ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

# Robust Vector Quantization for Burst Error Channels Using Genetic Algorithm

*Wen-Jyi Hwang*[1],*Chien-Min Ou*[2,3],*and Chin-Ming Yeh*[2]

[1]Graduate Institute of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, Taiwan 117, R.O.C.
[2]Department of Electrical Engineering,
Chung Yuan Christian University, Chungli, Taiwan 320, R.O.C.
[3]Department of Electronic Engineering,
Ching Yun Institute of Technology, Chungli, Taiwan 320, R.O.C.

**Abstract**. This paper presents a novel vector quantizer (VQ) design algorithm for a burst error channel (BEC). The algorithm minimizes the average distortion when the BEC is in normal state of operation, while maintaining a minimum fidelity when the BEC is in the undesirable state. An iterative design procedure is first derived in the algorithm for obtaining a local optimal solution. A novel genetic scheme is then proposed for attaining a near global optimal performance. Numerical results show that the algorithm significantly outperforms the VQ techniques optimizing the design only to the simple binary symmetric channels.

**Keywords:** Vector Quantization, Genetic Algorithm, Channel-Optimized Source Coding.

## 1 Introduction

Vector quantizers (VQ's) [2] have been found to be effective for data compression. The techniques remove redundancy in the source, and retain useful information for subsequent transmission. In the presence of channel noise, this removal of redundancy may cause significant performance degradation. One way to achieve some degree of robustness to channel errors is to employ the channel-optimized VQ (COVQ)[1] techniques, which optimizes a VQ under a specific noisy channel condition. The basic COVQ method [1, 3] constructs the codebook best matched to a given binary symmetric channel (BSC), and is therefore termed the COVQ-BSC technique. In many practical communication systems, the burst errors are likely to occur. Since the simple BSC model is not suited for describing the bursty error channels, the COVQ-BSC may not be effective for those communication systems. As compared with the BSC model, the Gilbert-Elliot (G-E) model are more effective for describing the burst error channels (BEC's). In the model,

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

there are two states: the good state is almost error free; the bad state has burst errors. A BEC can then be specified by the bit error rate (BER) in both states, and the state transition probabilities of the model. Therefore, to improve the robustness of the VQ's to burst errors, this paper proposes a novel COVQ design algorithm, termed COVQ-BEC technique, which optimizes the codebook design to the G-E models.

Unlike the COVQ-BSC scheme, where only one codebook is trained, the COVQ-BEC technique designs two codebooks, with one for each state of the given BEC. The selected codebook for encoding and decoding are dependent on the state information observed at the transmitter and receiver, respectively. Prior to the COVQ-BEC design, the algorithm allows the constraint of average distortion of the bad state to be pre-specified. Under the constraint, the COVQ-BEC minimizes the average distortion of the good state. The Lagrangian method is used to solve this optimization problem. Therefore, the cost function to this problem is the weighted sum of the average distortion of good and bad states. The necessary conditions for encoder and decoder minimizing the cost function are derived. Based on those necessary conditions, an iterative procedure is first proposed for the training of the codebook for each state, yielding a local optimal solution.

To obtain a near global optimal performance, a hybrid scheme combining the iterative design procedure and genetic algorithm (GA) [5] is proposed. Inspired by biological evolution, the GA has also been successfully used for global optimization [3]. The GA contains a set of genetic strings, which are evaluated by a fitness function. The fittest strings are then regenerated at the expense of the others. Moreover, crossover and mutation are employed to obtain better strings. The mutation operation changes individual elements of a string, and the crossover operation interchanges parts between strings. In our hybrid scheme, each string of the GA are the codebooks of the VQ. The resulting strings after the regeneration, crossover and mutation operations are used as the initial codebooks for the iterative design procedure. The codebooks designed by the iterative procedure are then used as the strings for the genetic operations of the next generation. The same process is repeated until the convergence of the algorithm. Numerical results show that the COVQ-BEC design based on the hybrid scheme has superior performance over that of the iterative scheme. In addition, the COVQ-BEC significantly outperforms the COVQ-BSC based on the same BEC channels. Our COVQ-BEC schemes therefore can be an effective alternative for robust transmission over burst noise channels.

