SWITCHED-ADAPTIVE INTERFRAME VECTOR PREDICTION WITH BINARY-TREE SEARCHED PREDICTORS

Davorka Petrinović, Davor Petrinović[1]
Faculty of Electrical Engineering and Computing,
Zagreb, Croatia

Abstract - An approach for reduction of complexity of the switched adaptive interframe vector prediction used for coding of the speech spectrum envelope is proposed in this paper. To facilitate the search through the set of switched predictors used for prediction of the input LSF (Line Spectral Frequencies) vector, the predictors are organized in a binary-tree structure. For the conventional full-searched SIVP coder with $N=2^b$ predictors, prediction must be performed for all of them to determine the best one, while only 2b predictions are sufficient for the proposed binary-tree searched coder. A design procedure for obtaining optimal binary-tree structured predictors is given. The effectiveness of the proposed coder is evaluated and results are compared to the baseline full-searched coders as a function of the number of predictors, and resolution of vector quantizers used for quantization of the prediction residual. Discussion of possible applications of the proposed coder is also given.

INTRODUCTION

Vector Quantization (VQ) has frequently been studied and applied in signal coding due to its significant advantage over scalar quantization. However, for most real time applications the compromise between quality and complexity still has to be made. In the past, this resulted in many different lower complexity sub-optimal VQ schemes, generally treated in [1], with split VQ (SVQ) and multistage VQ (MSVQ) being the most frequently used. The complexity of these techniques can further be reduced by utilizing tree-structured VQ (TSVQ) for searching through the codebooks, or by overlaying an auxiliary codebook over the full-searched one to accelerate the search [2].

Another approach for decreasing complexity of the VQ is by exploiting correlation of successive vectors of the vector process being quantized. For example, quantization of the speech spectrum envelope (LPC parameters) is usually based on the Line Spectral Frequencies (LSF) representation due to their favorable quantization and interpolation properties. If the LSFs of each LPC analysis frame are represented as a vector, then the consecutive vectors can be efficiently coded by means of Predictive Vector Quantization (PVQ) [1]. Each component of the current *p*-dimensional vector is predicted based on all components of a certain number of preceding vectors. Therefore, vector prediction

[1] This study was performed at the Department of Electronic Systems and Information Processing and was financed by the Ministry of Science and Technology of Croatia under project No. 036-024

is performed using a predictor whose coefficients are actually $p \times p$ matrices. The difference between the original and the predicted vector, so called prediction error vector, is then vector quantized and transmitted.

The technique known as Switched-Adaptive Interframe Vector Prediction (SIVP) [4] is a vector-adaptive linear prediction method that is based on selection of one predictor among the fixed set of pre-designed predictors. Selection is performed independently for each input LSF vector. It has been shown that VQ complexity can be reduced considerably by using more switched predictors (e.g. N = 32, 64, …), causing only slight degradation of the coder performance [5]. On the other hand, the highest possible prediction gain can only by achieved by exhaustively searching through all the switched predictors, thus adding a particular computational load to the coding procedure.

In this paper, a method for reducing the prediction complexity of the SIVP-VQ scheme is proposed. We introduce the binary-tree search principal for finding the closest prediction of the input LSF vector. The prediction complexity reduction offered by this technique is equal to $N / (2\log_2(N))$ that increases with N. However, tree-searched instead of full-searched prediction results in sub-optimal coder performance, which can be justified for high number of switched predictors and high LPC frame rate.

CLASSICAL SIVP SCHEME

The first order vector linear prediction $\tilde{\mathbf{x}}(n)$ of a p-dimensional input LSF vector $\mathbf{x}(n)$ based on only one preceding vector $\mathbf{x}(n\text{-}1)$ can generally be written as:

$$\tilde{\mathbf{x}}(n) = \mathbf{A}\,\mathbf{x}(n-1) \ , \tag{1}$$

where $\mathbf{A}$ is a matrix predictor. Prediction error (residual) vector sequence is defined as:

$$\mathbf{e}(n) = \mathbf{x}(n) - \mathbf{A}\mathbf{x}(n-1) \ . \tag{2}$$

In the classical SIVP-VQ scheme, predictor $\mathbf{A}$ is realized using a set of N switched predictor matrices $\mathbf{A}_m$, $m = 0, \ldots, N\text{-}1$, where N is usually an integer power of 2, i.e. $N = 2^b$, to simplify the coding of the index m. In the actual coders, the prediction is based on the quantized preceding vector $\hat{\mathbf{x}}(n-1)$ rather then on the original $\mathbf{x}(n\text{-}1)$. In the coding procedure, the switched predictor matrix that minimizes the Euclidean norm of the prediction error vector $\mathbf{e}(n)$ is chosen for prediction of $\mathbf{x}(n)$. Prediction residual is then quantized and its VQ codebook index as well as the index m are transmitted to the receiving side.

