SCELP: LOW DELAY AUDIO CODING WITH NOISE SHAPING BASED ON SPHERICAL VECTOR QUANTIZATION

Hauke Krüger and Peter Vary

Institute of Communication Systems and Data Processing
RWTH Aachen University, Templergraben 55, D-52056 Aachen, Germany
email: {krueger,vary}@ind.rwth-aachen.de

ABSTRACT

In this contribution a new wideband audio coding concept is presented that provides good audio quality at bit rates below 3 bits per sample with an algorithmic delay of less than 10 ms. The new concept is based on the principle of Linear Predictive Coding (LPC) in an analysis-by-synthesis framework, as known from speech coding. A spherical codebook is used for quantization at bit rates which are higher in comparison to low bit rate speech coding for improved performance for audio signals. For superior audio quality, noise shaping is employed to mask the coding noise. In order to reduce the computational complexity of the encoder, the analysis-by-synthesis framework has been adapted for the spherical codebook to enable a very efficient excitation vector search procedure. The codec principle can be adapted to a large variety of application scenarios. In terms of audio quality, the new codec outperforms ITU-T G.722 [4] at the same bit rate of 48 kbit/sec and a sample rate of 16 kHz.

1. 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. [2]. *Parametric audio coding* is based on a source model. In this paper we focus on the linear prediction (LP) approach, the basis for todays highly efficient speech coding algorithms for mobile communications, e.g. [3]: An all-pole filter models the spectral envelope of an input signal. Based on the inverse of this filter, the input is filtered to form the LP residual signal which is quantized. Often vector quantization with a sparse codebook is applied according to the CELP (Code Excited Linear Prediction, [1]) approach to achieve very high bit rate compression. Due to the sparse codebook and additional modeling of the speakers instantaneous pitch period, speech coders perform well for speech but can not compete with perceptual audio coding for non-speech input. The typical algorithmic delay is around 20 ms.

In this paper the ITU-T G.722 is chosen as a reference codec for performance evaluations. It is a linear predictive wideband audio codec, standardized for a sample rate of 16 kHz. The ITU-T G.722 relies on a sub band (SB) decomposition of the input and an adaptive scalar quantization according to the principle of adaptive differential pulse code modulation for each sub band (SB-ADPCM). The lowest achievable bit rate is 48 kbit/sec (mode 3). The SB-ADPCM tends to become instable for quantization with less than 3 bits per sample.

In this contribution we propose a new coding scheme for low delay audio coding. In this codec, the principle of linear prediction is pre-

served while a spherical codebook is used in a gain-shape manner for the quantization of the residual signal at a moderate bit rate. The spherical codebook is based on the apple-peeling code introduced in [5] for the purpose of channel coding and referenced in [6] in the context of source coding. The apple-peeling code has been revisited in [7]. While in that approach, scalar quantization is applied in polar coordinates for DPCM, we consider the spherical code in the context of vector quantization in a CELP like scheme.

The principle of linear predictive coding will be shortly explained in Section 2. After that, the construction of the spherical code according to the apple-peeling method is described in Section 3. In Section 4, the analysis-by-synthesis framework for linear predictive vector quantization will be modified for the demands of the spherical codebook. Based on the proposed structure, a computationally efficient search procedure with *pre-selection* and *candidate-exclusion* is presented. Results are finally shown in Section 5 in terms of a comparison with the G.722 audio codec.

2. BLOCK ADAPTIVE LINEAR PREDICTION

The principle of linear predictive coding is to exploit correlation immanent to an input signal $x(k)$ by decorrelating it before quantization. For short term block adaptive linear prediction, a windowed segment of the input signal of length L_{LPC} is analyzed in order to obtain time variant filter coefficients $a_1 \cdots a_N$ of order 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 form the LP residual signal $d(k)$. $d(k)$ is quantized and transmitted to the decoder as $\tilde{d}(k)$. The LP synthesis filter, $H_S(z) = (H_A(z))^{-1}$, reconstructs from $\tilde{d}(k)$ the signal $\tilde{x}(k)$ by filtering (all-pole filter) in the decoder.

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.

2.1 Closed Loop Quantization

A linear predictive closed loop scheme can be easily applied for *scalar* quantization (SQ). In this case, the quantizer is part of the linear prediction loop, therefore also called *quantization in the loop*. Compared to straight pulse code modulation (PCM) closed loop quantization allows to increase the signal to quantization noise ratio (SNR) according to the achievable prediction gain immanent to the input signal.

