TECHNICAL RESEARCH REPORT

Speech Coding over Noisy Channels

by N. Farvardin

T.R. 94-5

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Speech Coding over Noisy Channels†

Nariman Farvardin[1]

ABSTRACT This chapter contains a discussion of quantization over noisy channels. The effects of channel noise on the performance of vector quantizers are discussed and algorithms for the design of noisy-channel vector quantizers are presented. It is argued that in certain practical situations where delay and complexity place hard limits on system parameters, a combined source-channel coding approach might be preferable to the more traditional tandem source-channel coding. Examples of full-searched, multi-stage and finite-state vector quantization designed for a noisy channel are provided for coding of speech line spectrum pair parameters.

1.1 Introduction

A generic point-to-point communication system can be described by a source encoder/decoder, a channel encoder/decoder and a modulator/demodulator pair as depicted in Fig. 1.1. The modulator/demodulator essentially maps the waveform channel into a digital channel.

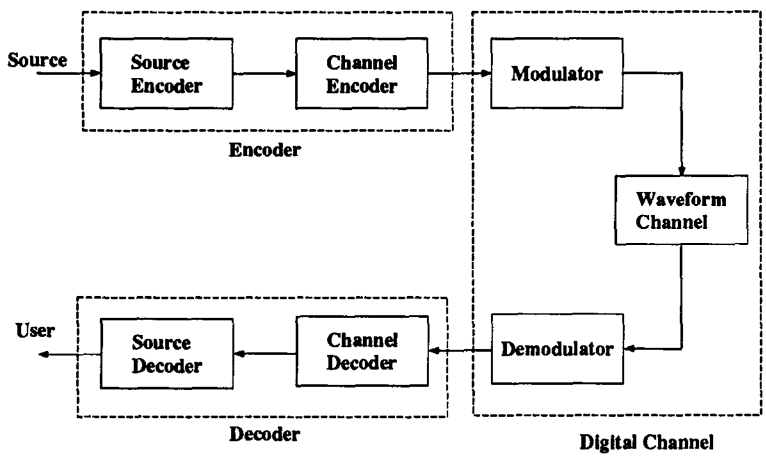

FIGURE 1.1. Block Diagram of a Generic Communication System.

†This work was supported in part by National Science Foundation grants NSFD MIP-9109109 and NSFD CD-8803012 and in part by NTT Corporation.

[1]Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742.

It is established by Shannon that the separation of the source and channel coding operations does not cause any loss of optimality in the limit of large block sizes [1], [2]. It is in view of this separation principle that the two areas of source and channel coding have evolved largely independently over the past three decades. Source coding researchers have pursued the quest for designing codes that achieve performance close to the source distortion-rate function while channel coding researchers have focused their effort on designing powerful error detection/correction schemes [3] - [6].

After a few decades of research, source and channel coding problems are fairly well understood and a number of powerful coding schemes have been developed. Within the class of block-structured codes, vector quantization techniques for source coding [4] and a variety of error correction codes (e.g., Golay, Reed-Solomon and BCH codes) [5], [6] for channel coding have emerged as promising methods. Generally speaking, larger block sizes (hence, larger delays and complexities) lead to better code performance. Practical considerations however impose certain constraints on the encoding delay and complexity, which, in turn, limit the performance of the code. These delay and complexity constraints are especially important in situations where real-time communication is needed and where power consumption, weight and cost of hardware are to be kept small. A good example of such a situation is two-way speech communication in a wireless communication network.

An important question that arises then is the following: How good (or close-to-optimal) is the performance of a *tandem source-channel code* which is obtained by concatenating a source encoder and a channel encoder subject to the practical constraints of encoding delay and complexity? This is a valid question because in the absence of Shannon's assumption of large blocks, there is no theoretical justification for the separation of source and channel coders. In particular, in a lossy coding situation where for example the ultimate objective is the minimization of average squared-error distortion between the source output and its reproduction, it is not clear why the bit (or symbol) error minimization criterion used for the channel code is an appropriate design criterion. A follow up to this question would be whether it is possible to design a *combined source-channel code* in which the design problem is set up in such a way that the overall average distortion caused by source coding and channel noise is minimized. Such a combined code will operate on the source output and generate a bit stream that will be delivered directly to the digital channel, as depicted in Fig. 1.2. If such a combined scheme can be designed, can it outperform a good tandem source-channel code with the same delay and complexity constraints?

FIGURE 1.2. Block Diagram of a Generic Communication System Using a Combined Source-Channel Code.

With the emergence of wireless communication technologies the above questions are becoming more important than ever. In these communication problems, because of the severe limitations on the available bandwidth, source coding is essential. On the other hand, wireless channels are typically very noisy and suffer from various multipath fading and bursty errors problems and therefore some type of error control is necessary. At the same time, the need for two-way communication places hard limits on the end-to-end delay. Finally, in many wireless applications the transmitter/receivers are required to be low-power (to better utilize battery life), light and inexpensive. These requirements place a constraint on the system complexity. Clearly, the wireless communication scenario presents an example where both source and channel coding are essential components of the system and where delay and complexity are severely constrained. It is therefore important to determine whether a combined approach to source-channel coding might not be preferable to the tandem approach under these conditions.

In the following sections we provide a brief discussion on combined source-channel coding and provide partial answers to some of the questions posed above. Our discussion will be limited to block-structured coding schemes. We begin by a discussion of full-searched vector quantization over a noisy channel and extend this to multi-stage and finite-state vector quantization. Some examples of applications to speech coding will also be discussed. Due to the limitation on the number of pages, our discussion will be mostly qualitative. Details can be found in the references.

1.2 Vector Quantization over Noisy Channels

In this section, we consider two issues: (i) How to assign indices to the codevectors of a *given* vector quantizer when they are to be transmitted over a noisy channel and (ii) how to design a vector quantizer which is intended for use over a noisy channel. For this discussion, we consider a real-valued, zero-mean, stationary and ergodic source $\{X_t;\ t = 0, 1, \ldots\}$.

1.2.1 Vector Quantizer Index Assignment

A k-dimensional, M-level vector quantizer (VQ) is described by an *encoder* mapping α and a *decoder* mapping β. The encoder $\alpha : \mathbb{R}^k \mapsto \mathcal{J}_M \equiv \{0, 1, \ldots, M-1\}$ is described in terms of a partition $\mathcal{P} = \{S_0, S_1, \ldots, S_{M-1}\}$ of $\mathbb{R}^k$ according to

$$\alpha(\mathbf{x}) = i, \text{ if } \mathbf{x} \in S_i,\ i \in \mathcal{J}_M, \tag{1.1}$$

where $\mathbf{x} \in \mathbb{R}^k$ is a typical source output vector. The decoder mapping $\beta : \mathcal{J}_M \mapsto \mathbb{R}^k$, is described in terms of a codebook $\mathcal{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{M-1}\}$, according to

$$\beta(i) = \mathbf{c}_i,\ i \in \mathcal{J}_M. \tag{1.2}$$

The quantization rule is $q \equiv \beta \circ \alpha$ where $q(\mathbf{x}) = \beta(\alpha(\mathbf{x}))$.

