ISSN 2319 - 6629

Volume 3, No.2, February – March 2014

International Journal of Wireless Communications and Networking Technologies

Available Online at http://warse.org/pdfs/2014/ijwcnt02322014.pdf

Combined Source Adaptive and Channel Optimized Matrix Quantization for Noisy Channels

Dr. Vasilios Bozantzis[1], Dr. Panos Philippopoulos[2]

[1]MIET, TEI of Peloponnesus (ex-TEI of Kalamata), Sparti, Greece, vbozantzis@teikal.gr

[2]Assistant Professor, TEI of Peloponnesus (ex-TEI of Kalamata), Sparti, Greece, pfilip@teikal.gr

ABSTRACT

Considering that most signal sources of practical interest are non-stationary, this paper introduces a technique which adapts Matrix Quantization to varying source statistics and optimizes it for noisy channels, thus designs a matrix quantizer/decoder that considers both non-stationary source and noisy channel statistics. Matrix Quantization (MQ) has already been successfully applied for speech signals and noiseless channels. MQ has also been shown in relevant literature to outperform well established Vector Quantization (VQ) when applied over noisy channels. The proposed algorithm, Combined Source Adaptive and Channel Optimized Matrix Quantization (CSACOMQ) is formally described and then evaluated for a source modelled as the non-stationary Wiener process and over both the memoryless Binary Symmetric Channel (BSC) and the Flat Fading Rayleigh Channel (FFRC). It is shown that CSACOMQ offers substantial Signal-to-Noise Ratio (SNR) performance improvement compared to Channel Optimized Matrix Quantization (COMQ), at the expense of minimal off-line additional computation complexity.

Key words: Adaptive Source Coding, Combined Source-Channel Coding, Matrix Quantization, Vector Quantization, Rayleigh Fading.

1. INTRODUCTION

VQ and Adaptive Vector Quantization (AVQ) [1] have been extensively studied and employed in practical applications, over the past decade. Applications of VQ and AVQ include modern audio codecs, such as the AMR-WB+ standard for 3G mobile audio services [2], the Broadvoice speech codec [3], the CELT speech and audio codec [4] and the OPUS interactive voice/audio codec [5], [6]. AVQ has been applied, among others, in video compression [7], where the compression scheme is based on adaptive vector quantization of multi-wavelet coefficients, and in image compression [8], where a fuzzy self-adaptive particle swarm optimization algorithm is described extracting a near-optimum VQ codebook. In [9], an AVQ scheme for speech coding is presented, where the VQ of the Line Spectral Frequency (LSF) parameters of speech is optimized for the probability density function (p.d.f.) of the LSF parameters using a mixture of Dirichlet distributions. In [10] a reordering VQ algorithm for temporally structured sources is presented, which outperforms mainstream VQ algorithms, when tested for Markov and speech sources.

In [11], [12] Bozantzis et al. presented the Combined Source Adaptive and Channel Optimized Vector Quantization (CSACOVQ) algorithm, which jointly adapts VQ to changing source statistics and optimizes it for the BSC and the FFRC respectively. In [13] Bozantzis et al. produce a novel and efficient approach on the Index Assignment (IA) scheme design for VQ. Finally, in [14], Bozantzis et al. present the joint VQ/IA design, which in a combined manner, is adapted to changing source statistics and is optimized for the noisy channels considered in [14].

Additionally to the well-known VQ, MQ has attracted research interest and is shown to be a very promising source coding technique, compared to VQ. In [15] it is shown that MQ can exploit better than VQ the correlation properties between consecutive speech frames with a multi-frame structure. In [16] the COMQ technique is presented, applied for the coding of Line Spectral Pair (LSP) parameters transmitted over noisy channels and is shown to outperform Channel Optimized Vector Quantization (COVQ) [17].

In this paper initially an Adaptive Matrix Quantization (AMQ) technique is presented, which, in accordance to the concept of AVQ, adapts the matrix quantizer/decoder to varying source statistics as the coding process progresses. The technique is based on the move-to-front generalized threshold replenishment (GTR) algorithm for VQ [18]. This AMQ technique alters the matrix quantizer/decoder codebooks for every transmitted matrix, according to a decision met at the transmitter side.

In the sequel, the CSACOMQ algorithm is introduced, which combines the AMQ technique with the COMQ algorithm. CSACOMQ jointly adapts a matrix quantizer/decoder to changing source statistics and optimizes it for a noisy channel, thus acting as a form of combined source-channel coding.

In Section 2 the move-to-front GTR algorithm for MQ is introduced and described providing the principles and preliminaries with the main system components and functionalities. Subsequently, in Section 3, the CSACOMQ algorithm for the BSC is presented, which jointly designs a matrix quantizer/decoder for changing source statistics and the BSC. In Section 4, the CSACOMQ algorithm for the FFRC is presented, which jointly designs a matrix quantizer/decoder for changing source statistics and the FFRC. Finally, in Section 5, simulation results are presented and discussed, and conclusions are drawn in Section 6.

