CHANNEL OPTIMIZED PREDICTIVE VQ

Jan Lindén†

Information Theory, Department of Signals and Systems,
Chalmers University of Technology, SE-412 96 Göteborg, Sweden
e-mail: linden@kane.ece.ucsb.edu

ABSTRACT

In this paper combined source-channel coding is considered for the case of predictive vector quantization. A design algorithm for channel optimized predictive vector quantizers is proposed. Under reasonable assumptions, the optimal encoder is presented and a sample iterative design method that simultaneously optimizes the predictor and the codebook is derived. We also demonstrate that this design method can be used to obtain index assignments that are advantageous to what is obtained by post process index assignment algorithms. Results are presented for a correlated Gauss-Markov process and for speech LSF parameters.

1. INTRODUCTION

Vector quantization (VQ) schemes that exploit interframe correlation have shown very promising results in many speech coding applications, e.g. quantization of the spectrum parameters [1, 2]. The most popular method is predictive VQ (PVQ) which is simply a vector generalization of scalar DPCM. It has however been argued that PVQ performance rapidly deteriorates when channel noise is introduced [2]. Recently, it has been shown that these problems can be circumvented, e.g. [1], and hence PVQ is an advantageous alternative to memoryless quantization also for noisy channels.

Noisy channel performance can be improved by finding an index assignment (IA) that minimizes the distance between codevectors with similar binary codewords [3]. Robustness against channel errors is thus obtained without using any explicit knowledge about the channel.

If some knowledge about the channel can be incorporated in the design, performance can be significantly improved. This is usually referred to as channel optimized VQ (COVQ) [3, 4]. Here we propose a new method for channel optimized predictive VQ (COPVQ) design. It is also demonstrated that the COPVQ design method can be used to obtain index assignments for PVQ that are advantageous to what is obtained by post process index assignment algorithms.

2. CHANNEL OPTIMIZED PVQ

The basic idea in COVQ design is to adopt a distortion measure that takes the channel characteristics into consideration. In this work we assume a discrete memoryless channel which can be described by its transition probabilities $p_{j_n|i_n}$. That is, the probability that index j_n is received given that index i_n was sent at time n.

† The author is currently with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, and Signalcom Inc., 7127 Hollister Ave. Suite 109, Goleta, CA 93117.

A predictive VQ scheme is depicted in Figure 1. The most common choice of predictor is a linear autoregressive predictor

$$\hat{\mathbf{x}}_n = \sum_{k=1}^{P} \mathbf{A}_k \tilde{\mathbf{x}}_n$$

where P is the predictor order and $\mathbf{A}_k$ are predictor matrices. We understand that the prediction vectors in the encoder and the decoder ($\hat{\mathbf{x}}_n$ and $\hat{\mathbf{x}}'_n$) are not necessarily equal if the channel is noisy. This fact must be compensated for in the design of the COPVQ encoder. Methods for improvement of predictors for noisy channels can be found in, e.g. [1, 5]. It is clear that the predictions, $\hat{\mathbf{x}}_n$ and $\hat{\mathbf{x}}'_n$, which determine the state of the encoder and decoder, are solely functions of the history of indices, $\mathbf{i}_0^{n-1} = \{i_k\}_{k=0}^{n-1}$ and $\mathbf{j}_0^{n-1} = \{j_k\}_{k=0}^{n-1}$, up to time $n-1$ and the initial states $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}'_0$. For simplicity, we assume these two initial states to be equal.

3. DISTORTION MEASURE FOR COPVQ

Assume that the input process is stationary with zero mean and that the expected value of the distortion at time n can be used as performance measure. Note that the indices, $\mathbf{i}_0^n$, of the encoder should be chosen all at the same time when the whole sequence to quantize is available in order to minimize the total distortion. This is of course not realistic. In this work, as most other work on PVQ, we will assume that the encoder must make a choice for the best index every frame and no extra delay is thus allowed. Therefore, we define the distortion measure to minimize as the expected value of the distortion, given the history of previous indices $\mathbf{i}_0^{n-1}$. Using this definition we obtain

$$\begin{aligned}\bar{D} &= E[d(\mathbf{X}_n, \tilde{\mathbf{X}}'_n)|\,\mathbf{i}_0^{n-1}] \\ &= \sum_{i_n}\sum_{\mathbf{j}_0^n} E[d(\mathbf{X}_n, \tilde{\mathbf{X}}'_n)|\,\mathbf{j}_0^n, \mathbf{i}_0^n]\, P(i_n, \mathbf{j}_0^n|\mathbf{i}_0^{n-1})\end{aligned}$$