At this point, we are not concerned with the design of the quantizer; we assume that the quantizer (i.e., the partition and codebook) is designed and fixed. In many practical situations the encoder output indices are mapped into binary words (codewords) and transmitted (or stored in storage situations). While in the absence of channel noise the assignment of the codewords to the VQ codevectors does not affect the average distortion, in the presence of channel noise this assignment plays an important role in determining the overall VQ performance.

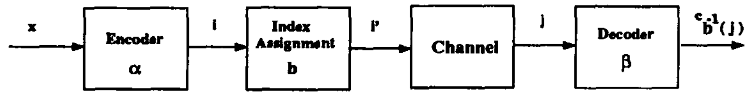

FIGURE 1.3. Block Diagram of the Vector Quantizer Based Coding System.

To be precise, consider Fig. 1.3 in which a one-to-one index (or codeword) assignment mapping b takes the output i of the encoder α and maps it into another integer $i' = b(i) \in \mathcal{J}_M$; the binary representation of i' is the codeword associated with S_i or $\mathbf{c}_i$. Let us assume that the channel is a discrete memoryless channel (DMC) with input and output alphabets $\mathcal{J}_M$ and with $P(j|i')$ denoting the probability that the index j is received given that i' is transmitted. We use $d(\mathbf{x}, \mathbf{y})$ to denote the distortion caused by representing the source vector $\mathbf{x}$ by a reproduction vector $\mathbf{y}$. Then the overall average distortion per sample $D(\mathcal{P}, \mathcal{C}; b)$ is given by

$$D(\mathcal{P}, \mathcal{C}; b) = \frac{1}{k} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(j|b(i)) \int_{S_i} p(\mathbf{x}) d(\mathbf{x}, \mathbf{c}_{b^{-1}(j)}) d\mathbf{x}, \tag{1.3}$$

where $p(\mathbf{x})$ is the k-fold probability density function (p.d.f.) of the source. While in general the contributions of the quantization noise and channel noise cannot be separated, in the special case where the squared-error distortion measure is used, $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||^2$, with the additional assumption that each VQ codevector is the centroid of its corresponding encoding region, it is easy to show that the average distortion in (1.3) can be decomposed into

$$D(\mathcal{P}, \mathcal{C}; b) = D_s + D_c, \tag{1.4}$$

where D_s, the contribution of the quantizer to the average distortion, is given by

$$D_s = \frac{1}{k} \sum_{i=0}^{M-1} \int_{S_i} p(\mathbf{x}) ||\mathbf{x} - \mathbf{c}_i||^2 d\mathbf{x}, \tag{1.5}$$

and D_c, the contribution of the channel noise to the overall distortion, is given by

$$D_c = \frac{1}{k} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P_i P(j|b(i)) ||\mathbf{c}_i - \mathbf{c}_{b^{-1}(j)}||^2, \tag{1.6}$$

where P_i is the probability that $\mathbf{x}$ is in S_i. Therefore, a good index assignment mapping is one that minimizes (1.6). In a transmission system modeled by a binary memoryless channel, binary words with a small Hamming distance are more likely to be confused with one another than those with a large Hamming distance. Therefore, intuitively, the index assignment mapping should be selected such that codevectors with a large (small) Euclidean distance are assigned codewords with a large (small) Hamming distance. Various index assignment algorithms are reported in [8], [9], [10]. The general conclusion is that if a VQ is designed for a noiseless channel but is to be used over a noisy channel, a judicious index assignment is important and could result in performance improvements which might be quite significant. The improvement is typically more significant for larger codebooks and for more highly correlated sources. Specific numerical results on Gauss-Markov sources can be found in [9].

1.2.2 Channel-Optimized Vector Quantization

Now we turn our attention to a slightly different problem in which the VQ encoder and decoder are themselves design variables. Specifically, with reference to the diagram of Fig. 1.3, for a given source, a given noisy channel, a fixed dimension k and a fixed codebook size M, we wish to choose $\mathcal{C}$, $\mathcal{P}$ and b in such a way as to minimize $D(\mathcal{P}, \mathcal{C}; b)$. Notice that in the diagram of Fig. 1.3 the decoder output is denoted $\mathbf{c}_{b^{-1}(j)}$. Since now the codebook is a design variable, with a slight abuse of notation, we will write $\mathbf{c}_{b^{-1}(j)}$ as $\mathbf{c}_j$ – the codevector associated with the received index j.

Upon rewriting (1.3) as

$$D(\mathcal{P}, \mathcal{C}; b) = \frac{1}{k} \sum_{i=0}^{M-1} \int_{S_i} p(\mathbf{x}) \{ \sum_{j=0}^{M-1} P(j|b(i)) d(\mathbf{x}, \mathbf{c}_j) \} d\mathbf{x}, \tag{1.7}$$

it becomes clear that for a fixed b, the problem of minimizing the average distortion is identical to the noiseless-channel VQ design problem with a *modified* distortion measure [11] (the term in the braces in (1.7)). Specifically, for a fixed b and a fixed $\mathcal{C}$, the optimum partition $\mathcal{P}^* = \{S_0^*, S_1^*, \ldots, S_{M-1}^*\}$ is such that

$$S_i^* = \{\mathbf{x} : \sum_{j=0}^{M-1} P(j|b(i)) d(\mathbf{x}, \mathbf{c}_j) \ \le \ \sum_{j=0}^{M-1} P(j|b(l)) d(\mathbf{x}, \mathbf{c}_j), \ \forall l\}, \ i \in \mathcal{J}_M. \tag{1.8}$$

We remark here that any change in $b(i)$ in (1.8) will only result in a relabeling of the elements of $\mathcal{P}^*$. In fact, the average distortion obtained after the application of (1.8) will be independent of the index assignment b. Thus, for this design problem, we believe the choice of b is only of limited importance.

Similarly, it is easy to show that for a fixed b and a fixed $\mathcal{P}$, the optimum codebook $\mathcal{C}^* = \{\mathbf{c}_0^*, \mathbf{c}_1^*, \ldots, \mathbf{c}_{M-1}^*\}$ must satisfy

$$\mathbf{c}_j^* = \arg \min_{\mathbf{y} \in \mathbb{R}^k} E\{d(\mathbf{X}, \mathbf{y}) | J = j\}, \ j \in \mathcal{J}_M. \tag{1.9}$$

A successive application of equations (1.8) and (1.9) results in a sequence of encoder-decoder pairs for which the corresponding average distortions form a non-increasing sequence of non-negative numbers which has to converge. Therefore, a straightforward extension of the noiseless-channel VQ design algorithm in [7] can be used for optimizing $\mathcal{P}$ and $\mathcal{C}$. From now on, we will refer to the encoder-decoder pair obtained from this modified algorithm as the channel-optimized VQ (COVQ).