Design of predictors is performed on a training base of input vectors in two phases [5]: first obtaining initial predictors with no quantization (so called open-loop design) and then in the second phase, by jointly optimizing predictors and VQ codebooks in the closed loop. The iterative design procedure used in both phases is essentially the same. The only difference is that in the open-loop design phase, the prediction is performed based on the unquantized LSF vectors, while in the second phase the quantized LSFs are used. In either case, in each iteration of the design procedure all input vectors are predicted with all N switched matrices. The prediction errors are then calculated and vectors are classified into N 'classes' each

associated with a predictor matrix yielding the least prediction error energy for that particular vector. At the end of an iteration, covariance matrices for each class are recalculated from preceding quantized input vectors $\hat{\mathbf{x}}(n-1)$ (for the closed-loop case) and unquantized input vectors $\mathbf{x}(n)$ assigned to that class, according to the expression:

$$\mathbf{C}_m = \sum_{n \in class\, m} \hat{\mathbf{x}}(n-1)\,\hat{\mathbf{x}}^{\mathrm{T}}(n-1) \quad , \quad \mathbf{\Gamma}_m = \sum_{n \in class\, m} \hat{\mathbf{x}}(n-1)\,\mathbf{x}^{\mathrm{T}}(n) \quad . \tag{3}$$

For the open-loop case $\hat{\mathbf{x}}(n-1)$ is replaced with $\mathbf{x}(n\text{-}1)$. The optimal predictor matrices $\mathbf{A}_m$ that minimize the sum of the prediction residual energy over the whole training base are obtained from (4) as described in [3]:

$$\mathbf{C}_m \cdot \mathbf{A}_m^{\mathrm{T}} = \mathbf{\Gamma}_m \quad , \quad m = 0, \ldots, N-1 \quad . \tag{4}$$

The procedure is repeated iteratively until the relative change of the prediction energy in two successive iterations is below the predefined threshold.

BINARY–TREE SEARCHED SWITCHED PREDICTORS

The structure of the proposed LSF coder with binary-tree searched switched predictors is shown in Fig.1.

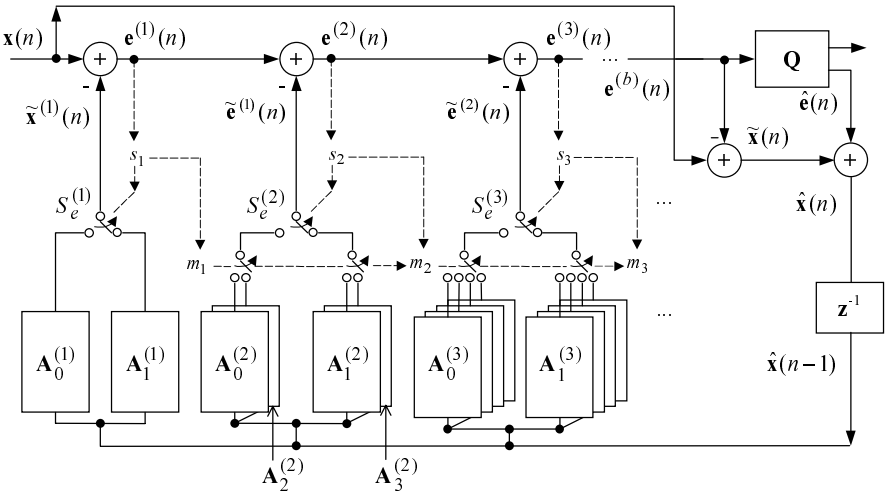

Figure 1. LSF coder with binary-tree searched switched predictors

The classical full-searched SIVP scheme has only one stage with all N predictors switched at that point. On the other hand, predictors of the binary-tree searched coder are distributed on b stages, with $N = 2^b$ predictors at the last stage. The number of predictors is doubled at each stage so the total number of predictors equals to $2N$-2. However, in the coding process only two predictions are performed at each stage so the total number of matrix multiplications is reduced from 2^b for the classical SIVP scheme to $2b$ for the proposed scheme.

