An Efficient Codebook for the SCELP Low Delay Audio Codec

Hauke Krüger and Peter Vary
Institute of Communication Systems and Data Processing
RWTH Aachen University, D-52056 Aachen, Germany
email: {krueger,vary}@ind.rwth-aachen.de

Abstract— **The SCELP (Spherical Code Excited Linear Prediction) audio codec has recently been proposed as a new candidate for low delay audio coding based on Linear Prediction (LP). The new codec applies closed-loop vector quantization employing a spherical code in a gain shape manner. The spherical code is based on the apple-peeling code construction rule and in general does not require a codebook table for the encoding and decoding process.**
In this contribution, however, we propose to employ auxiliary information gathered in advance to reduce the computational encoding and decoding complexity at runtime significantly. This auxiliary information can be considered as the SCELP codebook. Due to the consideration of the characteristics of the apple-peeling code construction principle, this codebook can be stored very efficiently in read-only-memory. With the proposed principle, low computational as well as low memory complexity can be achieved simultaneously in the SCELP codec.

I. Introduction

Lossy compression of audio signals can be roughly subdivided into two principles: *Perceptual audio coding* is based on transform coding. The signal to be compressed is firstly transformed by an analysis filter bank, and the sub band representation is quantized in the transform domain. A perceptual model controls the adaptive bit allocation for the quantization. The goal is to keep the noise introduced by quantization below the masking threshold described by the perceptual model. In general, the algorithmic delay is rather high due to large transform lengths, e.g. [3]. *Parametric audio coding* is based on a source model. One parametric approach is based on linear prediction (LP), the basis for todays highly efficient speech coding algorithms for mobile communications, for example [4]. An all-pole filter models the spectral envelope of the input signal. Based on the inverse of this filter, the input is filtered to produce the LP residual signal which is quantized. Often vector quantization with a sparse codebook is applied according to the CELP (Code Excited Linear Prediction [2]) analysis-by-synthesis approach to achieve very high compression. Due to the sparse codebook and additional modeling of the speaker's instantaneous pitch period, speech coders perform well for speech but cannot compete with perceptual audio coding for non-speech input. The typical algorithmic delay is around 20 ms.

In a previous contribution [1] we presented a new coding scheme for low delay parametric audio coding in which the principle of linear prediction is preserved while a spherical code is used in a gain-shape manner for the quantization of the residual signal at a moderate bit rate. This spherical code is based on the apple-peeling principle introduced in [5] for the purpose of channel coding and referenced in [6] in the context of a spherical code analysis. The apple-peeling code has been revisited in [7] for DPCM. In contrast to that approach, the SCELP codec employs the spherical code in a CELP scheme. The sphere construction principle according to the apple-peeling method in general enables the encoding and decoding of a signal vector without any codebook. In this contribution we propose to use auxiliary information which can be determined in advance during code construction. This auxiliary information is stored in read-only-memory (ROM) and can be considered as a compact vector codebook. At codec runtime it aids the process of transforming the spherical code vector index, used for signal transmission, into the reconstructed code vectors on encoder and decoder side. The compact codebook is based on a representation of the spherical code as a coding tree combined with a lookup table to store all required trigonometric function values for spherical coordinate transformation. Because both parts of this compact codebook are determined in advance the computational complexity for signal compression can be drastically reduced.

The properties of the compact codebook can be exploited to store it with only a small demand for ROM compared to an approach that stores a lookup table as often applied for trained codebooks [11].

In this paper, the principle of the SCELP audio coding scheme will be shortly reviewed in Section II. The construction of the spherical code according to the apple-peeling method is described in Section III. A representation of this code construction principle as spherical coding tree for code vector decoding is explained in Section IV. In Section V, the principle to efficiently store the coding tree and the lookup table for trigonometric function values for code vector reconstruction is presented. Results considering the reduction of the computational and memory complexity are given in Section VI.