For the squared-error distortion criterion, the optimum partition and the optimum codebook are given, respectively, by

$$S_i^* = \{\mathbf{x} : \sum_{j=0}^{M-1} P(j|b(i)) ||\mathbf{x} - \mathbf{c}_j||^2 \ \le \ \sum_{j=0}^{M-1} P(j|b(l)) ||\mathbf{x} - \mathbf{c}_j||^2, \ \forall l\}, \ i \in \mathcal{J}_M, \tag{1.10}$$

and

$$\mathbf{c}_j^* = \frac{\sum_{i=0}^{M-1} P(j|b(i)) \int_{S_i} \mathbf{x} p(\mathbf{x}) d\mathbf{x}}{\sum_{i=0}^{M-1} P(j|b(i)) \int_{S_i} p(\mathbf{x}) d\mathbf{x}}, \ j \in \mathcal{J}_M. \tag{1.11}$$

In this case, it is established in [12] that the optimum encoding regions are convex polyhedrons and that the encoding complexity is proportional to the number of *nonempty* encoding regions. It is important to note that although the encoder is allowed to have as many as M encoding regions, when the channel noise is high, the optimum system trades quantization accuracy for reduced sensitivity to channel noise by reducing the number of nonempty encoding regions. Generally, the more noisy the channel is, the smaller the number of nonempty encoding regions will be. Assuming that there are N nonempty encoding regions ($N \leq M$), only N codewords need to be transmitted; of course, any one of M binary words may be received and therefore the codebook remains of size M.

1.2.3 Numerical Results

We now present numerical results on the performance of COVQ and make comparison with the Linde, Buzo and Gray VQ (LBGVQ) whose design is based on a noiseless channel assumption [7]. We consider a Gauss-Markov source with correlation coefficients $\rho = 0.9$. The channel is assumed to be a Binary Symmetric Channel (BSC) with bit error rate (BER) ϵ. For $R = 1$ bit/sample, Signal-to-Noise Ratio (SNR) performance results are presented in Table 1.1. The number of encoding regions (as a measure of encoding complexity) for these different cases is included as well.

		$\epsilon = 0.00$	$\epsilon = 0.005$	$\epsilon = 0.01$	$\epsilon = 0.05$	$\epsilon = 0.10$
$k = 1$	LBGVQ	4.40 (2)	4.25 (2)	4.10 (2)	3.09 (2)	2.09 (2)
	COVQ	4.40 (2)	4.25 (2)	4.11 (2)	3.15 (2)	2.27 (2)
$k = 2$	LBGVQ	7.87 (4)	7.31 (4)	6.81 (4)	4.13 (4)	2.19 (4)
	COVQ	7.87 (4)	7.31 (4)	6.83 (4)	4.37 (4)	2.76 (4)
$k = 4$	LBGVQ	10.18 (16)	9.10 (16)	8.24 (16)	4.37 (16)	2.00 (16)
	COVQ	10.18 (16)	9.15 (16)	8.37 (16)	6.23 (11)	4.65 (9)
$k = 8$	LBGVQ	11.49 (256)	9.99 (256)	8.87 (256)	4.46 (256)	2.00 (256)
	COVQ	11.49 (256)	10.31 (249)	9.70 (230)	7.44 (98)	5.73 (61)

TABLE 1.1. SNR (in dB) Performance Results; Gauss-Markov Source; $\rho = 0.9$; $R = 1$ bit/sample; Numbers in Parentheses Indicate Number of Nonempty Encoding Regions.

The assignment of binary codewords to the codevectors of the designed LBGVQ is done via a simulated annealing algorithm described in [9]. The details of the COVQ design algorithm can be found in [12].

The results in Table 1.1 indicate that COVQ performs better than LBGVQ with a good index assignment; the performance improvements are more noticeable for larger dimensions and noisier channels. Also, it is shown in [12] that the improvements are larger for more strongly correlated channels. Note that, in fact, it is for these cases (e.g., $k = 8$, $\rho = 0.9$ and $\epsilon = 0.1$) that the largest reduction in the number of encoding regions (hence, encoding complexity) is observed. Needless to say, if the index assignment of LBGVQ is chosen randomly or inappropriately the performance degradations could be significant; specific numerical results can be found in [9].

The COVQ scheme described in this section is a simple combined source-channel code. The code is designed not to minimize the quantization distortion but to minimize the aggregate distortions caused by quantization and channel noise. To justify our use of the term "combined source-channel code," consider a channel-optimized *scalar quantizer* (i.e., $k = 1$) designed for a memoryless Gaussian source and a BSC with $\epsilon = 0.1$ at 4 bits/sample. Using a design algorithm similar to the one described here, the optimal partition, codeword assignment and codebook are obtained in [14] and summarized in Table 1.2. In this problem $M = 16$ but $N = 6$. That is, there are only 6 nonempty encoding regions (intervals) implying that only 6 codewords are used for transmission. Since the coding rate is 4 bits/sample, 16 codewords are available and by selecting 6 out of 16 possible codewords the encoder effectively increases the *average* Hamming distance between the used codewords. This is exactly what a good channel code attempts to do. In a sense, an operation similar to channel coding is implicitly included in the channel-optimized quantization operation. Since any one of 16 different binary words may be received at the receiver side, the codebook consists of 16 reproduction levels, one for each word.

Quantization Interval	Binary Codeword	Reconstruction Level
$(-\infty, -0.960]$	0011	-1.335
$(-0.960, -0.527]$	0111	-0.817
$(-0.527, -0.181]$	0001	-0.416
$(-0.181, 0.283]$	0000	0.070
$(0.283, 0.837]$	1000	0.576
$(0.837, \infty]$	1100	1.301
	1011*	-1.132
	0010*	-0.603
	1111*	-0.521
	0110*	-0.462
	0101*	-0.418
	1001*	0.158
	1010*	0.416
	0100*	0.648
	1101*	1.072
	1110*	1.156

TABLE 1.2. Encoding Regions, Codevectors and Codewords of a Channel-Optimized Scalar Quantizer; Memoryless Gaussian Source; 4 bits/sample; BSC with $\epsilon = 0.1$. A "*" Indicates Untransmitted Codeword.

One might criticize the comparisons between COVQ and LBGVQ in that the LBGVQ results are all based on a noiseless-channel assumption while the COVQ results are based on the exact knowledge of the channel error rate. If COVQ is a combined source-channel code, it should be compared against appropriately designed tandem source-channel codes. At this point it is not clear which of the two approaches results in a better performance.

Let us concentrate on this issue for a moment. For the channel used in Table 1.2 the capacity can be computed to be 0.53 bits/channel use. Therefore, so long as the coding rate is less than $4 \times 0.53 = 2.12$ bits/sample, an appropriately designed channel code can be found which renders the channel effectively noiseless. In this case the only source of distortion is the quantization noise. Thus, for the example of Table 1.2, a 4-level Lloyd-Max quantizer followed by such a channel code (thus, a tandem code) can be used. The resulting SNR will be 9.30 dB [17]. The SNR associated with the channel-optimized scalar quantizer of Table 1.2, however, is 5.60 dB [14]. Therefore, there exists some tandem source-channel code that outperforms the specific combined source-channel code used here. So, what is the merit of the combined source-channel coding approach? The problem is that in the above tandem scheme, in order to find a channel code to render the channel (approximately) noiseless, very large block sizes are needed. On the other hand, the channel-optimized quantizer uses only a block size of $k = 1$. So, the comparison is unfair in that the encoding delays of the two approaches are very different. Unfortunately, the codebook size and complexity of COVQ grows rapidly with the block size, making it impossible to design large block size COVQs. In the next section we describe a suboptimal, structured, noisy-channel VQ in which larger block sizes are possible and hence appropriate comparisons against tandem codes can be made.

