Soft-Decision COVQ for M-ary PAM Modulated AWGN and Rayleigh Fading Channels

by

Cynthia E. Thomas

A thesis submitted to the
Department of Mathematics and Statistics
in conformity with the requirements for
the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada
October, 2005

Abstract

Developments in vector quantization based joint source-channel coding have produced codes which efficiently and reliably transmit data signals over noisy channels. Further advancements through soft-decision decoding have shown improvements in signal-to-distortion ratio (SDR) over hard-decoding. We present a q-bit soft decision demodulator for the vector quantization of Gaussian and Gauss-Markov sources over M-Pulse Amplitude Modulated (M-PAM) additive white Gaussian noise (AWGN) channels and Rayleigh fading channels. Observations are made on the transition probability matrix of the equivalent channel corresponding to the concatenation of the modulator, physical channel and soft-decision demodulator. Gains of almost 0.5 and 0.45 dB in SDR are noted for the AWGN channel and the Rayleigh fading channel, respectively. We observe that most of the gain is achieved by increasing the demodulator resolution from $q = 1$ to $q = 2$, and that the gain is negligible for $q > 2$. Our system has a higher storage requirement at the decoder to store codewords. The system's decoding complexity is dependent on q but is still lower than other soft decoding schemes.

Acknowledgments

I thank my supervisors, Dr. Fady Alajaji and Dr. Tamás Linder, for their insight, guidance, and patience during the course of the research. Thank you for your supervision and careful proof-readings of this thesis. Dr. Alajaji: your enthusiasm is contagious and your listening ear was greatly appreciated. Thank you also for pulling this together at your busiest time. Dr. Linder: thank you for your motivating words in a time of need. Thank you for your help in a difficult time.

I thank Dr. Firouz Behnamfar for his numerous teaching and counseling sessions. Firouz, your continual support was pivotal for the completion of this thesis and your time and mentorship is immensely appreciated.

Many thanks to colleagues in the communications lab, fellow dwellers of Jeffrey Hall, and friends outside the department for all their support. Thanks to Dr. David Thomson for allowing me to use the Solar Lab processors for running code. Thanks to Jennifer Read of the Math & Stats Dept. for her support over the past few years and with smoothing out the final details. Also, thanks to Rose Silva in the School of Graduate Studies for her abundant generosity and cooperation.

I am grateful for financial support from the following entities: Dr. Alajaji,

Dr. Linder, the Department of Mathematics & Statistics, and the Dean of the School of Graduate Studies & Research's Canadian Visible Minority Award and R. S. McLaughlin Fellowship.

Last, but not least, thanks to my family. Abajan and Ma, thanks for keeping me in your thoughts and for unconditionally supporting me. To my sister Sal (chechi): I could not have been without your empathy and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In modern-day communications, typical problems faced by communication engineers involve data compression and signal protection against noise. The goals of an engineer are to minimize the overall distortion and maximize the performance of a system, all the while negotiating complexity and storage. Finding a reliable means of sending data whilst remaining efficient continues to be the motivation of many researchers. Data compression allows information to be represented efficiently, beneficial to both the transmission and storage of the data. The transformation of data into an efficient form by removing redundancies is called *source coding*. The removal of such redundancies can cause the data to become more vulnerable to channel noise or storage device errors. *Channel coding* aims to make signals robust, often by adding controlled redundancy.

1.1 Literature Review

Typically, data is passed through a source encoder followed by channel encoder, to apply source and channel coding, respectively. The independent and separate design and treatment of data in preparation for transmission is called a tandem system. Shannon's separation principle justifies the optimality of a tandem system through the following theorems. Shannon's source coding theorem says that allowing distortion D, the minimum rate needed to represent the source is $R(D)$ bits per sample, also known as the rate-distortion function. Shannon's channel coding theorem states that if the transmission rate R is less than the capacity of the channel C, then there exists a channel code with this rate whose probability of error approaches zero for a large enough block length n. On the other hand, if $R > C$, the probability of error is bounded away from zero for any channel code [9]. Unfortunately, optimality comes at the cost of very large (unbounded) block length, and in turn, very large (unbounded) encoding/decoding delay.

In practical systems where delay and complexity are limited, joint source-channel coding outperforms tandem coding systems, especially vector quantizers designed for noisy channels [25]. Combining source and channel coding may also be simpler for design and implementation [3].

The development of channel optimized vector quantization (COVQ) started with the necessary conditions for minimizing the squared-error distortion measure for scalar quantizers, which were established by Lloyd [22] and Max [24]. The design of quantizers was extended to k-dimensional vectors, known as vector quantizers [23]. Kurten-

bach and Wintz incorporated channel noise into the design of scalar quantizers in [20]. This work was extended to vector quantizers by Kumazawa et al. [19]. This design scheme eventually became known as COVQ. Farvardin and Vaishampayan [14], who studied the complexity of COVQ, found that the encoding complexity is proportional to the number of encoding regions, which is lower than the codebook size for noisy channels.

Recent works by Alajaji and Phamdo [1], [25] presented a soft-decision COVQ system which shows performance gain over hard-decision decoding (basic COVQ). The scheme has a higher encoding complexity than COVQ, and higher storage requirement. On the other hand, their scheme has a lower encoding complexity compared to other soft-decision schemes such as the Hadamard-based soft decoding of Skoglund [28]. Skoglund [30] also presented a suboptimal soft decoding scheme which lowers encoding/decoding complexity.

1.2 Contributions

The contributions of this thesis are as follows:

1. A soft-decision COVQ system is designed for both M-ary Pulse Amplitude Modulated (M-PAM) additive white Gaussian noise (AWGN) and Rayleigh fading channels. The work of Alajaji and Phamdo in [1] and [25] is extended to include any number of constellation signals $M = 2^h, h \in \mathbb{N}$, in one dimension. The design includes the channel transition probability matrix, contingent on the number of M-ary PAM signals, channel model, and soft-decision demodulator.

2. The numerical results of the soft-decision COVQ system were obtained and compared to hard-decision decoding (basic COVQ). Results in the form of capacity values and signal-to-distortion ratio (SDR) are presented for varying constellation sizes, source vector dimension k, and COVQ rate r, for both AWGN and Rayleigh fading channels. Results for both memoryless Gaussian and Gauss-Markov memory sources are presented.

1.3 Thesis Outline

The following thesis chapters are organized as follows. In Chapter 2, we begin by stating source and channel models to be used in our system design. We continue by introducing some terminology and concepts from information theory. We discuss quantizers and recall their necessary conditions for optimality. Additionally in Chapter 3, we study COVQ and techniques for codebook initialization. Previously studied soft-decision decoding algorithms are presented, followed by a summary of Blahut's algorithm for calculating capacity. In Chapter 4, we present the design of our soft-decision COVQ system for M-ary PAM modulated channels, including the channel transition probability matrix for both AWGN and Rayleigh fading channels. We present the numerical results of our soft-decision COVQ system and discuss their implications in Chapter 5. Capacity values of our system are tabulated against hard-decision COVQ values. Performances measured in SDR are plotted to show gains in our soft-decision design over hard-decision COVQ. Finally, we conclude the thesis with closing remarks in Chapter 6.