Predictors are labeled according to the scheme depicted in Fig. 2. It is obvious that there are 2^r predictors $\mathbf{A}_m^{(r)}$ at the stage r with $m \in \{0, 1, \ldots, 2^r\text{-}1\}$. However, the index of the predictor that can be chosen at the stage r, denoted with m_r, is not arbitrary, i.e. it depends on the index $m_{r\text{-}1}$ of the predictor selected at the previous stage and on the position of the switch $S_e^{(r)}$, according to the expression:

$$\begin{aligned} m_0 &= 0 \ , \\ m_r &= 2\, m_{r-1} + s_r \ , \quad r = 1, \ldots, b \ . \end{aligned} \tag{5}$$

The position of the switch $S_e^{(r)}$ is coded with a binary variable s_r (0 for left, 1 for right). Determination of s_r will be explained in sequel.

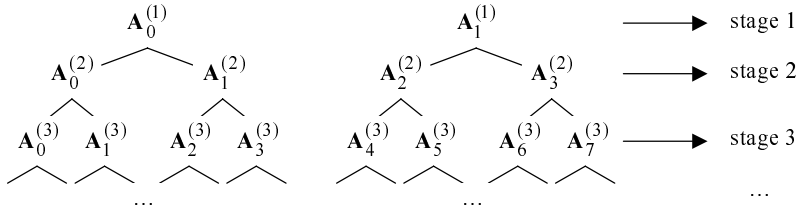

Figure 2. Labeling of the individual binary-tree searched switched predictors

The prediction of $\mathbf{x}(n)$ at the first stage, can be expressed as:

$$\tilde{\mathbf{x}}^{(1)}(n) = \mathbf{A}_{m_1}^{(1)}\, \hat{\mathbf{x}}(n-1) \ , \tag{6}$$

where $\mathbf{A}_{m_1}^{(1)}$ can be one of the two prediction matrices at this stage, $\mathbf{A}_0^{(1)}$ or $\mathbf{A}_1^{(1)}$, i.e. $m_1 = s_1$. Accordingly, the first stage prediction error vector is defined as:

$$\mathbf{e}^{(1)}(n) = \mathbf{x}(n) - \tilde{\mathbf{x}}^{(1)}(n) = \mathbf{x}(n) - \mathbf{A}_{m_1}^{(1)}\, \hat{\mathbf{x}}(n-1) \ . \tag{7}$$

In the coding process, selection between two predictors is based on minimization of the prediction error energy (8), thus determining the value of s_1 (position of the switch $S_e^{(1)}$):

$$\min_{s_1} \left(\left\| \mathbf{e}^{(1)}(n) \right\|^2 \right) \ \Rightarrow \ s_1 \ , \tag{8}$$

where $\| \ \|$ denotes the Euclidean norm. Since the prediction error $\mathbf{e}^{(1)}(n)$ is actually the input vector process at the second stage, its prediction is calculated based on corresponding switched predictors $\mathbf{A}_{m_2}^{(2)}$:

$$\tilde{\mathbf{e}}^{(1)}(n) = \mathbf{A}_{m_2}^{(2)}\, \hat{\mathbf{x}}(n-1) \ . \tag{9}$$

Although there are four predictors at this stage, the prediction is performed using only two of them for which the indexes satisfy the expression $m_2=2m_1+s_2$, with s_2

equal to either 0 or 1. The prediction error vector at the second stage is given by:

$$\mathbf{e}^{(2)}(n) = \mathbf{e}^{(1)}(n) - \tilde{\mathbf{e}}^{(1)}(n) = \mathbf{x}(n) - (\mathbf{A}^{(1)}_{m_1} + \mathbf{A}^{(2)}_{m_2})\,\hat{\mathbf{x}}(n-1) \ . \tag{10}$$

The value of s_2 defining the selected predictor $\mathbf{A}^{(2)}_{m_2}$ is determined analogously to (8) based on the minimal prediction error energy at that stage.

