doi:10.2498/cit.1001767

Gaussian Mixture Model-based Quantization of Line Spectral Frequencies for Adaptive Multirate Speech Codec

Tihomir Tadić[1] and Davor Petrinović[2]

[1] Research & Development Center, Ericsson Nikola Tesla d.d., Zagreb, Croatia
[2] Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

In this paper, we investigate the use of a Gaussian Mixture Model (GMM)-based quantizer for quantization of the Line Spectral Frequencies (LSFs) in the Adaptive Multi-Rate (AMR) speech codec. We estimate the parametric GMM model of the probability density function (*pdf*) for the prediction error (residual) of mean-removed LSF parameters that are used in the AMR codec for speech spectral envelope representation. The studied GMM-based quantizer is based on transform coding using Karhunen-Loève transform (KLT) and transform domain scalar quantizers (SQ) individually designed for each Gaussian mixture. We have investigated the applicability of such a quantization scheme in the existing AMR codec by solely replacing the AMR LSF quantization algorithm segment. The main novelty in this paper lies in applying and adapting the entropy constrained (EC) coding for fixed-rate scalar quantization of transformed residuals thereby allowing for better adaptation to the local statistics of the source. We study and evaluate the compression efficiency, computational complexity and memory requirements of the proposed algorithm. Experimental results show that the GMM-based EC quantizer provides better rate/distortion performance than the quantization schemes used in the referent AMR codec by saving up to 7.32 bits/frame at much lower rate-independent computational complexity and memory requirements.

Keywords: Gaussian mixture models (GMMs), Karhunen-Loève transform (KLT), line spectral frequency (LSF), Adaptive Multi-Rate (AMR), speech coding, transform coding, vector quantization (VQ), entropy constrained scalar quantizer (ECSQ)

1. Introduction

Efficient coding of the short-term speech spectral envelope, represented in the form of linear predictive coding (LPC) parameters, has been a significant topic of research in low bit-rate speech coding for several decades. These LPC parameters are generally quantized in terms of line spectral frequencies (LSFs) using a vector quantizer (VQ) (Paliwal & Atal, 1993). For a given bit-rate, full search vector quantizers generally achieve the lowest distortion, but they also require a large amount of searching and memory at high bit-rates. In order to cope with the computational and memory requirements, structural constraints have been imposed to vector quantizers. The imposed structural constraints return computational or memory savings (sometimes both), but as a consequence they introduce suboptimal coding performance.

The AMR speech codec is based on the Algebraic Code Excited Linear Prediction (ACELP) coding scheme which represents a linear predictive (LP) codec with algebraic codebook excitation (Ekudden et al., 1999). It implements eight different source coding modes at bit-rates between 4.75 Kb/s and 12.2 Kb/s and it is capable of switching its bit-rate from one speech frame of 20 ms to another upon command. For the purpose of spectral envelope quantization, the AMR speech codec converts the LP coefficients to the Line Spectral Frequency (LSF) domain. Quantization methods used by AMR codec are relatively simple and they are based on either Split matrix quantization (SMQ) or Split vector quantization (SVQ) of the Moving average (MA) mean-removed LSF vector prediction error. They both belong to the above mentioned structural constrained VQ schemes and, as such, their coding performance suffers because in split VQs intraframe correlation between sub-vectors is not exploited. The same problem is with SMQ that groups only two components of adjacent vectors into five 2×2 matrices, again losing much of the intraframe redundancy.

In recent years, a parametric coding approach based on Gaussian mixture models (GMM) has been proposed for low-complexity memoryless vector quantization of LPC parameters (Hedelin & Skoglund, 2000, Subramaniam & Rao, 2001, Subramaniam & Rao, 2003). In such approach, the joint density of telephone-band speech LSF vectors is approximated by a weighted mixture of Gaussian component densities. The Karhunen-Loève transform (KLT) is used to design transform coders optimized for each individual mixture component. The transform domain vectors are scalar quantized using optimum mixture-specific bit allocation schemes. To select the best performing transform coder (mixture component), an input LSF vector is quantized with all transform coders and the computationally intensive spectral distortion (SD) measure is evaluated. By joining two or more LSF vectors from adjacent frames into one concatenated vector, the same procedure can be used to exploit the interframe correlations as well (So & Paliwal, 2005). In (Xiaoyan et al., 2006) the computational complexity is further reduced at the cost of reduced quantization fidelity by selecting a smaller number of mixture components which compete for the best performing transform coder. A practical scheme for a GMM-based variable-rate VQ by combining transform domain lattice quantization and entropy coding (arithmetic coder) has been proposed in (Zhao et al., 2007).

In this paper, we use, adapt and combine several above mentioned approaches for low-complexity GMM-based fixed-rate vector quantization of LSF parameters in the AMR codec. We approximate the joint density of the prediction error (residual) of the mean-removed LSF vectors by means of GMM. We combine a KLT-based adaptive transformation (decorrelation) of a vector process with entropy constrained scalar quantization in a soft decision scheme for the best mixture component selection. The scalar quantization of the decorrelated vector components is followed by Huffman entropy coders designed specifically for each mixture component and each transformed vector component. The ultimate goal is a design of a fixed-rate adaptive transform coder which exploits the intraframe correlation of the differentially encoded mean-removed LSF parameters across the whole vector length. By applying entropy coders to the output indices of individual scalar quantizers, we have exploited the advantage of entropy constrained coding (over the resolution constrained coding) to better adapt to the local statistics of the source and achieve a reduction of the average bit-rate. As mentioned above, the idea of using GMM-based spectral envelope quantizers has already been described in several papers. However, applying such a concept in a real codec introduces the problem of fixed code length available for the spectral envelope coding, which does not go along with the concept of entropy coding. By implementing a GMM-based spectral envelope quantizer in a typical CELP-based codec, the interaction between the excitation model (fixed and adaptive codebook) and the quantized spectral envelope model (LPC/LSF) comes to the fore. The CELP codec's closed loop nature makes their corresponding analysis inseparable. For this purpose, we have implemented and evaluated the proposed GMM-based spectral envelope quantizer in a real CELP-based codec. We have chosen the AMR codec as a typical representative of such a codec, as it is widely used in the GSM and UMTS systems. We propose two techniques to adapt the entropy coding to the fixed-rate modes of the AMR codec, which represent the main novelty in this paper compared to the previous work. The performance of such a quantizer will be analyzed by evaluating the incurred spectral envelope distortion and also by measuring the performance of the entire CELP codec by using the PESQ (ITU-T Rec. P.862, 2001) quality measure. As the entropy coders are known as more sensitive to bit error propagation, application of the proposed algorithm is limited to systems which employ some kind of an error protection algorithm. To lower the computational complexity, we also apply open-loop selection criteria to select *2 best* mixture components which are then further processed for the best mixture component selection. Results demonstrate that the proposed procedure achieves relatively high compression level with low quantization complexity.