1.3 Multi-Stage Vector Quantization over Noisy Channels

1.3.1 Problem Formulation and Design Algorithm

A multi-stage VQ (MSVQ) performs the quantization operation based on a "successive approximation" approach. Instead of using a "full-rate" (M-level) VQ, the source vector, $\mathbf{x}$, is first approximated by a lower-rate (M_1-level) VQ with partition $\mathcal{P}_1$, codebook $\mathcal{C}_1$ and quantization rule $q_1 = \beta_1 \circ \alpha_1$ ($M_1 < M$). The quantization error of this stage, $\mathbf{e}_1 = \mathbf{x} - q_1(\mathbf{x})$, is then quantized by a second-stage (M_2-level) VQ with corresponding $\mathcal{P}_2, \mathcal{C}_2$ and $q_2 = \beta_2 \circ \alpha_2$, [15], [16]. The quantization error of the second stage, $\mathbf{e}_2 = \mathbf{e}_1 - q_2(\mathbf{e}_1)$, can be further quantized by a third-stage VQ, and so on. In general, an MSVQ can have S stages, with each stage having an intermediate rate $R_s = \frac{1}{k}\log_2 N_s$, a partition $\mathcal{P}_s$, a codebook $\mathcal{C}_s$ and a quantization rule $q_s = \beta_s \circ \alpha_s$ for $s = 1, 2, \ldots, S$. The overall rate of the quantizer is $R = R_1 + R_2 + \cdots + R_S$. We denote the overall encoder and decoder by α and β, respectively. The overall quantization rule is given by $q(\mathbf{x}) = \beta(\alpha(\mathbf{x})) = \sum_{s=1}^{S} q_s(\mathbf{e}_{s-1})$, with $\mathbf{e}_0 = \mathbf{x}$.

In the rate-distortion sense, the performance of MSVQ is inferior to that of LBGVQ [15]. The computational complexity of MSVQ however can be significantly less than that of LBGVQ.

Assuming that the encoder α and the decoder β are separated by a DMC characterized by the random mapping $\gamma : \mathcal{J}_M \mapsto \mathcal{J}_M$, the reconstructed vector is no longer $q(\mathbf{x}) = \beta(\alpha(\mathbf{x}))$ but is a composition of the three mappings: $q(\mathbf{x}) = \beta(\gamma(\alpha(\mathbf{x})))$.

We now proceed to describe an algorithm for designing an MSVQ-based scheme for a given noisy channel. The resulting scheme is thus called channel-matched MSVQ (CM-MSVQ). We avoid the term "channel-optimized" since, by its definition MSVQ is sub-optimal.

A. Problem Statement

Consider the block diagram of Figure 1.4. The first-stage (or primary) encoder, α_1, is described in terms of a partition $\mathcal{P}_1 = \{S_{1,0}, S_{1,1}, \ldots, S_{1,M_1-1}\}$ according to $\alpha_1(\mathbf{x}) = i$, if $\mathbf{x} \in S_{1,i}$. The output of this encoder is transmitted over a DMC described by the random mapping $\gamma_1 : \mathcal{J}_{M_1} \mapsto \mathcal{J}_{M_1}$ and the transition probability $Q_1(j|i) = \Pr\{\gamma_1(i) = j\}$. The primary decoder, β_1, is given in terms of the reproduction codebook, $\mathcal{C}_1 = \{\mathbf{c}_{1,0}, \mathbf{c}_{1,1}, \ldots, \mathbf{c}_{1,M_1-1}\}$ according to $\mathbf{z} \equiv \beta_1(j) = \mathbf{c}_{1,j}$.

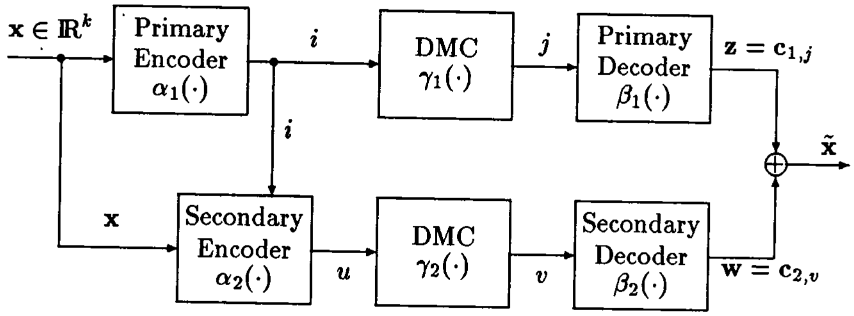

FIGURE 1.4. Block Diagram of the MSVQ Scheme Over Noisy Channels.

Here the source vector, $\mathbf{x}$, is directly sent to the second-stage (secondary) encoder. This is different from the noiseless-channel case, where the input to the second-stage encoder is the coding error, $\mathbf{x} - \mathbf{z}$, of the first stage. Since the channel here is noisy, the value of j and hence $\mathbf{z}$ is not known at the transmitter. For now, let us assume that the secondary encoder is the mapping $\alpha_2 : \mathbb{R}^k \times \mathcal{J}_{M_1} \mapsto \mathcal{J}_{M_2}$, described by

$$\alpha_2(\mathbf{x}, i) = u \quad \text{if } \mathbf{x} \in S_i^u, \quad i \in \mathcal{J}_{M_1}, u \in \mathcal{J}_{M_2}, \tag{1.12}$$

where $S_i^u \triangleq S_{1,i} \cap S_{2,u}$ and $\mathcal{P}_2 = \{S_{2,0}, S_{2,1}, \ldots, S_{2,M_2-1}\}$ is another partition of $\mathbb{R}^k$. The output of this encoder is transmitted over a second DMC described by $\gamma_2 : \mathcal{J}_{M_2} \mapsto \mathcal{J}_{M_2}$ and $Q_2(v|u) = \Pr\{\gamma_2(u) = v\}$. In reality, the outputs of the primary and secondary encoders are multiplexed and transmitted over a single DMC.

The secondary decoder, β_2, depends on the output of the second channel, v, and is described by the codebook, $\mathcal{C}_2 = \{\mathbf{c}_{2,0}, \mathbf{c}_{2,1}, \ldots, \mathbf{c}_{2,M_2-1}\}$, according to $\mathbf{w} \equiv \beta_2(v) = \mathbf{c}_{2,v}$. The reconstructed vector is $\tilde{\mathbf{x}} = \mathbf{z} + \mathbf{w}$.