2. ADAPTIVE MATRIX QUANTIZATION

The design objective of the move-to-front GTR algorithm for MQ, is matrix quantizer/decoder adaptation to varying source statistics, best described with the aid of block diagrams. Matrix encoder/decoder pair is depicted in Fig.1, while vector/matrix encoder structure is depicted in Fig.2.

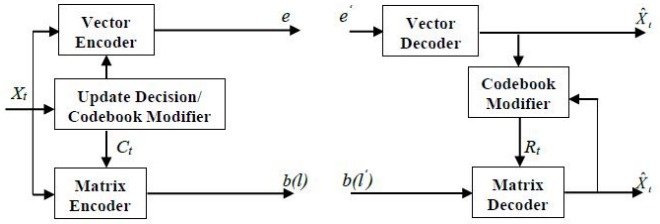

Figure 1. Structure of the Matrix Encoder/Decoder

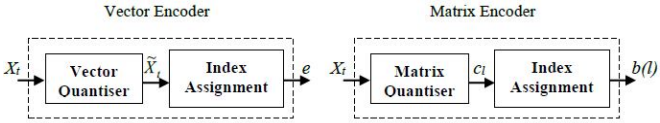

Figure 2. Vector Encoder and Matrix Encoder

At each time instance *t*, only one of the vector encoder/decoder or matrix encoder/decoder lines is active, according to a decision made by the Update Decision/Codebook Modifier block. It should be noted here that the main transmission line is the matrix encoder/decoder pair, whereas the vector encoder/decoder pair serves as an auxiliary line that is activated only in the case of a Codebook Update. More specifically, the source is a non-stationary, continuous amplitude and discrete-time process. The input matrix $X_t \in \Re^{N\times M}$, at time instance *t*, $X_t = (x_{t1},...,x_{tM})$, $x_{tm} \in \Re^{N}$, $m=1,...,M$, contains $N\times M$ successive samples originating from this process and is the input to the vector/matrix encoder and Update Decision/Codebook Modifier. The functionality of each block is described in the sequel.

Vector encoder: The vector encoder consists of a vector quantizer with codebook $V \in \Re^{N\times K}$, $V=(v_1,...,v_K)$, $v_g \in \Re^{N}$, $g=1,2,...,K$ and an index assignment scheme. The vector quantizer outputs, for each column vector x_{tm} of X_t, the codevector $\tilde{x}_{tm} \in V$ according to the following distortion-based only quantization rule:

$$\tilde{x}_{tm} = \arg\min_{v_g \in V} d(x_{tm}, v_g) \quad (1)$$

In the above formula $d(x_m,v_g)=\|x_m-v_g\|^2$ is the squared Euclidean distance between vectors x_m, $v_g \in \Re^{N}$. Therefore, the total output of the vector quantizer for input matrix X_t is matrix $\tilde{X}_t = (\tilde{x}_{t1},..., \tilde{x}_{tM})$. The index assignment scheme assigns to its input $\tilde{x}_{tm}$ the binary representation $b(e_m)$ of the index e_m of $\tilde{x}_{tm}$ in codebook *V*. As a result, the total produced binary sequence at time instance *t* is $e=b(e_1)b(e_2)...b(e_M)$. The index assignment mapping rule is given by:

$$b(e_m) = (e_m\text{-}1)_2 \quad (2)$$

Vector decoder: The decoder receives the binary sequence $e' = b(e_1')b(e_2')...b(e_M')$. It maps each received $b(e_m')$ to the e_m'-*th* element of *V*, $v_{e_m'} \in V$, based on the mapping rule (2). Thus decoder output at time *t* is matrix $\hat{X}_t = (v_{e_1'},..., v_{e_M'})$.

Matrix encoder: The codebook of the matrix quantizer is $C_t = \{c_1,c_2,...,c_L\}$, $C_t \in \Re^{N\times M\times L}$, where *L* is the number of the element codematrices $c_i \in \Re^{N\times M}$, $i=1,...,L$. At time instance *t*, it quantizes X_t into codematrix $c_l \in C_t$ according to the following rate-distortion quantization rule:

$$c_l = \arg\min_{c_i \in C_{t-1}} [D(X_t,c_i)+\lambda\cdot(-\log_2 p_{t-1}(i))],\ 1\le i \le L \quad (3)$$