This paper is organized as follows. In Section 2 we briefly review the LPC quantization methods used in the referent AMR codec (3GPP TS 26.104, v9.0.0), i.e. the computation and quantization methods of differentially encoded mean-removed LSF parameters (residuals). The concept of the proposed quantization method based on mean-removed KLT coding of Gaussian components is introduced in Section 3 together with brief descriptions of the fundamentals of Gaussian mixture models and KLT coding. In Section 4, the experiments and simulation results are summarized. The performance and complexity of the proposed LSF quantization scheme when applied to the spectrum coding of the AMR codec are discussed and evaluated. Section 5 presents conclusions.

2. Spectral Envelope Coding in the Referent AMR Speech Codec

The sampling frequency used in the AMR codec is 8 kHz and the speech encoding is performed on 20 ms speech frames. Therefore, each encoded speech frame accounts for 160 samples of the original speech. The LP coefficients are estimated once or twice for each speech frame, which makes the LPC analysis frequency either 50 or 100 times per second correspondingly. The LP coefficients are converted to the LSF domain and differentially encoded (3GPP TS 26.090). To exploit the inter-frame redundancies between succeeding LSF vectors, a mean vector is subtracted from each LSF vector and the 1st order MA linear prediction filter is applied. Thus, the actual quantization is performed on the prediction residual vectors.

For the 12.2 Kb/s mode, the AMR codec calculates two sets of LSF parameters. Two corresponding residual LSF vectors are joined into a matrix of dimension 10×2 that is jointly quantized using the SMQ. The matrix is split into 5 submatrices of dimension 2×2 (two elements from each vector) which are quantized with 7, 8, 8+1, 8, and 6 bits, respectively.

For all other modes, only one set of LSF parameters is calculated per each 20 ms speech frame. The residual LSF vectors are Split Vector quantized by partitioning the 10-dimensional vectors into subvectors of dimension 3, 3 and 4 and quantizing each of them with VQ resolutions between 7 to 9 bits per subvector. E.g. for the 10.2 Kb/s mode (the one with the highest rate), the first, the second, and the third subvectors are quantized with 8, 9, and 9 bits, respectively.

Early work (Farvardin & Laroia, 1989) showed that the adjacent LSF frames and neighboring LSF parameters in the same frame are strongly correlated. As a consequence of splitting the LSF coefficients from the same frame into multiple partitions, the SMQ and SVQ methods are not capable of exploiting their intraframe correlation entirely.

3. GMM-based LSF Vector Quantizer Description

The proposed quantizer block diagram is shown in Figure 1. The quantization scheme uses a GMM to parametrically model the *pdf* of the mean-removed LSF vector prediction errors (residuals). For the purpose of GMM estimation, these vectors are calculated and extracted from the referent AMR codec by processing the speech utterances from the training database. The estimated parameters of m linearly combined multivariate Gaussians (mixture components, or "soft clusters") are then used to design m KLT-based transform coders to decorrelate and encode the residuals. In order to select a suitable Gaussian component and the corresponding decorrelation matrix for a particular input vector, its probability of belonging to each of m components can be evaluated given the GMM model. The one that gives the highest probability should be a good choice in the sense of maximizing the transform coding gain (TCG) (Jayant & Noll, 1984) under the high-rate assumption (Gray, 1990). Alternatively, instead of probability evaluation, the selection criterion can be based on maximizing the TCG for individual input vector. As the mixture components overlap each other, in order to select the best mixture component for quantization of a particular LSF residual, a soft-decision scheme is used. This means that residual is quantized with the transform coder for each of m GMM components. Finally, quantized LSF vectors are reconstructed by inverse processing of quantized residuals and compared to the input LSF. The comparison is performed by computing the weighted LSP distortion measure, the same one that is also used in the referent AMR codec. Finally, the GMM component that incurs the least distortion is selected. Therefore, each of the mixture components competes to produce the

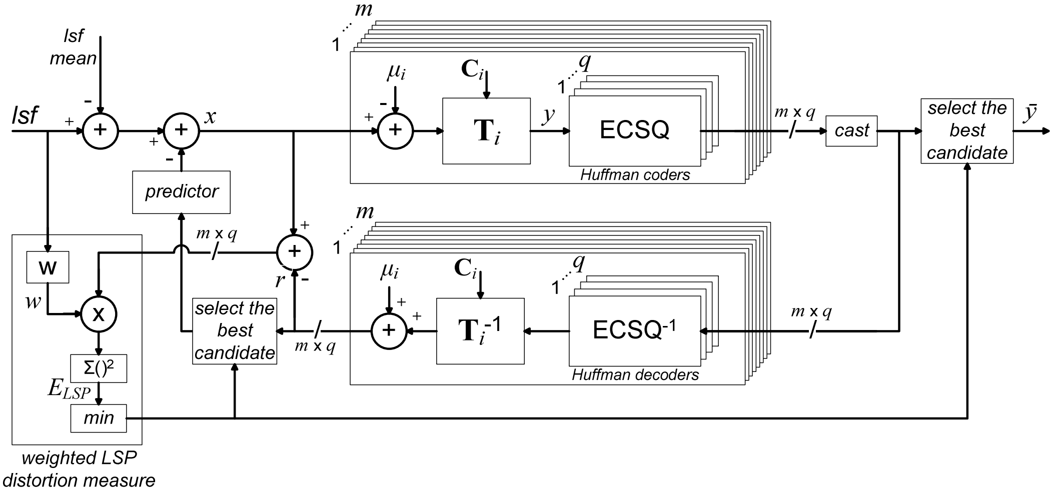

Figure 1. Block diagram of the GMM-based LSF vector quantizer ($m \times q$ version).

best quantized value given the past quantized value that is used in the MA predictor.

In order to reduce the algorithm complexity, we have also designed and evaluated an alternative system that uses the TCG principle to select a smaller number of transform domain residuals (M-best) that are chosen for distortion evaluation. Figure 2 shows a block diagram of such a modified GMM-based VQ system which selects 2 best mixture components among the m available using the TCG principle. The residual vector is quantized with only two transform coders for the two selected GMM components which then inverse processed for the weighted LSP distortion evaluation.