II. The SCELP audio codec

The SCELP codec as proposed in [1] is based on block adaptive linear prediction. Linear predictive coding in general exploits correlation immanent to an input signal $x(k)$ by decorrelating it before quantization. For short term block adaptive linear prediction of order N, a windowed segment of the input signal, $x_w(k)$, of length L_{LPC} is analyzed in order to obtain the filter coefficients $a_1 \cdots a_N$. Based on these filter coefficients the input signal is filtered with $H_A(z) = 1 - \sum_{i=1}^{N} a_i \cdot z^{-i}$, the LP analysis filter, to produce the LP residual signal $d(k)$ which is quantized and transmitted to the decoder as $\tilde{d}(k)$. In the decoder the signal $\tilde{x}(k)$ is reconstructed from $\tilde{d}(k)$ by filtering with the all-pole LP synthesis filter $H_S(z) = H_A^{-1}(z)$. Numerous contributions have been published concerning the principles of linear prediction, for example [8].

In the context of block adaptive linear predictive coding, the linear prediction coefficients must be transmitted in addition to signal $\tilde{d}(k)$. This can be achieved with only small additional bit rate as shown for example in [9]. The length of the signal segment used for LP analysis, L_{LPC}, is responsible for the algorithmic delay of the complete codec.

In the SCELP codec, Vector Quantization (VQ) is applied to the LP residual $d(k)$. Multiple samples of the signal $d(k)$ are combined in a vector $\mathbf{d} = \begin{bmatrix} d_0 & \cdots & d_{L_V-1} \end{bmatrix}$ of length L_V in chronological order with $l = 0 \cdots (L_V - 1)$ as vector coordinate index prior to quantization in L_V-dimensional coding space. In the SCELP audio codec, VQ is applied in closed-loop manner according to the principle of analysis-by-synthesis at the encoder side to find the optimal quantized excitation vector $\tilde{\mathbf{d}}$ for the LP residual.

The principle of the SCELP encoder is depicted in Figure 1. For analysis-by-synthesis the decoder is part of the encoder.

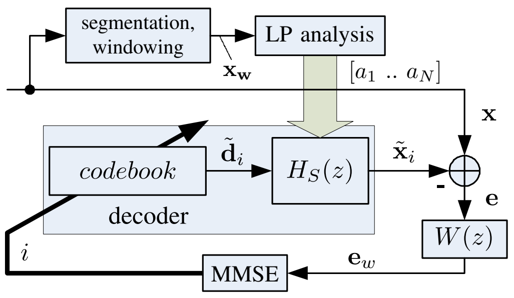

Fig. 1. *Principle SCELP Encoder.*

During the vector search procedure in the encoder for each index i corresponding to one of the available code vectors, an excitation vector $\tilde{\mathbf{d}}_i$ is generated first. That excitation vector is then fed into the LP synthesis filter $H_S(z)$. The resulting signal vector $\tilde{\mathbf{x}}_i$ is compared to the input signal vector $\mathbf{x}$ to find the index i_Q with minimum mean square error (MMSE)

$$i_Q = \arg\min_i \{\mathcal{D}_i = (\mathbf{x} - \tilde{\mathbf{x}}_i) \cdot (\mathbf{x} - \tilde{\mathbf{x}}_i)^T\}. \quad (1)$$

By the application of an error weighting filter $W(z)$, a perceptual masking of the quantization noise inherent to the decoded signal can be achieved.

The spherical code which is the basis for the vector quantization will be described in the next section. Because analysis-by-synthesis in general results in a high computational complexity for VQ, in [1] the analysis-by-synthesis framework has been modified to enable a very efficient vector search strategy that is based on Pre-Selection and Candidate-Exclusion.

