} - g^{(m)} c_{i(0)} \right]^2, \quad m = 1, \ 2, \ 3, \quad (1)
$$

where $w$ is the perceptually weighted vector, the superscript $T$ denotes the transpose, $x^{(m)} = w \cdot x^{(m)}$ and $c_{i(0)} = w \cdot c_{i(0)}$ mean the window-weighted target signal and the window-weighted codevector. To individually code the gain and codevector, the shape-gain vector quantization is performed. In other words, the index of the optimal codevector is the one that maximizes the normalized correlation term $R^i(m)$ as

$$
\hat{g}^{opt(m)} = \arg \max_i R^i(m) = \arg \max_i \left[ \frac{\langle c_{i(0)}, c_{i(0)} \rangle}{\| c_{i(0)} \|^2} \right]
$$

$$
\equiv \arg \max_i \frac{C_{i(m)}^2}{D_{i(m)}}, \quad m = 1, \ 2, \ 3, \quad (2)
$$

where the numerator $C_{i(m)} = \langle x_{i(0)}, c_{i(0)} \rangle$ is the inner product of the vectors $x_{i(0)}$ and $c_{i(0)}$, the denominator $D_{i(m)} = \| c_{i(0)} \|^2$. Subsequently, the optimal gain would be

$$
g^{opt(m)} = \frac{\langle x^{(m)}, c_{i(m)}^{opt} \rangle}{\| c_{i(m)}^{opt} \|^2}, \quad m = 1, \ 2, \ 3. \quad (3)
$$

The different search ranges have been defined according to the standard of iLBC for different codebooks. The optimal indices, $i_{rem}^{opt(m)}$ and $i_{subblock}^{opt(m)}$, for the $m$-th stage of the remaining section and subblocks respectively are

$$
i_{rem}^{opt(m)} = \arg \max_i \left[ R^{i_{rem}^{opt(m)}}, R^{i_{rem}^{opt(m)}} \right], \quad i_{subblock}^{opt(m)} = \arg \max_i \left[ R^{i_{subblock}^{opt(m)}}, R^{i_{subblock}^{opt(m)}} \right], \quad (4)
$$

where $R^{i_{base}^{opt(m)}}$, $R^{i_{exp}^{opt(m)}}$, $R^{i_{aug}^{opt(m)}}$, and $R^{i_{aug}^{expm}}$ denote the optimal indices for the coding of the base-codebook, the expanded-codebook, the augmented base-codebook and the augmented expanded-codebook respectively. Furthermore, the target vectors in the $m$-th stage, $x_{rem}^{(m)}$ and $x_{subblock}^{(m)}$ for the coding of the remaining section and subblocks respectively would be updated as

$$
x_{rem}^{(m)} = x_{rem}^{(m-1)} - g^{(m-1)} x_{rem}^{(m-1)} c_{i_{rem}^{opt(m-1)}},
$$

$$
x_{subblock}^{(m)} = x_{subblock}^{(m-1)} - g^{(m-1)} x_{subblock}^{(m-1)} c_{i_{subblock}^{opt(m-1)}}, \quad (5)
$$

where $g^{(m-1)}$ and $g^{(m-1)}$ are the quantized coding gains for the coded vectors, $c_{i_{rem}^{opt(m-1)}}$ and $c_{i_{subblock}^{opt(m-1)}}$ of the remaining section and subblocks, respectively. In addition, the base-codebook $[C_{base}]$ needs to be updated when each the 3-stage coding is finished to sequentially form the base-codebook buffer as

$$
[C_{base}] \rightarrow [C_{base}] : \sum_{m=1}^{3} [g_{rem}^{opt(m)}, g_{subblock}^{opt(m)}] \cdot (6)
$$

To analyze the computation distribution, Fig. 1 shows that the dynamic codebook search is the major computational load. Therefore, IETF provides several simplified schemes called the restricted mode in the document RFC3951. The simplified scheme is to restrict the search range for the expanded codebook in the interval

![Fig. 1 Computation distribution of the iLBC encoder. (20/30 ms modes)](image-url)
To adjust the computational complexity of the iLBC coder

3. Complexity Scalability Design

To adjust the computational complexity of the iLBC coder according to the capability of the computing kernel, the numerator of the criterion (2) can be modified as

\[
C_i^{(m)} = \sum_{s=0}^{S-1} \sum_{n=0}^{40(S-1)} x_i^{(m)}(40s + n) c_i(40s + n)
\]

where \( x_i^{(m)}(n) \) and \( c_i(n) \) are the elements of the vectors \( x_i^{(m)} \) and \( c_i^{(m)} \) respectively, \( S \) is the number of segments and

\[
E_s^{(m)} = \sum_{n=0}^{40S-1} x_s^{(m)}(40s + n) c_i(40s + n)
\]

is the inner product for the \( s \)-th segment. If the value \( E_s^{(m)} \) can be accumulated according to the order of importance, then the numerator of the criterion (2) can be rewritten as

\[
C_i^{(m)} = \sum_{s'=0}^{S-1} E_s^{(m)},
\]

