A Novel Hierarchical Decomposition Vector Quantization Method for High-Order LPC Parameters

Lin Wang, Zhe Chen, *Senior Member, IEEE*, and Fuliang Yin

***Abstract*—The paper investigates vector quantization coding of high-order (e.g., 20th-50th order) linear prediction coding (LPC) parameters, and proposes a novel hierarchical decomposition vector quantization method for a scalable speech coding framework with variable orders of LPC analysis. Instead of vector quantizing the whole group of LPC parameters in the linear spectral frequency (LSF) domain directly, the proposed method decomposes the high-order LPC model into several low-order (e.g., 10th-order) LPC models, and vector quantizes them in the LSF domain separately. For the decomposition, the high-order LPC model is converted into a group of reflection coefficients at first, and then the group is split into several subgroups and converted into multiple low-order LPC models. It is shown that the proposed method is naturally suitable for a scalable coding framework where the information of the decomposed low-order LPC models can be encoded into a multi-layered bitstream and can be combined in a progressive way to recover the high-order LPC information. Experiments in a scalable coding framework with variable LPC analysis orders (10-50) reveal that, compared to a direct vector quantization scheme, the proposed method can reduce the size of the codebook and the number of coding bits significantly, and can also efficiently reduce the computation cost.**

***Index Terms*—Line spectral frequency, linear prediction coding (LPC), reflection coefficient, scalable coding, vector quantization.**

I. Introduction

LINEAR prediction plays an important role in modern speech processing technology [1], [2]. Linear predictive coding (LPC) coefficients can represent the short-time spectral envelope information of speech signals efficiently and are widely used in code-excited linear prediction (CELP) speech codecs, such as G.728, G.729, AMR, and EVRC [3]–[6]. To reduce the bit rate, LPC parameters are generally vector quantized in the linear spectral frequency (LSF) domain, which shows advantages of robustness and interpolation [7]. Many mature vector quantization algorithms have already been proposed for the quantization of the LSF parameters, such as split vector quantization, multistage vector quantization, and predictive vector quantization [8]–[15]. The precision of LPC spectral estimation is closely related to the harmonic structure of source signals as well as the LPC order. In low-bit-rate speech codecs, which typically operate in the frequency range of 300–4000 Hz, a 10th-order LPC model can generally describe a speech signal well enough. However, for signals with a wider frequency range (e.g., 50–8000 Hz) or with more harmonic structures and spectral components (e.g., speech signals with background music), it is difficult to represent all the formant information with only 10 parameters. In this case, low-order LPC analysis will lead to unnatural regeneration of speech and hence it is necessary to increase the analysis order in order to capture all the formant information required for high-quality speech/audio synthesis [16], [17].

In recent years, the continuous increase of available data rates in telecommunication systems has allowed enhancing user experiences with for example the transmission of wideband speech/audio in high fidelity. In addition, to be compatible with different transmission bandwidth conditions, scalable speech codec techniques have emerged in recent years, which generate a multi-layered bitstream format with each additional layer successively improving the speech/audio quality [18]–[20]. Existing scalable speech codecs usually operate in low and high frequency bands separately and allocate most of the bandwidth to excitation parameters. However, in future speech codecs with higher bit rates, it is possible to allocate extra bandwidth to LPC information. This creates a more detailed LPC spectrum, which can further improve the speech synthesis quality. For now, high-order LPC analysis techniques have rarely been employed in a scalable speech codec. There are mainly two reasons for that: 1) The computation and memory cost increases significantly with the LPC analysis order when quantizing the LPC parameters in the LSF domain; 2) The current LPC quantization scheme (vector quantizing in the LSF domain directly) is unsuitable for the scalable speech coding framework.

This paper investigates vector quantization coding of high-order LPC parameters and proposes a novel vector quantization method that is naturally suitable for a scalable coding framework with variable LPC orders. The main idea is to decompose a high-order LPC model into several low-order ones, and then to vector quantize them in the LSF domain separately. The decomposition is performed with the help of an intermediate parameter

Manuscript received October 09, 2014; revised November 30, 2014; accepted December 05, 2014. Date of publication December 09, 2014; date of current version January 14, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61172107, 61172110, in part by the Fundamental Research Funds for the Central Universities China under Grant DUT13LAB06, in part by the National High Technology Research and Development Program (863 Program) of China, and in part by the Alexander von Humboldt Foundation in Germany. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Engin Erzin.

L. Wang is with the School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116023, China, and also with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: wanglin_2k@sina.com).

Z. Chen and F. Yin are with the School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: zhechen@dlut.edu.cn; flyin@dlut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2380352

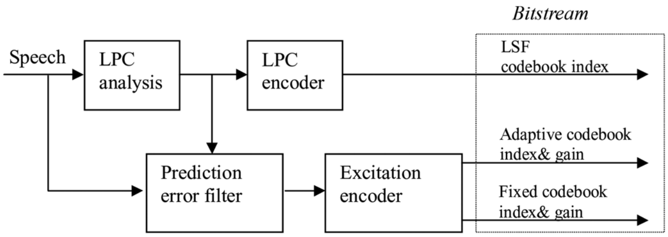


Fig. 1. Basic components of a CELP encoder.

(RC - reflection coefficient). The high-order LPC model is transformed to a group of RCs, which are then divided into several subgroups. Each subgroup of RCs is further transformed to a low-order LPC model and vector quantized in the LSF domain. The feasibility of the conversion between RC, LPC and LSF in each subgroup will be proved theoretically in the paper. The obtained low-order LPC models can be encoded in a multi-layered bitstream, which aims to successively improve the LPC analysis precision. Therefore, the proposed method is referred to as a *hierarchical decomposition vector quantization* method. By decomposing a high-order LPC model into low-order ones, the proposed method works more efficiently in a scalable coding framework than traditional methods, which vector quantize the high-order LPC model directly.

The rest of the paper is organized as follows. The theoretical background that is related to the proposed method is reviewed in Section II. The hierarchical decomposition vector quantization method is described in detail in Section III. Experiments are carried out to evaluate the performance of the proposed method in Section IV. Finally, conclusions are drawn in Section V.

II. Theoretical Background

In this section, the principle of CELP is introduced at first, after which some properties of LPC and its alternative representations, i.e., line spectral frequencies (LSFs) and reflection coefficients (RCs), are reviewed to help understand the proposed method.

A. CELP and LPC Order

Code-excited linear prediction has been widely used in speech coding [21]. The basic principle of a CELP encoder is shown in Fig. 1, which can be divided into two paths. In the upper path, the speech signal is segmented into short-time frames and LPC analyzed to compute the LPC parameters. The LPC parameters are converted into LSFs and vector quantized. In the lower path, the frames of speech are passed through a prediction error filter, which is constructed from the LPC parameters, obtaining the excitation residuals, which are subsequently encoded in the form of adaptive and fixed codebook indices and gains. The LPC encoder and the excitation encoder are two core blocks of the CELP encoder. The two blocks are relatively independent. The aim of the LPC encoder is to minimize the error between the original (unquantized) LPC spectrum and the reconstructed (quantized) LPC spectrum. The aim of the excitation encoder is to minimize the error between the original and reconstructed speech according to a perceptually weighted distortion measure. LPC analysis, which connects the LPC encoder block and the excitation encoder block, plays an important role on the performance of the whole CELP encoder.