Prediction error at the last stage $\mathbf{e}^{(b)}(n)$ is the input to the quantizer and is found by analogy with (8),(9),(10) for all succeeding stages:

$$\mathbf{e}^{(b)}(n) = \mathbf{x}(n) - \left(\sum_{r=1}^{b} \mathbf{A}^{(r)}_{m_r} \right) \hat{\mathbf{x}}(n-1) = \mathbf{x}(n) - \tilde{\mathbf{x}}(n) \ , \tag{11}$$

where $\tilde{\mathbf{x}}(n)$ denotes the total prediction of the input vector $\mathbf{x}(n)$.

As with the classical SIVP scheme only one prediction is needed on the decoder side, since according to (11), the sum of matrices selected along all possible paths of the binary-tree can be pre-computed and stored at the decoder. The predictor index from the final stage m_b uniquely defines the position of all switches $s_1, \ldots, s_b$, as well as the indexes at all preceding stages $m_{b\text{-}1}, m_{b\text{-}2}, \ldots, m_1$. Therefore, the binary-tree decoder matrices, denoted with $\mathbf{D}_{m_b}$, can be found for $m_b \in \{0, 1, \ldots, 2^b\text{-}1\}$ as in:

$$\mathbf{D}_{m_b} = \sum_{r=1}^{b} \mathbf{A}^{(r)}_{m_r} \ , \text{ where } m_r = \sum_{i=1}^{r} s_i 2^{r-i} \ . \tag{12}$$

Design of the binary-tree searched switched predictors

Design procedure of the binary-tree searched switched predictors is essentially the same as the one for the classical SIVP predictors, except for the fact that it is performed in several stages with 2^r predictors designed at stage r. In the first stage, input vectors are classified into two classes and covariance matrices $\mathbf{C}^{(1)}_0$, $\mathbf{C}^{(1)}_1$, $\mathbf{\Gamma}^{(1)}_0$ and $\mathbf{\Gamma}^{(1)}_1$ are calculated as in (3) for N=2, resulting with predictor matrices, $\mathbf{A}^{(1)}_0$ and $\mathbf{A}^{(1)}_1$, according to (4). Superscript $^{(1)}$ indicates that all given matrices belong to the first stage, i.e. labeling of covariance matrices is the same as for the predictor matrices depicted in Fig.2.

In all following stages, each of the 'parent' classes is split into two new sub-classes. Calculation of covariance matrices for any new sub-class pair, $\mathbf{C}^{(r)}_{2m_{r-1}+0}$ and $\mathbf{C}^{(r)}_{2m_{r-1}+1}$, is analogous to (3) except it is now based only on a subset of vectors associated with the 'parent' class having covariance matrix $\mathbf{C}^{(r-1)}_{m_{r-1}}$. Since all vectors of two sub-classes are contained in their parent class, the relation between covariance matrices is:

$$\mathbf{C}^{(r-1)}_{m_{r-1}} = \mathbf{C}^{(r)}_{2m_{r-1}+0} + \mathbf{C}^{(r)}_{2m_{r-1}+1}, \quad r = 2, \ldots, b \tag{13}$$

On the other hand, matrices $\mathbf{\Gamma}_{m_r}^{(r)}$ are calculated differently since the input vectors into stages 2 up to b are actually the prediction error vectors from previous stages, $\mathbf{e}^{(r-1)}(n)$. They are obtained based on the same subsets of vectors used for calculation of $\mathbf{C}_{m_r}^{(r)}$. The expression for $\mathbf{\Gamma}_{m_r}^{(r)}$ is:

$$\mathbf{\Gamma}_{m_r}^{(r)} = \sum_{n \in class\, m_r} \hat{\mathbf{x}}(n-1)\ \mathbf{e}^{(r-1)\,\mathrm{T}}(n) \quad , \quad r = 2, \ldots, b \quad . \tag{14}$$

Equation analogous to (13) between covariance matrices of any two sub-classes and their 'parent' class is also valid for matrix $\mathbf{\Gamma}$.

For the open-loop design procedure, the input vectors for any new pair of sub-classes are iteratively reclassified and covariance and predictor matrices are recalculated until the relative change of the total prediction error energy between two subsequent iterations is sufficiently low. When all sub-classes on a given stage are processed, the design is repeated for the next stage. All resulting $2N$-2 predictors are then used in the closed-loop design procedure as the initial predictors. The error vectors $\mathbf{e}^{(b)}(n)$ resulting from the final iteration are used as a training base for the initial VQ codebook design.