B. Necessary Conditions

Assuming that the primary encoder and decoder are given and fixed, our problem is to minimize the average distortion $D = \frac{1}{k}E[d(\mathbf{X}, \tilde{\mathbf{X}})]$ by appropriate design of $\mathcal{P}_2$ and $\mathcal{C}_2$. It is easy to show that the average distortion is given by

$$D = \frac{1}{k} \sum_{i,u} \int_{S_i^u} \left\{ \sum_{j,v} Q_1(j|i) Q_2(v|u) d(\mathbf{x}, \mathbf{c}_{1,j} + \mathbf{c}_{2,v}) \right\} p(\mathbf{x}) d\mathbf{x}. \tag{1.13}$$

The term in the braces is defined as the *modified* distortion measure [11], [14],

$$d'(\mathbf{x}; i, u) \triangleq \sum_{j,v} Q_1(j|i) Q_2(v|u) d(\mathbf{x}, \mathbf{c}_{1,j} + \mathbf{c}_{2,v}), \tag{1.14}$$

which can be interpreted as the expected distortion between $\mathbf{x}$ and $\tilde{\mathbf{X}}$ given that i and u are transmitted. With this modified distortion measure, it is clear that for a fixed codebook, the optimum partition, $\mathcal{P}_2^*$, must satisfy

$$S_{2,u}^* = \{\mathbf{x} : d'(\mathbf{x}; \alpha_1(\mathbf{x}), u) \le d'(\mathbf{x}; \alpha_1(\mathbf{x}), u') \quad \forall u' \in \mathcal{J}_{M_2}\}, \quad u \in \mathcal{J}_{M_2}. \tag{1.15}$$

Similarly, for a fixed partition, the optimum codebook, $\mathcal{C}_2^*$ is given by

$$\begin{aligned} \mathbf{c}_{2,v}^* &= \arg \min_{\mathbf{w} \in \mathbb{R}^k} E[d(\mathbf{X}, \tilde{\mathbf{X}})|v] \\ &= \arg \min_{\mathbf{w} \in \mathbb{R}^k} E[d(\mathbf{X}, \mathbf{Z} + \mathbf{w})|v], \qquad v \in \mathcal{J}_{M_2}. \end{aligned} \tag{1.16}$$

For the squared-error distortion measure, the modified distortion of (1.14) can be expressed as:

$$d'(\mathbf{x}; i, u) = ||\mathbf{x}||^2 - 2\langle \mathbf{x}, \mathbf{y}_{1,i} \rangle + \delta_{1,i} + \delta_{2,i,u} - 2\langle \mathbf{x} - \mathbf{y}_{1,i}, \mathbf{y}_{2,i,u} \rangle, \tag{1.17}$$

where $\mathbf{y}_{1,i} \triangleq \sum_j Q_1(j|i)\mathbf{c}_{1,j}$, $\mathbf{y}_{2,u} \triangleq \sum_v Q_2(v|u)\mathbf{c}_{2,v}$, $\delta_{1,i} \triangleq \sum_j Q_1(j|i)||\mathbf{c}_{1,j}||^2$ and $\delta_{2,u} \triangleq \sum_v Q_2(v|u)||\mathbf{c}_{2,v}||^2$. The sum of the first three terms in (1.17) is defined as $d_1(\mathbf{x}; i)$ which can be viewed as the expected distortion of the first stage in encoding $\mathbf{x}$ given that $\mathbf{x}$ is in $S_{1,i}$, where the expectation is taken over all possible channel transitions. The sum of the last two terms is defined as $d_2(\mathbf{x}; i, u)$, where $-d_2(\mathbf{x}; i, u)$ can be regarded as the amount of distortion reduction associated with the second stage given that $\mathbf{x}$ is in S_i^u.

The distortion d_2 corresponds exactly to the distortion measure used in the COVQ codebook search with $(\mathbf{x} - \mathbf{y}_{1,i})$ replacing $\mathbf{x}$. Note that $\mathbf{y}_{1,i}$ is the expected value of $\mathbf{Z}$ (the output of the primary decoder) given that i is transmitted. Thus, $(\mathbf{x} - \mathbf{y}_{1,i})$, the vector which is encoded by the secondary encoder, is nothing but the *expected* coding error of the first stage. This is in contrast with the ordinary MSVQ in which the second stage is designed to be optimal for the *actual* error of the first stage.

Finally, for the squared-error distortion measure, it can be readily shown that the optimum codebook of (1.16) reduces to

$$\mathbf{c}_{2,v}^* = \frac{\sum_{i,u} Q_2(v|u) \int_{S_i^u} (\mathbf{x} - \mathbf{y}_{1,i}) p(\mathbf{x}) d\mathbf{x}}{\sum_{i,u} Q_2(v|u) \int_{S_i^u} p(\mathbf{x}) d\mathbf{x}}, \quad v \in \mathcal{J}_{M_2}. \tag{1.18}$$

The above necessary conditions can be used to develop a channel-matched MSVQ design algorithm in which upon designing and fixing the first stage the second stage is obtained using the above necessary conditions. The details of the design algorithm including extension to S stages are omitted here but can be found in [18].

1.3.2 Comparisons Against a Tandem Source-Channel Code

Results on the performance of CM-MSVQ over BSC with different bit error rates are presented in [18]. The general conclusion is that when the channel is very noisy the channel-matched schemes outperform the ordinary MSVQ schemes with the same set of parameters. For the Gauss-Markov source (with $\rho = 0.9$), the performance of the channel-matched schemes tend to increase monotonically with k. Since the complexity of CM-MSVQ is smaller than that of COVQ of the same rate and dimension, for the same level of complexity larger dimensions are possible for CM-MSVQ. This, therefore, allows us to make a meaningful comparison between the CM-MSVQ (a combined source-channel code) and a tandem source-channel code with the same effective block size (hence, encoding delay).

The block diagram of a tandem source-channel code is depicted in Figure 1.5. Here, we assume that the source code is an LBGVQ with block size k_s and rate $R_s = m/k_s$ bits/sample. The source encoder thus produces m bits for every k_s source samples. The channel code is assumed to be a (q, lm) *linear block* code. Hence, the channel encoder accepts lm information bits and produces q bits for transmission ($q \geq lm$). That is, the channel encoder takes l successive m-bit codewords generated by the source encoder and produces a q-bit codeword. The overall rate of the tandem scheme is $R = q/(lk_s)$ transmitted bits per source sample and the effective block size (or delay) is $k = lk_s$ source samples. We have considered LBGVQ and the linear block code because they are both among the most well-known and conceptually simple block-structured source and channel codes, respectively.

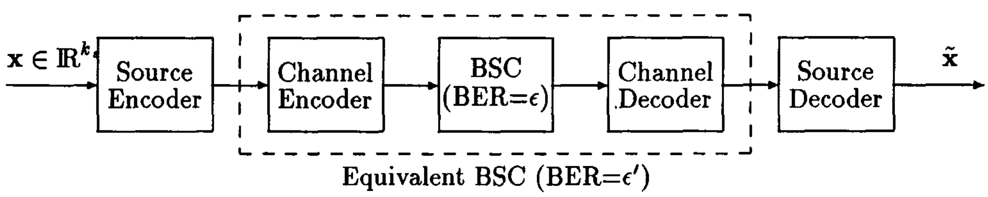

FIGURE 1.5. Block Diagram of a Tandem Source-Channel Coding Scheme Over a BSC and the Equivalent Channel Model.