The distortion measure $D(X,Y)$ between any two matrices $X=(x_1,...,x_M)$ and $Y=(y_1,...,y_M)$, $X,Y \in \Re^{N\times M}$, used also in the generalized Linde-Buzo-Gray (LBG) algorithm [1], is defined as:

$$D(X,Y) = \frac{1}{M}\sum_{m=1}^{M} d(x_m, y_m) \quad (4)$$

where $d(x_m,y_m)=\|x_m-y_m\|^2$ is the squared Euclidean distance between vectors $x_m, y_m \in \Re^{N}$. The weighing factor λ decides which of the components (rate or distortion), is important to the designer and $p_{t-1}(i)$ is the probability of occurrence of codematrix c_i at the output of the matrix quantizer at time instance *t-1*. These probabilities are unequal and most importantly time-variant; it is thus necessary to include the rate term in the quantization rule in (3). The index assignment scheme assigns to its input, $c_l \in C_t$, the binary representation $b(l)$ of its index *l*, $1\le l \le L$:

$$b(l) = (l\text{-}1)_2 \quad (5)$$

***Matrix decoder*:** The codebook of the matrix decoder is denoted as $R_t = \{r_1,r_2,...,r_L\}$, $R_t \in \Re^{N\times M\times L}$, where *L* is the number of the element reconstruction matrices $r_j \in \Re^{N\times M}$, $j=1,..L$. The matrix decoder follows the mapping rule of (5) and operates in exactly the reverse way as the index

assignment scheme, mapping the received binary sequence $b(l')$ to the reproduction matrix $r_{l'}$ of R_t. Therefore the decoder output at time instance t is matrix $\hat{X}_t = r_{l'}$.

Update Decision/Codebook Modifier (transmitter side): This is the key element that decides if a codebook update is necessary, according to a rate-decision criterion described in the sequel. If a codebook update is decided at time t, then the secondary transmission line (vector encoder/decoder pair) is activated, and the matrix quantizer codebook C_{t-1} is updated. Otherwise, the main transmission line (matrix encoder/decoder pair) is activated, and the matrix quantizer codebook C_{t-1} is only restructured as described in the following algorithm. In both cases, the block feeds the matrix quantizer at time instance t with its new codebook C_t.

Codebook Modifier (receiver side): The codebook modifier at the receiver side, alters the codebook of the matrix decoder according to the decision met at the transmitter side. If a codebook update is decided there at time instance t, then codebook R_{t-1} is accordingly updated, otherwise it is simply reconstructed, as described in the algorithm. In both cases, the block feeds the matrix decoder at time instance t with its new codebook R_t. It should be noted, that a specific binary codeword, or flag, is normally required to alert the receiver, each time a codebook update has occurred at the transmitter side. Since this flag is sent only when a codebook update occurs, its transmission does not cumber significantly the overall transmission rate.

The critical parameters of the move-to-front GTR algorithm for MQ are the already defined rate-distortion parameter λ and the windowing parameter ω, defined as the number of time instances (or codebook update decisions) prior to the current time instance t, considered in the calculation of the codematrices probabilities $p_t(i)$. The algorithm consists of the following steps:

<u>*Step 1:*</u> The initial time instance $t=1$ and initial codebooks C_0 and R_0 are set.
<u>*Step 2a:*</u> The input matrix X_t is quantized to matrix c_l of the codebook C_{t-1} according to the quantization rule described in (3), where the time subscript states that the codebook was formed by the codebook modifier during the previous time instance $t-1$. This metric minimisation includes the distortion measure $D(X_t,c_i)$ and the probability of occurrence $p_{t-1}(i)$ of the codematrix c_i in the codebook C_{t-1}.
<u>*Step 2b:*</u> The decision criterion ΔJ is calculated, which incorporates the distortion reduction $\Delta D=(X_t, \tilde{X}_t)-D(X_t ,c_l)$ and the rate increase $\Delta R=M\cdot log_2K$ bits per input matrix resulting from a codebook update. Thus the advantages and disadvantages of both decisions are merged into a criterion in order t o conclude which is the best option:

$$\Delta J = D(X_t, \tilde{X}_t) - D(X_t ,c_l)+\lambda\cdot M\cdot log_2K \quad (6)$$

If $\Delta J<0$ algorithm proceeds to Step 2c, otherwise Step 2d.
<u>*Step 2c (Codebook Update):*</u> The input matrix X_t replaces the last codematrix c_L of the codebook C_{t-1} and all elements of C_{t-1} are shifted one position to the right, such that X_t becomes the first element in the row, thus forming C_t. Binary sequence e is transmitted through the channel and $\hat{X}_t =(v_{e_1'},...,v_{e_M'})$ is the output. Matrix $\hat{X}_t$ replaces the last codematrix r_L of the codebook R_{t-1}, while all elements of R_{t-1} are shifted one place to the right such that $\hat{X}_t$ becomes the first element in the row, thus forming R_t. The codebook modifier at the transmitter side 'splits' the probability of codematrix c_l between c_l and X_t (or otherwise c_L, since X_t substitutes c_L). Therefore the number of times each codematrix is selected at the output of the matrix quantizer at time instance t is:

$$n_t(c_i) = \begin{cases} \omega \cdot p_{t-1}(c_i), \text{if } c_i \neq c_l, c_L \\ \omega \cdot p_{t-1}(c_i)/2, \text{if } c_i = c_l, c_L \end{cases} \quad (7)$$