During the training phase, the quantizer design, which includes the *pdf* modeling and transform and entropy coder design, is iterated several times to improve the overall closed-loop performance. The estimated quantizer parameters are then used to evaluate the quantizer performance on speech utterances from the evaluation data base.

3.1. PDF estimation using Gaussian mixture models

The fact that *pdf* functions of real-life sources are rarely Gaussian invariably causes a performance degradation in scalar quantizers that are designed with a common assumption on Gaussian sources. As an alternative to a presumption that the *pdf* of a source is a standard function

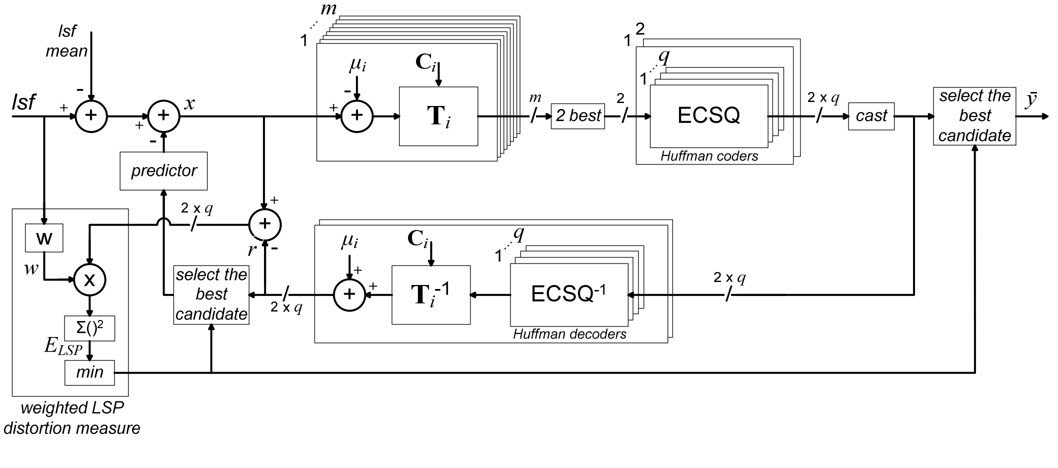

Figure 2. Block diagram of the GMM-based LSF vector quantizer with reduced computational complexity (*2-best* version).

such as Gaussian, the GMM can be used to *parametrically* model the *pdf* as close as desirable.

The joint-*pdf* of d-dimensional LSF residual vectors $\mathbf{X}$ can be approximated with a GMM model (Subramaniam & Rao, 2003) which is defined as a weighted sum of multivariate Gaussians given by

$$G(x|\Theta) = \sum_{i=1}^{m} \rho_i \mathrm{N}(x; \quad _i; \mathbf{C}_i), \tag{1}$$

$$\Theta = [m, \rho_1, \ldots, \rho_m, \quad _1, \ldots, \quad _m, \mathbf{C}_1, \ldots, \mathbf{C}_m], \tag{2}$$

$$\mathrm{N}(x; \quad ; \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^d det(\mathbf{C})}} e^{-\frac{1}{2}(x- \quad)^{\mathrm{T}}\mathbf{C}^{-1}(x- \quad)} \tag{3}$$

where $N(x; \quad ; \mathbf{C})$ is the normal multivariate distribution with mean vector and covariance matrix $\mathbf{C}$, m represents the number of clusters (mixture components), ρ_i is i-th mixture component weight, while d is the vector dimension. There is a trade-off related to the number of mixture components m. Larger number provides a more accurate *pdf* model, but may lead to an undue complexity and a risk of overfitting the estimated model thereby reflecting random properties associated with the limited training database.

For a given source database, the model parameters, Θ, are usually estimated using the well known expectation-maximization (EM) algorithm (Dempster et al., 1977) which iteratively computes the maximum likelihood estimate of Θ until the log likelihood converges.

For initialization of the GMM estimation procedure, the model parameters are initialized by applying the Linde-Buzo-Gray (LBG) algorithm (Linde et al., 1980) on the training vectors. As a result, m clusters are produced represented by their corresponding mean, , covariance matrix, $\mathbf{C}$, and cluster weight coefficient, ρ. These are then refined using the iterative EM algorithm.

3.2. Karhunen-Loève transform

When quantizing for minimum distortion under high-rate assumption (Gardner & Rao, 1995), the KLT is the optimal transformation for correlated Gaussian sources (Huang & Schultheiss, 1963). It is used here to exploit the intraframe correlation by decorrelating the LSF residuals. Each LSF residual vector is assigned to one of the Gaussian classes (mixture components), based on the classification measure. They are considered as approximately Gaussian and hence they can be best decorrelated using the KLT.

The covariance matrix of each Gaussian class can be diagonalized using the eigenvalue decomposition as

$$\mathbf{C}_i = \mathbf{T}_i diag(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,d})\mathbf{T}_i^T \tag{4}$$

where $i = 1, \ldots, m$ and $diag(\lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,d})$ is a diagonal matrix containing d descending eigenvalues. These actually represent variances $\lambda_{i,j}$ of the decorrelated components of cluster i, while $\mathbf{T}_i$ is a square orthogonal matrix with columns containing the corresponding eigenvectors of $\mathbf{C}_i$.

The orthogonal matrix $\mathbf{T}_i^T$ is actually the desired KLT transformation matrix of the i-th cluster that can be used to transform (decorrelate) the vector components obtained by subtracting the cluster mean, $\quad _i$, from the LSF residual vector x

$$y = \mathbf{T}_i^T(x - \quad _i) \tag{5}$$

Similarly, the reconstruction can be done by an inverse KLT by using the inverse relation

$$x = \mathbf{T}_i y + \quad _i \tag{6}$$

It should be noted that each cluster's own local statistics will produce its own unique transformation matrix, which is the main advantage of the GMM-based coding in comparison to the conventional transform coding with a single transformation matrix. In our experiments we have used a GMM with $m = 8$ mixture components.