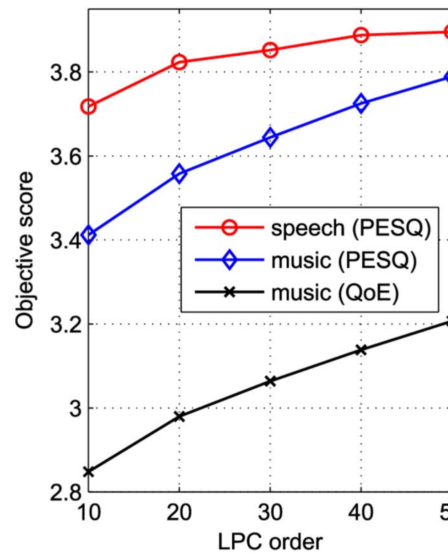


Fig. 2. Objective evaluation scores for speech (PESQ) and music (PESQ and QoE) encoding at different LPC analysis orders in a G.728 framework.

To show the influence of LPC analysis order on the speech codec quality, we use a modified G.728 speech codec, where the LPC order can be varied, to encode 5 segments of speech files and 5 segments of music files (each about 10 s long, 16 kHz sampling) at different LPC analysis orders increasing from 10 to 50, and calculate the perceptual evaluation of speech quality (PESQ) values of the decoded files [22]. (The details of the modified G.728 will be described in the Appendix.) Fig. 2 shows the average PESQ values of the speech and music files at different LPC orders. It can be clearly seen from Fig. 2 that speech obtains a higher PESQ value than music overall, and both of them show improved PESQ values with the increase of the LPC order. For speech, the improvement of PESQ becomes slow when the LPC order is greater than 40. For music, the PESQ improves continuously with the LPC order. Considering that PESQ is mainly designed for testing speech quality, we use another objective quality of experience (QoE) measure to evaluate the music quality [23]. The QoE measure is a linear combination of PESQ and perceptual evaluation of audio quality (PEAQ) [24], resulting in mean opinion score (MOS)-like values. The QoE results for music are also plotted in Fig. 2. Similar to the PESQ value for music, the QoE value also improves constantly with the LPC order. In a short summary, the observations above verify the utility of high-order LPC analysis in speech coding[1].

It should be noted that in G.728 standard, the 50-th order LPC information is not encoded and transmitted in the bitstream. When synthesizing speech in the decoder, the LPC information of each frame is approximated by the LPC parameters analyzed from previously decoded speech frames. In this way, the high computation and coding bits consumption of encoding high-order LPC parameters is avoided at the cost of inaccurate

[1]Some audio demos are available online [28].

LPC information. Although the aim of the paper is not to propose an alternate of G.728, we believe that if the LPC information is available in the bitstream, the speech codec quality can be further improved.

In this paper, we investigate how to efficiently encode the high-order LPC information in a scalable framework so that the spectral information can be reconstructed and improved progressively. The excitation encoder block will not be addressed in this paper since it is independent of the LPC encoding block.

B. LPC and LSF

Linear prediction analysis is a powerful tool used frequently in audio and speech signal processing for representing the spectral envelope. In linear prediction models, a short-time segment of speech is assumed to be generated as the output of an all-pole filter $H(z) = 1/A(z)$, where $A(z)$ is the prediction error filter defined by a set of M-order LPC parameters $\boldsymbol{a}_M = [1, a_{M,1}, \cdots, a_{M,M}]^{\mathrm{T}}$, i.e.,

$$A(z) = 1 + a_{M,1}z^{-1} + \cdots + a_{M,M}z^{-M}, \quad (1)$$

where $\boldsymbol{a}_M$ is generally calculated from the autocorrelation information of the signals with the Levinson-Durbin algorithm [1], [25], which is simply given below.

In the Levinson-Durbin algorithm, the autocorrelation vector for a time sequence $s(n)$ is defined as $\boldsymbol{r}_m = [r_0, r_1, \cdots, r_m]^{\mathrm{T}}$, with the i-th element $r_i = \mathrm{E}\{s(n)^{\mathrm{T}}s(n-i)\}$, where $\mathrm{E}\{\cdot\}$ denotes mathematical expectation. The M-order LPC parameters $\boldsymbol{a}_M = [1, a_{M,1}, \cdots, a_{M,M}]^{\mathrm{T}}$ can be estimated by using the recursions in (2):

$$\begin{cases} k_m = \frac{(\boldsymbol{a}^{\mathrm{B}}_{m-1})^{\mathrm{T}}\boldsymbol{r}_{m-1}}{P_{m-1}} \\ P_m = r_0 \prod_{i=1}^{m}(1-k_i^2) \\ \mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + k_m \begin{bmatrix} 0 \\ \mathbf{a}^{\mathrm{B}}_{m-1} \end{bmatrix} \end{cases}, \quad (2)$$

where m is the recursion order ascending from 1 to M; $\boldsymbol{a}_m = [1, a_{m,1}, \cdots, a_{m,m}]^{\mathrm{T}}$ is the vector of LPC parameters obtained at the m-th recursion and $\boldsymbol{a}^{\mathbf{B}}_m = [a_{m,m}, \cdots, a_{m,1}, 1]^{\mathrm{T}}$ is the order-reversed version of $\boldsymbol{a}_m$; k_m is the m-th reflection coefficient and always satisfies $|k_m| < 1$.

In speech coding, the LPC parameters are generally vector quantized in the LSF domain to reduce the data rate, because LSFs have a smaller dynamic range. In addition, the LSF representation is more robust to quantization than the LPC parameters and in difference to the LPC parameters, LSFs can be interpolated between frames without loss of stability of the predictor. In order to define LSF, the prediction error filter polynomial $A(z)$ is used to construct two polynomials:

$$\begin{cases} P(z) = A(z) + z^{-(M+1)}A(z^{-1}) \\ Q(z) = A(z) - z^{-(M+1)}A(z^{-1}) \end{cases}, \quad (3)$$

The roots of the polynomials $P(z)$ and $Q(z)$ are called the linear spectral frequencies (LSFs). The polynomials $P(z)$ and $Q(z)$ have two properties: 1) All zeros of $P(z)$ and $Q(z)$ lie on the unit circle; 2) Zeros of $P(z)$ and $Q(z)$ are interlaced with each other, so that the LSFs are in ascending order. These properties help in efficient numerical computation of the LSFs from $P(z)$ and $Q(z)$. It is shown in [1] that $A(z)$ has the minimum phase property, i.e., all zeros of $A(z)$ must be inside the unit circle in the z-plane, when its LSFs satisfy these two properties. In this case, the stability of LPC synthesis filter can be easily ensured by quantizing the LPC information in LSF domain. Thus, the minimum phase property of $A(z)$ is a prerequisite for the conversion between LPC parameters and LSFs. This requirement is called the *Minimum Phase Condition*.

C. LPC and Reflection Coefficient

As a by-product of the Levinson-Durbin algorithm, the reflection coefficients (RCs) in Eq. (2) are important parameters when representing an LPC predictor as a lattice structure. The relationship between RC and LPC is given below.