Considering *vector* quantization (VQ) multiple samples of the LP residual 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 index prior to quantization in L_V-dimensional coding space. Vector quantization can provide

significant benefits compared to scalar quantization.

For closed loop VQ the principle of analysis-by-synthesis is applied at the encoder side to find the optimal quantized excitation vector $\tilde{\mathbf{d}}$ for the LP residual, as depicted in Figure 1. For analysis-by-synthesis, the decoder is part of the encoder. For each index i corresponding to one entry in a codebook, 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)

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

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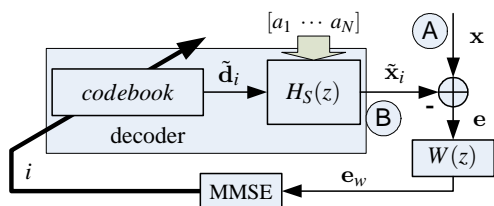

Figure 1: *Analysis-by-synthesis for VQ.*

$W(z)$ is based on the short term LP coefficients and therefore adapts to the input signal for perceptual masking similar to that in perceptual audio coding, e.g. [1].

The analysis-by-synthesis principle can be exhaustive in terms of computational complexity due to a large vector codebook.

3. SPHERICAL VECTOR CODEBOOK

Spherical quantization has been investigated intensively, for example in [6], [7] and [10]. The codebook for the quantization of the LP residual vector $\tilde{\mathbf{d}}$ consists of vectors that are composed of a gain (scalar) and a shape (vector) component. The code vectors $\tilde{\mathbf{c}}$ for the quantization of the shape component are located on the surface of a unit sphere. The gain component is the quantized radius $\tilde{R}$. Both components are combined to determine

$$\tilde{\mathbf{d}} = \tilde{R} \cdot \tilde{\mathbf{c}}. \tag{2}$$

For transmission, the codebook 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 codeword i_Q. In this section the design of the spherical codebook is shortly described first. Afterwards, the combination of the indices for the gain and the shape component is explained.

For the proposed codec a code construction rule named *apple-peeling* due to its analogy to peeling an apple in three dimensions is used to find the spherical codebook $\mathscr{C}$ in the L_V-dimensional coding space. Due to the block adaptive linear prediction, L_V and L_{LPC} are chosen so that $N_V = L_{LPC}/L_V \in \mathbb{Z}$. The concept of the construction rule is to obtain a minimum angular separation θ between codebook vectors on the surface of the unit sphere (centroids: $\tilde{\mathbf{c}}$) in all directions and thus to approximate a uniform distribution of all centroids on the surface as good as possible. As all available centroids, $\tilde{\mathbf{c}} \in \mathscr{C}$ have unit length, they can be represented in $(L_V - 1)$ angles $[\tilde{\varphi}_0 \cdots \tilde{\varphi}_{L_V-2}]$.

Due to the reference to existing literature, the principle will be demonstrated here by an example of a 3-dimensional sphere only, as depicted in Figure 2. There, the example centroids according to
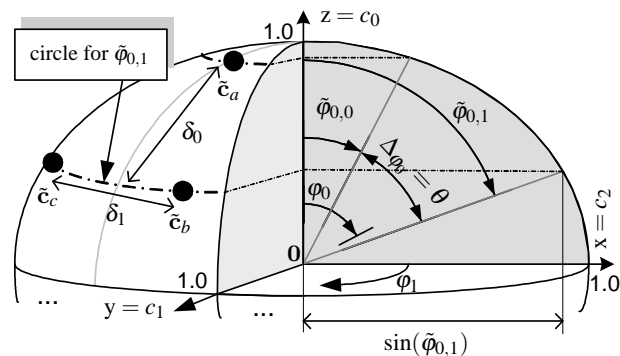

Figure 2: *3-dimensional sphere for apple-peeling code.*

the apple-peeling algorithm, $\tilde{\mathbf{c}}_a \ldots \tilde{\mathbf{c}}_c$, are marked as big black spots on the surface.