To associate a mixture component to each input vector during the quantizer training phase, the TCG principle was used. This is a common measure of coding efficiency in transform coding and it is computed for every mixture component as the ratio of the arithmetic to the geometric means of the transformed coefficient variances

$$G_i = \frac{\frac{1}{d}\sum_{j=1}^{d}\lambda_{i,j}}{\left(\prod_{j=1}^{d}\lambda_{i,j}\right)^{\frac{1}{d}}} = \frac{arithmetic\ mean}{geometric\ mean} = \frac{AM}{GM} \tag{7}$$

However, we could not use the above defined principle for classification since the association procedure must be performed for each individual vector, while this measure is defined in the expectation sense through variances. However, since these variances are expected value of squared magnitudes, simple replacement of squared magnitudes for variances is consistent with the concept of maximizing transform coding gain. As the arithmetic mean of squared magnitudes AM is invariant for any orthonormal transformation $\mathbf{T}_i^T$, the G_i is maximum when the GM is minimum. Thus, we simply select the mixture component which minimizes the geometric mean of squared transformed coefficients, which is equivalent to minimizing the product of the transform domain magnitudes. Thereby, we are applying a "TCG like" classification measure for mixture component selection:

$$\begin{aligned} m_{sel} &= \arg\min_i \left(\left(\prod_{j=1}^{d} y_{i,j}^2 \right)^{\frac{1}{d}} \right) \\ &= \arg\min_i \left(\prod_{j=1}^{d} |y_{i,j}| \right) \end{aligned} \tag{8}$$

Note that such criteria belongs to a kind of open-loop selection because the quantization performance is not considered during the mixture component selection process.

3.3. Scalar quantization of the transformed components

If the nonlinear dependences between components of the LSF residual vector are ignored, then after applying the KLT, the components of the transformed vector y represent a realization of independent Gaussian scalar variables which can be independently quantized. As an alternative to optimal constrained resolution (CR) scalar quantizers (SQ) with fixed length output codes, ordinary uniform scalar quantizers can be used, combined with entropy coders on their outputs. In such a case, scalar quantizers with uniform quantization levels provide the best codec performance. Combined with entropy coders of their output indices, those quantizers provide the lowest distortion for a given average rate, which has been theoretically proven under high rate assumption (Gish & Pierce, 1968). The significance of this approach lies in the ability to perform the quantization by a simple division (by the chosen quantization step) and rounding operation, with no memory consumption for codebooks. However, it is necessary to describe the entropy coder with a corresponding model (Huffman table or Arithmetic coder). Such quantizers solely constrain the average length (entropy) of the output code, hence their name Entropy Constrained Scalar Quantizers (ECSQ) (György & Linder, 2002). Although they are superior to the CRSQ in the rate-distortion (RD) sense, their inability to constrain the maximum code length represents their main drawback. Depending on the input signal, a specific vector containing very rare output indices can generate an entropy code that is much longer than the (constrained) average code length. This fact represents a major limitation for a direct application of ECSQ coders in typical speech signal coders whose spectral envelope code length is predetermined and fixed. All transformed vector components can be quantized using the same quantization step size s_q. Since statistical properties of individual transformed components are different, generated symbol indices are further entropy encoded by using a predesigned set of Huffman coders (Huffman, 1952) individually designed for each transformed vector component and each mixture component.

3.4. Fixed-rate consideration

It is well known that entropy coders produce variable bit-rate output bit strings. In the design process, only the average rate is constrained and thus the name "entropy-constrained". In order to apply such coding technique to the AMR codec, some modifications were necessary. As the AMR codec supports 8 different fixed bit-rates (modes), available rate reserved for spectral envelope quantization is fixed for each of the 8 modes. For example, for 10.2 Kb/s mode, only 26 bits are reserved in each frame for this purpose. Thus, the entropy code must not exceed the available fixed length.

In order to constrain the length of the code, two possible techniques can be used. First is the variation of the step size for scalar quantization that directly affects the entropy of output indices. Second technique is the simple truncation of the transformed vector i.e. casting it's length to the specified bit-rate. Due to energy compaction property of the KLT transform, the significance of the transformed vector components is decreasing in accordance with the decreasing variances $\lambda_{i,j}$. Thus, a given number of least significant components can be easily ignored in the coding process without affecting the overall performance significantly. Adaptation of the step size requires side information that must be forwarded to the decoder to enable reconstruction. On the other hand, vector truncation can be detected and decoded without the side information. While decoding, the receiver simply keeps the track of the current bit position in the encoded string. If the maximum length is exceeded prematurely, before reaching the terminal (leaf) node in the Huffman table for the current vector component, then this component and all succeeding components up to d are simply replaced with the zero value (actually with the mean of that mixture component after reconstruction).

Our proposed approach is based on combining both described techniques for code length limitation, thus compensating for shortcomings of one or the other. Step size adaptation is performed using the forward algorithm. To simplify encoding of the step size, a fixed chosen number of predefined step sizes was selected. In our case, we designed a system with $q = 4$ different step sizes for four target entropies. A set of four ECSQ quantizers was then designed, one for each rate. Each ECSQ quantizer actually comprises d individual quantizers, one for each component, but since they all share identical step-size, we are considering them as a single quantizer with aggregate entropy equal to the sum of individual entropies. These aggregate target entropies are related to the maximum available code length. To illustrate this concept, the same example of 10.2 Kb/s AMR mode will be elaborated. In this mode, exactly 26 bits are available to encode the selected Gaussian mixture component, the selected step size and entropy coded indices of all transformed vector components. With eight mixture components and four possible step sizes, the remaining available code length is 21 bits. Thus, target entropies of the four ECSQ quantizers can be chosen in the neighborhood of 21 bits, e.g. as: $21+6$, $21+2$, $21-2$ and $21-6$. In such a context, "complicated" input vectors that require low probability symbols with lengthy codes obviously cannot be encoded using the highest target rate of 27 bits, since most probably the encoded string would not fit the available slot of 21 bits. Hopefully, coarse step of the fourth entropy coder with the average target rate of 15 bits is sufficient to squeeze its code in the available slot. On the other hand, "simple" input vectors can be easily encoded with the first coder that ensures the maximum fidelity due to its finest step size. Proper selection of target entropies is crucial for achieving optimum performance with the fixed rate. We have experimented with many different strategies for their selection, but before discussing these results, we must also consider the effects of the second technique for code length limitation.

Since it is not possible to be sure in advance which of the four ECSQs produces the desired bounded code, the logical solution is to apply all four of them starting from the one with the highest rate. The first one that produces desired bounded code can be selected. Regrettably, it can not be guaranteed that even the last ECSQ with the lowest target rate will indeed generate a bounded code. Proposed solution is the application of the second technique that truncates the transformed vector to the given number of components thus generating bounded entropy code. This technique can actually be applied to all four ECSQ outputs, thus giving not one, but four valid results with bounded code. Consequently, even the "complicated" vectors can be encoded using the first ECSQ with the highest target rate if these vectors are sufficiently truncated. Again, we can select one of the four valid results by using a "delayed decision" approach, i.e. by reconstructing LSF vectors from all four solutions and finally selecting the one that gives the least distortion. If all m candidates for mixture component are used, this gives a total of mq possible solutions among which the best one can be selected.