On the one hand, the M-order LPC parameters $\boldsymbol{a}_M = [1, a_{M,1}, \cdots, a_{M,M}]^{\mathrm{T}}$ can be uniquely determined by a set of M reflection coefficients $\boldsymbol{k}_M = [k_1, \cdots, k_M]^{\mathrm{T}}$ by using the recursion in (4):

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + k_m \begin{bmatrix} 0 \\ \mathbf{a}^{\mathrm{B}}_{m-1} \end{bmatrix}, \quad (4)$$

where m is the recursion order ascending from 1 to M; $\boldsymbol{a}_m = [1, a_{m,1}, \cdots, a_{m,m}]^{\mathrm{T}}$ is the vector of LPC parameters obtained at the m-th recursion and $\boldsymbol{a}^{\mathrm{B}}_m = [a_{m,m}, \cdots, a_{m,1}, 1]^{\mathrm{T}}$ is the order-reversed version of $\boldsymbol{a}_m$; k_m is the m-th reflection coefficient. On the other hand, $\boldsymbol{k}_M$ can also be calculated from $\boldsymbol{a}_M$ by using the recursions in (5):

$$\begin{cases} k_m = a_{m,m} \\ \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} = \frac{\mathbf{a}_m - k_m\mathbf{a}^{\mathrm{B}}_m}{1-|k_m|^2} \end{cases}, \quad (5)$$

with the recursion order m descending from M to 1.

There are two important properties of RCs.

1) With Eqs. (4) and (5), RCs and LPC parameters can be converted to each other freely. In addition, it is shown in [26] that a prediction error filter $A(z)$ constructed from $\boldsymbol{a}_M$ is minimum phase if and only if its corresponding RC set $\boldsymbol{k}_M$ satisfies $|k_m| < 1$ for $m = 1, \cdots, M$.
2) The values of the RCs are independent of the LPC analysis order. For instance, two RC sets $\boldsymbol{k}_{M_1}$ and $\boldsymbol{k}_{M_2}$, as a result of applying M_1-order and M_2-order ($M_1 < M_2$) LPC analysis to a same signal respectively, will share the same first M_1 elements. This *Order Invariant Property* can be exploited for a scalable speech coding framework.

III. Hierarchical Decomposition Vector Quantization Method

Split vector quantization (SVQ) is an efficient method to vector quantize the high-order LPC parameters, which can balance between codebook size and bit rate [8]. However, this method is not suitable in a scalable coding framework where the LPC analysis precision is supposed to be improved progressively with a multi-layered bitstream. For such an application, a hierarchical decomposition vector quantization (HDVQ) method is proposed. We will introduce the split vector quantization method at first and then present the proposed method in detail.

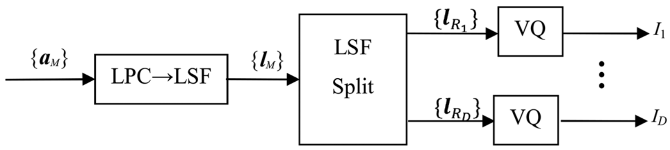


Fig. 3. Block diagrams of the encoding part of split vector quantization.

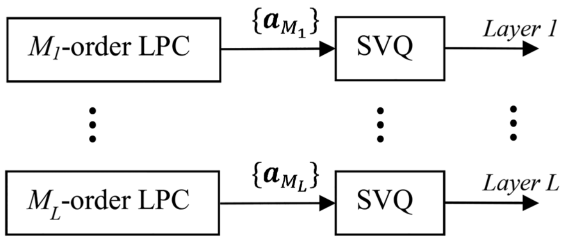


Fig. 4. Using split vector quantization in a scalable coding framework.

A. Split Vector Quantization

The block diagram of the encoding part of split vector quantization is shown in Fig. 3. After converting from LPC parameters to LSFs, the obtained LSF group $\boldsymbol{l}_M = \{l_{M,1}, \cdots, l_{M,M}\}$ is split into D subgroups $\{\boldsymbol{l}_{R_1}, \cdots, \boldsymbol{l}_{R_D}\}$, where the subscript R_d is the size of the d-th subgroup $\boldsymbol{l}_{R_d}$ and $\sum_{d=1}^{D} R_d = M$. Each LSF subgroup $\boldsymbol{l}_{R_d}$ is then vector quantized and represented with an index I_d of the corresponding codebook. With fewer LSFs in each subgroup, the size of the codebook for each subgroup decreases significantly. In this way, the LPC parameters can be well vector quantized even if the LPC order is high.

The application of split vector quantization in a multi-layered scalable LPC coding framework is shown in Fig. 4. In each layer, the LPC parameters obtained at a specific LPC order ($M_1 < \cdots < M_L$) are encoded separately with split vector quantization. The precision of LPC analysis improves with the increase of the layer and LPC order. The number of bitstream layers to be transmitted will depend on the network transmission status. Although split vector quantization can reduce the codebook size, it is still inefficient in this scalable coding framework since in essence the low-order LPC information has been encoded repeatedly in higher layers.

B. Proposed Method

With the aim of scalable coding, a hierarchical decomposition vector quantization method is proposed. The main idea is to decompose a high-order LPC model into several low-order LPC models, whose coefficients are then vector quantized in the LSF domain separately. Since an LPC model should satisfy the *Minimum Phase Condition* to be able to convert to LSFs, we use an intermediate variable (i.e., RC) to assist the decomposition. Specifically, the higher-order LPC model is converted to a group of RCs, which is then divided into subgroups. Next, each RC subgroup is converted to a low-order LPC model and then vector quantized in the LSF domain. It can be proved in Section III-C that with this intermediate variable the obtained low-order LPC models will satisfy the *Minimum Phase Condition* and hence can be converted to LSFs for vector quantization. The decomposed low-order LPC models can be encoded into different layers of the bitstream. Furthermore, based on the *Order Invariant Property* of reflection coefficients, the RCs from each subgroup can be combined to construct a high-order LPC model. In the following paragraphs, the proposed method is presented in encoding and decoding parts respectively.

Encoding Part: For an M-order LPC analysis with its LPC parameters $\boldsymbol{a}_M = \{1, a_{M,1}, \cdots, a_{M,M}\}$, the encoding part is shown in Fig. 5, consisting of 5 steps.

E1: Convert the LPC model $\boldsymbol{a}_M$ into a group of RCs $\boldsymbol{k}_M = \{k_{M,1}, \cdots, k_{M,M}\}$ by using the recursion (5).

E2: Split $\boldsymbol{k}_M$ into D subgroups, i.e., $\boldsymbol{k}_M = \{\boldsymbol{k}_{R_1}, \cdots, \boldsymbol{k}_{R_D}\}$, where R_d denotes the size of the d-th subgroup. We choose equal size for all the subgroups, i.e., $R_d = R$ for $d = 1, \cdots, D$, and $M = RD$. The components of the d-th subgroup is expressed as $\boldsymbol{k}_{R_d} = \{k_{R_d,1}, \cdots, k_{R_d,R_d}\} = \{k_{M,R(d-1)+1}, \cdots, k_{M,R(d-1)+R}\}$. Each subgroup is processed respectively in Step 3–5.

E3: Convert the d-th RC subgroup $\boldsymbol{k}_{R_d}$ into an R_d-order LPC model $\boldsymbol{a}_{R_d} = \{1, a_{R_d,1}, \cdots, a_{R_d,R_d}\}$ by using the recursion (4).

E4: Convert $\boldsymbol{a}_{R_d}$ to LSFs $\boldsymbol{l}_{R_d} = \{l_{R_d,1}, \cdots, l_{R_d,R_d}\}$.

E5: Vector quantize the LSFs $\boldsymbol{l}_{R_d}$, where the vector $\boldsymbol{l}_{R_d}$ can be represented by an index I_d of a codebook.