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 vector codebook, 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 sphere. The construction constraint to have a minimum separation angle θ in between neighbor centroids can be expressed also on the surface of the sphere: The distances between neighbor centroids in one direction is noted as δ_0 and δ_1 in the other direction. As the centroids are placed on a unit sphere and for small θ, the distances can be approximated by the circular arc according to the angle θ to specify the apple-peeling constraint:

$$\delta_0 \geq \theta, \ \delta_1 \geq \theta \text{ and } \delta_0 \approx \delta_1 \approx \theta \tag{3}$$

The construction parameter θ is chosen as $\theta(N_{sp}) = \pi/N_{sp}$ with the new construction parameter $N_{sp} \in \mathbb{Z}^+$ for codebook generation. By choosing the number of angles N_{sp}, the range of angle φ_0 is divided into N_{sp} angle intervals with equal size of $\Delta_{\varphi_0} = \theta(N_{sp})$. Circles (slash-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 $\varphi_0 \in \tilde{\varphi}_{0,i_0}$ and $\tilde{z} = \cos(\tilde{\varphi}_{0,i_0})$ in cartesian coordinates for all $\tilde{\mathbf{c}} \in \mathscr{C}$.

The radius of each circle 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 angle constraint, the separation angle Δ_{φ_1} is different from circle to circle and depends on the circle radius and thus $\tilde{\varphi}_{0,i_0}$:

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

With this, the number of intervals for each circle is

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

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 $\mathscr{C}$ for starting parameter N_{sp}.

For an efficient vector search described in the following section, with the construction of the sphere in the order of angles $\tilde{\varphi}_0 \to \tilde{\varphi}_1 \cdots \tilde{\varphi}_{L_V - 2}$, the coordinates of the sphere vector in cartesian must be constructed in chronological order, $\tilde{c}_0 \to \tilde{c}_1 \cdots \tilde{c}_{L_V - 1}$. As with angle $\tilde{\varphi}_0$ solely the cartesian coordinate in z-direction can be reconstructed, the z-axis must be associated to c_0, the y-axis to c_1 and the x-axis to c_2 in Figure 2.

Each centroid described by the tuple of $[i_0, i_1]$ is linked to a 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}. For centroid reconstruction, an index can easily be transformed into the corresponding angles $\tilde{\varphi}_0 \cdots \tilde{\varphi}_{L_V - 2}$ by sphere construction on the decoder side. For this purpose and with regard to a low computational complexity, an auxiliary codebook based on a coding tree can be used. The centroid cartesian coordinates c_l with vector index l are

$$\tilde{c}_l = \begin{cases} \cos(\tilde{\varphi}_l) \cdot \prod_{j=0}^{(l-1)} \sin(\tilde{\varphi}_j) & ; \, 0 \le l < (L_V - 1) \\ \prod_{j=0}^{(L_V - 2)} \sin(\tilde{\varphi}_j) & ; \, l = (L_V - 1) \end{cases}. \quad (8)$$

To retain the required computational complexity as low as possible, all computations of trigonometric functions for centroid reconstruction in Equation (8), $\sin(\tilde{\varphi}_{l/i})$ and $\cos(\tilde{\varphi}_{l/i})$, can be computed and stored in small tables in advance.

For the reconstruction of the LP residual vector $\tilde{\mathbf{d}}$, the centroid $\tilde{\mathbf{c}}$ must be combined with the quantized radius $\tilde{R}$ according to (2). 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 available M_R 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 as

$$i_Q = i_R \cdot M_{sp} + i_{sp} \quad (9)$$

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

$$2^r \ge M_{sp} \cdot M_R \quad (10)$$

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 codebook 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, codebooks are designed iteratively to provide the highest number of index combinations that still fulfill constraint (10).