Chapter 2

Background

2.1 Source and Channel Models

A source is thought to be the producer of signals which may be in the form of a sequence or waveform. A discrete memoryless source (DMS) can be represented mathematically by a random variable V, with finite alphabet $\mathcal{V}$ and probability mass function $p(v) = \Pr\{V = v\}, v \in \mathcal{V}$. A discrete random process is a sequence of random variables $\{V_t; t \in \mathcal{T}\}$, indexed according to time $\mathcal{T} = \{0, 1, 2, \ldots\}$. A discrete time process is said to be *independent and identically distributed* (i.i.d.) if each sample in the sequence has the same distribution function and all samples are independent.

A source with a continuous alphabet is said to be *Gaussian* if the samples in the sequence are Gaussian random variables. Thus, each sample is drawn according to the Gaussian probability density function (pdf), $V \sim \mathcal{N}(\mu, \sigma^2)$:

$$f_V(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(v-\mu)^2/2\sigma^2}$$

where μ is the mean and σ^2 is the variance of V. In this thesis, the memoryless source considered is the Gaussian source.

A source where a sample is dependent on previous samples is said to have memory. One such example is the first order *Gauss-Markov* source, of the form

$$V_t = \rho V_{t-1} + \sqrt{1-\rho^2} N_t$$

where V_t represents the Gauss-Markov source at time t, ρ is the correlation coefficient such that $0 \leq \rho < 1$, and the $\{N_t\}$ are i.i.d., $N_t \sim \mathcal{N}(0, \sigma^2)$. The source with memory used in this thesis is the first order Gauss-Markov source. We will see that our scheme outlined in Chapter 4 and others [1] have a better gain for sources with memory, due to redundancy in the source.

A source is said to be stationary if the joint distribution of any subset of the sequence of random variables is invariant under time shifts; i.e.,

$$\Pr\{V_1 = v_1, V_2 = v_2, \ldots, V_n = v_n\} = \Pr\{V_{1+m} = v_1, V_{2+m} = v_2, \ldots, V_{n+m} = v_n\}$$

for any time shift m and all $v_1, v_2, \ldots, v_n \in \mathcal{V}$. In this thesis, stationary sources are assumed.

A discrete communication channel is a system with a finite input alphabet $\mathcal{X}$, finite output alphabet $\mathcal{Y}$, and transition distribution $p(y|x)$. The distribution $p(y|x)$ represents the probability of receiving output symbol y, given input symbol x was sent [9]. Typically, the transition probabilities are stored in a channel transition probability matrix. Figure 2.1 shows a block diagram of such a communication channel. The encoder and decoder blocks in the diagram represent several possible operations applied to the source which will be discussed in the upcoming sections. Without

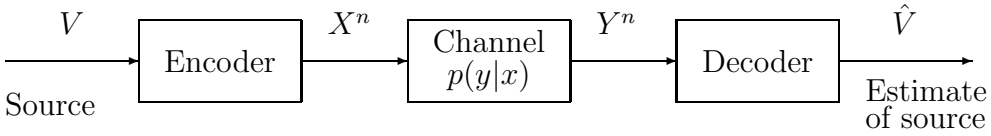

Figure 2.1: A communication channel.

loss of generality, for transmitted sequence $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and corresponding output sequence $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$, the channel transition probability matrix can be written as the n-dimensional distribution

$$p(\mathbf{y}|\mathbf{x}) \triangleq \Pr\{\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}\}, \quad \mathbf{x} \in \mathcal{X}^n, \text{ and } \mathbf{y} \in \mathcal{Y}^n. \tag{2.1}$$

Also generally speaking, the input and output alphabets do not necessarily have the same size. Thus, transition probabilities may be stored in matrices that are not necessarily square, as in our soft-decision demodulation scheme.

A *discrete memoryless channel* (DMC) has output symbols that are only statistically dependent on the corresponding input symbols at the time of transmission. It is independent of symbols previously sent or received. For example, the probability function of receiving symbol y_k, given symbol x_k was sent, at time k is

$$p(y_k|x_1, \ldots, x_k, y_1, \ldots, y_{k-1}) = p(y_k|x_k). \tag{2.2}$$

Also, if the channel does not have feedback, the channel input symbols do not depend on past output symbols. Thus, the channel transition probability of a discrete memoryless channel *without feedback* is given by

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{n} p(y_k|x_k). \tag{2.3}$$

Herein, when referring to a DMC, the discrete memoryless channel without feedback with finite input and output alphabets is implied.

2.1.1 AWGN Channel

One widely-known channel model that emulates the physical characteristics of a channel is the additive white Gaussian noise (AWGN) channel, shown in Figure 2.2. It is a time-discrete channel with continuous input and output alphabets. Its output at time t is simply defined as

$$Y_t = X_t + N_t, \quad N_t \sim \mathcal{N}(0, \sigma^2). \tag{2.4}$$

The noise N_t is assumed to be independent of the sent signal X_t and has a Gaussian distribution with variance σ^2. The AWGN channel is useful for modeling radio and satellite communication channels, though it does not describe distortion due to fading.

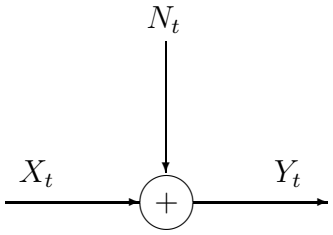

Figure 2.2: The Gaussian channel.

2.1.2 Rayleigh Fading Channel

Another channel model representing the physical effects of communication channels is the Rayleigh fading channel. Typically, signal fading occurs due to multipath propagation in mobile communication. Thus, an attenuation factor A_t denotes the fading that may occur to a signal. The Rayleigh fading channel is represented by

$$Y_t = A_t X_t + N_t. \tag{2.5}$$

The noise component N_t has a Gaussian distribution with zero-mean and σ^2 variance, just like the AWGN channel. The attenuation factor A_t has a Rayleigh distribution whose pdf is given by

$$f_A(a) = \begin{cases} 2ae^{-a^2}, & \text{if } a > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Note that $\{N_t\}$ and $\{A_t\}$ are i.i.d. processes that are independent of each other, and independent of $\{X_t\}$ as well. Also note that $E[{A_i}^2] = 1$.

2.2 Fundamentals from Information Theory

The first important concept from information theory is *entropy*. Entropy represents the amount of uncertainty of a random variable. The entropy $H(X)$ of a discrete random variable X is defined as [9]

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x), \tag{2.7}$$

where the probability mass function $p(x) = \Pr\{X = x\}, x \in \mathcal{X}$.