This brings us to the issue of encoding the selected combination. A simple binary code can be used for this purpose (e.g. 3 bits for GMM + 2 bits for the step size). However, since different combinations have different probabilities, it is even better to encode this information using the entropy code as well, which is also a natural approach since EC is used for quantizer indices.

To achieve the best possible results, we have individually designed q ECSQ quantizers for each of the transformed vector components and for each of the mixture components. Thus, we also have a total of mdq corresponding Huffman tables that fully describe the system.

3.5. Distortion measures for LPC parameters

As we have just described, the mixture component and the entropy coder selection in the proposed LSF quantization algorithm is done in a closed-loop manner. To select the GMM component and entropy coder that model a particular LSF residual vector x the best, the quantized LSF residual vector candidates are reconstructed. For every quantized candidate, the quantization indices are simply multiplied with the corresponding quantization step and reconstructed by the inverse KLT transform. Finally, the corresponding cluster means are added to form the reconstructed quantized LSF residual vector candidates r^i which approximate the original residual vector x. Now, we find the candidate index i which minimizes the weighted LSP distortion measure

$$m_{sel} = \arg\min_i E_{LSP}(r^i) \tag{9}$$

where

$$E_{LSP}(r^i) = \sum_{j=1}^{10} [x_j w_j - r^i_j w_j]^2. \tag{10}$$

This is the same measure that is used in the referent AMR codec for VQ entry selection. The weighting factors w_j are calculated from the corresponding unquantized LSF vectors according to (3GPP TS 26.090). For the 12.2 Kb/s mode, two sets of weighting coefficients are computed for the two LSF residuals. As the orthogonal bases of the transform space vary between mixture components, distance measure calculation, such as the one above, must be performed in the original space, hence the need for inverse processing of the quantized candidates.

So far, the GMM *pdf* modeling was based on unquantized residuals extracted from the referent AMR codec running on the training speech database. However, note that due to predictive nature of the MA structure, computation of these residuals requires the knowledge of quantized residuals of preceding LSF vectors. Thus, the database that was used for initial GMM training was affected by quantization properties of the original VQ algorithm. Furthermore, in order to train initial ECSQs which are optimized to individual mixture components, vectors from the database must be associated with mixture components and corresponding transform matrices. This initial association was performed based on the described TCG-like principle.

This initial GMM model with the corresponding initial set of ECSQs was then used to encode the training database. However, in this case the association was based on the described closed-loop selection criteria. This of course produces a new database of prediction residuals, since the quantized values are different from the original ones. The new database, together with the closed-loop associations were then used to train a new GMM model and a new set of ECSQs. Described procedure was iterated a few times.

3.6. Performance evaluation

Two measures were used to objectively quantify performance of the new quantizer. The first measure is a simple SD measure that is most commonly used for comparison of different LPC quantization techniques (Kleijn & Paliwal, 1995). It measures the RMS distance between log-spectral magnitude responses of the original and quantized LPC model for one speech frame. This measure is averaged for all frames of the training or evaluation database. Related to this measure are p2 and p4 outliers which were also computed, that give percentage of frames with SD distances above 2dB and 4dB respectively.

This distortion measure is useful for characterization of the spectral envelope quantization, but it is not necessarily related to the overall coder performance, especially for closed loop coders, such as AMR. Therefore, LSF vectors quantized using the proposed GMM based algorithm were inserted back into the original AMR implementation and used for speech coding and decoding using the standardized algorithm. Decoded speech utterances were then compared to the original (unquantized) speech sequences using the Perceptual Evaluation of Speech Quality (PESQ) algorithm which is an objective measurement tool defined in the ITU-T Recommendation P.862 for the evaluation of transmission quality. The same evaluation using SD and PESQ measures was performed for the original quantization algorithm as a baseline system.

4. Experiments and Simulation Results

We compare the performance of the referent AMR codec with several variations of the modified AMR codec using the GMM-based LSF quantization algorithm. The GMM-based LSF VQ is simulated in the Matlab environment. For evaluation purpose, the C source code of the referent AMR codec was modified in order to replace the original quantized LSF residual with the externally (Matlab) generated quantized residuals both in the encoder and decoder parts.

A training database of 56030 10-dimensinal LSF vectors was used. They were extracted using the referent AMR speech codec from various male and female speech utterances in several languages. Speech signals used for the training and evaluation database were all sampled with the conventional 8 kHz sampling rate. Unquantized LSF vectors and the corresponding residuals were extracted from the referent AMR implementation and used to estimate the initial GMM model with 8 mixture components with full covariance matrices. Estimation was performed on the training database using 100 iterations of the EM algorithm. Evaluation database was prepared in exactly the same way and comprised 18050 LSF vectors which were not part of the training database.

We have experimented with several different target entropy combinations for step size adaptation technique, as described in Section 3. They range from combinations having all the target entropies of the four ECSQ quantizers below or equal to the maximum available code length (e.g. for the 10.2 Kb/s mode: 26, $26-5$, $26-10$, $26-15$ bits), through the combinations having the target entropies symmetrically or asymmetrically placed around the maximum available code length (e.g. for the 10.2 Kb/s mode: $26+2$, $26+1$, 26, $26-1$ bits) to the combinations having all the target entropies greater or equal to the maximum available code length (e.g. for the 10.2 Kb/s mode: $26+6$, $26+4$, $26+2$, 26 bits). The experiments have shown that the combination using the highest target entropies (i.e. maximum available code length increased by 6, 4, 2 and 0 bits, respectively) shows the best quantization performance. This combination was used throughout all the simulations presented below. It can be noted that this combination of target entropies also provides the highest utilization of the selected fixed bit-rate, i.e. the least average number of unused bits.

The quantizer design procedure was iterated 10 times according to the procedure described in Section 3. It was observed that three iterations are sufficient for achieving optimal quantizer parameters in the sense of minimizing SD value and outliers percentage. Further iterations do not offer any additional improvement, due to slightly oscillatory behavior and no consistent convergence. This is due to the fact that both the model and ECSQs are modified in each iteration and the fact that each iteration uses a new database of LSF residuals.

Codec comparison using equal number of bits per LSF vector: First, we compare the referent and modified AMR codec using the same number of bits per frame, as it is defined by (3GPP TS 26.090). Performance was evaluated for all eight modes of the AMR codec. For 12.2 Kb/s mode, number of bits that are used for spectral envelope quantization is 38 bits; for 7.95 Kb/s it is 27 bits; then 26 bits are used for four modes: 10.2 Kb/s, 7.4 Kb/s, 6.7 Kb/s and 5.9 Kb/s, while the last two modes with the lowest rate: 5.15 Kb/s and 4.75 Kb/s use only 23 bits per frame. For our algorithm, these bits also include the entropy code for the selected mixture component and step-size selection code that indexes one of the four sets of Huffman tables for each mixture component.