4. OPTIMIZED EXCITATION SEARCH

Among the available code vectors constructed with the apple-peeling method the one with the lowest distortion according to Equation (1) must be found applying analysis-by-synthesis as depicted in Figure 1. This can be exhaustive for the large number of available code vectors that must be filtered by the LP synthesis filter to obtain $\tilde{\mathbf{x}}$. For the purpose of complexity reduction, the scheme in Figure 1 is modified as depicted in Figure 3. Positions are marked in both Figures with capital letters A and B in Figure 2 and C to M in Figure 3 to explain the modifications. The proposed scheme is applied for the search of adjacent signal segments of length L_V. For the modification, the filter $W(z)$ is moved into the signal paths marked as A and B in Figure 1. The LP synthesis
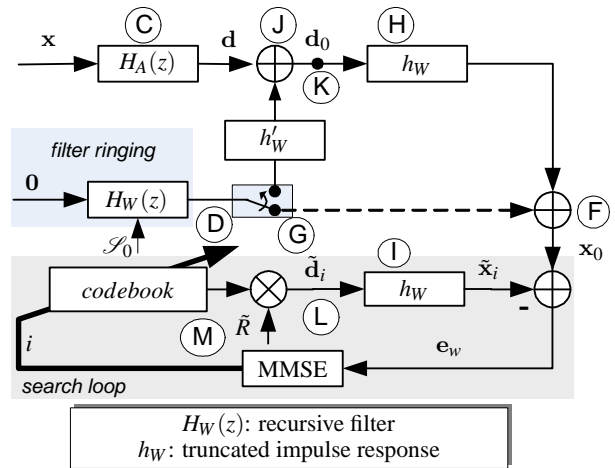

Figure 3: *Modified analysis-by-synthesis:*

filter is combined with $W(z)$ to form the weighted synthesis filter $H_W(z) = H_S(z) \cdot W(z)$ in signal path B. In signal branch A, $W(z)$ is replaced by the cascade of the LP analysis filter and the weighted LP synthesis filter $H_W(z)$:

$$W(z) = H_A(z) \cdot H_S(z) \cdot W(z) = H_A(z) \cdot H_W(z) \quad (11)$$

The newly introduced LP analysis filter in branch A in Figure 1 is depicted in Figure 3 at position C. The weighted synthesis filter $H_W(z)$ in the modified branches A and B have identical coefficients. These filters, however, hold different internal states: $\mathscr{S}$ according to the history of $d(k)$ in modified signal branch A and $\tilde{\mathscr{S}}$ according to the history of $\tilde{d}(k)$ in modified branch B. The *filter ringing* signal due to the states will be considered separately: As $H_W(z)$ is linear and time invariant (for the length of one signal vector), the filter ringing output can be found by feeding in a zero vector $\mathbf{0}$ of length L_V. For paths A and B the states are combined as $\mathscr{S}_0 = \mathscr{S} - \tilde{\mathscr{S}}$ in one filter and the output is considered at position D in Figure 3. The corresponding signal is added at position F if the switch at position G is chosen accordingly. With this, $H_W(z)$ in the modified signal paths A and B can be treated under the condition that the states are zero, and filtering is transformed into a convolution with the truncated impulse response of filter $H_W(z)$ as shown at positions H and I in Figure 3.

$$h_W = [h_{W,0} \quad \cdots h_{W,(L_V-1)}], h_W(k) \circ\!\!-\!\!\bullet H_W(z) \quad (12)$$

The filter ringing signal at position F can be equivalently introduced at position J by setting the switch at position G in Figure 3 into the corresponding other position. It must be convolved with the truncated impulse response h'_W of the inverse of the weighted synthesis filter, $h'_W(k) \circ\!\!-\!\!\bullet (H_W(z))^{-1}$, in this case.

Signal $\mathbf{d}_0$ at position K is considered to be the starting point for the *pre-selection* described in the following:

4.1 Complexity Reduction based on *Pre-selection*

Based on $\mathbf{d}_0$ the quantized radius $\tilde{R} = Q(\|\mathbf{d}_0\|)$, is determined first by means of scalar quantization Q and used at position M. Neighbor centroids on the unit sphere surface surrounding the unquantized signal after normalization ($\mathbf{c}_0 = \mathbf{d}_0 / \|\mathbf{d}_0\|$) are pre-selected in the next step to limit the number of code vectors considered in the *search loop*. Figure 4 demonstrates the result of the pre-selection in the 3-dimensional case: The apple-peeling centroids are shown as big spots on the surface while the vector $\mathbf{c}_0$ as the normalized input vector to be quantized is marked with a cross. The pre-selected

neighbor centroids are black in color while all gray centroids will not be considered in the search loop. The pre-selection can be
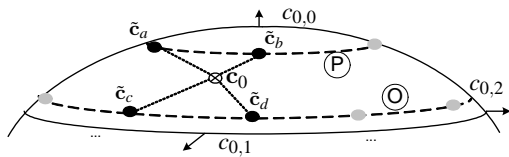