This definition leads us to Shannon's first coding theorem - the lossless source coding theorem. According to Shannon's Source Coding theorem, if H is the entropy of a discrete memoryless source, then the source output sequences can be represented without any loss of information by binary sequences of varying length with an average length arbitrarily close to, but not fewer than H binary digits per source digit [16]. Thus, the entropy of a source can represent the ultimate lossless data compression rate.

For sources with memory, such as Markov sources, the *entropy rate* describes how the entropy of a sequence grows with n when the sequence contains n random variables. Formally, the entropy rate of a source $\{X_i\}$ is defined as [9]

$$H\{X_i\} = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{2.8}$$

when the limit exists. Note, in the i.i.d. case, that $H(X_1, X_2, \ldots, X_n) = nH(X_1)$, thus $H\{X_i\} = H(X_1)$.

The single most important parameter in communication theory is channel capacity. The capacity of a channel describes the ultimate transmission rate per channel use and can be made a mathematical entity from the noise characteristics of the channel.

In order to understand the definition of channel capacity, one must be introduced to the *mutual information* between two random variables X and Y. Suppose X is the random variable at the channel input, and Y is the received random variable. The mutual information is defined as the relative entropy between the joint distribution

$p(x, y)$ and the product distribution $p(x)p(y)$, such that

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2.9}$$

In terms of the entropy and the conditional entropy of the random variable X, the mutual information between X and Y can be written as

$$I(X;Y) = H(X) - H(X|Y) \tag{2.10}$$

where $H(X|Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log p(y|x)$ is the conditional entropy of X, given Y.

This equation form allows us to interpret the mutual information between the channel input X and output Y as the average amount of uncertainty in X resolved by the observation of the outcome Y [16]. Thus, mutual information can represent a measure of the amount of information transferred between a channel's input and output.

This leads us to the definition of capacity. The *capacity* C of a DMC is defined as the largest mutual information $I(X;Y)$ that can be transmitted over the channel in one use, maximized over all input probability assignments. Written mathematically,

$$C = \max_{p(x)} I(X;Y) \tag{2.11}$$

where the maximum is taken over all possible input distributions $p(x)$. The above definition is specifically the "information" capacity. The rate R is the number of bits per transmission and is called *achievable* if information can be sent at rate R with an arbitrarily low probability of error. The "operational" capacity of a channel is defined

as the supremum of all achievable rates. Shannon's second theorem confirms that the "information" channel capacity is equal to the "operational" channel capacity [9].

Now that we have defined capacity and rate, we must understand the usefulness of such parameters in communication. Shannon presented the channel coding theorem in his 1948 paper [26] which states a benchmark for reliable transmission rates over noisy channels. The channel coding theorem states that for each fixed rate $R < C$, there exists a code of such rate with arbitrarily small probability of error.

In most cases, (2.11) does not have an analytic solution for the maximizing $p(x)$. In his 1972 paper [7], Blahut presents an algorithm for computing channel capacity using numerical techniques. The algorithm finds channel input probability vectors iteratively until the sequence of probability vectors converges to the vector that achieves capacity. The algorithm will be described in more detail in Section 3.4.

For non-discrete sources, the source sequence cannot be precisely reconstructed using a finite number of bits. In order to evaluate an encoder for such a source, both the encoding rate and some distortion measure must be taken into account. The number of bits per source sample (i.e. the encoding rate) will depend on the allowable distortion. A distortion measure is defined as a mapping

$$d\colon \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^{+} \tag{2.12}$$

from the source alphabet-reproduction alphabet pairs into the set of non-negative real numbers [9], where the reproduction alphabet is denoted by $\hat{\mathcal{X}}$. Thus, the distortion measure denoted by $d(X, \hat{X})$ measures the cost of representing the source X by its estimate $\hat{X}$, where $\hat{X}$ is chosen from $\hat{\mathcal{X}}$.

The study of the minimum expected distortion at a particular rate is called rate distortion theory. Rate distortion theory is pertinent to all lossy coding schemes (i.e. those that allow distortion, thus cannot perfectly recover the original signal).

The *rate distortion function* $R(D)$ is defined as the infimum of rates R achievable at a given distortion D [9], where D is the expected distortion with respect to the probability distribution of X. The *information* rate distortion function of a discrete memoryless source X with given distortion measure $d(X, \hat{X})$ is defined as [9]

$$R^{(I)}(D) = \min_{p(\hat{x}|x):\ \sum_{(x,\hat{x})} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}), \tag{2.13}$$

where the minimization is over all conditional distributions $p(\hat{x}|x)$ for which the joint distribution $p(x, \hat{x}) = p(x)p(\hat{x}|x)$ satisfies the distortion constraint [9].

The *information distortion rate function* is defined as

$$D^{(I)}(R) = \min_{p(\hat{x}|x):\ I(X;\hat{X}) \leq R} E[d(X, \hat{X})]. \tag{2.14}$$

Shannon's rate distortion theorem states that for an i.i.d. source X with distribution $p(x)$ and bounded distortion function $d(x, \hat{x})$, the rate distortion function $R(D)$ is equal to the information rate distortion function $R^{(I)}(D)$ [9].

2.3 Source and Channel Coding

As stated earlier, a communication system transmits data from a source to a destination. Before the data is sent over a channel, it may be transformed into an efficient and/or reliable form through an *encoder*. After the signal is sent over the channel, it is received and processed through a *decoder*, which attempts to reproduce the sent

data as close as possible to the original. A generic view of a communication system is shown in Figure 2.1. The encoder and decoder blocks could represent a host of processes which compress, protect, and retrieve the data, while the channel block could represent a noise disturbance model.

Two possible procedures encapsulated by the encoder block are source and channel coding. Source coding consists of compressing data from the source output in order to minimize the number of channel uses needed to transmit information. Channel coding prioritizes accurate signal reproduction at the receiver. By dividing the encoder into these two categories, we can view the system as a tandem communication system, where source coding is first applied to the source output, followed by channel coding. A block diagram of a tandem source-channel coding system is shown in Figure 2.3. The optimality of separately-designed source and channel coders in a tandem system is justified through Shannon's source-channel separation principle [26], [27]. The constraints of Shannon's source and coding theorems render large delay and complexity, in practice.

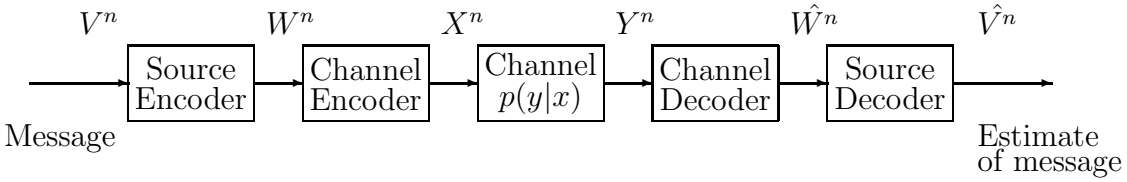

Figure 2.3: A communication system with tandem encoding and decoding.