The new codematrix probabilities $p_t(i)$, $1\leq i \leq L$ are given by:

$$p_t(c_i) = \frac{n_t(c_i)}{\sum_{j=1}^{L} n_t(c_j)} \quad (8)$$

After completion, the algorithm proceeds to Step 2e.
<u>*Step 2d (No Codebook Update):*</u> Codematrix c_l is moved to the front of C_{t-1}, index l is transmitted through the noisy channel, and reproduction matrix $r_{l'}$ is moved to the front of R_{t-1}. The new codematrix probabilities $p_t(i)$, $1\leq i \leq L$ at time instance t are then given by:

$$p_t(c_i) = \begin{cases} \omega \cdot p_{t-1}(c_i)/(\omega+1), \text{if } c_i \neq c_l \\ (\omega \cdot p_{t-1}(c_i)+1)/(\omega+1), \text{if } c_i = c_l \end{cases} \quad (9)$$

After completion, the algorithm proceeds to Step 2e.
<u>*Step 2e:*</u> It is set $C_t=C_{t-1}$, $R_t=R_{t-1}$. If $t<I$ (I is the length of training sequence) then $t=t+1$ and the algorithm returns to Step 2a.

The main difference of the concept of AMQ from the concept of the AVQ algorithm [18] is that the indices produced by the matrix encoder have a fixed length because the transmission channel is assumed to be fixed-rate.

3. CSACOMQ ALGORITHM FOR THE BSC

The CSACOMQ algorithm combines the AMQ algorithm presented in Section 2 with the COMQ algorithm [16]. The following steps provide its definition:

<u>*Step 1:*</u> The stopping threshold of the algorithm ε, and the initial codebooks $C^{(0)}$ and $R^{(0)}$ are selected. An initial value $D^{(0)}$ for the overall mean square error (MSE) between the source-input matrix and the sink-output matrix is set. The initial set $\{p_0(i), 1\leq i \leq L\}$ of probabilities of occurrence of codematrices $c_i \in C^{(0)}$, the rate-distortion parameter λ, and the

windowing parameter ω are also selected. The parameter λ is selected according to the weighing of rate, or distortion in the algorithm, and ω is selected according to the adaptation requirement of the matrix encoder/decoder. The iteration parameter k is set $k=1$.

Step 2: Initial codebooks of the move-to-front GTR algorithm are set $C_0=C^{(k-1)}$, and $R_0=R^{(k-1)}$. The time is set $t=1$.

Step 3: The *Steps 2a-2e* of the move-to-front GTR algorithm of Section 2 are executed and generate codebooks C_t and R_t.

Step 4: The iteration parameter is set $n=1$ and initial codebooks are set $C^{(0)}= C_t$ and $R^{(0)}= R_t$.

Step 4a: From the set $R^{(n-1)}=\{r_1^{(n-1)}, r_2^{(n-1)},..., r_L^{(n-1)}\}$ the optimal partition $\{S_i^{(n)}, 1\le i \le L\}$ is calculated according to the following partition rule:

$$S_i^{(n)} = \{X_t : \sum_{j=1}^{L} p_{j/i} \cdot D(X_t, r_j^{(n-1)}) \le \sum_{j=1}^{L} p_{j/u} \cdot D(X_t, r_j^{(n-1)}),\ u=1,2,..,L\} \quad (10)$$

The channel transition probability $p_{j/i}$ is the *a-posteriori* probability that index j is received when index i is transmitted:

$$p_{j/i} = (crossp)^x (1-crossp)^{\log_2 L - x} \quad (11)$$

where x is the Hamming distance between the binary representations *b(i), b(j)* of indices i and j respectively and *crossp* is the crossover probability of the BSC. The partitioning calculated in (10), minimises the distortion measure between source input matrix X_t and sink output matrix $\hat{X}_t$.

Step 4b: The optimal set of reconstruction matrices $R^{(n)}$ is calculated according to:

$$r_j^{(n)} = \frac{\sum_{i=1}^{L} p_{j/i} \cdot E(X_t \mid X_t \in S_i^{(n)})}{\sum_{i=1}^{L} p_{j/i} \cdot P_i^{(n)}},\ j=1,\ldots,L \quad (12)$$

where $P_i^{(n)}$ is the probability that input matrix X_t is an element of the partition $S_i^{(n)}$.

Step 4c: The overall MSE is computed:

$$D^{(n)} = \sum_{i=1}^{L} \sum_{j=1}^{L} p_{j/i} \cdot E[D(X_t, r_j^{(n)}) \mid X_t \in S_i^{(n)}] \quad (13)$$

If $(D^{(n-1)}-D^{(n)})/D^{(n)}<\varepsilon$ then,

(i) the codebook $R^{(k)}$ is set $R^{(k)}=R^{(n)}$,

(ii) the codebook $C^{(k)}$ is set as the current set of centroids of $S_i^{(n)}$, $c_i^{(k)}=E(X_t|X_t \in S_i^{(n)})$, $i=1,...L$,

(iii) the overall MSE $D^{(k)}$ is set $D^{(k)}=D^{(n)}$,

and the algorithm proceeds to Step 5.