From now on we will use (n', k') to denote (q, lm). The channel is assumed to be a BSC. For an (n', k') linear block code operating over a BSC with BER ϵ, the probability of correct decoding is upper bounded by

$$P_c = \sum_{i=0}^{t} \binom{n'}{i} \epsilon^i (1-\epsilon)^{n'-i} + A_t \epsilon^{t+1} (1-\epsilon)^{n'-t-1}, \tag{1.19}$$

where t is the greatest integer such that $A_t \triangleq 2^{n'-k'} - \sum_{i=0}^{t}(n'i) \geq 0$. This is known as the sphere-packing bound and is achievable if and only if the (n', k') code is quasi-perfect [6].

To analyze the performance of the tandem scheme, we assume that the channel code is quasi-perfect and model the channel encoder, the BSC and the channel decoder as an *equivalent BSC* (the dashed box in Figure 1.5). This assumption leads to a simple analysis of the tandem scheme. Here, we assume that, in the equivalent channel, the probability that a block of k' consecutive bits are received without error is equal to P_c. Thus the BER of this channel can be computed as

$$\epsilon' = 1 - P_c^{1/k'}. \tag{1.20}$$

With this equivalent model, the analysis for the tandem source-channel coding scheme reduces to that of LBGVQ operating over a BSC with BER ϵ' [13],[12].

Using the equivalent model, we have evaluated the performance of the tandem scheme for $k_s = 6$ and $l = 4$. These results (for the Gauss-Markov source with $\rho = 0.9$) are plotted in Figure 1.6. For the LBGVQ of the tandem scheme, a simulated annealing algorithm [9] was used for index assignment. We have chosen $m = 3, 4, 5$ and 6 (corresponding to $R_s = 0.5, 0.67, 0.83$ and 1.0). In all cases, we have chosen $q = lk_s$ so that the overall rate is always 1 bit/sample. We have also plotted in this graph the optimum performance theoretically attainable (OPTA) obtained by equating the rate-distortion function to the channel capacity [19] and the performances of the CM-MSVQ ($M = 1$) and the multiple candidate CM-MSVQ ($M = 8$) [18] with the same k as that of the tandem scheme (thus, similar encoding delays). In this graph, the performance of the multiple candidate CM-MSVQ is always better than the tandem scheme. It is clear that when the channel is relatively noise-free, all of the available bits should be allocated to source coding. As the channel becomes noisier, more and more bits should be allocated to channel coding. In any case, even with an optimum allocation of bits between the source and channel codes, the tandem scheme is always inferior to CM-MSVQ for the cases considered.

An important point to make is that CM-MSVQ is a suboptimal combined source-channel coding scheme, whereas the tandem scheme consists of a source code which, by itself, is optimum for its block size and a channel code which, by itself, is optimum (according to the equivalent model) for its block size.

We should make a note that our analysis of the tandem scheme was somewhat optimistic. First, a quasi-perfect code does not exist for all values of (n', k') [6]. Secondly, even if a quasi-perfect code does exist for (n', k'), there is still the question about the validity of the equivalent BSC model and its implications on the performance analysis. We conjecture that our analysis of the tandem scheme *overestimates* its actual performance. To support this conjecture, we have simulated the tandem scheme with $R_s = 0.5$ using the (24,12) extended Golay code [6]. These simulation results, which are reported in [18], establish that the simulation results coincide with the analytical results when the channel is relatively clean, but that the analytical results significantly overestimate the actual performance for very noisy channels. At $\log_{10} \epsilon = -1$, they differ by more than 3 dB. This observation further supports the usefulness of the CM-MSVQ scheme as a combined source-channel code. Additional results on mismatch and complexity issues in CM-MSVQ are provided in [18]. An application of the CM-MSVQ approach to coding of speech line spectrum pair (LSP) parameters over noisy channels is presented in [20] in which the superior performance of CM-MSVQ to a tandem code is demonstrated.

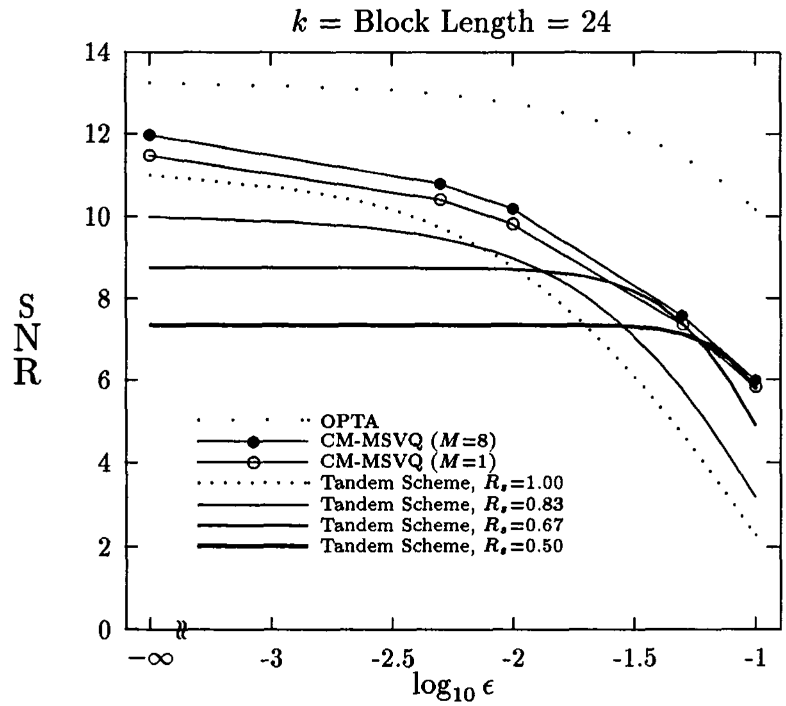

FIGURE 1.6. SNR (in dB) Performances of the Tandem Source-Channel Coding Scheme and the CM-MSVQ Scheme for the Gauss-Markov Source with Correlation Coefficient 0.9; Overall Rate = 1 Bit/Sample; M = Number of Candidates of CM-MSVQ; R_s = Rate of Source Code of Tandem Scheme.

1.4 Finite-State Vector Quantization over Noisy Channels

In some practical applications where there is a large correlation between the source output vectors, increasing the VQ block size or encoding multiple vectors simultaneously could potentially lead to performance improvements. A good example of such a situation is in coding of speech LSP parameters where there is a significant inter-vector correlation in addition to the intra-vector correlation [24]. As mentioned before, increasing the block size of COVQ results in an increase in encoding complexity which quickly becomes prohibitive. For CM-MSVQ, larger block sizes are possible, but for a given encoding rate, this comes with an increase in the number of stages and therefore further suboptimality.

An alternative to increasing the block size in these situations is to incorporate some type of feedback mechanism in the code that utilizes the source inter-vector memory. One such technique is finite-state vector quantization (FSVQ) [4], [15], [21], [22]. An

FSVQ is a finite-state machine with one VQ associated with each state; the encoder and the decoder share the same state space and the decoder can track the encoder state sequence based on the received encoder output sequence.