2.4 Source Coding

Source signals are typically transmitted over communication channels or stored in media, but initially must be transformed into a suitable form. Source coding extracts vital information from source signals, either analog or digital, by removing redundancies in the data. Representing the original signal as a binary sequence prepares it for transmission or storage. Source coding is also known as signal compression.

There are two kinds of signal compression: lossy and lossless. Lossy compression introduces a tolerable amount of distortion in order to achieve beneficial compression rates; thus, it cannot perfectly reproduce the signal. Lossless compression does not introduce any distortion; therefore, the precise original signal is recovered. Lossy compression is inevitable when an analog or non-discrete source must be converted into a digital form. Digital data is more reliable than analog data for transmission over noisy channels when detection techniques such as matched filters are used [17].

Lossy compression can be achieved by sampling the source, quantizing the samples, and converting the quantized samples into binary data. Quantization is one of the most important and developed forms of lossy compression. Its advancements are discussed in upcoming sections.

Mathematically speaking, the two types of compression can be determined by the type of source redundancy removed. Lossless compression tackles statistical redundancy - the redundancy due to a source's non-uniform distribution, source memory, or both. Lossy compression exploits both statistical and non-statistical redundancies. Examples of non-statistical data include frequencies not captured by the human eye

and ear in images and sound, respectively.

Statistical redundancy is related to the entropy or the entropy rate of the source [8]. The total statistical redundancy can be expressed as

$$\rho_{tot} = \log_2 |\mathcal{X}| - H\{X_i\} \tag{2.15}$$

and can also be written as

$$\rho_{tot} = \rho_{dist} + \rho_{mem} \tag{2.16}$$

where $\rho_{dist} = \log_2 |\mathcal{X}| - H(X_1)$ is the redundancy due to a source's non-uniform distribution, while $\rho_{mem} = H(X_1) - H\{X_i\}$ is the redundancy due to source memory, and $H\{X_i\}$ is the entropy rate.

A useful tool in analyzing lossy compression is the rate-distortion function. To recapitulate, the rate distortion function gives a measure for the minimum number of bits required to allow a particular distortion.

2.4.1 Scalar Quantization

Quantization is the basis for analog-to-digital conversion. Simply speaking, quantization is a type of lossy compression that takes a single number and selects the nearest number from a predetermined finite set.

An N-point scalar quantizer Q is a mapping, typically from the real line $\mathbb{R}$ to a finite set of values $\mathcal{C}$, such that

$$Q \colon \mathbb{R} \to \mathcal{C}. \tag{2.17}$$

The output set or codebook $\mathcal{C}$ is

$$\mathcal{C} = \{y_1, y_2, \ldots, y_N\} \subset \mathbb{R} \tag{2.18}$$

and has size $|\mathcal{C}| = N$. The elements or y_i's in the codebook are also known as output points or codewords.

Every N-point quantizer has N quantization cells that form a partition of the real line $\mathbb{R}$. The quantization cells $S_i, i = 1, 2, \ldots, N$ are described as

$$S_i = \{x \in \mathbb{R} \colon Q(x) = y_i\} \equiv Q^{-1}(y_i), \quad i = 1, 2, \ldots, N. \tag{2.19}$$

The important properties of the cells are

$$\bigcup_{i=1}^{N} S_i = \mathbb{R} \tag{2.20}$$

and

$$S_i \bigcap S_j = \emptyset \quad \text{for } i \neq j. \tag{2.21}$$

One important parameter of a scalar quantizer is the rate R, defined as

$$R = \log_2 N \tag{2.22}$$

and it represents the number of bits needed to identify a specific quantized output.

Every quantizer can be viewed as two successive mappings: the encoder followed by the decoder. The encoder-decoder structure of the N-point quantizer Q is shown in Figure 2.4.

The encoder is defined as the mapping γ such that

$$\gamma \colon \mathbb{R} \to \mathcal{I} \text{ where } \gamma(x) = i \text{ iff } x \in S_i \tag{2.23}$$

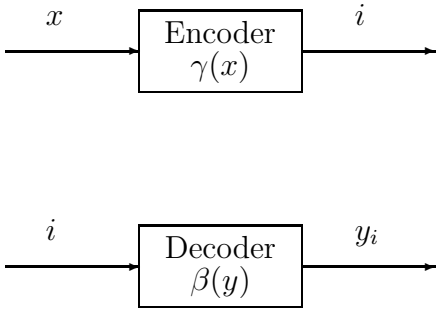

Figure 2.4: Quantizer encoding and decoding structure.

where $\mathcal{I} = \{1, 2, \ldots, N\}$ and the decoder β is defined as

$$\beta \colon \mathcal{I} \to \mathcal{C} \text{ where } \beta(i) = y_i. \tag{2.24}$$

Thus, a scalar quantizer can be completely defined by its codebook $\mathcal{C}$ and partition $\{S_i\}$.

In order to assess the performance of a quantizer, the overall quality degradation can be quantified using a variety of distortion measures. The most popular distortion measure is the squared error defined by

$$d(x, y) = (x - y)^2. \tag{2.25}$$

In general, the statistical average (expected) distortion is defined as

$$D = E[d(X, Q(X))] = \int_{-\infty}^{\infty} d(x, Q(x)) f_X(x)\, dx \tag{2.26}$$

where $f_X(x)$ is the pdf of X. The statistical average of the squared error distortion measure, for a given input random variable X and N-point quantizer $Q = \{y_i, S_i; i = 1, 2, \cdots, N\}$, can be written as

$$D = E[(X - Q(X))^2] = \sum_{i=1}^{N} \int_{S_i} (x - y_i)^2 f_X(x)\, dx. \tag{2.27}$$

The average squared error distortion measure is the most popular performance measure for a quantizer and is used in our design of a soft-decision channel-optimized vector quantizer, outlined in Chapter 4.

If the input sequence is stationary and ergodic, and the statistical average distortion has value D, then the ergodic theorem implies with probability one, that the limiting time average distortion is D, or [17]

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} d(X_i, Q(X_i)) = D. \tag{2.28}$$

Conditions for Optimality

From an engineering stand-point, improving performance is of the utmost importance. Since the rate function R is fixed for a scalar quantizer, the distortion is the main focus. The distortion function is a single expression indicative of the signal degradation due to quantization. Since the input is unknown, the source is assumed to be a random variable, usually described by its pdf.

The goal of the scalar quantizer design is to minimize the distortion D, given a fixed number of output levels N and a particular source pdf. The optimal quantizer is hoped to be achieved by appropriately picking the reproduction levels y_i and the partition cells S_i. In general, there are no known closed-form solutions to find the optimal quantizer, but there are necessary conditions for optimality. Finding a partition $\{S_i\}$ allows the quantizer to have an encoding rule, while a codebook $\mathcal{C}$ gives the quantizer a decoding rule. Again, since there are no closed-form expressions for optimality, the best we can do is optimize one rule, given the other one is provided.