During closed-loop predictor design, coding is performed as depicted in Fig.1. utilizing binary search through predictors and vector quantization of the prediction residual. Each input vector $\mathbf{x}(n)$ is associated with an index $m_b(n)$ from the last stage of the prediction, uniquely defining the classes at all stages. After all vectors are coded, indexes $m_b(n)$ are used for an update of covariance matrices and corresponding predictors, that are forwarded to the following iteration. Update of the VQ codebook is also performed as with the conventional SIVP closed-loop design [5].

SIMULATIONS

The training speech database used for design of the LSF coder contains 20 minutes of speech spoken by 4 speakers, recorded with a low quality condenser microphone. Testing of the coder was performed using the evaluation database with 10 utterances, each 1 minute long, spoken by 10 other speakers recorded with a studio dynamic microphone. Speech signals were sampled at 8kHz and the full speech spectrum bandwidth (0-4kHz) was analyzed. Both databases were analyzed using robust LPC analysis, with prediction order p=10 combined with fixed pre-emphases and bandwidth expansion of 10 Hz. LPC analysis was performed on speech segments of 25 ms, with a frame update of 50 or 100 frames/s.

Open-loop evaluation

Firstly, the proposed binary-searched prediction scheme was evaluated using an ideal quantizer (open-loop) for number of stages b=1, 2, 3, 4 and 5 (N=2,4,8,16 and 32). Prediction gain given by (15) was used as a performance measure.

$$G = 10 \log_{10} (\mathrm{E} (\| \mathbf{x}(n) \|^2) / \mathrm{E} (\| \mathbf{e}(n) \|^2)) \quad , \ [\mathrm{dB}] \ , \tag{15}$$

where E denotes expectation. The same expression was used for both cases: the classical SIVP where $\mathbf{e}(n)$ represents the prediction error as in (2), as well as for the proposed method where $\mathbf{e}(n)$ is actually the last stage prediction error $\mathbf{e}^{(b)}(n)$. The resulting prediction gain values are shown in Fig. 3. as a function of b.

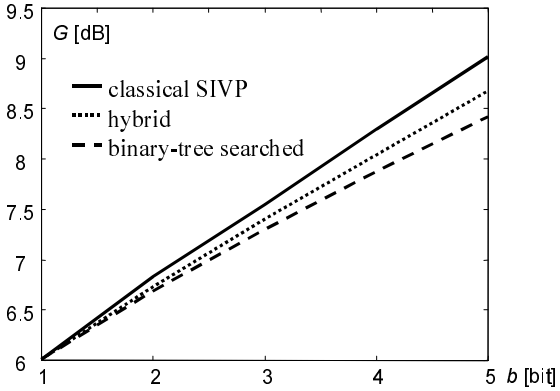

Figure 3. Prediction gain vs. b, for three types of open-loop coders at 50 frame/s

The sub-optimality of the binary-searched scheme compared to the full-searched one is obvious in Fig 3. and it increases with the number of predictors. To investigate the reasons for this behavior, another test was performed. A new hybrid coder was designed that uses the binary-tree decoder matrices $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_{2^b-1}$, in the full-searched prediction scheme. It can be seen in Fig.3 that the prediction gain of this hybrid coder lays somewhere in the middle of the other two.

The results for the hybrid coder are worse then those obtained by the classical approach although the exhaustive search was performed in both cases. This is due to the fact that matrices $\mathbf{D}_i$ are not optimal for the full-searched scheme. On the other hand, the sub-optimality induced by the binary-tree search can be assessed by comparing hybrid and binary-tree coders. It can be explained by the fact that once the 'parent' predictor matrix is selected as best for predicting a vector on a certain stage, prediction in all subsequent stages will be performed based on that branch of the predictor matrix tree. The selection is performed without anticipating the energy of the prediction residual in the following stages. Therefore, if the number of stages is higher, the higher is the chance that the wrong path will be chosen, so the sub-optimality increases.

Closed-loop evaluation

For the closed loop evaluation, baseline coders and the coders with binary-tree searched predictors were designed using the classical SIVP and the proposed binary-tree design procedure respectively, both for $N = 16$, 32 and 64.