where $d(\mathbf{X}_n, \tilde{\mathbf{X}}'_n)$ is an arbitrary distortion measure. The optimal coding system is now defined as the coder that minimizes $\bar{D}$. In [6] it is shown that if the channel is assumed to be a discrete memoryless channel the distortion becomes

$$\bar{D} = \sum_{i_n} \int_{R_{i_n|\mathbf{i}_0^{n-1}}} \left\{ \sum_{\mathbf{j}_0^n} P(\mathbf{j}_0^n|\mathbf{i}_0^n) d(\mathbf{x}_n, \tilde{\mathbf{x}}'_n) \right\} f_{\mathbf{X}|\mathbf{i}_0^{n-1}} d\mathbf{x}_n$$

where $R_{i_n|\mathbf{i}_0^{n-1}}$ is the encoding region for index i_n. We rewrite the expression in braces using the fact that the channel is memoryless

$$\alpha(i_n, \mathbf{x}_n|\mathbf{i}_0^{n-1}) = \sum_{\mathbf{j}_0^n} d(\mathbf{x}_n, \tilde{\mathbf{x}}'_n) \prod_{l=0}^{n} p_{j_l|i_l}$$

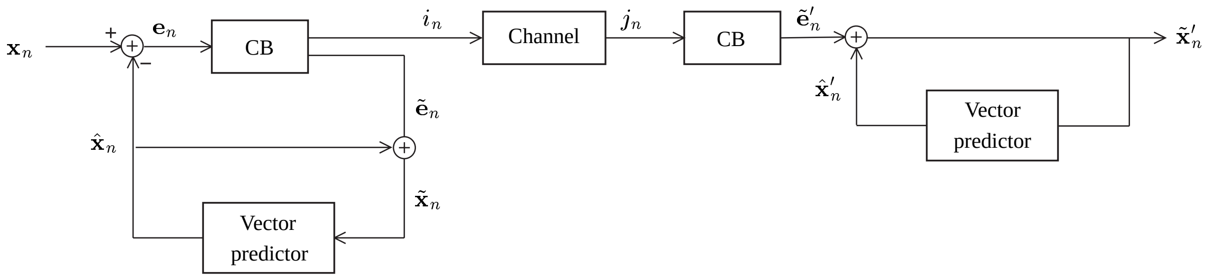

Figure 1: Predictive VQ operating over a noisy channel. A prediction error vector is formed by subtracting from the input vector a prediction of it. The prediction error vector is then quantized by a memoryless VQ.

which is the distortion measure that defines the encoder for a given codebook. Hence, all possible decoder states must be investigated and the probabilities of the decoder states given the encoder state must be known in order to calculate the distortion.

The number of operations involved in the calculation of the distortion is growing with time and becomes unmanageable after only a few samples. However, if the distortion measure that is employed is the weighted squared Euclidean measure

$$d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^t\mathbf{W}(\mathbf{x})(\mathbf{x}-\mathbf{y})$$

the distortion can be calculated by simply altering the encoder predictor [6]. The weighting matrix $\mathbf{W}(\mathbf{x})$ is assumed to be diagonal with non-negative components. Introduce

$$\tilde{\mathbf{x}}''_n = E[\tilde{\mathbf{X}}'_n \,|\, \mathbf{i}_0^{n-1}] = \sum_{k=1}^{P}\mathbf{A}_k\tilde{\mathbf{x}}''_{n-k} + \bar{\mathbf{c}}_{i_n}$$

and

$$\hat{\mathbf{x}}''_n = E[\hat{\mathbf{X}}'_n \,|\, \mathbf{i}_0^{n-1}] = \sum_{k=1}^{P}\mathbf{A}_k\tilde{\mathbf{x}}''_{n-k} = \tilde{\mathbf{x}}''_n - \bar{\mathbf{c}}_{i_n}$$