Otherwise the iteration parameter n is set $n=n+1$ and the algorithm returns to Step 4a.

Step 5: If $(D^{(k-1)}-D^{(k)})/D^{(k)} <\varepsilon$, then the algorithm has reached a solution and stops. Otherwise iteration parameter k is set $k=k+1$ and the algorithm returns to the Step 2.

4. CSACOMQ ALGORITHM FOR THE FFRC

The objective here is the joint adaptation of the matrix quantizer/decoder for changing source statistics and the FFRC. The FFRC is modelled by a Markov finite-state model with Q states, as described and explained in [12], [19]. The modelling idea is the calculation of (i) the probability p_q, $q=1,2,\ldots,Q$, of occurrence of the q-th state and (ii) the crossover probability $(crossp)_q$, $q=1,2,\ldots,Q$, of the q-th state based on a set of received signal SNR thresholds A_q, $q=1,2,\ldots,Q-1$ that satisfy the following order:

$$0= A_0<A_1< A_2<\ldots< A_{Q-1}< A_Q= \infty \quad (14)$$

The SNR of the received signal is measured at the receiver side and the channel is considered to be at the q-th state when:

$$A_{q-1}<SNR< A_q \quad (15)$$

The channel transition probability $p_{j,q/i}$ of receiving index j when i was transmitted and the channel is in the q-state is:

$$p_{j,q/i} = (crossp)_q^x [1-(crossp)_q]^{\log_2 L - x} \quad (16)$$

where x is the Hamming distance between the binary representations *b(i), b(j)* of indices i and j respectively.

The matrix decoder codebook R_t here is of dimensions NxMx$(L\cdot Q)$, because it contains Q subcodebooks, each for every one of the Q states of the Markov model. Therefore the matrix decoder codebook is structured as $R_t =\{R_{t,1}, R_{t,2} ,\ldots, R_{t,Q}\}$, where $R_{t,q}$, $q=1,2,\ldots,Q$, is the matrix decoder codebook for the q-th state of the Markov model. The latter is denoted as $R_{t,q} =\{r_{1,q} , r_{2,q} ,\ldots, r_{L,q}\}$.

For the case of the FFRC, *Step 2c (Codebook Update)* and *Step 2d (No Codebook Update)* of the AMQ algorithm presented in Section 2 are modified as follows:

Modification of Step 2c (Codebook Update): The binary sequence e is transmitted through all the Q states of the Markov model and the matrix decoder receives for the q-th state of the Markov model the binary sequence e'_q. The resulting reconstruction of the input for the q-th state of the Markov model $\hat{X}_{t,q}$ is inserted in the front of $R_{t-1,q}$ and the last element of $R_{t-1,q}$ is deleted.

Modification of Step 2d (No Codebook Update): The binary sequence l is passed through the Q states of the Markov model and sequence l'_q is received by the decoder for the q-th state of the model. The l'_q-th reconstruction vector of the $R_{t-1,q}$ is moved to the front of $R_{t-1,q}$.

Before the CSACOMQ algorithm for the FFRC is presented, the COMQ algorithm for the FFRC modelled by the Markov model is presented:

Step 1. Determine the initial values of A_q, $q=1,2,\ldots,Q-1$, a stopping threshold ε, and let the initial overall distortion to be $D^{(0)}= \infty$. Choose initial matrix decoder codebook $R^{(0)}$. Set $m=1$.

Step 2. Calculate probabilities probability p_q and $(crossp)_q$, using equations (1)-(4) of [12] for the current values of A_q.
Step 3. From the set $R^{(m-1)}$ define the optimal partition $\{S_i^{(m)}, 1 \le i \le L\}$ according to:

$$S_i^{(m)} = \{X_t : \sum_{q=1}^{Q}\sum_{j=1}^{L} p_{j,q/i} \cdot p_q \cdot D(X_t, r_{j,q}^{(m-1)}) \le \sum_{q=1}^{Q}\sum_{j=1}^{L} p_{j,q/u} \cdot p_q \cdot D(X_t, r_{j,q}^{(m-1)}),\ u = 1,2,..,L\} \quad (17)$$

where $D(X_t, r_{j,q}^{(m-1)})$ is the distance between matrices X_t and $r_{j,q}^{(m-1)}$ as given in (4).

Step 4. Find the optimal set of reconstruction matrices $R^{(m)}$ according to:

$$r_{j,q}^{(m)} = \frac{\sum_{i=1}^{L} p_{j,q/i} \cdot E(X_t \mid X_t \in S_i^{(m)})}{\sum_{i=1}^{L} p_{j,q/i} \cdot P_i^{(m)}},\ j=1,2,\ldots,L \quad (18)$$

where $P_i^{(m)}$ is the probability that the input matrix X_t is an element of $S_i^{(m)}$.
Step 5. Compute the overall MSE given by:

$$D^{(m)} = \sum_{q=0}^{Q}\sum_{i=1}^{L}\sum_{j=1}^{L} p_{j,q/i} E[D(X_t, r_j^{(m)}) \mid X_t \in S_i^{(m)}] \quad (19)$$

If $\frac{D^{(m-1)} - D^{(m)}}{D^{(m)}} < \varepsilon$ then the algorithm has converged and is stopped. Otherwise go to Step 6.
Step 6. For the new structure of the matrix quantizer/decoder obtained in Steps 3-4, define the new set of SNR thresholds A_q. The thresholds are found numerically using the global random search method [20], under the constraint of equation (14). Set *m=m+1* and return to Step 2.