Quantization of the prediction residual was performed using split vector quantization (SVQ) with 4-6 split and full searched codebooks for all coders. The evaluation was performed for two typical frame rates (*FR*): 50 and 100 frame/s. The sizes of the VQ codebooks were varied in order to obtain total bit rates *B*: from 19 to 23 bits/frame for *FR*=50 and from 15, to 19 bits/frame for *FR*=100. Coding as well as the VQ design was performed based on the Weighted Mean Squared Error (WMSE) between the original and the quantized LSF vectors. The weights were calculated from the analytical spectral sensitivity of each input LSF vector [6].

The coder performance was evaluated using the average Log Spectral Distortion (SD) measure found as the RMS error between the original and the quantized LPC filter spectrum. The average SD for the evaluation database, *FR*=50 is plotted in Fig. 4. as a function of the total number of bits/frame. The differences between baseline coders having different number of switched predictors are rather small, even smaller compared to the results reported in [5]. The proposed coders are sub-optimal as expected, with largest increase of SD for the coder with the highest number of predictors, as already discussed in the open-loop evaluation.

The results for *FR*=100 are similar to those shown on Fig. 4., however the differences in SD between binary-tree searched and baseline coders are slightly higher compared to the case with *FR*=50. This can be explained by the fact that for higher frame rate the influence of the prediction part of the coder is more pronounced. The necessary increase of the number of bits per frame required for achieving the baseline quality with the proposed coders having *b*=4, 5 and 6 stages is approximately 0.6, 0.9 and 1.25 bit at *FR*=50 and 0.7, 1.0 and 1.4 bit at *FR*=100.

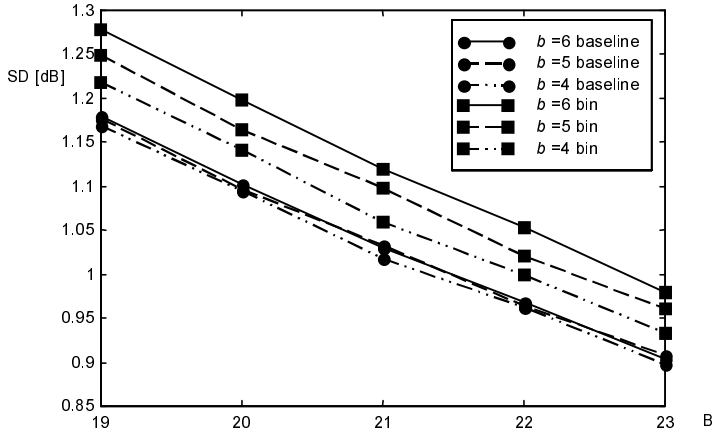

Figure 4. Average SD vs. *B* bits/frame for baseline coders and coders with binary-tree searched switched predictors, *FR*=50

The proposed LSF coders with binary-tree searched predictors are primarily targeted for applications with strict complexity constraint, i.e. where the increase of the bit-rate can be sacrificed for complexity reduction. Discussion of the LSF coder complexity as a function of spectral distortion and type of prediction will be given next, based on results shown in Figures 5 a) for *FR*=50 and 5 b) for *FR*=100.

Complexity is expressed as a total number of operations (add., mult. and compar.) for the whole closed loop LSF coding procedure (prediction and quantization).