where $\bar{\mathbf{c}}_{i_n} = E[\mathbf{c}_{j_n} \,|\, i_n] = \sum_{j_n}\mathbf{c}_{j_n}p_{j_n|i_n}$. If the encoder predictor is replaced by $\hat{\mathbf{x}}_n = \hat{\mathbf{x}}''_n$ the input to the VQ is no longer $\mathbf{e}_n$ but rather $\hat{\mathbf{e}}''_n = \mathbf{x}_n - \tilde{\mathbf{x}}''_n$. It turns out that by replacing $\mathbf{e}_n$ by $\mathbf{e}''_n$ and use the standard COVQ distance measure [4] for the memoryless VQ is equivalent to employing the distortion measure derived for COPVQ. The distortion for this case is

$$\begin{aligned}&\alpha(i_n,\mathbf{x}_n|\mathbf{i}_0^{n-1}) = \alpha(i_n,\mathbf{e}''_n)\\&= \sum_{j_n}(\mathbf{e}''_n-\mathbf{c}_{j_n})^t\mathbf{W}(\mathbf{x}_n)(\mathbf{e}''_n-\mathbf{c}_{j_n})p_{j_n|i_n}\end{aligned}$$

Hence, we have now found a very simple way to implement the COPVQ distortion measure for the weighted squared Euclidean measure by altering the encoder predictor.

4. SAMPLE ITERATIVE TRAINING METHOD

Examples of training procedures for COVQ are [4] which is a method based on the generalized Lloyd algorithm (GLA) and [7] which is a sample iterative (stochastic gradient) procedure. Here we present a sample iterative method for simultaneous update of the predictor and the codebook for COPVQ. In [6] a block iterative training method for COPVQ is also presented. Due to the high complexity of the block iterative method we confine the discussion to the sample iterative method.

For PVQ it is customary to design the predictor first without taking quantization into consideration and then design the codebook for the given predictor. However, when the channel is noisy it is very important that the predictor is redesigned to prevent error propagation. Hence, it is natural to design the predictor for the noisy channel using the same strategy as for the codebook design. The basic idea is to, for each incoming vector, update the parameters in the direction of the negative gradient of the instantaneous distortion. In our case this implies that all code vectors and the predictor are updated for each training vector. In the following we derive formulas to update the codebook and the predictor for the case when the squared Euclidean distance measure is employed. The update of the codebook and the predictor can be performed simultaneously but of course it is also possible to update only the codebook for a given predictor and vice versa.

For each training sample, a search for the best vector (called winning vector) in the current prediction error codebook is performed in order to determine the instantaneous distortion. Denoting the index of the winning codevector for the n-th training vector ι_n, the generic formula for updating the codevectors $\mathbf{c}_k^{(n+1)}$ (codebook vector number $k = 1, 2, \ldots, M$) at time $n+1$ can be written

$$\mathbf{c}_k^{(n+1)} = \mathbf{c}_k^{(n)} - \mu_Q(n)\nabla_{\mathbf{c}_k^{(n)}}\alpha(\iota_n,\mathbf{x}_n|\boldsymbol{\iota}_0^{n-1})$$

where $\mu_Q(n)$ is an annealing function that is decreasing with time. The gradient of the distortion can now be calculated

$$\begin{aligned}&\nabla_{\mathbf{c}_k^{(n)}}\alpha(\iota_n,\mathbf{x}_n|\boldsymbol{\iota}_0^{n-1}) = \nabla_{\mathbf{c}_k^{(n)}}\alpha(\iota_n,\mathbf{e}''_n)\\&= \sum_{j_n}\nabla_{\mathbf{c}_k^{(n)}}(\mathbf{e}''_n-\mathbf{c}_{j_n})^t\mathbf{W}(\mathbf{x}_n)(\mathbf{e}''_n-\mathbf{c}_{j_n})p_{j_n|\iota_n}\\&= -2p_{k|\iota_n}\mathbf{W}(\mathbf{x}_n)(\mathbf{e}''_n-\mathbf{c}_k^{(n)})\end{aligned}$$

since we have assumed $\mathbf{W}(\mathbf{x}_n)$ to be symmetric. Hence, the equation for updating the codebook becomes

$$\mathbf{c}_k^{(n+1)} = \mathbf{c}_k^{(n)} + 2\mu_Q(n)P_{k|\iota_n}\mathbf{W}(\mathbf{x}_n)(\mathbf{e}''_n-\mathbf{c}_k^{(n)})$$

It is clear that codevectors that are far from the winning vector in the code space, i.e. with low transition probability, are updated much less than those that are close.