Let us be more precise. A k-dimensional K-state code is specified by a state space $\mathcal{S} \equiv \mathcal{J}_K$, an encoder mapping $\alpha : \mathbb{R}^k \times \mathcal{S} \to \mathcal{J}_M$, a decoder mapping β: $\mathcal{J}_M \times \mathcal{S} \to \hat{\mathcal{A}}$ and a next state function f: $\mathcal{J}_M \times \mathcal{S} \to \mathcal{S}$; where $\hat{\mathcal{A}}$ is the reproduction space.

Let $\{\mathbf{x}_n\}_{n=0}^{\infty}$ denote the input vector sequence, where $\mathbf{x}_n \in \mathbb{R}^k$. Similarly, let $\{i_n\}_{n=0}^{\infty}$, $\{s_n\}_{n=0}^{\infty}$ and $\{\hat{\mathbf{x}}_n\}_{n=0}^{\infty}$ denote the channel symbol sequence, state sequence and reproduction vector sequence, respectively. Given an initial state s_0, the input sequence determines the sequence of channel symbols, reproduction vectors and states according to:

$$i_n = \alpha(\mathbf{x}_n, s_n), \tag{1.21}$$

$$\hat{\mathbf{x}}_n = \beta(i_n, s_n), \tag{1.22}$$

$$s_{n+1} = f(i_n, s_n), \quad n = 0, 1, \ldots. \tag{1.23}$$

The next state depends only on the present state and the output channel symbol; therefore, given the initial state and correct channel symbol sequence, the decoder can track the state sequence. Here, $\mathcal{C}_s \equiv \{\beta(i,s), i \in \mathcal{J}_M\}$ is the codebook associated with state s and $\hat{\mathcal{A}} = \bigcup_{s=0}^{K-1} \mathcal{C}_s$. As defined in [21], an FSVQ is a finite-state code with α given by the minimum distortion rule

$$\alpha(\mathbf{x}, s) = \arg\min_{i \in \mathcal{N}} d(\mathbf{x}, \beta(i,s)), \; s \in \mathcal{S}. \tag{1.24}$$

The average distortion incurred in an FSVQ system is given by $\frac{1}{L}E[d(\mathbf{X}, \hat{\mathbf{X}})]$, where the expectation is taken with respect to the source distribution. The rate is given by $R = \frac{1}{k}\log_2 M$, bits/sample. Details of FSVQ design can be found in [4], [21], [22].

Clearly, the ability of the decoder to track the encoder state sequence *critically* depends on the availability of the *exact* replica of the transmitted codewords. Even a single error occurring in the transmitted codeword can lead to an incorrect decoder state. Once the decoder state is different from the encoder state, the decoder state sequence can remain "derailed" for a long time. For the sake of this discussion, consider an example with a Gauss-Markov source as before, encoded by LBGVQ (with an appropriate index assignment), COVQ and FSVQ (as described in [21]) all operating over a BSC with BER ϵ. The SNR performance results for $k = 4$ and $R = 1$ bit/sample are tabulated in Table 1.3. Clearly, for a clean channel FSVQ gives the best SNR as it can utilize the inter-vector correlation in the source; for noisy channels however the error propagation (caused by incorrect tracking of the encoder state sequence) eclipses this advantage of FSVQ and leads to a severe degradation in performance.

At this point it is natural to ask whether it is possible to design a *finite-state* combined source-channel code along the lines of FSVQ while avoiding (or minimizing) the state derailing problem. Such a system can exploit the inter-vector correlation of the source vectors without introducing excessive sensitivity to channel errors. In what follows, we outline the basic ideas behind such a code, hereafter referred to as the channel-matched FSVQ (CM-FSVQ); the details can be found in [23].

The basic idea behind CM-FSVQ consists of two components. First, for $M > K$, instead of a general next-state function it has a structured next-state function where

Coder	$\epsilon = 0.000$	$\epsilon = 0.005$	$\epsilon = 0.010$	$\epsilon = 0.050$	$\epsilon = 0.100$
LBGVQ	10.18	9.10	8.24	4.37	2.00
COVQ	10.18	9.15	8.37	6.23	4.65
FSVQ	11.31	4.30	1.98	-1.67	-2.53
CM-FSVQ	10.84	10.04	9.23	7.26	5.55

TABLE 1.3. SNR Performance of LBGVQ, COVQ, FSVQ and CM-FSVQ over a Binary Symmetric Channel; Gauss-Markov Source ($\rho = 0.9$); $k = 4$; $K = 8$; 1 bit/sample.

the next-state s_{n+1} is given by $f(i_n)$. That is, for any state, the selected index (codeword) contains the information of the next state as part of the codeword. Imposing this structure on the next-state function, of course, leads to some performance degradation for noiseless channels [23]. In the presence of channel noise, however, this next-state function significantly reduces the code's sensitivity to channel noise. This is because the state information is automatically corrected as soon as a correct codeword is received. Second, this structured next-state function can be used to design a CM-FSVQ scheme in which the state VQs are essentially COVQs as described in Section 1.2. We refer the interested reader to [23] for the details of CM-FSVQ. However, to complete our example, we have included in Table 1.3 the SNR results of CM-FSVQ. Clearly, CM-FSVQ outperforms COVQ in all cases. This is done by exploiting the source inter-vector correlation. Needless to say, CM-FSVQ is more effective in applications where there is significant inter-vector correlation and therefore it might be a good candidate for coding of speech spectral parameters over noisy channels.

1.5 Coding of Speech LSP Parameters over Noisy Channels

We opened this chapter by a question about the possible application of combined source-channel codes instead of tandem codes in two-way speech communication over noisy channels. As a concrete example, we have studied the coding of LSP parameters (used in the U.S. Government Standard FS1016 Codebook Excited Linear Predictive (CELP) Coder [25] to represent the short-term spectrum) using COVQ, CM-FSVQ [23] and CM-MSVQ [18], [20]. First, we considered a CM-MSVQ based system designed based on certain assumptions on the channel noise and then applied it to a variety of channels. This encoder, whose performance in terms of average spectral distortion vs. bit rate (per 10 LSP parameters) is depicted in Fig. 1.7, is shown to perform consistently better than a judiciously designed tandem source-channel code [20]. This, by itself, supports the usefulness of the combined source-channel coding approach in a practical situation. Furthermore, for comparison, we have simulated the performance of COVQ and CM-FSVQ applied to coding of LSP parameters over the same noisy channels. In this case, following Paliwal and Atal's approach [24], to reduce the design and encoding complexity of the VQs, the LSP vectors are split into three subvectors; each subvector is encoded by a COVQ or a CM-FSVQ, separately. These results are also included in Fig. 1.7. Note that the COVQ and CM-FSVQ scheme are designed separately for each value of channel noise ϵ, whereas the CM-MSVQ results are all based on one design. Our general conclusion from these results is that the CM-FSVQ scheme gives the best results and provides an attractive alternative to tandem source-channel coding strategies for speech LSP parameters over noisy channels. Our informal listening tests on a CELP-type coder also confirm the superiority of the CM-FSVQ scheme for encoding the LSP parameters over noisy channels. We have conducted similar experiments on a Rayleigh fading channel and observed the same general trend. We have not yet considered the application of the proposed combined source-channel codes to encoding of other CELP parameters such as the pitch, gain, and stochastic codebook indices. These issues are currently being investigated.