III. Employed Spherical Code

Spherical quantization in general has been investigated intensively, for example in [6], [7] and [10]. For the construction of code vectors for the quantization of the LP residual, $\tilde{\mathbf{d}}$, the apple-peeling principle is applied in the SCELP audio codec in a gain-shape manner: Each code vector is composed of a gain (scalar) and a shape (vector) part. The code vectors $\tilde{\mathbf{c}}$ for the quantization of the shape part are located on the surface of a unit sphere. The gain component is the quantized radius $\tilde{R}$. Both components are combined to determine

$$\tilde{\mathbf{d}} = \tilde{R} \cdot \tilde{\mathbf{c}}. \quad (2)$$

For transmission, the index i_{sp} and the index i_R for the reconstruction of the shape part of the vector and the gain factor respectively must be combined to form the codeword i_Q. The principle of the spherical code construction according to the apple-peeling code as applied in the SCELP codec has been described in the literature, for example in [5]. It is based on the representation of the centroids for the shape part, $\tilde{\mathbf{c}} \in \mathcal{C}$, in $(L_V - 1)$ angles $[\tilde{\varphi}_0 \;\cdots \tilde{\varphi}_{L_V-2}]$ in polar coordinates. In the following, the construction principle will be shortly demonstrated by the example of a 3-dimensional sphere, as depicted in Figure 2. The example centroids constructed with respect to the apple-peeling algorithm, $\tilde{\mathbf{c}}_a$, $\tilde{\mathbf{c}}_b$, $\tilde{\mathbf{c}}_c$, are marked as big black bullets on the surface of the unit sphere.

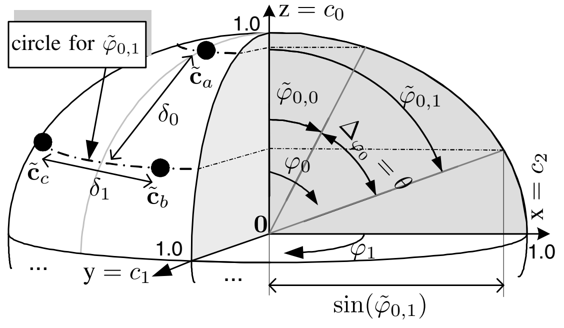

Fig. 2. *Apple-peeling Code for a 3-dimensional Sphere.*

The sphere has been cut in order to display the 2 angles, φ_0 in x-z-plane and φ_1 in x-y-plane. Due to the symmetry properties of the spherical code, only the upper half of the sphere is shown. For code construction, the angles will be considered in the order of φ_0 to φ_1, $0 \leq \varphi_0 < \pi$ and $0 \leq \varphi_1 < 2\pi$ for the complete unit sphere. The apple-peeling construction constraint to have a minimum separation angle θ in between neighbor centroids is expressed on the surface of the sphere: The distance between neighbor centroids in one direction is denoted as δ_0 and in the other direction as δ_1. As the centroids are placed on a unit sphere and further assuming small θ, the distances can be approximated by the circular arc according to the angle θ to specify the apple-peeling constraint:

$$\delta_0 \geq \theta,\; \delta_1 \geq \theta \text{ and } \delta_0 \approx \delta_1 \approx \theta \quad (3)$$

The minimum separation angle θ is chosen as $\theta = \pi / N_{sp}$ with the code construction parameter $N_{sp} \in \mathbb{N}$. By choosing the number of angle quantization levels N_{sp}, the range of angle φ_0 is divided into N_{sp} angle quantization intervals with equal size of $\Delta_{\varphi_0} = \theta$. Circles of latitude (dash-dotted line) on the surface of the unit sphere at

$$\varphi_0 = \tilde{\varphi}_{0,i_0} = (i_0 + 1/2) \cdot \Delta_{\varphi_0} \tag{4}$$

are linked to index $i_0 = 0 \cdots (N_{sp} - 1)$. The centroids of the apple-peeling code are constrained to be located on these circles which are spaced according to the distance δ_0, hence $\varphi_0 \in \tilde{\varphi}_{0,i_0}$ for all $\tilde{\mathbf{c}} \in \mathcal{C}$.