Finally the CSACOMQ algorithm for the FFRC, modelled by the Markov model, is presented:

Step A. The initialisations are identical with these of the CSACOMQ algorithm for the BSC. Additionally, an initial set of thresholds A_q, *q=1,2,…,L-1* is selected.
Step B. Initial codebooks of the move-to-front GTR algorithm are set $C_0=C^{(k-1)}$, and $R_0=R^{(k-1)}$. The time is set *t=1*.
Step C. The *Steps 2a-2e* of the algorithm of Section 2 are executed and generate codebooks C_t and R_t. Steps *2c* and *2d* are modified as described earlier in this section.
Step D. Set *m=1* and $R^{(m-1)} = R_t$.
Step E. Steps 2-6 of the COMQ algorithm for the FFRC are executed.
Step F. Set $R^{(k)} = R^{(m)}$. Codebook $C^{(k)}$ is formed by the centroids of the regions of partition $S_i^{(m)}$. The distortion of the *k*-th iteration is set $D^{(k)} = D^{(m)}$. If $\frac{D^{(k-1)} - D^{(k)}}{D^{(k)}} < \varepsilon$ then the algorithm has converged and is stopped, otherwise set *k=k+1* and go to Step B.

5. SIMULATION RESULTS AND DISCUSSION

Simulation results presented in this section document the performance improvement of the CSACOMQ algorithm over the COMQ algorithm, in terms of Signal-to-Noise Ratio (SNR) gain. The non-stationary source applied, was the Wiener process with $\sigma=1$, which generates sequences of $5.000.000 \cdot N \cdot M$ samples for the training and testing of both algorithms. For comparison purposes, both algorithms were trained with the same training samples sequence, and were in the sequel tested using various sample sequences.

The stopping threshold ε in the convergence criteria is set to 0.001. The windowing parameter is chosen ω=1000, which provides a sufficient frame of past input matrices. The size of the auxiliary vector encoder/decoder codebook K=512, which provides a sufficient resolution for quantizing the input matrix in case of a Codebook Update. The initial values of the set of probabilities of occurrence $\{p_0(i),\ 1 \le i \le L\}$ of the codematrices at the output of the matrix quantizer, were assumed to be equal.

Results were produced for rate-distortion parameter values ranging from λ=1800 to λ=4500. In this range, a reasonable trade-off between rate and distortion is observed. The reference COMQ algorithm for the case of the BSC was simulated as described in [16], whereas for the case of the FFRC as described in Section 4. The initial codebooks $R^{(0)}$ and $C^{(0)}$ for both algorithms were generated using the Generalized LBG algorithm [1]. It was assumed that the information regarding a codebook update is passed to the decoder side without distortion. Any possible temporal loss of matrix encoder/decoder synchronisation due to disruption of this information, is cancelled at the next iteration of the algorithm.

In Table 1, simulation results for the BSC of crossover probability *crossp* are presented. The displayed *SNR* performance gain of the CSACOMQ over the COMQ algorithm is the average SNR gain for values of the rate-distortion parameter from λ=1800 το λ=4500. The chosen design parameters are *N=2,3*, *M=4,6* and *L=256,512,1024,2048*. The parameters are chosen in such a manner that enables as to achieve low source coding rates.
In Table 2, simulation results for the FFRC of average error probability *p* are presented. The displayed *SNR* performance gain of the CSACOMQ over the COMQ algorithm is the average SNR gain for values of the rate-distortion parameter from λ=1800 το λ=4500. The chosen design parameters are

N=2,3, M=4,6, L=256,512,1024,2048 and *Q=16*. The number of states of the finite-state Markov model (*Q=16*) is considered to provide a good approximation for the FFRC in our algorithms.