Figure 4: *Neighbor centroids due to pre-search.*

considered as a construction of a small group of candidate code vectors among the vectors in the codebook on a sample by sample basis. For the construction a representation of c_0 in angles is considered: Starting with the first unquantized normalized sample, $c_{0,l=0}$, the angle φ_0 of the unquantized signal can be determined, e.g. $\varphi_0 = \arccos(c_{0,0})$. Among the discrete possible values for $\tilde{\varphi}_0$ (defined by the apple-peeling principle, Eq. (4)), the lower $\tilde{\varphi}_{0,lo}$ and upper $\tilde{\varphi}_{0,up}$ neighbor can be determined by rounding up and down. In the example for 3 dimensions, the circles O and P are associated to these angles.

Considering the pre-selection for angle φ_1, on the circle associated to $\tilde{\varphi}_{0,lo}$ **one** pair of upper and lower neighbors, $\tilde{\varphi}_{1,lo/up}(\tilde{\varphi}_{0,lo})$, and on the circle associated to $\tilde{\varphi}_{0,up}$ **another** pair of upper and lower neighbors, $\tilde{\varphi}_{1,lo/up}(\tilde{\varphi}_{0,up})$, are determined by rounding up and down. In Figure 4, the code vectors on each of the circles surrounding the unquantized normalized input are depicted as $\tilde{c}_a$, $\tilde{c}_b$ and $\tilde{c}_c$, $\tilde{c}_d$ in 3 dimensions.

From sample to sample, the number of combinations of upper and lower neighbors for code vector construction increases by a factor of 2. The pre-selection can hence be represented as a binary code vector construction tree, as depicted in Figure 5 for 3 dimensions. The pre-selected centroids known from Figure 4 each correspond to one path through the tree. For vector length L_V, $2^{(L_V-1)}$ code vec-
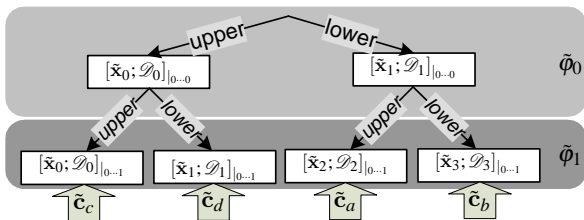

Figure 5: *Binary tree representing pre-selection.*

tors are pre-selected.

For each pre-selected code vector $\tilde{c}_i$, labeled with index i, signal $\tilde{x}_i$ must be determined as

$$\tilde{x}_i = \tilde{d}_i \star h_W = (\tilde{R} \cdot \tilde{c}_i) \star h_W. \tag{13}$$

Using a matrix representation

$$H_{w,w} = \begin{bmatrix} h_{w,0} & h_{w,1} & \cdots & h_{w,(L_V-1)} \\ 0 & h_{w,0} & \cdots & h_{w,(L_V-2)} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & h_{w,0} \end{bmatrix} \tag{14}$$

for the convolution, Equation (13) can be written as

$$\tilde{x}_i = (\tilde{R} \cdot \tilde{c}_i) \cdot H_{w,w}. \tag{15}$$

The code vector $\tilde{c}_i$ is decomposed sample by sample:

$$\begin{aligned} \tilde{c}_i &= \begin{bmatrix} \tilde{c}_{i,0} & 0 & 0 & \cdots & 0 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & \tilde{c}_{i,1} & 0 & \cdots & 0 \end{bmatrix} \\ &+ \dots \\ &+ \begin{bmatrix} 0 & 0 & 0 & \cdots & \tilde{c}_{i,(L_V-1)} \end{bmatrix} \\ &= \tilde{c}_{i,0} + \tilde{c}_{i,1} + \cdots + \tilde{c}_{i,(L_V-1)} \end{aligned} \tag{16}$$

With regard to each decomposed code vector $\tilde{c}_{i,l}$, signal vector $\tilde{x}_i$ can be represented as a superposition of the corresponding *partial convolution output vectors* $\tilde{x}_{i,l}$:

$$\tilde{x}_i = \sum_{j=0}^{L_V-1} \tilde{x}_{i,j} = \sum_{j=0}^{L_V-1} (\tilde{c}_{i,j} \cdot H_{w,w}). \tag{17}$$