The two necessary conditions for optimality were observed independently by Lloyd,

originally in 1957 [22], and Max in 1960 [24]. Max states that to minimize D for fixed N, the necessary conditions are derived by differentiating D with respect to endpoints of the partition cells and output levels, then setting the equations equal to zero [24].

The first condition is the nearest neighbour condition, which produces the optimal partition, given the codebook already exists.

Nearest Neighbour Condition

For a given codebook $\mathcal{C}$ with N output levels, the optimal partition cells satisfy

$$S_i = \{x \colon d(x, y_i) \leq d(x, y_j);\ \text{for all } j \neq i\}, \forall i \tag{2.29}$$

so that the mapping Q becomes

$$Q(x) = y_i \text{ if } d(x, y_i) \leq d(x, y_j) \text{ for all } j \neq i, \tag{2.30}$$

where the distortion function $d(x, y_j)$ could represent any distortion measure, including the square Euclidean distance measure. For a quantizer with a finite output set, the indexing of interval endpoints $\{x_i\}$ and output values $\{y_i\}$ are assumed to be ordered according to

$$x_0 < y_1 < x_1 < y_2 < x_2 < \cdots < y_N < x_N. \tag{2.31}$$

For square or absolute error, the endpoint of the partition cell that is implied by the nearest neighbour rule is the midpoint between two output levels [24], i.e.,

$$x_i = (y_i + y_{i+1})/2, \tag{2.32}$$

where x_i represents the endpoint between cell S_i and S_{i+1}. To fully complete the encoding rule, an arbitrary assignment rule is necessary for points on the cell boundary, since the average distortion will not be affected [17]. A typical solution for

boundary cell assignment is to define the partition cell as a half-closed interval, like $S_i = \{x\colon x_i < x \leq x_{i+1}\}$.

The second necessary condition for optimality is the centroid condition, which specifies the reproduction levels or codebook, given a partition.

Centroid Condition

For a given partition $\mathcal{P} = \{S_i;\, i = 1, \ldots, N\}$, the optimal output levels, with respect to the mean squared error (MSE) for a noiseless channel, satisfy

$$y_i = \text{cent}(S_i) = \frac{\int_{S_i} x f_X(x)\, dx}{\int_{S_i} f_X(x)\, dx} \tag{2.33}$$

$$= \arg\min_{y_i} E[\|X - y_i\|^2 | X \in S_i] \quad \text{for all } i = 1, \ldots, N. \tag{2.34}$$

Quantizers obeying the previous conditions are usually called Lloyd-Max quantizers. Additional conditions for optimality were given by Lloyd for the scalar quantizer case with a discrete alphabet [17].

Generally, the two necessary conditions stated above are not sufficient. In specific cases, the conditions can be shown to be sufficient, for example when $\log f_X(x)$ is a concave function of its argument [15]. One random variable with such a pdf is the Gaussian random variable.

The design algorithm used to create quantizers is based on the two necessary conditions for optimality derived by Lloyd, and also by Max [17]. The basic iteration of the algorithm creates a new codebook, improving upon the older one. The Lloyd algorithm, as outlined in [17], follows.

Lloyd algorithm

Step 1. Begin with the initial codebook $\mathcal{C}_1$. Set $m = 1$.

Step 2. Given the codebook $\mathcal{C}_m$, perform the following *Lloyd iteration* to generate the improved codebook $\mathcal{C}_{m+1}$:

- Given the codebook $\mathcal{C}_m = \{y_i\}$, find the optimal partition using the Nearest Neighbour Condition:

$$S_i = \{x \colon d(x, y_i) \leq d(x, y_j); \text{ for all } j \neq i\}.$$

- Using the Centroid Condition, compute the centroids for each of these cells to create a new codebook $\mathcal{C}_{m+1} = \{\text{cent}(S_i), i = 1, \ldots, N\}$.

Step 3. Compute the average distortion for $\mathcal{C}_{m+1}$. If it has changed by a small enough distortion since the last iteration, then stop; else increment m by 1 and go to step 2.

The centroid condition calls for the expectation operation, thus implying the source's pdf is known. For the case when it is not, the following result for empirical data applies. By the strong law of large numbers, the empirical cdf converges with probability one to the actual cdf of the random input variable for n sufficiently large, where n is the number of training points [17]. Thus, the Lloyd iteration can be used to produce a quantizer that is nearly optimal for the true distribution of the input random variable.

2.4.2 Vector Quantization

Vector Quantization (VQ) generalizes the scalar case to allow the quantization of vectors instead of quantizing just a real number. It extends the one-dimensional design concept to multiple dimensions. A vector quantizer maps a sequence of continuous or discrete vectors to a predetermined set of vectors, or codevectors. A typical use for VQ is data compression, including image and audio data.

Vector quantization shows vast improvements in performance over scalar quantization, even if the source is memoryless [18, p.4]. The performance of VQ tends to the rate-distortion bound as the block length approaches infinity [23]. Indeed, this is not possible in practice but necessary conditions for optimality were extended from the scalar case to apply to vector quantizers.

A k-dimensional vector quantizer Q is defined as a mapping of a vector in the k-dimensional Euclidean space $\mathbb{R}^k$ to a finite set or codebook $\mathcal{C} = \{\mathbf{y}_i, i = 1, \ldots, N\}$, with N reproduction vectors, otherwise known as codewords. Thus, the vector quantizer Q can be expressed as the mapping

$$Q\colon \mathbb{R}^k \to \mathcal{C}. \tag{2.35}$$

Every N-point VQ partitions the k-dimensional Euclidean space $\mathbb{R}^k$ into N cells or regions. The i-th cell is defined as $S_i = \{\mathbf{x} \in \mathbb{R}^k\colon Q(\mathbf{x}) = \mathbf{y}_i\}$. Thus, the partition $\mathcal{P} = \{S_i; i = 1, \ldots, N\}$ is a set of mutually exclusive subsets that cover $\mathbb{R}^k$. With this notation, the mapping $Q(\cdot)$ can also be written as

$$Q(\mathbf{x}) = \mathbf{y}_i \text{ if } \mathbf{x} \in S_i \tag{2.36}$$

and the partition $\mathcal{P}$ satisfies

$$\bigcup_i S_i = \mathbb{R}^k, \text{ and } S_i \bigcap S_j = \emptyset \text{ when } i \neq j. \tag{2.37}$$