The derivation of the predictor update is a little more complicated than the codebook update. When the simplified distortion measure presented in Section 3 was derived a term was disregarded that contains the current predictor since it does not affect the choice of current codevector. When differentiating the distortion with respect to the predictor this term must be included. Thus, we start with the original distortion measure rather than the simplified one.

The gradient of $\alpha(\iota_n, \mathbf{x}_n|\boldsymbol{\iota}_0^{n-1})$ with respect to the k-th predictor matrix is

$$\begin{aligned}
&\nabla_{\mathbf{A}_k^{(n)}} \alpha(\iota_n, \mathbf{x}_n|\boldsymbol{\iota}_0^{n-1}) \\
&= \nabla_{\mathbf{A}_k^{(n)}} \sum_{\mathbf{j}_0^n} (\mathbf{x}_n - \tilde{\mathbf{x}}_n')^t \mathbf{W}(\mathbf{x}_n)(\mathbf{x}_n - \tilde{\mathbf{x}}_n') \prod_{l=0}^{n} p_{j_l|\iota_l} \\
&= \sum_{\mathbf{j}_0^n} \prod_{l=0}^{n} p_{j_l|\iota_l} \nabla_{\mathbf{A}_k^{(n)}} (\mathbf{x}_n - \tilde{\mathbf{x}}_n')^t \mathbf{W}(\mathbf{x}_n)(\mathbf{x}_n - \tilde{\mathbf{x}}_n') \\
&= -2\mathbf{W}(\mathbf{x}_n) \sum_{\mathbf{j}_0^n} \prod_{l=0}^{n} p_{j_l|\iota_l} (\mathbf{x}_n - \tilde{\mathbf{x}}_n')(\tilde{\mathbf{x}}_{n-k}')^t
\end{aligned}$$

We note that this is the difference (i.e. error) between straightforward estimates of the correlation between the current input and the previous outputs and the current output and previous outputs. Finally, we write

$$\begin{aligned}
&\nabla_{\mathbf{A}_k^{(n)}} \alpha(\iota_n, \mathbf{x}_n|\boldsymbol{\iota}_0^{n-1}) \\
&= -2\mathbf{W}(\mathbf{x}_n)\left(\mathbf{x}_n(\tilde{\mathbf{x}}_{n-k}'')^t - \sum_{\mathbf{j}_0^n} \tilde{\mathbf{x}}_n'(\tilde{\mathbf{x}}_{n-k}')^t \prod_{l=0}^{n} p_{j_l|\iota_l}\right)
\end{aligned}$$

This expression is very difficult to calculate for other predictor orders than one [6]. However, we have in our simulations seen that by simply replacing the last term by $\tilde{\mathbf{x}}_n''(\tilde{\mathbf{x}}_{n-k}'')^t$ good COPVQs are obtained. The resulting update formula for the predictors can now be simplified to

$$\mathbf{A}_k^{(n+1)} = \mathbf{A}_k^{(n)} + 2\mu_P(n)\mathbf{W}(\mathbf{x}_n)(\mathbf{e}_n'' - \bar{\mathbf{c}}_{\iota_n})(\tilde{\mathbf{x}}_{n-k}'')^t$$

In order to ensure stability throughout the training and convergence of the algorithm, $\mu_P(n)$ should not be a scalar function but rather a matrix function. For simplicity we have used scalar linearly decreasing functions for both step sizes, $\mu_Q(n)$ and $\mu_P(n)$.

5. SIMULATIONS

5.1. Gauss-Markov Source

In this section we briefly investigate performance of COPVQ for a blocked scalar process. Such a process arise when a scalar valued random sequence is partitioned into blocks of d samples, each block defining a d-dimensional vector. We examine a scalar Gauss-Markov source with correlation coefficient $\rho = 0.9$. The training sequence consists of $1\,000\,000$ vectors and the evaluation sequence of $200\,000$ vectors from another realization. The results for COPVQs with rate=1 bit/sample are shown in Table 1 and can be compared with the results of memoryless COVQ for the same source in Table 2. All quantizers are designed for the actual channel bit error rate. The COPVQ clearly outperforms the memoryless COVQ for all error rates.

We have also compared the results obtained here with another memory based COVQ method. In [8], methods for noisy channel optimized finite-state VQ (FSVQ) are proposed. The results from these two investigations can be compared for the case when the dimension equals 4. The results presented here are 0.3-0.7 dB better with the largest difference for small error probabilities.