## 2 The COVQ-BEC Algorithm

Figure 1 shows the basic structure of the VQ designed by the COVQ-BEC algorithm. Let $s_1$ and $s_2$ denotes the good and bad states of the G-E model, respectively. Let $s_j$ be the actual state of the channel, and let $s_m$ and $s_n$ be the estimated state of the channel at the transmitter and receiver, respectively. The transmitter (receiver) of the VQ has two encoders (decoders) with one for

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

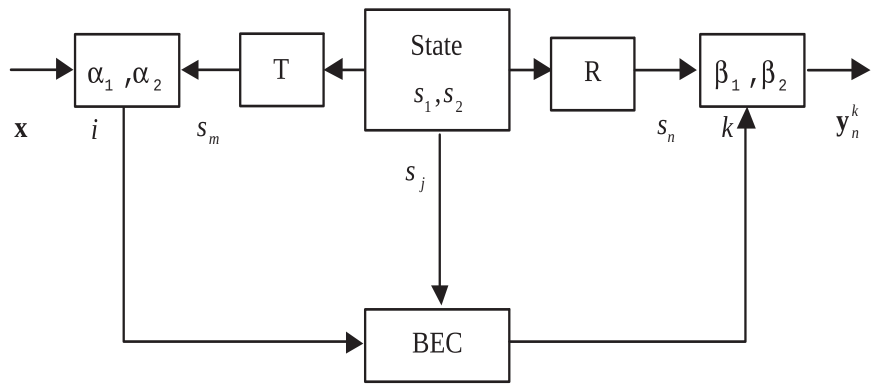


Figure 1: Basic structure of VQ designed by the COVQ-BEC algorithm.

each state. Let $\alpha_m$ and $\beta_n$ be the encoder and decoder used by the transmitter and receiver when their estimated state are $s_m$ and $s_n$, respectively. Let $B_n = \{\mathbf{y}_n^1, ..., \mathbf{y}_n^N\}$ be the codebook used by $\beta_n$, where $\mathbf{y}_n^k$ is the $k$-th codeword of $B_n$, and $N$ is the number of codewords. All the codewords are with dimension $q$.

Let $\mathbf{x} \in \mathcal{R}^q$ be the $q$-dimensional source vector to be encoded. Given $\mathbf{x}$ and the estimated state $s_m$ in the transmitter, an index $i = \alpha_m(\mathbf{x})$ is generated and delivered. Let $P_j(k/i)$ be the probability that the receiver obtains index $k$, when index $i$ is transmitted, and the actual state of channel is $s_j$. Since the estimated state at the receiver is $s_n$, after receiving the index $k$, the decoder $\beta_n$ reproduces the corresponding codeword $\mathbf{y}_n^k$ from its codebook $B_n$.

Let $\mathcal{T} = \{\mathbf{x}^l, l = 1, ..., t\}$ be the set of training vectors,where $t$ is the number of training vectors. Let $D_j(m, n)$ be the average distortion of the VQ measured on $\mathcal{T}$, given that $s_j$ is the actual state of the channel, and $s_m$ and $s_n$ are the estimated state of the channel at the transmitter and receiver, respectively. The $D_j(m, n)$ is then given by

$$D_j(m,n) = \frac{1}{t}\sum_{l=1}^{t}\sum_{k=1}^{N} P_j(k/\alpha_m(\mathbf{x}^l))d(\mathbf{x}^l, \mathbf{y}_n^k), \quad j, m, n = 1, 2, \tag{1}$$

where $d(\mathbf{u}, \mathbf{v})$ is the squared distance between vectors $\mathbf{u}$ and $\mathbf{v}$.

Since the observation of channel states is subject to errors, let $t_{mj}$ ($r_{nj}$) be the state estimation probability that state $s_m$ ($s_n$) is perceived at transmitter (receiver) when the state of the channel is $s_j$. Consequently, $D_j$, the average distortion of the VQ when the channel is in state $s_j$, is given by

$$D_j = \sum_{m=1}^{2}\sum_{n=1}^{2} t_{mj} r_{nj} D_j(m,n), \quad j = 1, 2. \tag{2}$$

Given a BEC described by the G-E model, the objective of the COVQ-BEC

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

algorithm can be formulated as follows:

$$\text{minimize} \quad D_1, \qquad \text{subject to} D_2 \leq \bar{D}. \tag{3}$$

That is, the algorithm minimizes the average distortion of the good state $s_1$, subject to $\bar{D}$, the constraint of the average distortion of the bad state $s_2$. We convert this constrained optimization problem to an unconstrained optimization problem by solving the following Lagragian:

$$J = D_1 + \lambda D_2, \tag{4}$$

where $\lambda \geq 0$ is the Lagrange multiplier.