One important measure of VQ performance is the rate function, which is defined as

$$R = \frac{1}{k} \log_2 N, \tag{2.38}$$

otherwise known as the encoding rate with unit bits per source sample. Similar to the scalar quantizer, a VQ can be regarded as a cascade of two operations: the vector *encoder* and vector *decoder* operations. The encoder operation γ maps a vector from $\mathbb{R}^k$ to the index set $\{1, 2, \ldots, N\}$, defining the mapping as $\gamma(\mathbf{x}) = i$ if $\mathbf{x} \in S_i$. Sequentially, the decoder mapping β maps the index set to the codebook $\mathcal{C}$ with the mapping $\beta(i) = \mathbf{y}_i$ for $i \in \{1, 2, \ldots, N\}$ and $\mathbf{y}_i \in \mathcal{C}$. The overall operation is described by

$$Q(\mathbf{x}) = \beta(\gamma(\mathbf{x})). \tag{2.39}$$

VQ Optimality Conditions

The typical *single letter* distortion commonly used to measure VQ performance is the squared error distortion. This *per-letter* distortion function calculates the squared Euclidean distance between the source and quantized vectors and is defined for $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$ as

$$\begin{aligned} d(\mathbf{x}, \mathbf{y}) &= \|\mathbf{x} - \mathbf{y}\|^2 \\ &= \sum_{i=1}^{k} |x_i - y_i|^2. \end{aligned} \tag{2.40}$$

Given $\mathbf{X}$ is a random vector in $\mathbb{R}^k$ with a particular pdf $f_{\mathbf{X}}(\mathbf{X})$, the average distortion per sample for a particular VQ with a defined partition set and codebook can be expressed as

$$D = \frac{1}{k} E d(\mathbf{X}, Q(\mathbf{X})) = \frac{1}{k} \int_{\mathbb{R}^k} d(\mathbf{x}, Q(\mathbf{x}))\, f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}, \tag{2.41}$$

$$= \frac{1}{k} \sum_{j=1}^{N} \int_{S_j} d(\mathbf{x}, \mathbf{y}_j)\, f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}, \tag{2.42}$$

$$= \frac{1}{k} \int \sum_{j=1}^{N} P(\mathbf{X} \in S_j) d(\mathbf{x}, \mathbf{y}_j)\, f_{\mathbf{X}|j}(\mathbf{x})\, d\mathbf{x}, \tag{2.43}$$

$$= \frac{1}{k} \sum_{j=1}^{N} P(\mathbf{X} \in S_j) E[d(\mathbf{x}, \mathbf{y}_j) | \mathbf{X} \in S_j]. \tag{2.44}$$

The goal of designing a VQ is to find a codebook and a partition set to improve the overall performance, which can be assessed as one which minimizes distortion. Similar to the scalar case, Linde et al. proposed a solution to finding an at least locally optimal VQ through an iterative algorithm [23]. The algorithm is typically called the LBG or the Generalized Lloyd-Max algorithm. It extends the necessary conditions for optimality of the scalar quantizer case, and also uses them in the iterative algorithm. In vector form, the necessary conditions for optimality are stated below.

Nearest Neighbour Condition

For a given codebook $\mathcal{C}$ with N k-dimensional reproduction vectors, the optimal partition cells satisfy

$$S_i = \{\mathbf{x} \colon d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j);\ \text{for all } j\}, \tag{2.45}$$

such that the mapping Q becomes

$$Q(\mathbf{x}) = \mathbf{y}_i \text{ if } d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \text{ for all } j, \tag{2.46}$$

where the distortion function $d(\mathbf{x}, \mathbf{y}_j)$ could represent any distortion measure, including the squared error distortion measure. The partition cell defined in the nearest neighbour condition is also called a Voronoi region [17].

The second necessary condition for optimality in a VQ is the centroid condition, which specifies the codevectors, or reproduction levels, given a partition.

Centroid Condition

For a given partition $\mathcal{P} = \{S_i;\, i = 1, \ldots, N\}$, the optimal codevectors that minimize the squared error distortion measure for a noiseless channel satisfy

$$\mathbf{y}_i = \text{cent}(S_i) = \frac{\int_{S_i} \mathbf{x} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}}{\int_{S_i} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}} \tag{2.47}$$

$$= \arg\min_{\mathbf{y}_i} E[\|\mathbf{X} - \mathbf{y}_i\|^2 | \mathbf{X} \in S_i] \quad \text{for all } i = 1, \ldots, N. \tag{2.48}$$

Note that in both necessary conditions for optimality of a VQ, the equations simplify to those of the scalar quantizer case when the dimension of the vector is one, i.e. $k = 1$. Though these two conditions are necessary for the optimality of a VQ, they are not sufficient.

An iterative algorithm based on the necessary conditions for the optimality improves the codebook, and hence the performance of VQ, and is the generalization of Lloyd's method for designing scalar quantizers [23]. The algorithm improves the codebook and partition set based on a training sequence and is identical to the scalar case, except vectors are used in place of real numbers.

The codebook that is used at the beginning of the algorithm presents another discussion. Linde et al. use what they call a 'splitting' algorithm that shows improve-

ment on the performance of an LBG-VQ compared to an arbitrary initial codebook. This and other techniques for developing the initial codebook are discussed in Chapter 3. The Generalized Lloyd-Max algorithm involves exhaustive codebook searches which significantly contribute to the complexity of the algorithm. Gray [18] outlines alternate VQs which trade off performance to reduce computation or memory requirements. Typically, these types of algorithms use suboptimal codebooks, encoding rules, or constrained code structures to ease the searching or lessen the memory required at the price of suboptimal performance. Examples of these VQs include Tree-searched VQs, Multistage VQ, Product codes, and Lattice VQ.

2.5 Channel Coding

The primary purpose of source coding is to remove redundancies from the source signal in order to compress it. On the other hand, channel coding adds controlled redundancy to signals before they are transmitted over presumably noisy channels. The purpose of channel coding is to protect the most vulnerable data from distortion caused by channel noise. Channel codes are also referred to as error correcting codes, and are generally categorized into block codes and convolutional codes. Block codes combine data bits and redundant parity bits to create codewords. Convolutional codes are described by their generator sequences or impulse responses, which are used to generate the discrete-time convolution of the input data sequence [31]. Both types of channel codes have practical applications in mobile communications.

Chapter 3

VQ-Based Joint Source-Channel Coding

Though Shannon's fundamental result allows the separation of source and channel coding in order to achieve the optimal compression and transmission rates, these limits are only achievable with very large block length, and thus, very large encoding/decoding complexity. Motivation for investigating joint source and channel coding is drawn from avoiding complexity due to separately optimizing source and channel coders. Methods of tackling the combined coding problem are modifying the Lloyd-Max algorithm or assigning binary codewords to VQ codewords. Dunham and Gray [10] proved that joint source and channel trellis encoding systems can perform arbitrarily close to the source distortion-rate function at the channel capacity but with no indication of how to design a trellis encoding system [3]. Ayanoglu and Gray [3] applied joint source and channel coding to trellis waveform coders by using the generalized Lloyd-Max algorithm. By using the squared error distortion measure and

operating over an AWGN channel, Ayanoglu and Gray found jointly optimized codes performing close to or better than optimized tandem systems under the same trellis code conditions.