The vector

$$\tilde{x}_i \big|_{[0 \cdots l_0]} = \sum_{j=0}^{l_0} \tilde{x}_{i,j} \tag{18}$$

is defined as the *superposed convolution output vector for the first $(l_0 + 1)$ coordinates* of the code vector

$$\tilde{c}_i \big|_{[0 \cdots l_0]} = \sum_{j=0}^{l_0} \tilde{c}_{i,j}. \tag{19}$$

Considering the characteristics of matrix $H_{w,w}$, with the first $(l_0 + 1)$ coordinates of the codebook vector $\tilde{c}_i$ given, the first $(l_0 + 1)$ coordinates of the signal vector $\tilde{x}_i$ are equal to the first $(l_0 + 1)$ coordinates of the superposed convolution output vector $\tilde{x}_i \big|_{[0 \cdots l_0]}$. We therefore introduce the partial distortion

$$\mathscr{D}_i \big|_{[0 \cdots l_0]} = \sum_{j=0}^{l_0} \left(x_{0,j} - \tilde{x}_{i,j} \big|_{[0 \cdots l_0]} \right)^2. \tag{20}$$

For $(l_0 + 1) = L_V$, $\mathscr{D}_i \big|_{[0 \cdots l_0]}$ is identical to the distortion $\mathscr{D}_i$ (Equation 1) that is to be minimized in the search loop.

With definitions (18) and (20), the pre-selection and the search loop to find the code vector with the minimal quantization distortion can be efficiently executed in parallel on a sample by sample basis: We therefore consider the binary code construction tree in Figure 5: For angle $\tilde{\varphi}_0$, the two neighbor angles have been determined in the pre-selection. The corresponding first cartesian code vector coordinates $\tilde{c}_{i^{(0)},0}$ for lower and upper neighbor are combined with the quantized radius $\tilde{R}$ to determine the superposed convolution output vectors and the partial distortion as

$$\begin{aligned} \tilde{x}_{i^{(0)}} \big|_{[0 \cdots 0]} &= \tilde{c}_{i^{(0)},0} \cdot H_{w,w} \\ \mathscr{D}_{i^{(0)}} \big|_{[0 \cdots 0]} &= \left(x_{0,0} - \tilde{x}_{i^{(0)},0} \big|_{[0 \cdots 0]} \right)^2 \end{aligned} \tag{21}$$

Index $i^{(0)} = 0, 1$ at this position represents the two different possible coordinates for lower and upper neighbor according to the pre-selection in the apple-peeling codebook in Figure 5. The superposed convolution output and the partial distortion are depicted in the square boxes for lower/upper neighbors.

From tree layer to tree layer and thus vector coordinate $(l-1)$ to vector coordinate l, the tree has branches to lower and upper neighbor. For each branch the superposed convolution output vectors and partial distortions are updated according to

$$\begin{aligned} \tilde{x}_{i^{(l)}} \big|_{[0 \cdots l]} &= \tilde{x}_{i^{(l-1)}} \big|_{[0 \cdots (l-1)]} + \tilde{c}_{i^{(l)},l} \cdot H_{w,w} \\ \mathscr{D}_{i^{(l)}} \big|_{[0 \cdots l]} &= \mathscr{D}_{i^{(l-1)}} \big|_{[0 \cdots (l-1)]} + \left(x_{0,l} - \tilde{x}_{i^{(l)},l} \big|_{[0 \cdots l]} \right)^2 \end{aligned} \tag{22}$$

In Figure 5 at the tree layer for $\tilde{\varphi}_1$, index $i^{(l=1)} = 0 \cdots 3$ represents the index for the four possible combinations of $\tilde{\varphi}_0$ **and** $\tilde{\varphi}_1$. The index $i^{(l-1)}$ required for Equation (22) is determined by the backward reference to upper tree layers. [1]

The described principle enables a very efficient computation of the distortion for all $2^{(L_V-1)}$ pre-selected code vectors compared to an approach where all possible pre-selected code vectors are determined and processed by means of convolution. If the distortion has been determined for all pre-selected centroids, the index of the vector with the minimal distortion can be found.

4.2 Complexity Reduction based on *Candidate-exclusion*(CE)

The principle of *candidate-exclusion* can be used in parallel to the pre-selection. This principle leads to a loss in quantization SNR. However, even if the parameters for the candidate-exclusion are setup to introduce only a very small decrease in quantization SNR still an immense reduction of computational complexity can be achieved.