The radius of each circle of latitude depends on $\tilde{\varphi}_{0,i_0}$. The range of φ_1, $0 \leq \varphi_1 < 2\pi$, is divided into $N_{sp,1}$ angle intervals of equal length Δ_{φ_1}. In order to hold the minimum distance constraint, the separation angle Δ_{φ_1} is different from circle to circle and depends on the circle radius, $\sin(\tilde{\varphi}_{0,i_0})$, and thus $\tilde{\varphi}_{0,i_0}$:

$$\Delta_{\varphi_1}(\tilde{\varphi}_{0,i_0}) = \frac{2\pi}{N_{sp,1}(\tilde{\varphi}_{0,i_0})} \gtrsim \frac{\theta(N_{sp})}{\sin(\tilde{\varphi}_{0,i_0})} \tag{5}$$

With this, the number of intervals for each circle is

$$N_{sp,1}(\tilde{\varphi}_{0,i_0}) = \lfloor \frac{2\pi}{\theta(N_{sp})} \cdot \sin(\tilde{\varphi}_{0,i_0}) \rfloor . \tag{6}$$

In order to place the centroids onto the sphere surface, the according angles $\tilde{\varphi}_{1,i_1}(\tilde{\varphi}_{0,i_0})$ associated with the circle for $\tilde{\varphi}_{0,i_0}$ are placed in analogy to (4) at positions

$$\tilde{\varphi}_{1,i_1}(\tilde{\varphi}_{0,i_0}) = (i_1 + 1/2) \cdot \frac{2\pi}{N_{sp,1}(\tilde{\varphi}_{0,i_0})} \tag{7}$$

Each tuple $[i_0, i_1]$ identifies the two angles and thus the position of one centroid of the resulting code $\mathcal{C}$ for construction parameter N_{sp}.

A tuple of $[i_0, i_1]$ is mapped to an overall sphere index $i_{sp} = 0 \cdots (M_{sp}(N_{sp}) - 1)$ with the number of centroids $M_{sp}(N_{sp})$ as a function of the start parameter N_{sp}. This index mapping will be described in detail in the next section.

While the example here is limited to a unit sphere in three dimensions, in practical cases a higher dimension $L_V > 3$ is used. Considering a representation of the code vectors in spherical polar coordinates, $L_V - 1$ angles are considered, the last angle in the range of $0 \leq \tilde{\varphi}_{L_V-2} < 2\pi$ and all other angles in the range of $0 \leq \tilde{\varphi}_l < \pi$, $l = 0 \cdots (L_V - 3)$.

With respect to the complete codeword i_Q for a signal vector of length L_V, a budget of $r = r_0 \cdot L_V$ bits is available with r_0 as the effective number of bits available for each sample. Considering M_R available indices i_R for the reconstruction of the radius and M_{sp} indices i_{sp} for the reconstruction of the vector on the surface of the sphere, the indices can be combined in a codeword i_Q according to

$$i_Q = i_R \cdot M_{sp} + i_{sp} \tag{8}$$

for the sake of coding efficiency. In order to combine all possible indices in one codeword, the condition

$$2^r \geq M_{sp} \cdot M_R \tag{9}$$

must be fulfilled.

A possible distribution of M_R and M_{sp} is proposed in [7]. The underlying principle is to find a bit allocation such that the distance $\theta(N_{sp})$ between code vectors on the surface of the unit sphere is as large as the relative step size of the logarithmic quantization of the radius. In order to find the combination of M_R and M_{sp} that provides the best quantization performance at the target bit rate r, the spherical code is designed iteratively with increasing N_{sp} to provide the highest number of index combinations that still fulfill constraint (9).

IV. Spherical Coding Tree for Decoding

For an efficient spherical decoding procedure we propose to employ a spherical coding tree in this contribution. In the context of the decoding process for the spherical vector quantization the incoming vector index i_Q is decomposed into index i_R and index i_{sp} with respect to equation (8). The reconstruction of the radius $\tilde{R}$ requires to read out an amplitude from a coding table due to scalar logarithmic quantization.