The RC-LPC-LSF conversion and vector quantization (the steps 3–5) are repeated for all the D subgroups. Finally $\mathbf{a}_M$ is represented by D indexes $\{I_1, \cdots, I_D\}$. For scalable coding, each index I_d can be encoded into a separate layer of the bitstream. Regarding the value of D, we typically choose $R = 10$ and $D = M/10$, because a large amount of works have already been done on the vector quantization of 10-th order LPC models and can be used directly. Apart from this, other values of D can also be used as appropriate.

For vector quantization at each subgroup, there are two schemes to design the codebook.

Scheme 1: Considering the fact that in each subgroup we have the same number of LSF parameters distributed in the interval $(0, \pi)$, a shared codebook can be designed for all the subgroups. This scheme can reduce the total size of the codebook significantly. However, the vector quantization efficiency is low since the probability distribution of the LSF parameters in each subgroup may be different.

Scheme 2: Considering the fact that the probability distribution of the LSF parameters in each subgroup may be different, an individual codebook can be designed for each subgroup separately. This scheme can increase the vector quantization efficiency. However, the total size of the codebook also increases with the number of subgroups.

Decoding Part: The decoding part of the proposed method is shown in Fig. 6

D1: Given the indexes $\{I_1, \cdots, I_D\}$, the D subgroups of LSFs $\{\boldsymbol{l}_{R_1}, \cdots, \boldsymbol{l}_{R_D}\}$ can be recovered with inverse vector quantization.

D2: At each subgroup, the LSF vector $\boldsymbol{l}_{R_d}$ is converted to the LPC model $\boldsymbol{a}_{R_d}$ and then to the RC set $\boldsymbol{k}_{R_d}$ using the recursion (4).

D3: The D subgroups of RC are combined to get a whole group $\boldsymbol{k}_M = \{\boldsymbol{k}_{R_1}, \cdots, \boldsymbol{k}_{R_D}\}$.

D4: Finally, the RC set $\boldsymbol{k}_M$ is converted to the LPC model $\boldsymbol{a}_M$ using the recursion (5).

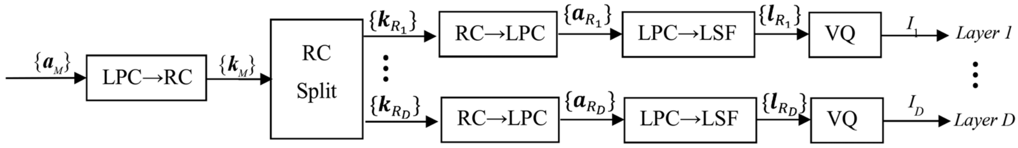


Fig. 5. Block diagram of the encoding part of the proposed hierarchical decomposition vector quantization method.

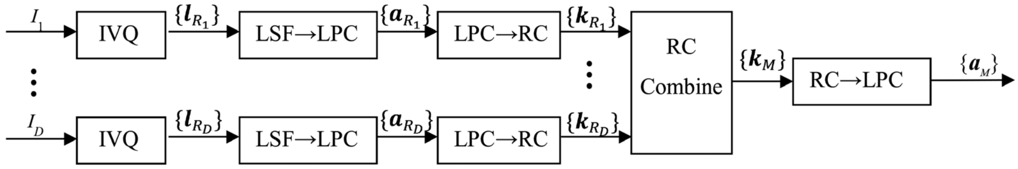


Fig. 6. Block diagram of the decoding part of the proposed hierarchical decomposition vector quantization method.

The LSF-LPC-RC-LPC conversion in the decoding part is completely an inverse procedure of the encoding part. Based on the *Order Invariant Property*, the information of the RCs at different layers can be combined to improve the LPC analysis precision progressively in the decoding part.

C. Remarks on the Proposed Algorithm

In the encoding part of the proposed method, each subgroup of RCs is converted to LPC parameters and then converted to LSFs for vector quantization. As mentioned in Section II, LPC parameters and RCs can be converted to each other freely, while the conversion between LPC parameters and LSFs should satisfy the *Minimum Phase Condition*. A proof of the equivalence between LPC parameters and LSFs in each subgroup is provided below.

In the step (E1), the M-order LPC model $\boldsymbol{a}_M$ obtained by the Levinson-Durbin algorithm satisfies the *Minimum Phase Condition*, meanwhile the corresponding RC set $\boldsymbol{k}_M$ also satisfies that $|k_{M,m}| < 1$, for $m = 1, \cdots, M$.

In the step (E2), since all the elements of the d-th RC subgroup $\boldsymbol{k}_{R_d}$ come from the whole group $\boldsymbol{k}_M$, they also satisfy that $|k_{R_d,m}| < 1$, for $m = 1, \cdots, R_d$.

In the step (E3), a new R_d-order LPC model $\boldsymbol{a}_{R_d}$ is uniquely determined from the RC set $\boldsymbol{k}_{R_d}$. Based on the first property for RCs in Section II-C, the new LPC model $\mathbf{a}_{R_d}$ also satisfies the *Minimum Phase Condition* and thus can be converted to LSFs successfully.

In a short summary, by using RCs as intermediate variables, the proposed method decomposes a high-order LPC model into several low-order LPC models and then vector quantizes them separately. The main difference between the proposed hierarchical decomposition vector quantization method and the split vector quantization method, which also splits the LSFs into subgroups, is as below.

1) In HDVQ, the LSFs of each subgroup are contained within the same interval $(0, \pi)$. Exploiting this feature, either a shared or individual codebook can be designed for each subgroup. In contrast, the LSFs obtained by SVQ in each subgroup distribute in different but smaller intervals and thus an individual codebook must be designed for each subgroup. For SVQ, the LSFs in each subgroup have smaller dynamic range and are easier to be vector quantized. Therefore, for LPC analysis at a fixed order, SVQ performs more efficiently than HDVQ.
2) The advantage of HDVQ lies in that it can decompose a high-order LPC model into several low-order LPC models for encoding and thus it is fit for a scalable coding framework. In contrast, SVQ has to encode LPC models at different orders separately, and thus is inefficient in terms of codebook size and coding bits.
3) A major computation consuming operation in LPC vector quantization is the conversion from LPC parameters to LSFs, which requires calculating the roots of the polynomials in Eq. (3) with numerical analysis methods [27]. The computation complexity of this procedure typically increases exponentially with the LPC order. As shown in Fig. 3, SVQ involves a direct conversion from LPC parameters to LSFs and hence is computationally demanding especially for high-order LPC. In contrast, as shown in Fig. 5, HDVQ can decompose a high-order LPC-LSF conversion into the conversion between several low-order models, and hence can reduce the computation complexity significantly.

It should be noted that in Fig. 5 and Fig. 6 we aim to present a framework for scalable LP coding. This framework can be used for arbitrary model orders. For encoding the low-order model in each subgroup, existing vector quantization schemes (such as split vector quantization, differential vector quantization, multistage vector quantization, or a combination of them [8]–[15]), which are already mature and efficient, can be utilized for this task. For simplicity, we only use split vector quantization in the following experiments and the conclusion can be generalized to other vector quantizers.