For the explanation of the principle, the binary code construction tree in Figure 6 for dimension $L_V = 5$ is considered. During the pre-selection, *candidate-exclusion* positions are defined such that each vector is separated into sub vectors. After the pre-selection
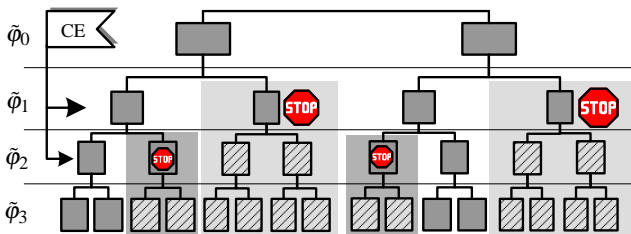

Figure 6: *Candidate-exclusion (CE)*.

according to the length of each sub vector a candidate-exclusion is accomplished, in Figure 6 shown at the position where four candidates have been determined in the pre-selection for $\tilde{\varphi}_1$. Based on the partial distortion measures $\mathscr{D}_{i^{(1)}} \mid_{0 \cdots 1}$ determined for the four candidates $i^{(1)}$ at this point, the two candidates with the highest partial distortion are excluded from the search tree, indicated by the STOP-sign. An immense reduction of the number of computations can be achieved as with the exclusion at this position, a complete **sub tree** will be excluded. In Figure 6, the excluded sub trees are shown as boxes with the light gray background and the diagonal fill pattern. Multiple exclusion positions can be defined for the complete code vector length, in the example, an additional CE takes place for $\tilde{\varphi}_2$.

5. RESULTS

The proposed codec principle is the basis for a low delay (around 8 ms) **audio** codec, realized in floating point arithmetic. Due to the codecs independence of a source model, it is suitable for a variety of applications specifying different target bit rates, audio quality and computational complexity. In order to rate the codecs achievable quality, it has been compared to the G.722 audio codec at 48 kbit/sec (mode 3) in terms of achievable quality for **speech**. The proposed codec has been parameterized for a sample rate of 16 kHz at a bit rate of 48 kbit/sec (2.8 bit per sample ($L_V = 11$) plus transmission

of $N = 10$ LP parameters within 30 bits). Speech data of 100 seconds was processed by both codecs and the result rated with the wideband PESQ measure. The new codec outperforms the G.722 codec by 0.22 MOS (G.722 (mode 3): 3.61 MOS; proposed codec: 3.83 MOS). The complexity of the encoder has been estimated as 20-25 WMOPS using a weighted instruction set similar to the fixed point ETSI instruction set. The decoders complexity has been estimated as 1-2 WMOPS. Targeting lower bit rates, the new codec principle can be used at around 41 kbit/s to achieve a quality comparable to that of the G.722 (mode 3). The proposed codec provides a reasonable audio quality even at lower bit rates, e.g. at 35 kbit/sec.

6. CONCLUSION

A new low delay audio coding scheme has been presented that is based on Linear Predictive coding as known from CELP, applying a spherical codebook construction principle named apple-peeling algorithm. This principle can be combined with an efficient vector search procedure in the encoder. Noise shaping is used to mask the residual coding noise for improved perceptual audio quality.

The proposed codec can be adapted to a variety of applications demanding compression at a moderate bit rate and low latency. It has been compared to the G.722 audio codec, both at 48 kbit/sec, and outperforms it in terms of achievable quality. Due to the high scalability of the codec principle, higher compression at bit rates significantly below 48 kbit/sec is possible.

REFERENCES

[1] M. Schroeder, B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates", Proc. ICASSP'85, pp. 937-940, 1985.

[2] T. Painter, "Perceptual Coding of Digital Audio", Proc. of IEEE, vol. 88. no. 4, 2000.

[3] European Telecomm. Standards Institute, "Adaptive Multi-Rate (AMR) speech transcoding" ETSI Rec. GSM 06.90 (1998).

[4] ITU-T Rec. G722, "7 kHz audio coding within 64 kbit/s" International Telecommunication Union (1988).

[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.

---

[1] Note that for the last angle in the pre-selection, two coordinates of the vector $\tilde{\mathbf{c}}$ are determined.