where \( s' \) is the reordered index according to the pilot function. However, a pilot function must be found to decide the order of accumulation (7). The pilot function will pick the coarse information around each segment and is suggested as

\[
p(s) = \sum_{n \in R_k} \Lambda \left( x_i^{(m)}(n) \right), \quad s = 0, 1, \ldots, S - 1,
\]

where \( R_k \) denotes the \( k \)-th region of the coding frame and \( \Lambda (\cdot) \) is a condensing function. Because the maximal amplitude of the LPC residual signal needs to be compensated, the pilot function can be designed according to the magnitude of target signal. Considering the computational load, a simplest pilot function is suggested as

\[
p(s) = \text{Max}_{n \in R_k} |x(n)|, \quad s = 0, 1, \ldots, S - 1.
\]

Figure 2 illustrates an example of the target signal and the corresponding pilot function in the case of \( S = 8 \). The pilot function indicates a better order of the accumulation (2) for this subblock. In Fig. 2, the number marked in each segment indicates the priority of the accumulation (2) of the segment in this coding subblock.

When the performance of the computation kernel is poor, the first \( N \) terms of \( E_s^{(m)} \) can achieve a good approach for (7). Thus, the criterion (2) is modified as the proposed complexity scalability iLBC (CS-iLBC)

\[
p_{\text{opt}}^{(m)} = \arg \max_i \left( \frac{\sum_{s'=0}^{N-1} E_s^{(m)}}{D_i} \right), 1 \leq N \leq S,
\]

where \( s' \) is the sorted index according to the pilot function. In the modified criterion (10), the parameter \( N \) results in the complexity scalability property of the dynamic codebook search. The value, \( 1 - \frac{N}{S} \), is the ratio of the complexity reduction. The unselected \( S - N \) segments will cause the quality degradation. In this approach, the \( N \) segments are considered as the important parts and selected to perform (10). In other words, the parameter \( N \) controls the trade-off between the computational complexity and the quality of the coded speech. It is noted that since the remaining section contains only 22/23 samples, to divide it into \( S \) segments is not necessary. Therefore, the proposed CS-iLBC is only performed in the coding of the 40-sample subblocks.

Because the codevectors \( c_{w(i)} \) are extracted from the sequential coded residual samples, the value \( D_i \) in (2) can be successively computed at a cost of only two multiplications and two additions. Considering the goal of complexity reduction, it is not necessary to modify the computation of \( D_i \). However, the mismatch of the number of elements in \( \sum_{s'=0}^{N-1} E_s^{(m)} \) and \( D_i \) may degrade the coding performance.

To find the trade-off for the computation reduction and the performance, the value \( N \) in (10) can be decided stage-by-stage and subblock-by-subblock. The proposed dynamic CS-iLBC (DCS-iLBC) is to dynamically decide the value \( N \) according to the pilot function. For example, when the descending sorted values of the pilot function become smaller than \( r \) times the global maximum \( p_{\text{max}}(s) \), the accumulating process of the criteria (10) can be terminated. Therefore, the selected indices of the sorted pilot function are
When the ratio is \( r = 0 \), the criterion (10) is identical to (2). A simple and reasonable setting is \( r = 0.5 \) so that the factor \( r \) can be implemented by a right-shift operation. Furthermore, the proposed DCS-iLBC can be integrated with RiLBC to turn into the DCS-RiLBC method.

4. Performance Analysis

The performance comparison for a fast algorithm of the speech codec is focusing on the quality of the coded speech and the reduction of the complexity. In the following, the fixed-point implementation specified by IETF\[7\] is chosen to analyse the performance. According to our informal mean opinion score (MOS) test, the proposed approaches can reduce the computational complexity of the dynamic codebook search in IETF-iLBC with perceptually intangible degradation for speech quality. In addition, the MOS-listening quality objective (MOSLQO) score specified by ITU-T P.862.1\[18\] is used to evaluate the speech quality. ITU-T P.862.1 has mapped the row P.862\[19\] scores, perceptual evaluation of speech quality (PESQ), to MOSLQO scores and provides a linear comparison with MOS. To subjectively compare the performance, the normalized degradation (ND) of MOSLQO is proposed and defined as

\[
ND(\%) = \frac{1}{K} \sum_{k=1}^{K} \frac{MOSLQO(k) - MOSLQO_{iLBC}(k)}{MOSLQO_{iLBC}(k)} \times 100\%,
\]

where \( K \) is the number of test speech files, \( MOSLQO(k) \) and \( MOSLQO_{iLBC}(k) \) are the MOSLQO values for the \( k \)-th test file using the desired method and the iLBC, respectively.

Figure 3 and 4 show the normalized degradation of MOSLQO at different modes for the different values of \( S \) and \( N \) by using 23 test speech files provided by ITU-T P.862. These test speech files can be downloaded on the website of ITU\[20\]. As shown in Fig. 3 and 4, the performance of RiLBC is worse than that of iLBC because the search range is narrowed. The performance of the proposed CS-iLBC and CS-RiLBC can be improved by increasing the number of segments \( S \). Furthermore, the performance of the CS-iLBC and CS-RiLBC can be enhanced while the number of used segments \( N \) increases under the fixed \( S \), i.e. the ratio \( N/S \) increases. Furthermore, the DCS-iLBC and DCS-RiLBC can both improve the performance for CS-iLBC and CS-RiLBC slightly.