5.2. LSF Parameters

In this section we investigate the performance of COPVQ for a vector process. We have designed COPVQs for quantization of

Table 1: COPVQ performance for Gauss-Markov process with $\rho = 0.9$. First order predictors.

BER	dimension d				
[%]	2	3	4	5	6
0.0	11.12	11.52	11.72	11.86	12.00
0.1	10.89	11.30	11.45	11.56	11.57
1	9.44	9.79	9.99	10.08	10.21
2	8.48	8.85	9.05	9.13	9.50
5	6.93	7.25	7.53	7.71	7.82
10	5.14	5.58	5.95	5.96	5.98

Table 2: COVQ performance for Gauss-Markov process with $\rho = 0.9$. Rate=1 bit/sample.

BER	dimension d				
[%]	2	3	4	5	6
0.0	7.92	9.39	10.22	10.68	11.01
0.1	7.80	9.21	9.99	10.42	10.71
1	6.85	8.05	8.66	9.04	9.39
2	6.12	7.16	7.71	8.07	8.65
5	4.71	5.59	6.00	6.66	6.96
10	3.35	3.93	4.51	4.94	5.14

line spectrum frequencies (LSF), which is one of the major applications for VQ in speech coding.

The training database consists of 86 minutes of speech and the evaluation database has a length of 7 minutes. Three-split VQ are used for all quantizers. A description of the databases and experimental setup can be found in [1]. We have used first order predictors (P=1) since the gain of using higher orders for this application is negligible [1].

Table 3: Performance of 21 bit COPVQs. In the first case the predictor is optimized for noisefree performance and only the codebooks are trained and in the second case predictor and codebooks are trained simultaneously.

BER	Only CB training			Sim. training		
[%]	$\overline{\text{SD}}$ $[dB]$	2-4 dB [%]	> 4 dB [%]	$\overline{\text{SD}}$ $[dB]$	2-4 dB [%]	> 4 dB [%]
0	1.02	2.8	0	1.02	2.0	0
0.1	1.14	7.1	0.6	1.07	4.0	0.3
0.5	1.46	16	1.9	1.27	9.4	0.7
1	1.75	28	2.8	1.42	16	0.9
2	2.10	40	5.4	1.64	24	1.4
5	2.71	55	13	2.12	42	4.3
10	3.47	58	29	2.77	60	13

The first experiment is conducted to investigate the importance of optimizing the predictor for a certain channel. In Table 3 average spectral distortion (SD) as well as outlier measures are presented for 21 bit COPVQs. A significant performance improvement is obtained by also optimizing the predictor for the noisy channel while no difference is visible for noisefree channel.

In our second experiment we compare COPVQ performance for LSF quantization with traditional memoryless COVQ. The results in Figure 2 indicate that a gain of 4 bits is achieved for this application. For high error rates the performance of the 21 bit COPVQ is actually comparable to a 26 bit COVQ which indicates a gain of 5 bits. We see that by designing a PVQ scheme for channel errors, significant performance gains are achievable compared to memoryless VQs. The gain over memoryless COVQ for the

FSVQ schemes in [8] was in the order of 2-3 bits. Hence, the FSVQ scheme that is more complex and requires more storage capabilities is outperformed by the COPVQ scheme proposed here.

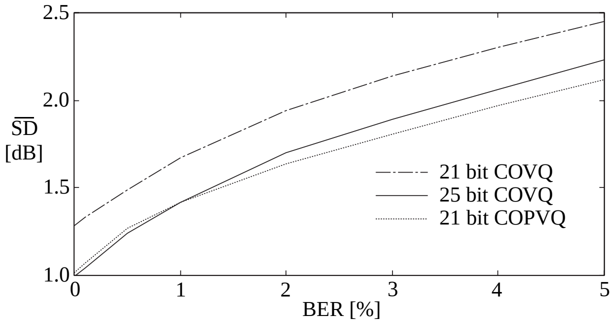

Figure 2: Average SD for 21 bit COPVQ compared with 21 bit and 25 bit memoryless COVQs.