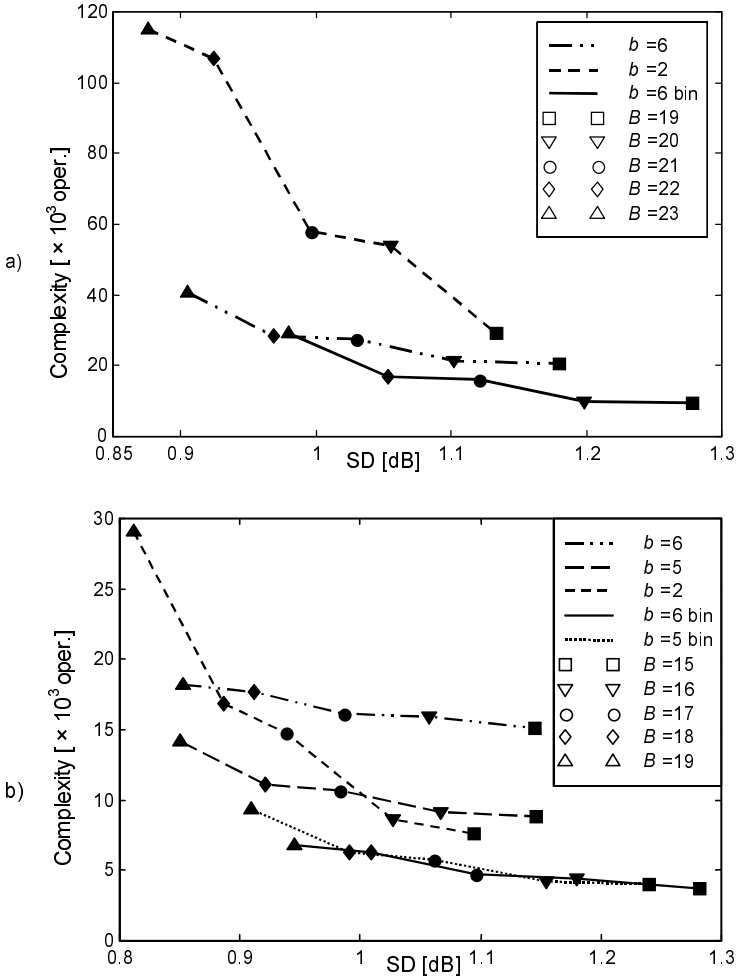

Figure 5. Overall coder complexity vs. average SD for several baseline and binary searched coders with a) *FR*=50 and b) *FR*=100

For both frame rates, baseline coders with only two switched predictors are shown as a reference on these graphs since they have minimum distortion for any given bit-rate [5] (dashed line). Having higher number of predictors (e.g. *N*=64) reduces complexity significantly at *FR*=50 (dash-dot line). However, the application of the binary searched prediction is not that effective (solid line), since the overall complexity at this frame rate is still dominated by the VQ. On the other hand, for higher frame rates the binary searched scheme successfully reduces the overall coder complexity for *b*=5 or 6. For example, the complexity of a binary searched coder with *b*=6 and SD=0.95 dB is more than two times lower than of the reference coder with *N*=2 having the same spectral distortion. This reduction in

complexity is paid by the increase of the LSF coding bit rate by two bits. Generally, the benefit of the binary-tree searched prediction increases with frame-rate as well as bit-rate (quality). Further overall complexity reduction is achievable by combining the proposed prediction technique with simpler VQ techniques (e.g. TSVQ) representing the basis for possible future work.

CONCLUSION

A new, computationally efficient switched-adaptive vector linear prediction scheme is proposed in this paper. It is based on a cascaded prediction with switched predictors organized in a binary-tree fashion. The binary-tree prediction scheme was integrated into an interframe LSF coder frequently used in speech coding, replacing the original full-searched SIVP switched predictor. The quality of the proposed coder was evaluated and compared to the baseline coder for several interesting bit-rates and number of switched predictors.

It was found that the sub-optimality increases with the number of prediction stages b. On the other hand, b should be higher since the prediction complexity reduces by a factor of $2b/2^b$. Since the overall complexity of the LSF coder depends on the VQ part as well it was found that the main benefit of the proposed algorithm is for applications with higher frame-rates and higher required quality. This type of coder can be applied in systems with stringent requirements on complexity, what is quite common in real-time communication and multimedia applications.

References

[1] Gersho A., Gray R. M., *Vector quantization and signal compression*, Kluwer Academic Publishers, 1992

[2] Zhou J., Shoham Y., Akanasu, A., Bunnell, H.T., Idsardi, W., "Simple fast vector quantization of the line spectral frequencies", *Proc. of Int. Conf. on Spoken Language Processing*, 1996, vol. 2, pp.945-948

[3] Chen J.H., Gersho A., "Covariance and autocorrelation methods for vector linear prediction", in *Proc. ICASSP*, 1987, pp.1545-1548

[4] Yong M., Davidson G., Gersho A., "Encoding of LPC spectral parameters using switched-adaptive interframe vector prediction", in *Proc. ICASSP*,1988, vol.1, pp. 402-405

[5] Wang S., Paksoy E., Gersho A., "Product code vector quantization of LPC parameters", in *Speech and Audio Coding for Wireless and Network Applications*, editors: B. S. Atal, V. Cuperman, A. Gersho, Kluwer Academic Publishers, 1993, pp. 251-258

[6] Gardner, W.R., Rao, B.D., "Theoretical analysis of the high-rate vector quantization of LPC parameters", *IEEE Transaction on Speech and Audio Processing*, vol. 3, no. 5, pp. 367-381, 1995