## 2.1 The Iterative Design Procedure

Substituting eqs.(1)(2) into eq.(4), we can rewrite the cost function $J$ as

$$J = \frac{1}{t} \sum_{j=1}^{2} \lambda_j \sum_{m=1}^{2} \sum_{n=1}^{2} t_{mj} r_{nj} \sum_{l=1}^{t} \sum_{k=1}^{N} P_j(k/\alpha_m(\mathbf{x}^l)) d(\mathbf{x}^l, \mathbf{y}_n^k). \tag{5}$$

where $\lambda_1 = 1$ and $\lambda_2 = \lambda$. In our design, $t_{mj}$, $r_{nj}$, $P_j(k/i)$ and $\lambda$ should be prespecified. The cost function $J$ therefore is minimized by finding the optimal $\alpha_m$ and $B_n$, which can be accomplished by an iterative procedure shown below.

### 2.1.1 The optimal encoder design

Assume $B_1$ and $B_2$ are fixed. From eq.(5), we rewrite $J = \sum_{m=1}^{2} J(m)$, where

$$J(m) = \frac{1}{t} \sum_{j=1}^{2} \lambda_j \sum_{n=1}^{2} t_{mj} r_{nj} \sum_{l=1}^{t} \sum_{k=1}^{N} P_j(k/\alpha_m(\mathbf{x}^l)) d(\mathbf{x}^l, \mathbf{y}_n^k), \quad m = 1, 2. \tag{6}$$

Therefore, given a set of codebooks $B_1$ and $B_2$, the minimization of $J$ is equivalent to the independent minimization of each $J(m)$, which depends only on the encoder $\alpha_m$. From eq.(6), it follows that, given codebooks $B_1$ and $B_2$, the optimal encoder $\alpha_m$ should satisfy

$$\alpha_m(\mathbf{x}^l) = \arg \min_{1 \leq i \leq N} \sum_{j=1}^{2} \lambda_j \sum_{n=1}^{2} t_{mj} r_{nj} \sum_{k=1}^{N} P_j(k/i) d(\mathbf{x}^l, \mathbf{y}_n^k), \quad m = 1, 2. \tag{7}$$

### 2.1.2 The optimal decoder design

Assume the encoders $\alpha_1$ and $\alpha_2$ are fixed. Since each decoder $\beta_n$ is completely characterized by the codebook $B_n$, designing the optimal $B_n$ is equivalent to designing the optimal $\beta_n$. For a given set of $\alpha_1$ and $\alpha_2$, it can be shown that each codeword $\mathbf{y}_n^k$ minimizing $J$ is evaluated as

$$\mathbf{y}_n^k = \frac{\sum_{j=1}^{2} \lambda_j \sum_{m=1}^{2} t_{mj} r_{nj} \sum_{l=1}^{t} P_j(k/\alpha_m(\mathbf{x}^l)) \mathbf{x}^l}{\sum_{j=1}^{2} \lambda_j \sum_{m=1}^{2} t_{mj} r_{nj} \sum_{l=1}^{t} P_j(k/\alpha_m(\mathbf{x}^l))}, \quad n = 1, 2, \ k = 1, ..., N. \tag{8}$$

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

#### 2.1.3 The complete iterative design procedure

The eqs.(7) and (8) are the necessary conditions for the optimal encoder and decoder, respectively. Based on eqs.(7) and (8), the complete iterative procedure for COVQ-BEC design is listed below:

| | |
|---|---|
| Step 0 | Given $t_{mj}$, $r_{nj}$, $P_j(k/i)$, $\lambda$, initial $B_1$ and $B_2$,<br>Set $f = 0$,$J^0 = \infty$, $\epsilon > 0$. |
| Step 1 | Fix current $B_1$ and $B_2$, find the optimal $\alpha_1$ and $\alpha_2$ using eq.(7). |
| Step 2 | Fix current $\alpha_1$ and $\alpha_2$, find the optimal $B_1$ and $B_2$ using eq.(8). |
| Step 3 | Set $f \leftarrow f + 1$.<br>Compute the new value of the Lagrangian $J$, denoted by $J^f$. |
| Step 4 | If $(J^f - J^{f-1})/J^f < \epsilon$, then stop, else goto Step 1. |