IV. Experimental Results

In this section, we give some experimental results to demonstrate the advantages of the proposed method. The test sequence used in the experiment consists of 45 minutes of random broadcasting speeches sampling at 16 kHz, where 40 minutes of the data is used for codebook training and the remaining 5 minutes of data is used for testing. After applying a high-pass filter at 100 Hz and a pre-emphasis filter of $H(z) = 1 - 0.68z^{-1}$, the Levinson-Durbin algorithm is performed every 15 ms after applying a 30-ms Hanning window. LPC parameters at 5 different

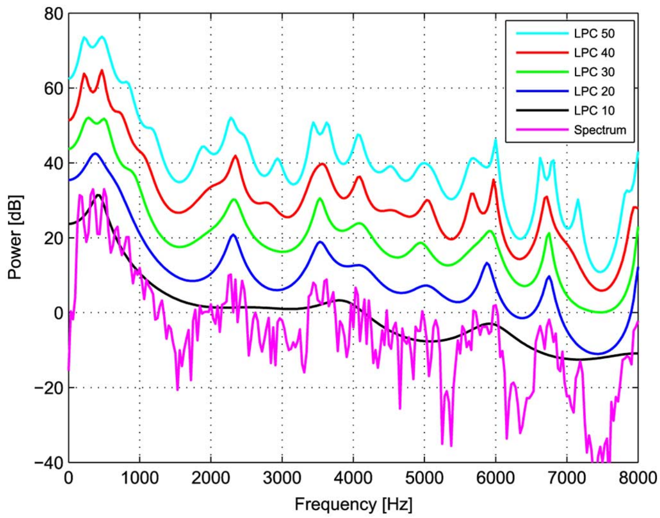


Fig. 7. Spectrum of 30 ms speech for different orders of linear prediction analysis.

orders (10, 20, 30, 40 and 50) are calculated and vector quantized with SVQ and HDVQ, respectively. The performance of the two methods is compared in a scalable coding framework where the LPC information at the 5 different orders is encoded into 5 layers respectively.

The experiment is divided into two parts. The first part investigates the probability distribution of the obtained LSFs in each subgroup when applying high-order LPC analysis. The second part compares SVQ and HDVQ in a scalable coding framework in terms of vector quantization performance and computation complexity.

A. High-Order LPC Analysis

Fig. 7 shows the spectrum of a frame (30 ms) of wideband speech using different orders (10-50) of linear prediction analysis. A shifted version of the spectrum is depicted so that the details of the spectrum curves can be clearly observed and compared. The 10th-order LPC can only capture the variation of the speech spectrum roughly, with a lot of important formant information missing as compared to the original speech spectrum. For instance, the formants at frequencies around 2.2 kHz, 3.5 kHz and 6.7 kHz are completely invisible at LPC-10. One main reason for the information missing is the fact that number of formants is greater than the LPC order. When the LPC order is increased to 20, some formants, which are missed in LPC-10, are well captured in LPC-20. Further increasing the LPC order will introduce more spectral details into the LPC spectrum. The spectrum of LPC-50 is quite close to the original speech spectrum. With Fig. 7, the spectral difference between different LPC orders can be clearly observed, with more spectral (formant) details captured by using higher LPC orders. Referring to the observation in Fig. 2, it can be confirmed that by increasing the LPC order and introducing more spectral details into the LPC spectrum, the speech codec quality can be continuously improved.

The 50th-order LPC model is decomposed into five 10th-order LPC models using the proposed method, with their spectra shown in Fig. 8. The first panel of Fig. 8 depicts the LPC spectrum that is represented using only k1-10; the second panel depicts the LPC spectrum that is represented using only k11-k20; and so on. As indicated in the proposed method, each low-order LPC model contains the information of a subgroup of reflection coefficients. It can be seen from Fig. 8 that the first low-order LPC model (k1-k10), which is equivalent to the 10th-order LPC model in Fig. 7, can only represent the speech spectrum roughly. Regarding the other 4 models (k11-k20, k21-k30, k31-k40, k41-k50), it seems that their spectra are unrelated to the speech spectrum. However, when we compare them with the spectrum of LPC-50 in Fig. 7, it is found that the positions of the spectral peaks and valleys in these models are consistent with those in the LPC-50 spectrum. This observation implies a possibility to incorporate a perceptual model in the vector quantization procedure of each subgroup. Specifically, when designing a codebook for vector quantization, a weighted Euclidean distance measure can be employed to improve the perceptual performance, because the spectral peaks play a higher role than spectral valleys in auditory perception [8]. This is achieved by assigning more weights to the LSFs corresponding to the high-amplitude formants than those corresponding to the low-amplitude formants when calculating the Euclidean distance between quantized and unquantized LSFs. As a possible extension in our future work, we can apply this spectral weighting strategy to the spectral peaks and valleys of each decomposed low-order LPC model in order to improve the perceptual performance.

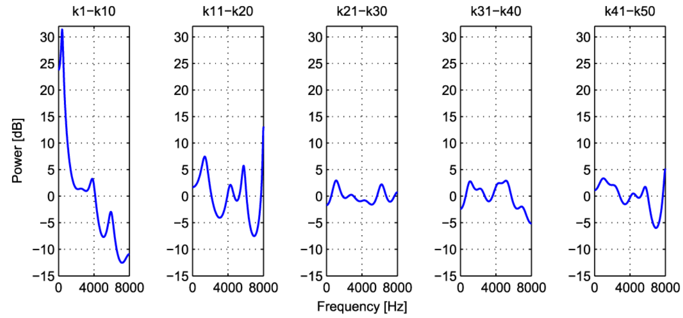


Fig. 8. Spectra of the five 10th-order LPC models obtained by decomposing a 50th-order LPC model. Each column depicts the LPC spectrum that is represented using only 10 LSFs (k1-10, k11-k20, k21-k30, k31-k40, and k41-k50).

It is additionally observed in Fig. 8 that the dynamic range of the spectrum of the first (k1-k10) model, which varies within the range between -10 dB and 30 dB, is much wider than the dynamic range of the spectra of the 2nd-5th models, which all vary within the range between -5 dB to 5 dB. We thus presume that the 2nd-5th models may be easier to vector quantize than the first one. To verify this assumption, we plot the probability distribution function (PDF) of the LSF parameters of each low-order model in Fig. 9. As shown in each panel in Fig. 9, the 10 LSF parameters of each model distribute in the interval $(0, \pi)$. The PDFs of the LSFs in the first model show irregular shapes, each spanning a wide dynamic range. In contrast, the PDFs of the LSFs in the 2nd-5th models show regular Gaussian-like shapes, each spanning a much narrower dynamic range. The phenomena above can confirm our assumption: the LSF parameters in the 2nd-5th models span smaller dynamic range than the ones in the first model and hence can be vector

TABLE I
REQUIRED OPERATIONS BY SPLIT VECTOR QUANTIZATION AND HIERARCHICAL DECOMPOSITION VECTOR QUANTIZATION IN A SCALABLE CODING FRAMEWORK

LPC order	HDVQ		SVQ	
	Coding	Decoding	Coding	Decoding
10	A10→L10	L10→A10	A10→L10	L10→A10
20	(A20→K20)+(K10→A10→L10)×2	(L10→A10→K10)×2+(K20→A20)	A20→L20	L20→A20
30	(A30→K30)+(K10→A10→L10)×3	(L10→A10→K10)×3+(K30→A30)	A30→L30	L30→A30
40	(A40→K40)+(K10→A10→L10)×4	(L10→A10→K10)×4+(K40→A40)	A40→L40	L40→A40
50	(A50→K50)+(K10→A10→L10)×5	(L10→A10→K10)×5+(K50→A50)	A50→L50	L50→A50