The average error probability *p* of the FFRC, when coherent BPSK is adopted as the digital modulation scheme and ρ is the average received SNR is given by:

$$p = \sum_{q=1}^{Q} p_q(crossp)_q = \frac{1}{2}\left(1 - \sqrt{\frac{\rho}{\rho+1}}\right) \quad (20)$$

Table 1: Performance comparison between CSACOMQ and COMQ for the Binary Symmetric Channel

Average overall SNR gain of CSACOMQ over COMQ (dB)	Crossover Probability *crossp* of the BSC			
	0.001	***0.005***	***0.01***	***0.05***
N=2, M=4, L=256	1.32	1.31	0.83	0.54
N=2, M=4, L=512	1.45	1.41	0.89	0.56
N=2, M=4, L=1024	1.53	1.49	0.92	0.58
N=2, M=4, L=2048	1.57	1.50	1.04	0.59
N=2, M=6, L=256	1.18	1.16	0.72	0.42
N=2, M=6, L=512	1.22	1.20	0.76	0.46
N=2, M=6, L=1024	1.27	1.25	0.79	0.47
N=2, M=6, L=2048	1.30	1.26	0.80	0.50
N=3, M=4, L=256	1.16	1.13	0.68	0.39
N=3, M=4, L=512	1.21	1.19	0.71	0.43
N=3, M=4, L=1024	1.23	1.20	0.76	0.44
N=3, M=4, L=2048	1.27	1.24	0.79	0.47
N=3, M=6, L=256	1.02	0.99	0.59	0.29
N=3, M=6, L=512	1.06	1.02	0.61	0.29
N=3, M=6, L=1024	1.11	1.08	0.63	0.30
N=3, M=6, L=2048	1.14	1.11	0.67	0.31

In both tables, the overall (source-to-sink) *SNR* is defined as:

$$SNR=10\cdot log_{10}[E(X_t^2)/E(D(X_t, \hat{X}_t))]\text{dB} \quad (21)$$

The mean square value of the input matrix X_t is defined as:

$$E(X_t^2) = \frac{1}{M}\sum_{m=1}^{M} E(x_{tm}^2) \quad (22)$$

where $E(x_{tm}^2) = \|x_m\|^2$ is the squared Euclidean norm of x_m.

It should be noted that $E(D(X_t, \hat{X}_t))$ is the mean value of the distortion between source input matrix X_t and sink output matrix $\hat{X}_t$.

The matrix quantization rate (source coding rate) is defined as:

$$R_{MQ} = (log_2 L)/(N*M) \text{ bits/source symbol} \quad (23)$$

Table 2: Performance comparison between CSACOMQ and COMQ for the Flat Fading Rayleigh Channel

Average overall SNR gain of CSACOMQ over COMQ (dB)	Average error probability *p* of the FFRC			
	0.001	***0.005***	***0.01***	***0.05***
N=2, M=4, L=256	2.74	2.41	1.93	1.57
N=2, M=4, L=512	2.87	2.63	1.98	1.59
N=2, M=4, L=1024	3.06	2.68	2.01	1.64
N=2, M=4, L=2048	3.21	2.71	2.03	1.66
N=2, M=6, L=256	2.43	2.26	1.62	1.34
N=2, M=6, L=512	2.51	2.31	1.72	1.39
N=2, M=6, L=1024	2.62	2.36	1.81	1.43
N=2, M=6, L=2048	2.69	2.40	1.89	1.55
N=3, M=4, L=256	2.39	2.24	1.61	1.32
N=3, M=4, L=512	2.50	2.30	1.68	1.37
N=3, M=4, L=1024	2.59	2.32	1.79	1.40
N=3, M=4, L=2048	2.67	2.38	1.85	1.52
N=3, M=6, L=256	2.21	2.07	1.42	1.23
N=3, M=6, L=512	2.24	2.12	1.49	1.27
N=3, M=6, L=1024	2.28	2.18	1.53	1.29
N=3, M=6, L=2048	2.34	2.23	1.58	1.31

From the results demonstrated at Tables 1 and 2, it can be concluded that (i) the SNR gain increases as the matrix quantization rate increases (ii) for identical matrix quantization rate, the SNR gain increases as the number of rows (*N*) of each element matrix of the codebook decreases and (iii) the SNR gain decreases as the channel becomes noisier. These SNR gains are achieved at the expense of minimal (at the order of 10^{-2}) additional off-line computational complexity, due to the combining nature of the CSACOMQ algorithm. However, CSACOMQ has exactly the same on-line computational requirements as COMQ and clearly designs more effectively the matrix quantizer/decoder pair for variable source statistics and noisy channel environments.

6. CONCLUSIONS AND FUTURE WORK

In this paper a novel move-to-front GTR algorithm for Matrix Quantization is initially presented. Subsequently, the novel CSACOMQ algorithm is introduced, which designs a matrix quantizer/decoder pair taking into account both the non-stationary nature of the source and the noisy nature of the channel. The CSACOMQ algorithm is compared to the reference COMQ algorithm for various values of the design parameters of the matrix quantizer/decoder pair and for both considered channels. Simulation results show gains of up to 1.57 dB for source coding rate ranging from 4/9 to 11/8 bits

per source symbol for the case of the BSC. For the case of the FFRC, performance gains of up to 3.21 dB for source coding rate ranging from 4/9 to 11/8 bits per source symbol are achieved. The SNR gain is achieved at the expense of minimal off-line additional computational complexity (at the order of 10^{-2}), while no additional on-line computational complexity is required.