Kurtenbach and Wintz [20] incorporate source and channel coding where they address the sources of error in the system and find the quantization levels that minimize the total MSE. Kurtenbach and Wintz combine source and channel coding by extending the necessary conditions for optimality by Lloyd and Max for scalar quantization to incorporate channel noise. The quantizer structure presented in [20] minimizes the squared error distortion for any probability density on the input data $f_X(x)$ and channel transition probability matrix $p(\mathbf{y}|\mathbf{x})$. Using notation from [20], the squared error distortion of a scalar quantizer for noisy channels can be written as

$$\epsilon^2 = E(x - z)^2 \tag{3.1}$$

$$= \epsilon_q^2 + \epsilon_c^2 + \epsilon_m \tag{3.2}$$

where ϵ_q^2 represents the distortion due to representing x as quantization level y, ϵ_c^2 represents the cost of making incorrect decisions at the receiver by choosing z instead of y, and finally, ϵ_m is the mutual error. Totty and Clark [33] express these sources of error mathematically as

$$\epsilon_q^2 = E(x - y)^2 \tag{3.3}$$

$$\epsilon_c^2 = E(y - z)^2 \tag{3.4}$$

$$\epsilon_m = 2E[(x - y)(y - z)]. \tag{3.5}$$

Totty and Clark also show that the mutual error ϵ_m vanishes when the quantizer levels are chosen according to the optimal conditions from Max that minimize the squared

error distortion. Kumazawa et al. [19] extend the scalar quantizer for noisy channels to the case of vector quantizers. Their results are presented in the next section.

One method of improving quantizer performance is to incorporate the channel transition matrix into the design of the quantizer, as previously discussed. Another method is to systematically assign binary codewords to the reproduction vectors of a VQ. This is also known as index assignment. It has been shown that index assignment has an impact on system performance [12], and specifically, it improves performance over randomly selected codewords [14].

3.1 Channel Optimized Vector Quantization

Joint source and channel coding has shown performance improvements over tandem source and channel coders. Specifically, channel optimized vector quantization (COVQ) incorporates channel knowledge into the design of the vector quantizer. Basically, in the design of a COVQ, the source is described through its pdf and the channel is described through the channel transition probability matrix. For the noiseless channel, the transition matrix is the identity matrix, which reduces to the vector quantization results.

Kurtenbach and Wintz [20] incorporated the channel information into the design of a scalar quantizer. Kumazawa et al. [19] presented the multidimensional generalization of Kurtenbach and Wintz's work when they designed a vector quantizer for noisy channels.

Assume the source to be encoded is a real-valued, stationary, and ergodic process

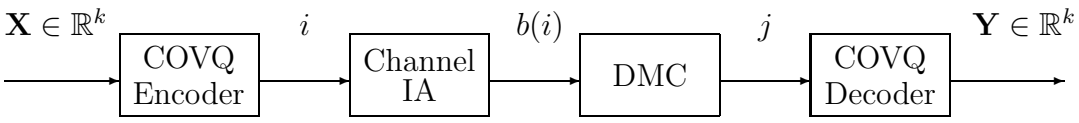

Figure 3.1: The COVQ system with channel index mapping.

$\{X_t;\, t = 0, 1, \ldots\}$ with zero-mean and variance ${\sigma_X}^2$. The source is to be encoded via a vector quantizer and sent over a noisy channel. Consider a k-dimensional N-level VQ and a DMC with input and output alphabets $\mathcal{I} = \{1, 2, \ldots, N\}$ and $\mathcal{J} = \{1, 2, \ldots, N\}$, respectively.

For now, we will assume the input and output alphabets are the same size. We will see in the next section that this assumption is limiting. The generalization allows the channel transition matrix to be non-square.

Figure 3.1 shows a block diagram of a COVQ communication system. The system components include an encoder mapping γ, a channel index mapping b, and a decoder mapping β. The encoder mapping $\gamma\colon \mathbb{R}^k \mapsto \mathcal{I}$ is described in terms of a partition $\mathcal{P}$ such that

$$\gamma(\mathbf{x}) = i, \quad \text{if } \mathbf{x} \in S_i, \quad i \in \mathcal{I}, \tag{3.6}$$

where $\mathbf{x} = (x_{nk}, x_{nk+1}, \ldots, x_{nk+k-1})$ is a typical source output vector. The partition $\mathcal{P} = \{S_1, S_2, \ldots, S_N\}$ is a set of disjoint subsets that cover the entire vector space $\mathbb{R}^k$. The channel index assignment (IA) mapping $b\colon \mathcal{I} \mapsto \mathcal{I}$ is a one-to-one mapping whose purpose is to assign the encoder output to a non-random index to be transmitted over the DMC. Thus, the index assignment mapping is described as $i' = b(i) \in \mathcal{I}$. The DMC can be described by its channel transition probability matrix $p(j|i')$, whose

entries represent the probability that j is received, given i' was sent.

The last mapping is the decoder mapping $\beta\colon \mathcal{J} \mapsto \mathbb{R}^k$, which is described by the codebook $\mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, such that

$$\beta(j) = \mathbf{y}_j, \quad j \in \mathcal{J}. \tag{3.7}$$

Again, the encoding rate is defined as

$$R = \frac{\log_2 N}{k} \text{ bits/source sample.} \tag{3.8}$$

We shall denote the distortion caused by representing a source vector $\mathbf{x}$ by a codevector $\mathbf{y}$ as $d(\mathbf{x}, \mathbf{y})$, a non-negative distortion measure. The performance of the COVQ system can be measured by the average distortion per sample $D(\mathcal{P}, \mathcal{C}, b)$, calculated according to the following equation [14]:

$$D(\mathcal{P}, \mathcal{C}, b) = \frac{1}{k} \sum_{i=1}^{N} \sum_{j=1}^{N} p(j|b(i)) \int_{S_i} d(\mathbf{x}, \mathbf{y}_j) f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{3.9}$$

Thus, for a given source, noisy channel, dimension k, and fixed codebook size N, we wish to minimize $D(\mathcal{P}, \mathcal{C}, b)$ by carefully choosing $\mathcal{P}$, $\mathcal{C}$, and b. Note, that this distortion measure is just a modified version of that used to measure VQ performance.

For ease of notation, we shall omit the index assignment mapping b since a change in $b(i)$ only results in a relabeling of the partition $\mathcal{P}$ [14]. Thus, the average overall distortion per source sample can be written as

$$D = \frac{1}{k} \sum_{i=1}^{N} \int_{S_i} \{\sum_{j=1}^{N} p(j|i) d(\mathbf{x}, \mathbf{y}_j)\} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{3.10}$$

An algorithm based on simulated annealing that assigns binary codewords to the codevectors of a VQ is presented in a later section.