Since the sequence $\{J^f\}$ is nonincreasing, and is bounded below by zero, the convergence of the sequence is guaranteed. The resulting encoders and codebooks after the convergence of $\{J^f\}$ minimizes (locally) the cost function $J$. After the design, the average distortions $D_1$ and $D_2$ of the VQ can be computed by eq.(2). By repeating the iterative design procedure with different $\lambda$ values, we obtain a plot of attainable $D_1$ versus $D_2$.

## 2.2 The Hybrid Design Procedure

The objective of the hybrid design procedure is to solve the problem of local optima for the minimization of cost function $J$ by combining the iterative design procedure with the GA. In the hybrid scheme, there are $P$ genetic strings for the genetic operation. Each string $r$ represents a set of $2N$ codewords $\{\mathbf{y}_1^1, ..., \mathbf{y}_1^N, \mathbf{y}_2^1, ..., \mathbf{y}_2^N\}_r$, where $\{\mathbf{y}_n^1, ..., \mathbf{y}_n^N\}$ are the codewords of the codebook $B_n$. Let $S(k)$ and $C(k)$ denote the set of $P$ strings and the value of *current minimum* $J$ value after the execution of the $k$-th iteration of hybrid scheme, respectively. Let $s^*$ be the *current optimum string* during the course of hybrid scheme. In the initial step of hybrid scheme, we let $C(0) = \infty$, and initialize $s^*$ as *null*. In addition, we can randomly select vectors from training data as the codewords of strings in $S(0)$.

During the course of hybrid design for minimizing $J$ given a fixed $\lambda$, suppose that the $(k-1)$-th iteration is completed, and the execution of the $k$-th (note that $k \geq 1$) is to be done. We then perform the following genetic operations sequentially on the strings in $S(k-1)$.

**Regeneration**: Since each string in $S(k-1)$ for the genetic operations is in fact a VQ, their corresponding $J$ value can be computed using eq.(5). The inverse of $J$ is used as fitness function for each string. The regeneration process is then conducted using the roulette-wheel technique [5]. That is, for offspring generation, we spin a simulated biased roulette wheel whose slots have different sizes proportional to the fitness values of the individual strings. The results of the spin gives a reproduction candidate. Once a string has been selected

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

for reproduction, an exact replica of it is made as a regeneration string. This regeneration string will then be used for crossover and mutation. In our, $P$ regeneration strings are created after the regeneration operation.

**Crossover**: On each regeneration string $r$, $\{\mathbf{y}_1^1, ..., \mathbf{y}_1^N, \mathbf{y}_2^1, ..., \mathbf{y}_2^N\}_r$, two-point crossover is applied with probability $P_c$. Out of the total population, a partner string $\hat{r}$, $\{\hat{\mathbf{y}}_1^1, ..., \hat{\mathbf{y}}_1^N, \hat{\mathbf{y}}_2^1, ..., \hat{\mathbf{y}}_2^N\}_{\hat{r}}$ is randomly chosen. Then two integer random numbers $h_1$ and $h_2$ between 1 and $N$ are generated. The strings $r$ and $\hat{r}$ are mutually exchanged in accordance with the following equation:

$$\{\mathbf{y}_1^1, ..., \mathbf{y}_1^N, \mathbf{y}_2^1, ..., \mathbf{y}_2^N\}_r \to \{\mathbf{y}_1^1, ..., \mathbf{y}_1^{h_1}, \hat{\mathbf{y}}_1^{h_1+1}, ..., \hat{\mathbf{y}}_1^N, \mathbf{y}_2^1, ..., \mathbf{y}_2^{h_2}, \hat{\mathbf{y}}_2^{h_2+1}, ..., \hat{\mathbf{y}}_2^N\}_r$$
$$\{\hat{\mathbf{y}}_1^1, ..., \hat{\mathbf{y}}_1^N, \hat{\mathbf{y}}_2^1, ..., \hat{\mathbf{y}}_2^N\}_{\hat{r}} \to \{\hat{\mathbf{y}}_1^1, ..., \hat{\mathbf{y}}_1^{h_1}, \mathbf{y}_1^{h_1+1}, ..., \mathbf{y}_1^N, \hat{\mathbf{y}}_2^1, ..., \hat{\mathbf{y}}_2^{h_2}, \mathbf{y}_2^{h_2+1}, ..., \mathbf{y}_2^N\}_{\hat{r}}$$

**Mutation**: Mutation is performed on each codeword $\mathbf{y}_n^k$ of each string with a small probability $P_m$. From $\mathbf{y}_n^k$, one of the $q$ components is chosen at random. Then a random number, taking the binary values $b$ or $-b$, is generated, and is added to the chosen component.