To evaluate the performance of the DCS-iLBC and DCS-RiLBC in detail, Table 1 and 2 show the experimental results at different modes. In these tables, the average MOSLQO, ND, the average \( N \) and the corresponding complexity are compared. The standard deviation of ND is
Table 1 Performance comparison of DCS-iLBC and DCS-RiLBC at the 20 ms mode.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOSLQO</th>
<th>ND (%)</th>
<th>Std of ND</th>
<th>Avg. N</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>iLBC</td>
<td>4.0824</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>100 %</td>
</tr>
<tr>
<td>CS-iLBC (S = 20)</td>
<td>4.0761</td>
<td>-0.1498</td>
<td>0.6253</td>
<td>9</td>
<td>45.0%</td>
</tr>
<tr>
<td>DCS-iLBC (S = 20)</td>
<td>4.0826</td>
<td>+0.0172</td>
<td>0.7040</td>
<td>8.8628</td>
<td>44.3%</td>
</tr>
<tr>
<td>RiLBC</td>
<td>4.0778</td>
<td>-0.1141</td>
<td>0.5663</td>
<td>-</td>
<td>68.2%</td>
</tr>
<tr>
<td>CS-RiLBC (S = 20)</td>
<td>4.0538</td>
<td>-0.7012</td>
<td>1.0071</td>
<td>9</td>
<td>30.7%</td>
</tr>
<tr>
<td>DCS-RiLBC (S = 20)</td>
<td>4.0594</td>
<td>-0.5591</td>
<td>0.7511</td>
<td>8.8943</td>
<td>30.3%</td>
</tr>
</tbody>
</table>

*: Experimental result

also listed to indicate the performance variation. Since the RiLBC is recommended by IETF, its performance can be considered as a comparison basis. In addition, two cases, CS-iLBC and CS-RiLBC with \( S = 20 \) and \( N = 9 \), are also compared because the number of used segments for DCS-iLBC and DCS-RiLBC is within the interval \( 8.8 \sim 8.9 \) in average. The worst case of the used segments is about \( N = 12 \) in some subblocks. However, theoretically, there will be \( N = 10 \) in average for the case of \( S = 20 \). The experimental result is \( N = 8.8 \sim 8.9 \) in average. It indicates that the DCS-iLBC and DCS-RiLBC methods have lower complexity than CS-iLBC and CS-RiLBC methods.

According to Table 1 and 2, when the segmentation reaches \( S = 20 \), the proposed DCS-iLBC can reduce the computational load for coding subblocks about 55.7% at the 30 ms and the 20 ms modes and the MOSLQO score remains almost the same as that of RiLBC. The DCS-RiLBC can reduce the computational load for coding subblocks about 70% at the 30 ms and the 20 ms modes. However, its MOSLQO score is slightly worse than that of RiLBC, but such degradation in MOSLQO is perceptually negligible. In addition, the DCS-iLBC and DCS-RiLBC also improve the deviation of the performance for CS-iLBC and CS-RiLBC according to the observation on the standard deviation of ND.

5. Conclusions

In this paper, the complexity scalability of IETF-iLBC speech coder is proposed. The proposed approaches can reduce the computational complexity of the coding of dynamic codebook for the speech coder IETF-iLBC with perceptually negligible degradation of the speech quality. The experimental results using the MOSLQO score of the speech quality assessment specified by ITU-T P.862.1 for the proposed approaches have verified the claims. Furthermore, based on the property of scalability in computational complexity, the proposed approaches would be especially helpful in reducing power consumption and relaxing the CPU load of the hand-held mobile devices.

Acknowledgments

This work was supported by the National Science Council of Republic of China under the research contract 97-2221-E-218-02.

References

2nd IASTED Int. Conf. on Communications, Internet and Information Technology, pp.372–375, Nov. 2003.


Fu-Kun Chen was born in Tainan, TAIWAN, in 1967. He received the B.S. degree from the National Taiwan Ocean University in 1991, the M.S. and Ph.D. degrees from the National Cheng-Kung University in 1993 and 2001, respectively, all in electrical engineering. From 1993 to 1995, he joined the Chinese Army to be a Lieutenant. And then, he worked for Holtek Semiconductor Ins. as an ASIC design engineer during 1995–1996, and worked for Chunghwa Telecom Co. Ltd. in the GSM/DCS maintenance center during 1998–2002. Currently, he is an associate professor of Southern Taiwan University. His main teaching and research are speech/video coding, adaptive signal processing, and fast algorithm design. He is the member of IEEE.

Kuo-Bao Kuo was born in Tainan, Taiwan, in 1985. He received the B.S. degree from the Southern Taiwan University in 2009. He is interested in the speech coding and the interactive website design.