The results for two algorithm versions and referent AMR codec are presented in Table 1. The first version ($m{\times}q$ – Figure 1) evaluates the LSP distortion measure for all 32 candidates (8 mixture components × 4 Huffman coders per component) to select the best quantization vector using the closed-loop approach. The second version (*2-best* – Figure 2) first selects 2 best mixture components among the 8 available using the described TCG-like principle, and then evaluates only 8 candidates (2 mixture components × 4 Huffman coders per component) in the closed-loop. PESQ score differences between the referent codec and the $m{\times}q$ version of the proposed GMM based algorithm are within -0.012 to 0.013, depending on the particular mode (bit-rate). If the results are averaged across all 8 modes, the average PESQ difference is only 0.003 in favor of the GMM-based quantizer. Thus, it can be concluded that the modified version performs almost equally to the referent codec in the PESQ score sense. However, much greater differences can be observed for the SD measure. The proposed quantizer reduces the SD distortion between 0.105 dB up to 0.292 dB, depending of the selected mode. In average, the SD improvement of 0.208 dB can be observed. Significant improvement can also be observed in outlier percentage reduction. In general, it could be noted that improvement of the GMM-based VQ for LSF quantization is more pronounced for the lower LSF bit-rates (e.g. 0.292 dB SD reduction for 4.75 Kb/s mode with 23 bits/frame).

With the same performance trends across the supported modes, the *2-best* version expectedly shows slightly inferior performance compared to the $m{\times}q$ version, but still clearly outperforms the referent codec. The reduced computational complexity comes at the cost of 0.018 worse average PESQ score, and 0.107 dB higher average SD value compared to the $m{\times}q$ version.

Although the spectral envelope is clearly quantized more accurately with our proposed algorithm, this gain cannot be observed in the PESQ score. This fact actually shows that the performance is bounded by the excitation quantization error (fixed and adaptive codebook) and not by the envelope error. Thus, using equal rate for our quantizer, makes little sense, since the rate is wasted on something that doesn't improve the overall codec performance. To verify this assumption, we have also designed a GMM-based quantizer that uses lower rate for spectral quantization than the referent codec.

Codecs comparison using GMM VQ with reduced rate: In order to investigate the possibility for rate reduction with the proposed GMM-based LSF quantization algorithm, empirical RD curves were computed, by varying the desired fixed LSF rate. For the 12.2 Kb/s mode that quantizes two LSF vectors as a single entity, the fixed rate was varied from 28 to 41 bits/frame. For the remaining modes, the fixed rate was varied between 18 and 29 bits/frame.

The measured RD pairs for the referent and modified AMR codecs in 12.2 Kb/s mode are shown in Figure 3a. The results show that the proposed $m{\times}q$ codec version outperforms the referent codec by 7.32 bits/frame in the average SD sense. Similarly, the *2-best* version achieves

AMR mode (Kb/s)	*Referent AMR codec*					*Modified AMR codec* ($m{\times}q$)					*Modified AMR codec (2-best)*				
	bits/ frame	PESQ	Avg. SD (in dB)	Outliers (in %)		bits/ frame	PESQ	Avg. SD (in dB)	Outliers (in %)		bits/ frame	PESQ	Avg. SD (in dB)	Outliers (in %)	
				2-4 dB	> 4 dB				2-4 dB	> 4 dB				2-4 dB	> 4dB
12.2	38	4.002	0.983	1.864	0.072	38	4.012	0.762	1.220	0.042	38	3.989	0.821	2.286	0.106
10.2	26	3.914	1.217	4.100	0.069	26	3.927	1.029	1.416	0.037	26	3.916	1.130	5.060	0.324
7.95	27	3.721	1.074	2.164	0.058	27	3.709	0.969	1.305	0.058	27	3.701	1.055	3.447	0.186
7.4	26	3.704	1.217	4.100	0.069	26	3.714	1.029	1.416	0.037	26	3.691	1.130	5.060	0.324
6.7		3.614					3.619					3.583			
5.9		3.488					3.488					3.466			
5.15	23	3.365	1.552	15.656	0.106	23	3.365	1.260	5.007	0.164	23	3.349	1.411	7.324	0.149
4.75		3.306					3.307					3.295			

Table 1. Performance comparison between the referent and modified AMR codecs using equal number of bits/frame for coding the LSF parameters.

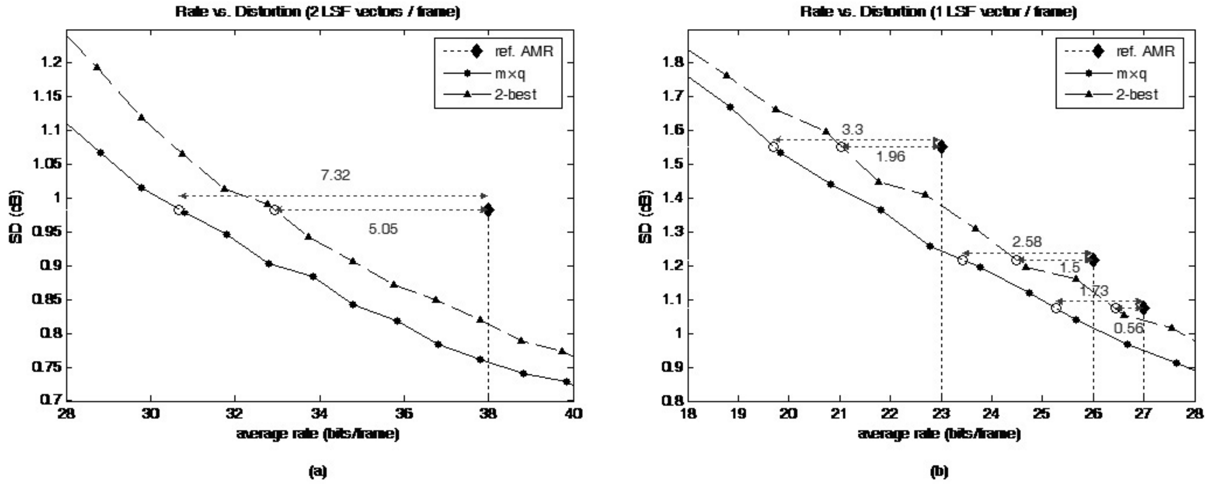

Figure 3. SD performance comparison: (a) between the ref. AMR codec in 12.2 Kb/s mode using 38 bits/frame (diamond marker) and modified AMR codec with GMM-based quantization designed for fixed rates between 28 and 40 bits/frame. (b) the same comparison for other AMR modes.

the same average SD as the referent codec by using 5.05 bits/frame less in average. As already discussed in the previous section, both modified versions achieve similar PESQ scores at reduced bit-rates as the referent codec at full rate (-0.017 PESQ difference for the 7 bit rate reduction in the $m \times q$ version and -0.019 difference for the 5 bit rate reduction in the *2-best* version).