The total of these three operations is called a generation. After a generation, we then apply the iterative design procedure to each string with the codewords in that string as the initial conditions. The $P$ strings after the iterative design are then the strings of the set $S(k)$. The $J$ value of each string in $S(k)$ is computed. Let $r^*$ be the string in $S(k)$ having minimum $J$ value, and $J^*$ be the $J$ value of $r^*$. We then compare $J^*$ with $C(k-1)$. If $J^*$ is smaller than $C(k-1)$, then $C(k) \leftarrow J^*$, and $s^* \leftarrow r^*$. Otherwise, $C(k) \leftarrow C(k-1)$, and the *current optimum string* $s^*$ is retained the same. This completes the execution of the $k$-th iteration for the hybrid scheme.

The iteration continues until the convergence of the sequence $\{C(k)\}$. In practice, we stop the design algorithm after the observation of $I$ consecutive iterations yielding identical $C(k)$ value. The *current optimum* string $s^*$ after the completion of hybrid scheme is then the desired codebooks. To show the convergence of $\{C(k)\}$, we first observe that the sequence is nonincreasing. In addition, from eq.(5), it follows that $C(k) > 0$ for all integers $k > 0$. Therefore, $\{C(k)\}$ is bounded below, and is guaranteed to converge.

# 3 Simulation Results

This section presents some numerical results of the COVQ-BEC algorithm. Two Gauss-Markov sources with identical parameter $\rho = 0.9$ and identical length (96000 samples) are used for the VQ training and performance measurement, respectively. The dimension of vector is $q = 8$. The number of training and test vectors therefore are 12000. Two cases are considered for the implementation of the COVQ-BEC: both transmitter and receiver do not have state information (denoted as COVQ-BEC I), and only receiver has accurate state information (denoted as COVQ-BEC II). The number of codewords of each state for the COVQ-BEC design is $N = 32$. The BEC considered in this experiment is modeled by the G-E channel with BER $\epsilon_1 = 0.0001$ in state $s_1$, and BER $\epsilon_2 = 0.1$ in the state $s_2$. The state transition probabilities are given by $p_{11} = 0.9$,

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

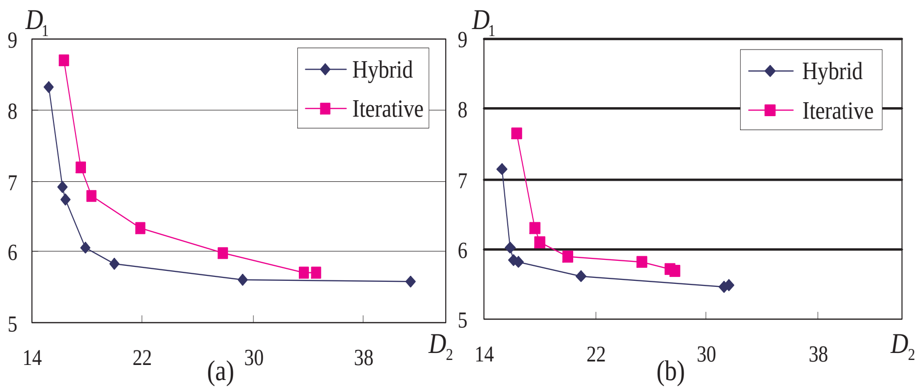


Figure 2: Performance of COVQ-BEC I and II realized by iterative and hybrid schemes: (a) COVQ-BEC I,(b) COVQ-BEC II.