The CSACOMQ algorithm can be utilised by modern audio and speech codec standards [2]-[6], [9], [10], image compression techniques [8] and video compression techniques [7] where common ground is the need for low bit-rate communication systems, quick and robust adaptation to varying source statistics and optimization to fast changing noisy channels. The concept of CSACOMQ can be generalised to combine other AMQ techniques with the COMQ algorithm. The authors also intend to investigate the performance of the CSACOMQ algorithm for high bit-rate communication systems.

REFERENCES

1. A. Gersho, and R. Gray. ***Vector Quantization and Signal Compression***, Springer, 1992, Ch. 10, pp. 309-343, Ch. 11, pp. 345-405, Ch. 16, pp. 587-629.
2. J. Makinen, B. Bessette, S. Bruhn, P. Ojala, and R. Salami. **AMR-WB+: A new audio coding standard for 3rd generation mobile audio services**, in *Proceeding of 2005 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005, Vol. 2, pp. 1109-1112.
3. J.H. Chen, and J. Thyssen. **The Broadvoice Speech Coding Algorithm**, in *Proceeding of 2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, Vol. 4, pp. 537-540.
4. J.M. Valin, T.B. Terriberry, C. Montgomery, and G. Maxwell. **A High-Quality Speech and Audio Codec With Less Than 10 ms delay**, *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, No. 1, pp. 58-67, January 2010.
5. J.M. Valin, G. Maxwell, T.B. Terriberry, and K. Vos. **High-Quality, Low-Delay Music Coding in the Opus Codec**, in *Proceedings of the 135th Convention of the Audio Engineering Society (AES-13)*, New York, NY, US, October 2013, Paper No. 8942.
6. K. Vos, K.V. Sorensen, S.S. Jensen, and J.M. Valin. **Voice Coding with Opus**, in *Proceedings of the 135th Convention of the Audio Engineering Society (AES-13)*, New York, NY, US, October 2013, Paper No. 8941.
7. S. Esakkirajan, T. Veerakumar, and P. Navaneethan. **Adaptive Vector Quantization based Video Compression Scheme**, in *International Conference on Multimedia, Signal Processing and Communication Technologies*, 2009, pp. 40-43.
8. M. Kumar, R. Kapoor, and T. Goel. **Vector Quantization based on Self-Adaptive Particle Swarm Optimization**, *International Journal of Nonlinear Sciences*, Vol. 9, No. 3, pp. 311-319, June 2010.
9. M. Zhanyu, A. Leijon, and W.B. Kleijn. **Vector quantization of LSF parameters with a mixture of dirichlet distributions**, *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 21, No. 9, pp. 1777-1790, September 2013.
10. M.C. Chu, and D.V. Anderson. **Likelihood codebook reordering vector quantization**, in *International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 5114-5117.
11. V. Bozantzis, and Ali. **Combined Source Adaptive and Channel Optimized Vector Quantization Algorithm**, *IEE Electronics Letters*, Vol. 35, No. 17, pp. 1455-1456, August 1999.
12. V. Bozantzis, and Ali. **Combined Source Adaptive and Channel Optimized Vector Quantization (CSACOVQ) for a Flat Fading Rayleigh Channel**, in *Proceedings of the IEEE Symposium on Communications and Vehicular Technology*, IMEC, Leuven (BE), October 2000, pp. 201-208.
13. V. Bozantzis, and Ali. **Combined Vector Quantization and Index Assignment with Embedded Redundancy for Noisy Channels**, *IEE Electronics Letters*, Vol. 36, No. 20, pp. 1711-1713, September 2000.
14. V. Bozantzis, F. H. Ali, and E. Stipidis. **Joint Source Adaptive Vector Quantization and Index Assignment with Embedded Redundancy**, *IET Proceedings on Communications*, Vol. 1, No. 5, pp. 1023-1030, October 2007.
15. C. Xydeas, and C. Papanastasiou. **Efficient Coding of LSP Parameters Using Split Matrix Quantization**, in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1995, pp. 740-743.
16. J.L. Perez-Cordoba, A. Rubio, J.M. Lopez-Soler, and V.E. Sanchez. **Channel optimized matrix quantization (COMQ) of LSP parameters over waveform channels**, in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, Vol. 2, pp. 725-728.
17. N. Farvardin, and V. Vaishampayan. **On the performance and complexity of Channel-Optimized Vector Quantizers**, *IEEE Transactions on Information Theory*, Vol. 37, No. 1, pp. 155-160, February 1991.
18. J.E. Fowler. **Generalized Threshold Replenishment: An Adaptive Vector Quantization Algorithm for the Coding of Non- stationary Sources**, *IEEE Transactions on Image Processing*, Vol. 7, Issue 10, pp. 1410-1424, October 1998.
19. Hong Shen Wang. **Finite-state modeling, capacity, and joint source/channel coding for time-varying channels**, Graduate School-New Brunswick Rutgers, The State University of New Jersey, May 1992.
20. Singiresu S. Rao. ***Engineering Optimization: Theory and Practice***, John Wiley & Sons, 1996, Chapter 7, pp. 428-555.