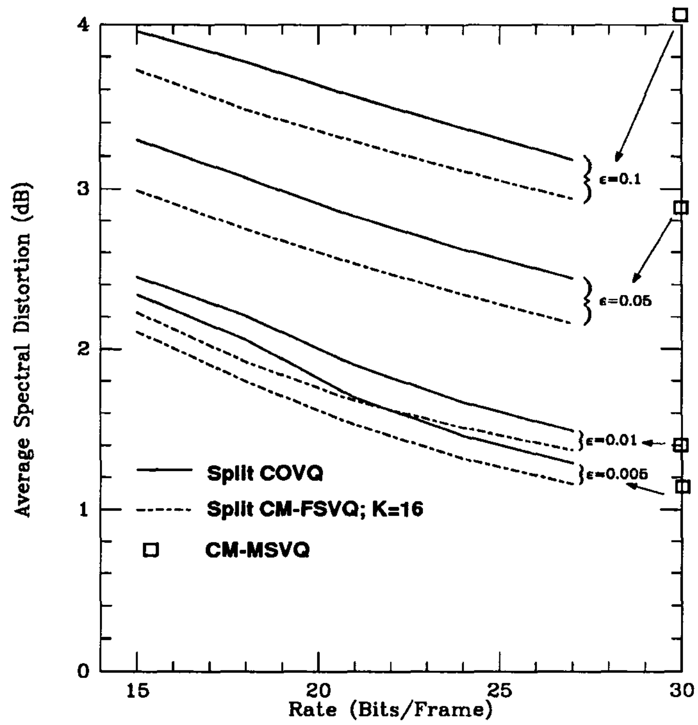

FIGURE 1.7. Performance of Split COVQ, Split CM-FSVQ and CM-MSVQ in coding Speech LSP Parameters over a Noisy Channel.

1.6 Concluding Remarks

In this chapter, we have provided a discussion of quantization for noisy channels and presented methodologies for noisy-channel full-searched, multi-stage and finite-state vector quantizer design. These noisy-channel quantization schemes, which constitute examples of combined source-channel codes, are compared against tandem source-channel codes. Our main conclusion is that in practical situations were due to delay and complexity constraints very large block sizes are not possible, the traditional block-structured tandem source-channel codes may be far from optimal and combined source-channel codes might therefore be preferable. Speech coding in wireless environments is a communication problem which might benefit from this combined approach to source-channel coding.

There remain many interesting open problems. Examples are: (i) Integration of modulation with source-channel coding, (ii) design of combined source-channel codes for more complicated (but realistic) channels, such as channels with memory and (iii) design of sliding block source-channel codes.

1.7 Acknowledgement

I would like to express my thanks to Vinay Vaishampayan, Nam Phamdo, Yunus Hussain and Takehiro Moriya for their help at various stages of this work.

1.8 References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.,* vol. 27, pp. 379-423 and 623-656, 1948.

[2] C. E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion," *IRE Nat. Conv. Rec.,* pp. 142-163, March 1959.

[3] N. S. Jayant and P. Noll, *Digital Coding of Waveforms, Principles and Applications to Speech and Video*, Englewood Cliff, NJ: Prentice-Hall, 1984.

[4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston, MA: Kluwer Academic Publishers, 1992.

[5] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[6] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes.* Cambridge, MA: The MIT Press, 1972.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization Design," *IEEE Trans. Commun.*, vol. 28, pp. 84-95, Dec. 1980.

[8] J. R. B. De Marca and N. S. Jayant, "An Algorithm for Assigning Binary Indices to the Codevectors of a Multidimensional Quantizer," *Proc. IEEE Int. Comm. Conf.*, Seattle, WA, pp. 1128-1132, June 1987.

[9] N. Farvardin, "A Study of Vector Quantization for Noisy Channels," *IEEE Trans. Inform. Theory,* vol. 36, pp. 799-809, July 1990.

[10] K. Zeger and A. Gersho, "Pseudo-Gray Coding," *IEEE Trans. Commun.*, vol. 38, pp. 2147-2158, Dec. 1990.

[11] E. Ayanoglu and R. M. Gray, "The Design of Joint Source and Channel Trellis Waveform Coders," *IEEE Trans. Inform. Theory,* vol. 33, pp. 855-865, Nov. 1987.

[12] N. Farvardin and V. Vaishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Trans. Inform. Theory,* vol. 37, pp. 155-160, Jan. 1991.

[13] H. Kumazawa, M. Kasahara and T. Namekawa, "A Construction of Vector Quantizers for Noisy Channels," *Electronics and Engineering in Japan,* vol. 67-B, pp. 39-47, 1984.

[14] N. Farvardin and V. Vaishampayan, "Optimal Quantizer Design for Noisy Channels: An Approach to Combined Source-Channel Coding," *IEEE Trans. Inform. Theory,* vol. 33, pp. 827-838, Nov. 1987.

[15] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4-29, April 1984.

[16] B. H. Juang and A. H. Gray, Jr., "Multiple Stage Vector Quantization for Speech Coding," *Proc. ICASSP-82*, pp. 597-600, April 1982.

[17] J. Max, "Quantizing for Minimum Distortion, "*IRE Trans. Inform Theory,*" vol. IT-6, pp. 7-12, March 1960.

[18] N. Phamdo, N. Farvardin, and T. Moriya, "A Unified Approach to Tree-Structured and Multi-Stage Vector Quantization for Noisy Channels," *IEEE Transactions on Inform. Theory*, vol. 39, pp. 835–850, May 1993.

[19] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[20] N. Phamdo, N. Farvardin and T. Moriya, "Combined Source-Channel Coding of LSP Parameters Using Multi-Stage Vector Quantization," in *Speech and Audio Coding for Wireless and Network Applications,* ed. B. S. Atal, V. Cuperman and A. Gersho, Boston, MA: Kluwer Academic Publishers, pp. 181-190, 1993.

[21] J. Foster, R. M. Gray and M. O. Dunham, "Finite-State Vector Quantization for Waveform Coding," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 348-359, May 1985.

[22] M. O. Dunham and R. M. Gray, "An Algorithm for the Design of Labeled-Transition Finite-State Vector Quantizers," *IEEE Trans. Commun.*, vol. COM-33, pp. 83-89, Jan. 1985.

[23] Y. Hussain and N. Farvardin, "Finite-State Vector Quantization for Noisy Channels," *IEEE Trans. Speech Proc.*, submitted for publication, 1993.

[24] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *IEEE Trans. Speech Proc.*, vol. 1, pp. 3-14, Jan. 1993.

[25] J. Campbell, T. Tremain and V. Welch, "The DoD 4.8 kbps Standard (Proposed Federal Standard 1016)," in *Advances in Speech Coding*, ed. B. S. Atal, V. Cuperman and A. Gersho, Boston, MA: Kluwer Academic Publishers, pp. 121-134, 1991.