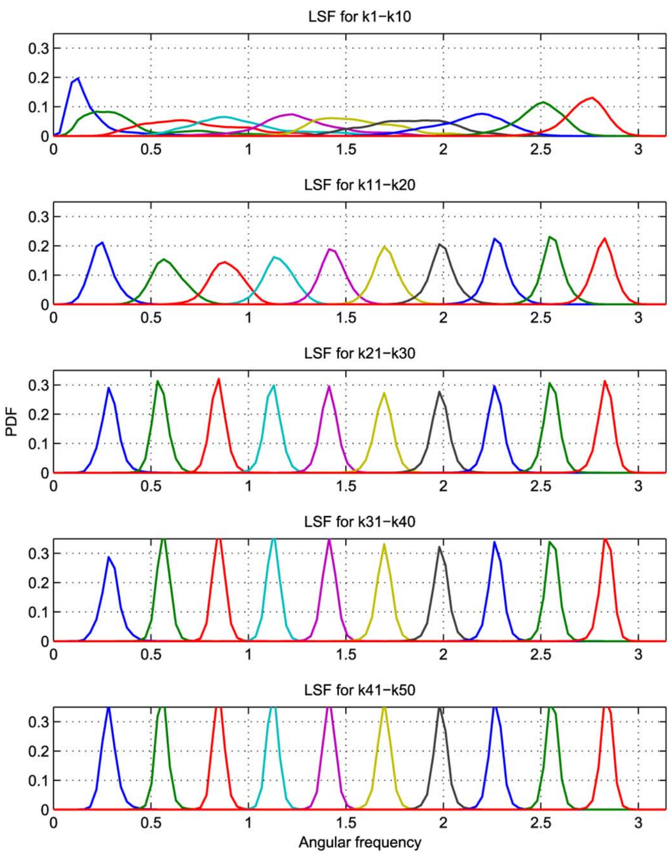


Fig. 9. Probability density functions of the LSF parameters at the five decomposed low-order LPC models.

quantized more efficiently. It is additionally observed that the PDFs of the LSFs in the 2nd-5th models are very similar. For instance, the first LSFs of the 2nd-5th models all distribute in the same interval (0, 0.5); similar phenomena can be observed for all the other LSFs. Based on this observation, we propose to design an individual codebook for the first model while a shared codebook for the 2nd-5th models.

B. Comparison Between SVQ and HDVQ

Vector Quantization Performance: The vector quantization performance of HDVQ and SVQ are compared in a scalable coding framework. For SVQ, the LPC information at the orders of 10, 20, 30, 40 and 50 are encoded into five different layers separately. For HDVQ, the decomposed five 10th-order LPC models are encoded into five different layers separately. The required operations by the two methods are shown in Table I, where for example the term "$(\text{A}20 \rightarrow \text{K}20) + (\text{K}10 \rightarrow \text{A}10 \rightarrow \text{L}10) \times 2$ " denotes one LPC-LSF (of order 20) conversion plus twice RC-LPC-LSF (of order 10) conversions.

The vector quantization performance is compared in terms of codebook size, coding bits and quantization error. The quantization error is measured by log spectral distortion (LSD), which is often used in evaluation of LPC quantization [8]. For the i-th frame, the spectral distortion is defined in dB as

$$D_i = \sqrt{\frac{1}{f_s}\int_0^{f_s} \left[10\log_{10} P_i(f) - 10\log_{10} \hat{P}_i(f)\right]^2 df}, \quad (6)$$

where f_s is the sampling frequency in Hz, $P_i(f) = 1/|A_i(f)|^2$ and $\hat{P}_i(f) = 1/|\hat{A}_i(f)|^2$ are the coded and decoded LPC power spectra in the i-th frame, respectively, with $A_i(f)$ and $\hat{A}_i(f)$ calculated from the coded and decoded LPC parameters. An averaged LSD is calculated over all the testing frames. For HDVQ, two LSD measures can be calculated: a local LSD measure for each low-order LPC model and a global LSD measure for the reconstructed high-order LPC model, which combines the low-order models. For SVQ, only the global LSD measure for the high-order LPC model is calculated.

For these two types of vector quantization methods, the codebooks are designed under a criterion that the two methods can achieve comparable global LSD at all the LPC orders. The parameters of the designed codebooks for SVQ and HDVQ are given by Table II and Table III, respectively. As shown in Eq. (6), the global SD for each LPC order is calculated between the (same-order) quantized and unquantized LPC power spectra. As the LPC order grows, the spectrum of the LPC model becomes complicated. This increases the challenge of quantization and results in more spectral distortion even if more coding bits are used. In SVQ, each subgroup of the 10 LSF parameters is processed with another round of SVQ, i.e., the 10 LSFs are further split into subgroups of (3, 3, 4) and vector quantized separately. The codebook size, coding bits, and global LSD by SVQ are also shown in Table II. In HDVQ, each subgroup of the LSF parameters is processed in a similar way, i.e., the 10 LSFs are split into subgroups of (3, 3, 4) and vector quantized separately. The codebook size, coding bits, local and global LSDs by HDVQ are also shown in Table III. It should be noted that the parameters of the codebook for each subgroup, e.g., codebook size, are determined heuristically and may not be optimal. Other advanced techniques for codebook design can also be employed. However, in this paper, we focus on the comparison between the two methods and the design of an optimal codebook will be left for future work.

One big difference between SVQ and HDVQ in a scalable coding framework is that SVQ has to encode the full LPC information at different orders into the corresponding layer while HDVQ only needs to encode the decomposed low-order LPC information into each layer. We compare the two methods in two

TABLE II
CODEBOOK FOR SPLIT VECTOR QUANTIZATION

LPC order	VQ scheme	Codebook size	Coding bits	Global LSD [dB]
10	(3+3+4)×1	4096×3+4096×3+4096×4	36	0.70
20	(3+3+4)×2	(1024×3+1024×3+2048×4)×2	62	1.07
30	(3+3+4)×3	(1024×3+1024×3+1024×4)×3	90	1.12
40	(3+3+4)×4	(1024×3+1024×3+1024×4)×4	120	1.20
50	(3+3+4)×5	(512×3+512×3+1024×4)×5	140	1.30

TABLE III
CODEBOOK FOR HIERARCHICAL DECOMPOSITION VECTOR QUANTIZATION

RC subgroup	VQ scheme	Codebook size	Coding bits	Local LSD [dB]	Global LSD [dB]
k_1-k_{10}	(3+3+4)×1	4096×3+4096×3+4096×4	36	0.70	0.70
k_{11}-k_{20}	(3+3+4)×1	2048×3+2048×3+4096×4	34	0.39	1.03
k_{21}-k_{30}	(3+3+4)×1	0	34	0.34	1.17
k_{31}-k_{40}	(3+3+4)×1	0	34	0.31	1.28
k_{41}-k_{50}	(3+3+4)×1	0	34	0.30	1.38