To compare index assignments obtained with a post process IA algorithm and by the COPVQ design algorithm we have performed the following experiment: The IA algorithm presented in [9] has been applied to a PVQ that was designed for a noisefree channel. Two cases were considered, one in which the predictor was not scaled and one in which the predictor was scaled such that a small degradation of 0.04 dB was allowed for noisefree channel. The design error rates for the COPVQ designs were chosen such that the same two conditions for noisefree performance were met while trying to improve noisy channel performance. Note that in this experiment, the same quantizers are used for all error probabilities regardless of design error probability. In Figure 3, average SD for these four different designs are compared. Clearly, much more channel robust PVQs are obtained by the COPVQ design method compared to post process index assignment.

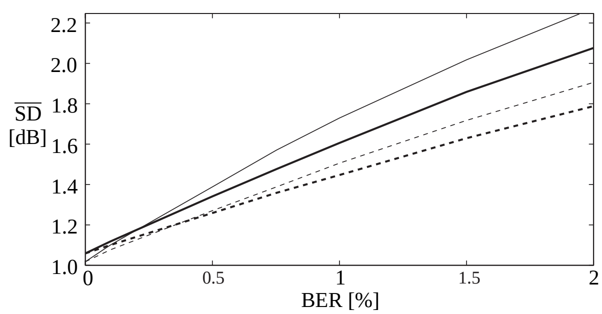

Figure 3: Average SD for 21 bit PVQs with different index assignments. The solid lines correspond to post process IA and the dashed lines to IA obtained by COPVQ design. Two cases are considered: no degradation of noisefree performance allowed (thin lines) and a small degradation of noisefree performance is allowed to improve noisy channel performance (thick lines).

5.3. Subjective Evaluation

We have demonstrated using objective measures that the performance of a 21 bit COPVQ is comparable with that of a 25 bit memoryless COVQ. In the present section we compare these two quantizers in a simple listening experiment.

Synthetic speech was produced for each of the quantizers using the following procedure: A prediction residual was formed by inverse filtering the speech signal using an unquantized prediction filter. Synthetic speech was then generated by exciting the quantized production filter with the prediction error signal from the unquantized inverse filter. The experiment was carried out for a bit error probability of 2%.

The speech signal was obtained as a concatenation of material spoken by four speakers, two male and two female, each reading continuous text for one minute. The listener could choose which of the two coded versions to listen to interactively throughout the experiment. The task for the five listeners was then to state which of the two versions that was preferred.

All five listeners voted the 21 bit COPVQ as the winner. They were all confident of having made a correct choice and had a clear preference for this coder. Furthermore, they all stated that the difference was most prominent for the male speakers.

6. SUMMARY

We have in this work presented an efficient sample iterative design algorithm for channel optimized PVQ. Performance was investigated for a blocked scalar process as well as a true vector process. In both cases, COPVQ clearly outperformed memoryless COVQ. It was also found that channel optimized finite-state VQ was also outperformed by the proposed COPVQ. The validity of the results was strengthened by an informal listening test.

7. REFERENCES

[1] J. Skoglund and J. Lindén, "Predictive VQ for noisy channel spectrum coding: AR or MA?," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, (Munich, Germany), pp. 1351–1354, 1997.

[2] H. Ohmuro, T. Moriya, K. Mano, and S. Miki, "Vector quantization of LSP parameters using moving average interframe prediction," *Electronics and Communications in Japan, Part 3*, vol. 77, pp. 12–26, 1994.

[3] P. Hedelin, P. Knagenhjelm, and M. Skoglund, "Theory for transmission of vector quantization data," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), pp. 347–396, Elsevier Science, 1995.

[4] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 799–809, 1990.

[5] J. Lindén and J. Skoglund, "Channel optimization of predictive VQ for spectrum coding," in *Proc. IEEE Workshop on Speech Coding for Telecommunications*, (Pocono Manor, USA), pp. 93–94, 1997.

[6] J. Lindén, *Interframe quantization for noisy channels*. Ph.D. dissertation, Chalmers University of Technology, 1998.

[7] P. Knagenhjelm, "A recursive design method for robust vector quantization," in *Proc. International Conference on Signal Processing Applications and Technology*, (Boston, USA), pp. 948–954, 1992.

[8] Y. Hussain, *Design and performance evaluation of a class of finite-state vector quantizers*. PhD dissertation, University of Maryland, College Park, 1992.

[9] P. Knagenhjelm and E. Agrell, "The Hadamard transform - a tool for index assignment," *IEEE Transactions on Information Theory*, vol. 42, no. 4, pp. 1139–1151, 1996.