For the decoding of the shape part of the excitation vector, $\tilde{\mathbf{c}} = \begin{bmatrix} \tilde{c}_0 & .. & \tilde{c}_{(L_V-1)} \end{bmatrix}$, the sphere index i_{sp} must be transformed into a code vector in cartesian coordinates. For this transformation the spherical coding tree is employed. The example for the 3-dimensional sphere in Figure 3 demonstrates the correspondence of the spherical code vectors on the unit sphere surface with the proposed spherical coding tree.

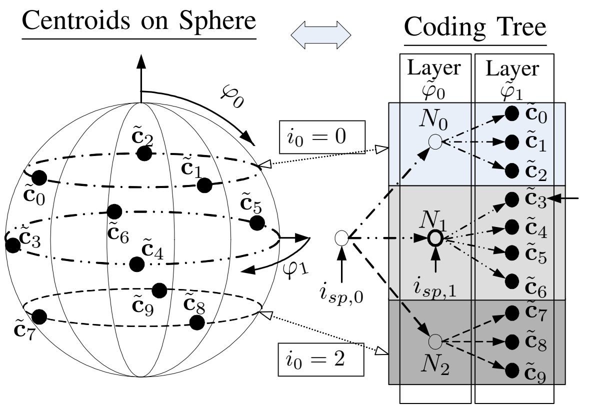

Fig. 3. *Correspondence: Code Vector and Coding Tree.*

The coding tree on the right side of the Figure contains branches, marked as non-filled bullets, and leafs, marked as black colored bullets. One layer of the tree corresponds to the angle $\tilde{\varphi}_0$, the other one to angle $\tilde{\varphi}_1$. The depicted coding tree contains three subtrees, marked as horizontal boxes in different gray colors. Considering the code construction, each subtree represents one of the circles of latitude on the sphere surface, marked with the dash-dotted, the dash-dot-dotted, and the dashed line. On the layer for angle $\tilde{\varphi}_0$, each subtree corresponds to the choice of index i_0 for the quantization reconstruction level of angle $\tilde{\varphi}_{0,i_0}$. On the tree layer for angle $\tilde{\varphi}_1$ each coding tree leaf corresponds to the choice of index i_1 for the quantization reconstruction level of $\tilde{\varphi}_{1,i_1}(\tilde{\varphi}_{0,i_0})$. With each tuple of $[i_0, i_1]$ the angle quantization levels for $\tilde{\varphi}_0$ and $\tilde{\varphi}_1$ required to find the code vector $\tilde{\mathbf{c}}$ are determined. Therefore each leaf corresponds to one of the centroids on the surface of the unit sphere, $\tilde{\mathbf{c}}_{i_{sp}} = [\tilde{c}_{i_{sp},0} \quad \tilde{c}_{i_{sp},1} \quad \tilde{c}_{i_{sp},2}]$ with the index $i_{sp} = 0..9$ in Figure 3.

For decoding, the index i_{sp} must be transformed into the coordinates of the spherical centroid vector. This transformation employs the spherical coding tree: The tree is entered at the coding tree root position as shown in the Figure with incoming index $i_{sp,0} = i_{sp}$. At the tree layer for angle $\tilde{\varphi}_0$ a decision must be made to identify the subtree to which the desired centroid belongs to find the angle index i_0. Each subtree corresponds to an index interval, in the example either the index interval $i_{sp} \mid_{i_0=0} = 0, 1, 2$, $i_{sp} \mid_{i_0=1} = 3, 4, 5, 6$, or $i_{sp} \mid_{i_0=2} = 7, 8, 9$. The determination of the right subtree for incoming index i_{sp} on the tree layer corresponding to angle $\tilde{\varphi}_0$ requires that the number of centroids in each subtree, N_0, N_1, N_2 in Figure 3, is known. With the code construction parameter N_{sp}, these numbers can be determined by the construction of all subtrees. The index i_0 is found as

$$i_0 = \begin{cases} 0 & \text{for } 0 \leq i_{sp,0} < N_0 \\ 1 & \text{for } N_0 \leq i_{sp,0} < (N_0 + N_1) \\ 2 & \text{for } (N_0 + N_1) \leq i_{sp,0} < (N_0 + N_1 + N_2) \end{cases} \quad (10)$$