The comparison of RD performance for modes which quantize only one LSF vector/frame is shown in Figure 3b. It is visible that the modified codecs outperform the referent AMR codec by 1.73 up to 3.3 bits/frame for the $m \times q$ version, depending of the specific AMR mode (bitrate). For the *2-best* version, the improvement is between 0.56 and 1.96 bits/frame in average. As already commented earlier for the case of equal LSF rate, greater rate reduction is achievable for lower bit-rate modes. This can also be observed from annotations in Figure 3b that explicitly show the rate reduction capabilities for each operating mode.

It was shown that the modified codec versions perform almost equally (using equal number of bits per LSF vector) or similar (for reduced bitrates at which they achieve the same average SD as the referent AMR codec) to the referent codec in the PESQ score sense. However, further reduction of the spectral envelope coding rate shows a clear PESQ score degradation (Figure 4). The relatively flat slope of PESQ score curves shows that the CELP codec is able to produce satisfying total results even in conditions when spectral envelope quantization causes a relatively high difference between the ideal envelope (determined by the LPC analysis) and the envelope described by the quantized LSF vector. This is a consequence of the CELP codec closed-loop structure. Such a codec is able to find an excitation which will synthesize a speech segment close enough to the input sequence, even with a big mismatch in the corresponding spectral envelopes. This fact is more visible for higher-rate codecs, as they use a high percentage of their total rate for excitation cod-

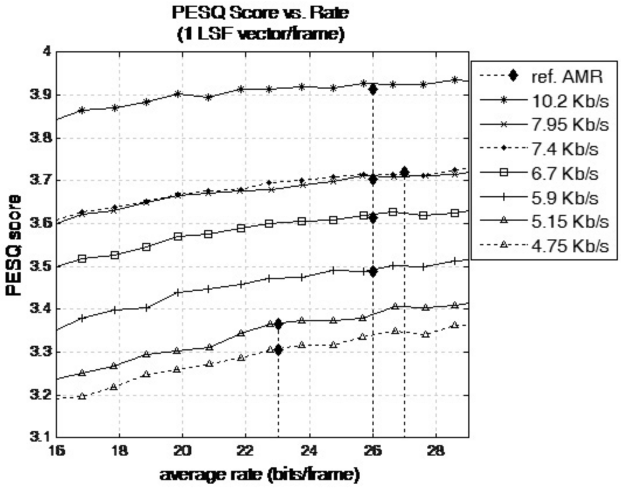

Figure 4. PESQ score vs. spectral envelope coding rate for the $m \times q$ version of the GMM-based LSF VQ. The curves represent different AMR codec modes which calculate 1 LSF vector/frame.

	SVQ (referent AMR codec)		*GMM-based LSF VQ (modified AMR codec)*			
	encoder	*decoder*	*encoder*		*decoder*	
			$m\times q$ ver.	*2-best ver.*	*$m\times q$ ver.*	*2-best ver.*
computational complexity (flops)	17405	0	10527	3990	270	
memory requirements (floats)	4352		2660		4780	

Table 2. Computational complexity and memory requirements comparison between the LSF VQs used in the referent and modified AMR codec versions.

ing (87% for the 10.2 Kb/s mode), which gives them the ability to find an appropriate excitation in order to compensate for the spectral envelope model imperfection. That is the reason why the curve slope becomes steeper for lower-rate codecs, as well as for lower spectral envelope coding rates.

Computational complexity and memory requirements comparison: Next, we illustrate the computational complexity and memory requirements of the GMM-based LSF VQ in comparison to the SVQ used in the referent AMR codec for the 10.2 Kb/s mode. Their complexities are evaluated in terms of floating point operations (flops) needed to quantize an LSF residual vector, i.e. in flops/frame. The evaluation results are summarized in Table 2.

To quantize a 10-dimensional LSF residual vector, the referent AMR codec uses 26 bits and needs 17405 flops/frame. It also needs 4352 floats to store the codebooks of the three subvector quantizers. For a given vector dimensionality d, these numbers grow exponentially in respect to the VQ resolution, i.e. the number of bits used to encode the vector (Gray & Neuhoff, 1998). On the contrary, the GMM-based VQ shows a computational advantage by functioning with a rate-independent complexity, which is linear in respect to the number of mixture components ($O(m)$) used to model the *pdf* of a source (Subramaniam & Rao, 2003). To quantize an LSF residual, the $m\times q$ version of the GMM-based LSF VQ needs 10527 flops/frame while the *2-best* version needs 3990 flops/frame for the same task. It takes 270 flops/frame to decode the quantized residual at the decoder side. Storage requirements are identical for both GMM versions. In order to store the mixture component means and KLT transformation matrices, a total of $m(d^2 + d) = 880$ floats is needed. Additional storage requirements for Huffman tables for encoding and decoding purposes are 1780 locations at the encoder side and 3900 locations at the decoder side.

As a conclusion, the $m\times q$ quantizer version appears to be 1.6 less complex comparing to the SVQ which is used for the same task in the referent AMR encoder. The *2-best* version shows even lower computational requirements, as in comparison with the SVQ it needs 4.6 times less flops to quantize an LSF vector. On the decoder side, the computational complexity of the LSF VQ is invariant to the modified codec version. There is a significant increase in decoding computational complexity compared to the simple reading of three indexed subvectors in the SVQ case. However, this increased LSF decoding complexity is completely negligible in comparison to the complexity of the whole AMR decoder. The memory requirements are also invariant to the modified codec version. On the encoder side, in comparison to the SVQ storage requirements of the referent AMR codec, the GMM-based LSF quantizers have about 1.6 times less storage requirements. On the decoder side, the GMM-based LSF quantizers have similar memory requirements as the SVQ of the referent AMR codec. It should be noted that about 60% of the encoder memory requirements and about 80% of the decoder memory requirements are dedicated to the Huffman tables in the GMM-based LSF quantizers. As they mainly contain integers representing symbols and row indices, they can be efficiently stored in narrower memory locations. For example, if we assume storing the SVQ codebook entries in 32-bit locations (float), by storing the Huffman table elements in 8-bit locations the above mentioned GMM-based LSF quantizer memory requirements can be further reduced by an approximate factor of 2.