$p_{21} = 0.1$, $p_{12} = 0.9$ and $p_{22} = 0.1$, where $p_{ij}$ denotes the probability that given the current state of the channel is $s_j$, the next state of the channel will be $s_i$. Let $p_j$ be the probability that the channel is in state $s_j$. Therefore, $p_1 = 0.9$ and $p_2 = 0.1$. The average BER of the BEC channel, $\epsilon$, is then given by $\epsilon = 0.01009$.

Figure 2 shows the location of the sample points of the iterative and hybrid schemes for the two cases in the $D_1$-$D_2$ plane. Different sample points are obtained by varying $\lambda$ values in eq.(4). We set $P = 6, P_c = 0.8$ and $P_m = 0.3$ for the implementation of the hybrid scheme. Form Figure 2, we observe that the hybrid scheme outperforms the iterative scheme for each case. The hybrid scheme has superior performance because it can attain a near global optimal performance using the GA algorithm. On the contrary, the iterative scheme may fall into a poor local optima when improper sets of initial codewords are chosen. Figure 3 compares the performance of the COVQ-BEC I and II realized by the hybrid scheme. The performance of the COVQ-BSC is also included for the comparison purpose. The implementation of the COVQ-BSC is optimized to the BSC with the same average BER $\epsilon = 0.01009$ as that of the BEC. From the figure, it is observed that the COVQ-BEC II outperforms COVQ-BEC I and the COVQ-BSC. This is because the state information is available at receiver in that case. The performance of the COVQ-BEC may be degraded when the state information become unavailable. Nevertheless, even when both the transmitter and receiver do not have state information, the COVQ-BEC still outperforms COVQ-BSC. As shown in the figure, when $D_2$ is 26.81, the $D_1$ of the COVQ-BEC II is 5.67, which is lower than that of COVQ-BSC by 0.68.

## 4 Conclusion

Numerical results have shown that the performance of the COVQ-BEC can be effectively improved using the GA algorithm. In addition, its performance can be enhanced further when channel status information becomes available. Given

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 267-274

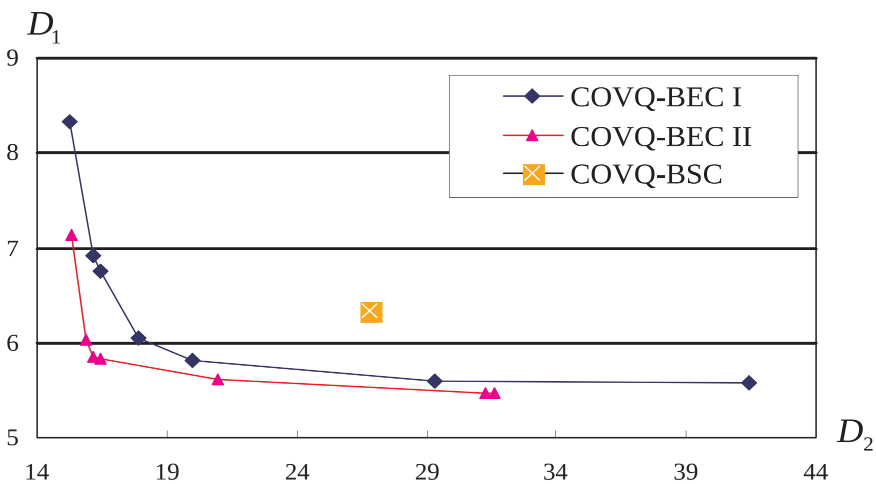


Figure 3: Comparison of COVQ-BEC I,II and COVQ-BSC schemes.

the same BEC channel, the COVQ-BEC algorithm also significantly outperforms the COVQ-BSC algorithm even when the state observation is noisy. The COVQ-BEC therefore can be an attractive alternative for the applications where the robust transmission over burst error channels are desired.

# References

[1] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Information Theory*, Vol. 37, pp.155-160, 1991.

[2] A. Gersho and R.M. Gray, Vector quantization and signal compression, Kluwer, Amsterdam, The Netherlands, 1992.

[3] W.J. Hwang and S.L. Hong, "Genetic entropy-constrained vector quantization design algorithm," *Optical Engineering*, pp.233-239, 1999.

[4] W.J. Hwang, F.J. Lin and C.T. Lin, "Fuzzy channel-optimized vector quantization," *IEEE Communications Letters,* Vol. 4, pp.408-410, 2000.

[5] C.T. Lin and C.S. Lee, *Neural fuzzy systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1996.