With index i_0 the first code vector reconstruction angle $\tilde{\varphi}_{0,i_0}$ and hence also the first cartesian coordinate, $\tilde{c}_{i_{sp},0} = \cos(\tilde{\varphi}_{0,i_0})$, can be determined. In the example in Figure 3, for $i_{sp} = 3$, the middle subtree, $i_0 = 1$, has been found to correspond to the right index interval.

For the tree layer corresponding to $\tilde{\varphi}_1$ the index $i_{sp,0}$ must be modified with respect to the found index interval according to the following equation:

$$i_{sp,1} = i_{sp,0} - \sum_{i=0}^{(i_0-1)} N_i. \quad (11)$$

As the angle $\tilde{\varphi}_1$ is the final angle, the modified index corresponds to the index $i_1 = i_{sp,1}$. With the knowledge of all code vector reconstruction angles in polar coordinates, the code vector $\tilde{\mathbf{c}}_{i_{sp}}$ is determined as

$$\begin{aligned} \tilde{c}_{i_{sp},0} &= \cos(\tilde{\varphi}_{0,i_0}) \\ \tilde{c}_{i_{sp},1} &= \sin(\tilde{\varphi}_{0,i_0}) \cdot \cos(\tilde{\varphi}_{1,i_1}) \\ \tilde{c}_{i_{sp},2} &= \sin(\tilde{\varphi}_{0,i_0}) \cdot \sin(\tilde{\varphi}_{1,i_1}) \end{aligned} \quad (12)$$

For a higher dimension $L_V > 3$, the index modification in (11) must be determined successively from one tree layer to the next.

The subtree construction and the index interval determination must be executed on each tree layer for code vector decoding. The computational complexity related to the construction of all subtrees on all tree layers is very high and increases exponentially with the increase of the sphere dimension $L_V > 3$. In addition, the trigonometric functions used in (12) in general are very expensive in terms of computational complexity.

In order to reduce the computational complexity the coding tree with the number of centroids in all subtrees is determined in advance and stored in ROM. In addition, also the trigonometric function values will be stored in lookup tables, as explained in the following section.

Even though shown only for the decoding, the principle of the coding tree and the trigonometric lookup tables can be combined with the Pre-Search and the Candidate-Exclusion methodology described in [1] very efficiently to reduce also the encoder complexity.

V. Efficient Storage of the Codebook

Under consideration of the properties of the apple-peeling code construction rule the coding tree and the trigonometric lookup tables can be stored in ROM in a very compact way:

A. Storage of the Coding Tree

For the explanation of the storage of the coding tree, the example depicted in Figure 4 is considered.