5. Conclusions and Summary

In this paper, we have investigated the use of a GMM-based VQ for quantization of LSF vectors in the AMR speech codec. By applying the KLT on LSF residuals, correlation between LSF residual vector components within each frame is better exploited, leading to better quantization. Compared to the previous work, the main novelty in this paper lies in applying and adapting the entropy constrained coding for fixed-rate scalar quantization of transform domain LSF vector components thereby allowing for better adaptation to the local statistics of the source. Comparison of SD and corresponding outlier results show that the quantization performance of the modified AMR speech codecs significantly outperform the baseline referent AMR codec. However, the fact that this gain could not be observed in the PESQ score shows that the overall AMR performance is bounded by the excitation quantization error rather than the envelope error. As a conclusion, the GMM-based LSF VQ can be used to achieve referent AMR codec performance at lower bit-rates. For the modified $m \times q$ codec variant in 12.2 Kb/s mode, a saving of up to 7.32 bits/frame in average was achieved, while the other modes meet the referent AMR codec performance with an average LSF rate reduction of up to 3.3 bits/frame. Furthermore, in comparison to the referent codec, the GMM-based LSF VQ reduces the encoder computational complexity by 1.6 up to 4.6 times (depending on the modified codec version), along with the encoder memory requirements reduction by a factor of 1.6.

References

[1] 3GPP TS 26.090 V9.0.0 Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions, (2009).

[2] 3GPP TS 26.104 V9.0.0 ANSI-C code for the floating-point Adaptive Multi-Rate (AMR) speech codec, (2009).

[3] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39 (1977), pp. 1–38.

[4] E. Ekudden, R. Hagen, I. Johansson, J. Svedberg, The Adaptive Multi-rate Speech Coder. *Proceedings of the IEEE Workshop on Speech Coding*, (1999) Porvoo, Finland, pp. 117–119.

[5] N. Farvardin, R. Laroia, Efficient encoding of speech LSP parameters using the discrete cosine transformation. *Proceedings ICASSP*, (1989), pp. 168–171.

[6] W. R. Gardner, B. D. Rao, Theoretical analysis of the high-rate vector quantization of LPC parameters. *IEEE Transactions on Speech and Audio Processing*, 3 (1995), pp. 367–381.

[7] H. Gish, J. N. Pierce, Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14 (1968), pp. 676–683.

[8] R. M. Gray, *Source coding theory*. Kluwer Academic Publisher, 1990.

[9] R. M. Gray, D. L. Neuhoff, Quantization. *IEEE Transactions on Information Theory* 44 (1998), pp. 2325–2384.

[10] A. György, T. Linder, On the structure of optimal entropy constrained scalar quantizers. *IEEE Transactions on Information Theory*, 48 (2002), pp. 416–427.

[11] P. Hedelin, J. Skoglund, Vector quantization based on Gaussian mixture models. *IEEE Transactions on Speech and Audio Processing*, 8 (2000), pp. 385–401.

[12] Y. Huang, P. M. Schultheiss, Block quantization of correlated Gaussian random variables. *IEEE Transactions on Communications Systems*, 11 (1963), pp. 289–296.

[13] D. A. Huffman, A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40 (1952), pp. 1098–1101.

[14] ITU-T Rec. P.862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, (2001).

[15] N. S. Jayant, P. Noll, *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[16] W. B. Kleijn, K. K. Paliwal, *Speech Coding and Synthesis*. Elsevier, Amsterdam, 1995.

[17] Y. Linde, A. Buzo, R. M. Gray, An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 1 (1980), pp. 84–95.

[18] S. So, K. K. Paliwal, Multi-frame GMM-based block quantisation of line spectral frequencies. *Speech Communication*, 47 (2005), pp. 265–276.

[19] K. K. Paliwal, B. S. Atal, Efficient vector quantization of LPC parameters at 24 bits/frame. *IEEE Transactions on Speech and Audio Processing*, 1 (1993), pp. 3–14.

[20] A. D. Subramaniam, B. D. Rao, PDF optimized parametric vector quantization of speech line spectral frequencies. *IEEE Transactions on Speech and Audio Processing*, 11 (2003), pp. 130–142.

[21] A. D. SUBRAMANIAM, B. D. RAO, Speech lsf quantization with rate independent complexity, bit scalability and learning. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2 (2001), pp. 705–708.

[22] D. XIAOYAN, Z. YONGGANG, T. KUN, Improved Memoryless GMM VQ for Speech Line Spectral Frequencies. Presented at the *8th International Conference on Signal Processing*, Beijing, 2006.

[23] D. Y. ZHAO, J. SAMUELSSON, M. NILSSON, GMM-based Entropy-Constrained Vector Quantization. *Proceedings of ICASSP 2007*, (2007) Hawaii, USA.

Received: December, 2009
Revised: March, 2011
Accepted: June, 2011

Contact addresses:
Tihomir Tadić
Research & Development Center
Ericsson Nikola Tesla d.d.
Krapinska 45
10000 Zagreb
Croatia
e-mail: tihomir.tadic@ericsson.com

Davor Petrinović
Department of Electronic Systems and Information Processing
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3
10000 Zagreb
Croatia
e-mail: davor.petrinovic@fer.hr

TIHOMIR TADIĆ received the Dipl. ing. degree in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, in 1999 and the M.Sc. degree in electrical engineering from the same institution in 2010. His research interests include information theory, signal compression and speech processing. Since 1999 he is with the Ericsson Nikola Tesla d.d., Croatia, where he is currently working as a software engineer in the Research & Development Center.

DAVOR PETRINOVIĆ received the Dipl. ing. degree in electrical engineering from the Faculty of Electrical Engineering (currently, the Faculty of Electrical Engineering and Computing), University of Zagreb, Croatia, in 1988 and the M.Sc. and Ph.D. degree in electrical engineering from the same institution in 1996 and 1999, respectively. In 2005, he was appointed an Associate Professor at the Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb. In 2000-2001, he was a Fulbright Post-doctoral Scholar at the SCL Laboratory, University of California, Santa Barbara, and a Visiting Researcher at Sound and Image Processing Lab, School of Electrical Engineering, KTH, Stockholm, Sweden in 2005-06. His current research interests include speech and audio modeling, processing and coding.