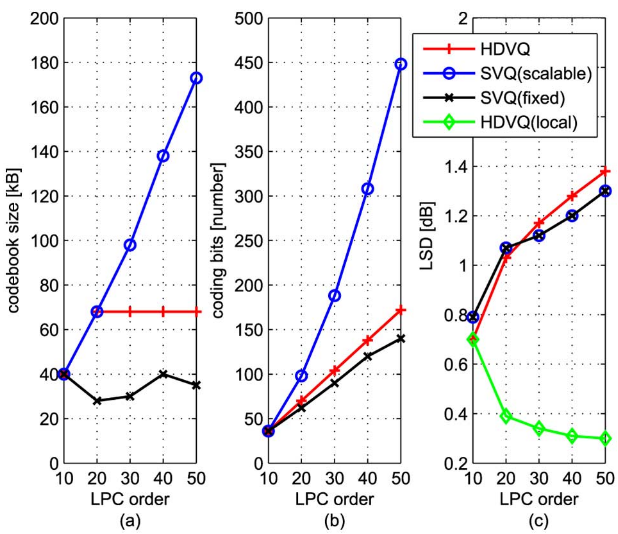


Fig. 10. Performance comparison between split vector quantization and hierarchical decomposition vector quantization in a scalable coding framework.

scenarios: LPC quantization with fixed LPC order (i.e., only one LPC order is considered) and LPC quantization with scalable LPC orders (i.e., all the variable LPC orders are considered). As a summary of Table II and Table III, Fig. 10 compares the two methods in both scenarios in terms of codebook size, coding bits and LSD. In each panel of Fig. 10 three curves are depicted: one for HDVQ (which performs equivalently for both fixed and variable LPC orders), one for SVQ with a fixed LPC order, and one for SVQ with variable LPC orders. In Fig. 10(c), the local LSD obtained by HDVQ for each low-order LPC model is additionally given as "HDVQ (local)" for reference.

As mentioned above, the sizes of the codebooks are especially chosen for the two methods so that they can show comparable global LSD performance at all the variable LPC orders. As shown in Fig. 10(c), with the designed codebook, HDVQ can achieve good local LSD performance inside each low-order model. For instance, very small local LSD (less than 0.4 dB) can be obtained for the last four models (order 20–50) even if a shared codebook is employed for them. However, since the quantization error at each low-order LPC model accumulates when reconstructing a high-order LPC model, degraded global LSD performance will be observed with the increasing LPC order. In contrast, SVQ processes the LPC information at different orders separately and hence performs more efficiently than HDVQ for a fixed LPC order. As shown in Fig. 10(a)–(c), once the LPC order is fixed, SVQ (fixed) requires smaller codebooks and fewer coding bits than HDVQ for a comparable LSD performance.

The advantage of HDVQ is mainly manifested for scalable LPC orders. Since the codebooks and coding bits of SVQ accumulate with the number of variable LPC orders, the total consumption (in terms of codebook size and coding bits) of SVQ (scalable) tends to increase exponentially with the increasing LPC order, just as shown in Fig. 10(a)–(b). In contrast, HDVQ only needs to encode the decomposed low-order models progressively, and the consumption of the codebook and coding bits generally increases linearly with the increasing LPC order. As shown in Fig. 10(a), since HDVQ uses an individual codebook for the first low-order model while a shared codebook for the remaining low-order models, no new codebook is required for the 3rd-5th layers.

In a short summary, HDVQ performs less efficiently than SVQ for fixed LPC orders; however, it shows greater advantages in a scalable framework by reducing the total codebook size and coding bits significantly.

Computation Complexity: The computation complexity of SVQ and HDVQ are compared also in two scenarios: LPC quantization with fixed LPC orders and LPC quantization with scalable LPC orders. The computation complexity is measured by computation time. When calculating this measure, only the operations listed in Table I are considered. The program was coded in Matlab and run on an Intel 64 Q8400@2.66 GHz. The encoding time and decoding time for 10000 frames are counted for the SVQ and HDVQ in both scenarios and are shown in the two panels of Fig. 11, respectively. In each panel three curves are depicted: one for HDVQ (which performs equivalently for both fixed and variable LPC orders), one for SVQ with a fixed LPC order, and one for SVQ with variable LPC orders.

As shown in Fig. 11(a), for fixed LPC orders, the encoding time of SVQ (fixed) increases exponentially with the increasing LPC order. This is mainly because the computation complexity of the conversion from LPC parameters to LSFs exponentially

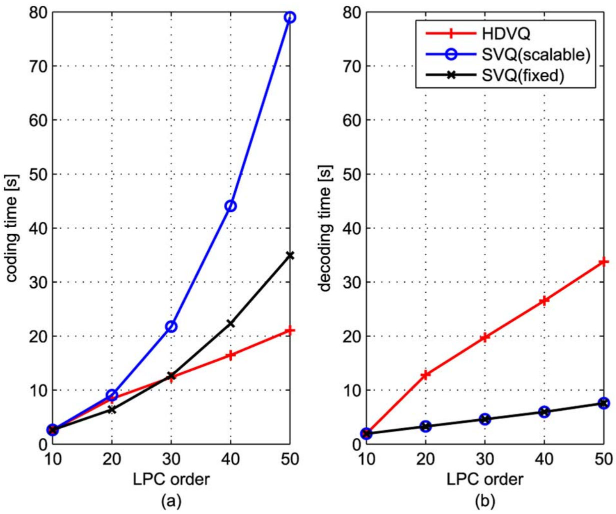


Fig. 11. Computation time (for 10000 frames) comparison between split vector quantization and hierarchical decomposition vector quantization in a scalable coding framework.

grows with the LPC order. HDVQ decomposes a high-order LPC-LSF conversion into multiple low-order conversions; as a result, its encoding time increases linearly with the increasing LPC order. The advantage of HDVQ decomposition can be evidently observed especially for high-order LPC. As shown in Fig. 11(b), the decoding time of SVQ (fixed) and HDVQ both increases linearly with the increasing LPC order. Additionally, SVQ (fixed), which involves only one LSF-LPC conversion, requires less decoding time than HDVQ, which involves a relatively more complicated LPC reconstruction procedure.

For scalable LPC orders, SVQ (scalable) needs to encode for all the bitstream layers, thus its encoding time increases exponentially with the LPC order. As shown in Fig. 11(a), the encoding time by SVQ (scalable) reaches an extremely high value at the 50th LPC order. For the decoding time shown in Fig. 11(b), SVQ (scalable) needs only one LSF-LPC conversion (the same operation as in SVQ (fixed)), and hence requires the same time as SVQ (fixed). HDVQ performs equivalently for both scalable and fixed LPC orders. It requires much less encoding time than SVQ (scalable) while a bit more decoding time.

In a short summary, compared with SVQ, HDVQ can reduce the computation time significantly in a scalable coding framework. Furthermore, by decomposing the high-order LPC-LSF conversion into low-order ones, HDVQ can further reduce computation complexity.

V. Conclusion

With the help of an intermediate parameter, reflection coefficient, it is possible to decompose a high-order LPC model into several low-order LPC models and quantize them separately in the LSF domain. This scheme is naturally fit for a scalable coding scheme with variable orders of LPC analysis since the decomposed low-order LPC models can be combined in a progressive way to recover the high-order LPC information. A shared codebook can be designed for the vector quantization of these low-order LPC models, which can significantly reduce the total size of a codebook. Comparison with the traditional split vector quantization method demonstrates that

1) For a fixed LPC order, SVQ works more efficiently than the proposed HDVQ method.
2) In a scalable coding framework, HDVQ evidently outperforms SVQ by reducing the size of the codebooks and the number of coding bits significantly and reducing the computation time efficiently.
3) By decomposing the high-order LPC-LSF conversion, which is highly computation demanding, into low-order LPC-LSF conversions, HDVQ can further reduce the computation complexity.

The proposed method is feasible for arbitrary LPC analysis orders and has shown great potential in a scalable coding framework.