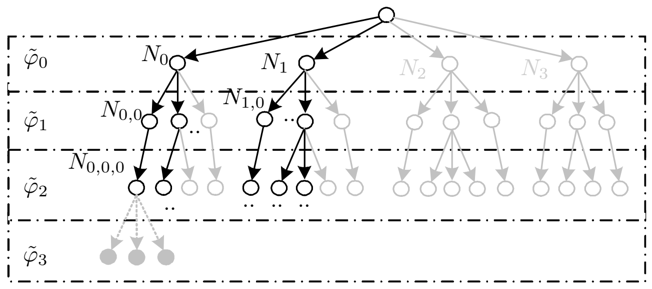

Fig. 4. *Compact Realization of the Coding Tree.*

Compared to Figure 3 the coding tree has 4 tree layers and is suited for a sphere of higher dimension $L_V = 5$. The number of nodes stored for each branch are denoted as N_{i_0} for the first layer, N_{i_0,i_1} for the next layer and so on. The leafs of the tree are only depicted for the very first subtree, marked as filled gray bullets on the tree layer for $\tilde{\varphi}_3$. The leaf layer of the tree is not required for decoding and therefore not stored in memory. Considering the principle of the sphere construction according to the apple-peeling principle, on each remaining tree layer for $\tilde{\varphi}_l$ with $l = 0, 1, 2$ the range of the respective angle, $0 \leq \tilde{\varphi}_l < \pi$, is separated into an even or odd number of angle intervals by placing the centroids on sub spheres according to (4) and (7). The result is that the coding tree and all available subtrees are symmetric as shown in Figure 4. It is hence only necessary to store half of the coding tree and also only half of all subtrees. In Figure 4 that part of the coding tree that must be stored in ROM is printed in black color while the gray part of the coding tree is not stored. Especially for higher dimension only a very small part of the overall coding tree must be stored in memory.

B. Storage of the Trigonometric Functions Table

Due to the high computational complexity for trigonometric functions, the storage of all function values in lookup tables is very efficient. These tables in general are very large to cover the complete span of angles with a reasonable accuracy. Considering the apple-pealing code construction, only a very limited number of discrete trigonometric function values are

required as shown in the following:
Considering the code vectors in polar coordinates, from one angle to the next the number of angle quantization levels according to equation (6) is constant or decreases. The number of quantization levels for $\tilde{\varphi}_0$ is identical to the code construction parameter N_{sp}. With this a **limit** for the number of angle quantization levels $N_{sp,l}$ for each angle $\tilde{\varphi}_l$, $l = 0 \cdots (L_V - 2)$ can be found:

$$N_{sp,l}(\tilde{\varphi}_{0,i_0} \cdots \tilde{\varphi}_{0,i_{l-1}}) \leq \begin{cases} N_{sp} & 0 \leq l < (L_V - 2) \\ 2N_{sp} & l = (L_V - 2) \end{cases} \quad (13)$$

The special case for the last angle is due to the range of $0 \leq \tilde{\varphi}_{L_V-2} < 2\pi$. Consequently, the number of available values for the quantized angles required for code vector reconstruction according to (4) and (7) is limited to

$$\tilde{\varphi}_l \in \begin{cases} (j + \frac{1}{2}) \cdot \frac{\pi}{N_{sp,l}} & \text{for } l < (L_V - 2) \\ (j + \frac{1}{2}) \cdot \frac{2\pi}{N_{sp,l}} & \text{for } l = (L_V - 2) \end{cases}, \quad (14)$$

with $j = 0 \cdots (N_{sp,l} - 1)$ as the index for the angle quantization level.
For the reconstruction of the vector $\tilde{\mathbf{c}}$ in cartesian coordinates according to (12) only those trigonometric function values are stored in the lookup table that may occur during signal compression/decompression according to (14). With the limit shown in (13) this number in practice is very small. The size of the lookup table is furthermore decreased by considering the symmetry properties of the cos and the sin function in the range of $0 \leq \tilde{\varphi}_l < \pi$ and $0 \leq \tilde{\varphi}_{L_V-2} < 2\pi$ respectively.

VI. Results

The described principles for an efficient spherical vector quantization are used in the SCELP audio codec to achieve the estimated computational complexity of 20-25 WMOPS as described in [1][1]. Encoding without the proposed methods is prohibitive considering a realistic real-time realization of the SCELP codec on a state-of-the-art General Purpose PC. The complexity estimation in the referenced contribution has been determined for a configuration of the SCELP codec for a vector length of $L_V = 11$ with an average bit rate of $r_0 = 2.8$ bit per sample plus additional bit rate for the transmission of the linear prediction coefficients. In the context of this configuration a data rate of approximately 48 kbit/sec for audio compression at a sample rate of 16 kHz could be achieved.
Considering the required size of ROM, the new codebook is compared to an approach in which a lookup table is used to map each incoming spherical index to a centroid code vector. The iterative spherical code design procedure results in $N_{sp} = 13$. The number of centroids on the surface of the unit sphere is determined as $M_{sp} = 18806940$ while the number of quantization intervals for the radius is $M_R = 39$. The codebook for the quantization of the radius is the same for the compared approaches and therefore not considered. In the approach with the lookup table M_{sp} code vectors of length $L_V = 11$ must be stored in ROM, each sample in 16 bit format. The required ROM size would be

$$M_{\text{ROM,lookup}} = 18806940 \cdot 16 \text{ Bit } \cdot 11 = 394.6 \text{ MByte}. \quad (15)$$

For the storage of the coding tree as proposed in this paper, only 290 KByte memory is required. With a maximum of $N_{sp,l} = 13$ angle quantization levels for the range of $0 \cdots \pi$ and $N_{sp,(L_V-2)} = 26$ levels for the range of $0 \cdots 2\pi$, the trigonometric function values for code vector reconstruction are stored in 2 KByte ROM in addition to achieve a resolution of 32 Bit for the reconstructed code vectors. Comparing the two approaches the required ROM size can be reduced with the proposed principles by a factor of

$$\frac{M_{ROM,lookup}}{M_{ROM,tree}} \approx 1390. \quad (16)$$

VII. Conclusion

It was shown in our previous contribution [1] that the SCELP low delay audio codec outperforms the G.722 audio codec in terms of achievable audio quality with an algorithmic delay of below 10 ms. In this contribution an auxiliary codebook has been proposed to reduce the computational complexity of the spherical code as applied in the SCELP. This codebook not only reduces the computational complexity of encoder and decoder simultaneously, it actually is a mandatory building block to achieve a realistic performance of the SCELP codec.

The codebook is based on a coding tree representation of the apple-peeling code construction principle and a lookup table for trigonometric function values for the transformation of a codeword into a code vector in cartesian coordinates. Considering the storage of this codebook in ROM, the required memory can be downscaled in the order of magnitudes with the new approach compared to an approach that stores all code vectors in one table as often used for trained codebooks.

References

[1] H. Krüger and P. Vary, “SCELP: Low Delay Audio Coding with Noise Shaping based on Spherical Vector Quantization”, submitted and accepted for EUSIPCO 2006.
[2] M. Schroeder, B. Atal, “Code-excited linear prediction (CELP): High-quality speech at very low bit rates”, ICASSP’85, pp. 937-940, 1985.
[3] T. Painter, “Perceptual Coding of Digital Audio”, Proc. of IEEE, vol. 88. no. 4, 2000.
[4] European Telecomm. Standards Institute, ”Adaptive Multi-Rate (AMR) speech transcoding” ETSI Rec. GSM 06.90 (1998).
[5] E. Gamal, L. Hemachandra, I. Shperling, V. Wei ”Using Simulated Annealing to Design Good Codes”, IEEE Trans. Information Theory, Vol. it-33, no. 1, 1987.
[6] J. Hamkins, ”Design and Analysis of Spherical Codes”, PhD Thesis, University of Illinois, 1996.
[7] J. B. Huber, B. Matschkal, ”Spherical Logarithmic Quantization and its Application for DPCM”, 5th Intern. ITG Conf. on Source and Channel Coding, pp. 349-356, Erlangen, Germany, 2004.
[8] Jayant, N.S., Noll, P., ”Digital Coding of Waveforms”, Prentice-Hall, Inc., 1984.
[9] K. Paliwal, B. Atal, ”Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame”, IEEE Trans. Speech and Signal Proc., vol. 1, no. 1, pp. 3-13, 1993.
[10] J.-P. Adoul, C. Lamblin, A. Leguyader, ”Baseband Speech Coding at 2400 bps using Spherical Vector Quantization”, Proc. ICASSP’84, pp. 45 - 48, March 1984.
[11] Y. Linde, A. Buzo, R.M.Gray, ”An Algorithm for Vector Quantizer Design”, IEEE Trans. Communications, 28(1):84-95, Jan. 1980.

[1]Complexity measured in a floating point implementation with a weighted instruction set similar to the weighted instruction set specified by ETSI for fixed point.