In the experiment, a 50th-order LPC analysis is employed to validate the proposed method. It should be noted that an LPC-50 model might not be optimal for a practical speech codec, where the bandwidth efficiency has to be considered. Applying the proposed method to a high-quality scalable coding system, determining the optimal LPC analysis order, and evaluation with comprehensive subjective listing tests will be our future work.

Appendix
A Modified G.728 With Variable LPC Order

G.728 is an ITU-T standard for low-delay CELP speech coding operating at 16 kbit/s [3]. It involves a 50th-order LPC analysis and thus is inherently suitable to be modified as a codec with variable LPC orders from 10 to 50. Although optimally designed to work at an 8 KHz sampling rate, it can also process the speech stream sampling at 16 kHz, by directly treating the latter one as a stream sampling at 8 kHz.

We use the G.728 ANSI C code obtained from ITU [29]. The *Order Invariant Property* of RC in Section II-C will be employed for code modification. Specifically, to encode with an M-order ($M < 50$) LPC analysis, we only need to modify the Levin-Durbin algorithm in G.728, setting all the reflection coefficients k_m that satisfy $m > M$ to 0 during the iteration. In this way, we get 50th-order LPC parameters which contain only the M-order information, without changing the processing of other blocks in the G.728 project. Finally, the executable files of the modified G.728 are given online, along with the PESQ executable files and audio demos [28].

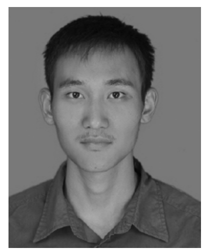
Acknowledgment

The authors would like to thank the anonymous reviewers for their detailed comments, which have been extremely helpful in improving the clarity and quality of this paper.

References

[1] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. New Jersey, NJ, USA: Prentice-Hall, 1978.
[2] B. S. Atal, “The history of linear prediction,” *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 154–161, Mar. 2006.
[3] *Coding of speech at 16 kbit/s using low-delay code excited linear prediction (LD-CELP)*, ITU Rec. G.728, 1994.
[4] *Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)*, ITU Rec. G. 729, 1996.

[5] B. Bessette, R. Salami, and R. Lefebvre *et al.*, "The adaptive multi-rate wideband speech codec (AMR-WB)," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 620–636, Nov. 2002.
[6] *Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems*, TIA/EIA/IS-127, 1996.
[7] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 5, pp. 562–574, Oct. 1980.
[8] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 1, pp. 3–14, Jan. 1993.
[9] F. K. Soong and B. H. Juang, "Optimal quantization of LSP parameters," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 1, pp. 15–24, Jan. 1993.
[10] W. P. LeBlanc, B. Bhattacharya, S. A. Mahmond, and V. Cuperman, "Efficient search and design procedure for robust multi-stage VQ of LPC parameters for 4 kb/s speech coding," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 4, pp. 373–385, Oct. 1993.
[11] J. Pan and R. F. Thomas, "Vector quantization of speech line spectrum pair parameters and reflection coefficients," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 106–115, Mar. 1998.
[12] L. M. da Silva and A. Alcaim, "Differential coding of speech LSF parameters using hybrid vector quantization and bidirectional prediction," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 208–211, Mar. 2000.
[13] W. J. Han, E. K. Kim, and Y. H. Oh, "Multicodebook split vector quantization of LSF parameters," *IEEE Signal Process. Lett.*, vol. 9, no. 12, pp. 418–421, Dec. 2003.
[14] D. A. Subramaniam and D. R. Bhaskar, "PDF optimized parametric vector quantization of speech line spectral frequencies," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 2, pp. 130–142, Mar. 2003.
[15] S. So and K. K. Paliwal, "Efficient product code vector quantisation using the switched split vector quantiser," *Digital Signal Process.*, vol. 17, no. 1, pp. 138–171, 2007.
[16] S. So and K. K. Paliwal, "A comparative study of LPC parameter representations and quantisation schemes for wideband speech coding," *Digital Signal Process.*, vol. 17, no. 1, pp. 114–137, 2007.
[17] D. M. Howard and J. Angus, *Acoustics and Psychoacoustics*. New York, NY, USA: Taylor & Francis, 2009.
[18] B. Geiser, H. Kruger, H. W. Lollmann, and P. Vary *et al.*, "Candidate proposal for ITU-T super-wideband speech and audio coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, 2009, pp. 4121–4124.
[19] S. Ragot, B. Kovesi, and R. Trilling *et al.*, "ITU-T G. 729.1: An 8-32 kbit/s scalable coder interoperable with G. 729 for wideband telephony and Voice over IP," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Honolulu, HI, USA, 2007, vol. 4, pp. 529–532.
[20] *Frame error robust narrowband and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s*, ITU-T Rec. G. 718, 2008.
[21] W. C. Chu, *Speech coding algorithms: Foundation and evolution of standardized coders*. New York, NY, USA: Wiley, 2004.
[22] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, USA, 2001, vol. 2, pp. 749–752.
[23] M. Graubner, P. S. Mogre, R. Steinmetz, and T. Lorenzen, "A new QoE model and evaluation method for broadcast audio contribution over IP," in *Proc. ACM 20th Int. Workshop Netw. Operating Syst. Support Digital Audio Video*, Amsterdam, The Netherlands, 2010, pp. 57–62.
[24] P. Kabal, An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality TSP Lab Tech. Rep. Dept. Elect. and Comput. Eng., McGill Univ., Montreal, QC, Canada, 2002.
[25] P. P. Vaidyanathan, "The theory of linear prediction," *Synth. Lectures Signal Process.*, vol. 2, no. 1, pp. 1–184, 2007.
[26] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[27] P. Kabal and R. P. Ramachandran, "The computation of linespectral frequencies using chebyshev polynomials," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. ASSP-34, no. 6, pp. 1419–1426, Dec. 1986.
[28] [Online]. Available: https://sites.google.com/site/linwangsig/celp
[29] [Online]. Available: http://www.itu.int



Lin Wang received the B.S. degree in electronic engineering from Tianjin University, Tianjin, China, in 2003 and the Ph.D. degree in signal processing from Dalian University of Technology, Dalian, China, in 2010. From 2011 to 2013, he was an Alexander von Humboldt Fellow at the University of Oldenburg, Germany. Since 2014, he has been a Postdoctoral Researcher in Queen Mary University of London, U.K. His research interests include video and audio compression, blind source separation, and 3D audio processing.

Zhe Chen (M'03–SM'11) received the B.S. degree, the M.S. degree in signal and information processing, and the Ph.D. degree in signal and information processing from Dalian University of Technology (DUT), Dalian, China, in 1996, 1999, and 2003, respectively.

He joined the Department of Electronic Engineering, DUT, as a Lecturer in 2002, and became an Associate Professor in 2006. His research interests include digital signal processing, speech processing, image processing, and broadband wireless communication.

Fuliang Yin received the B.S. degree in electronic engineering and the M.S. degree in communications and electronic systems from Dalian University of Technology (DUT), Dalian, China, in 1984 and 1987, respectively. He joined the Department of Electronic Engineering, DUT, as a Lecturer in 1987 and became an Associate Professor in 1991. He has been a Professor at DUT since 1994, and the Dean of the School of Electronic and Information Engineering of DUT from 2000 to 2009. His research interests include digital signal processing, speech processing, image processing, and broadband wireless communication.