Kumazawa et al. [19] present the equations for the optimal reconstruction vectors $\mathbf{y}_j \in \mathcal{C} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ and hence, one of the necessary conditions for minimizing (3.9). Given the partition $\mathcal{P} = \{S_1, S_2, \ldots, S_N\}$, the reconstruction vector of cell S_i that minimizes (3.9) is

$$\mathbf{y}_l = \frac{\sum_{i=1}^{N} p(l|i) \int_{S_i} \mathbf{x} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}}{\sum_{i=1}^{N} p(l|i) \int_{S_i} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}}. \tag{3.11}$$

The second necessary condition for minimizing (3.9) is as follows. Given the reconstruction alphabet or codebook $\mathcal{C}$, then the subset S_l that minimizes (3.9) is [19]

$$S_l = \{\mathbf{x}\colon \sum_{j=1}^{N} p(j|l) d(\mathbf{x}, \mathbf{y}_j) \leq \sum_{j=1}^{N} p(j|i) d(\mathbf{x}, \mathbf{y}_j) \text{ for all } i \neq l\}. \tag{3.12}$$

A discrete modification of the necessary conditions, with the mean square error distortion measure explicitly stated are

$$\mathbf{y}_l = \frac{\sum_{i=1}^{N} p(l|i) \sum_{v\colon \mathbf{x}_v \in S_i} \mathbf{x}_v / n}{\sum_{i=1}^{N} p(l|i) |S_i| / n} \tag{3.13}$$

$$= \frac{\sum_{i=1}^{N} p(l|i) \sum_{v\colon \mathbf{x}_v \in S_i} \mathbf{x}_v}{\sum_{i=1}^{N} p(l|i) |S_i|}, \tag{3.14}$$

and

$$S_l = \{\mathbf{x}\colon \sum_{j=1}^{N} p(j|l) \{\|\mathbf{x} - \mathbf{y}_j\|^2\} \leq \sum_{j=1}^{N} p(j|i) \{\|\mathbf{x} - \mathbf{y}_j\|^2\} \text{ for all } i \neq l\} \tag{3.15}$$

where $\{\mathbf{x}_v\}$ is the training set, n is the total number of training vectors, and $|S_i|$ is the number of training vectors in S_i. This form is useful, especially when the source distribution is not known and training sequences are used in the quantizer design.

The design of the COVQ follows the same format as the VQ design, with the exception of the modified average distortion calculation and modified necessary conditions for optimality. The initial codebook $\mathcal{C}_1$ used at the beginning of the design

algorithm has been shown to heavily influence the performance of such a quantizer. Techniques for developing a superior initial codebook are discussed in a later section.

It is interesting to note the result by Farvardin and Vaishampayan [14] concerning the complexity of COVQ algorithms. Farvardin and Vaishampayan state that the complexity of encoding is proportional to the number of nonempty encoding regions in the partition $\mathcal{P}$. This is an interesting fact, especially when combined with the knowledge that the number of nonempty encoding regions drops below the cardinality of the codebook at low SNR, where SNR is the signal-to-noise ratio. Thus, there is a reduced encoding complexity at low SNR.

3.2 Soft-Decision COVQ

In much of the previous work on joint source and channel coding algorithms, discrete memoryless channels were used with hard-decision demodulation assumed at the receiver. Vaishampayan and Farvardin began the use of soft decoders for VQs over noisy channels in [34]. Similar to our scheme to be presented in Chapter 4, Vaishampayan and Farvardin include the signal constellation in the design of the joint quantizer. The difference lies in the encoder where in [34] a linear estimate-based decoder is used. Liu et al. also use estimation theory to extend the soft decoding problem for VQs by providing a nonlinear optimal decoder in [21].

A Hadamard-based soft decoder scheme was developed by Skoglund [29], which also has an estimator-based decoder. Unfortunately, it has higher computational complexity than hard-decision decoding; thus, in the same paper, Skoglund lowers

complexity by developing suboptimal schemes. In [1] and [25], Alajaji and Phamdo present a soft-decision scheme whose decoder is detection-based, with a look-up decoding table. Our scheme is based on that presented by Alajaji and Phamdo. It should be noted that our scheme and the schemes in [1] and [25] trade complexity for the storage requirement (for the codebook). Both Skoglund's as well as Alajaji and Phamdo's schemes are briefly discussed below.

3.2.1 Soft-Decision Demodulation for COVQ

Alajaji and Phamdo proposed a scheme that used soft-decision information in the design of the COVQ algorithm. The scheme was used for both AWGN and Rayleigh fading channels where binary phase-shift keying (BPSK) modulation was used ([1], [25]). For both memoryless Gaussian and Gauss-Markov sources, the results proved the scheme performs better than hard-decision demodulation. A block diagram of the soft-decision COVQ system from [1] is shown in Figure 3.2. The input source vector

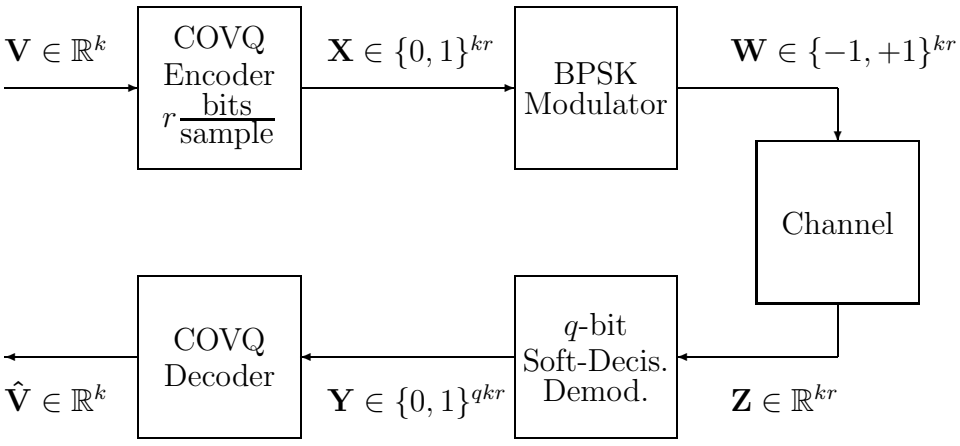

Figure 3.2: The soft-decision COVQ system.

$\mathbf{V}$ is a real k-tuple, while the COVQ encoder has a rate of r bits per source dimension. The COVQ encoder produces a binary index vector $\mathbf{X} \in \{0,1\}^{kr}$ for each source vector to be transmitted. The binary index is transmitted via a BPSK modulation; thus, each of the kr bits is represented by a BPSK constellation signal $\mathbf{W} \in \{-1,+1\}$. Once the signal is modulated, it is sent over the noisy channel, which can be described by its non-square transition matrix. The channel transition probability matrix describes the concatenation of the modulator, channel, and demodulator.

At the receiver, the received vector $\mathbf{Z} \in \mathbb{R}^{kr}$ is put through the q-bit soft-decision demodulator, where the output is a binary vector $\mathbf{Y} \in \{0,1\}^{qkr}$. A q-bit soft decision demodulator takes soft information of signals transmitted over a channel and demodulates the signals with a uniform scalar quantizer, producing q bits for every bit sent. Thus, the transition matrix represents a 2^{kr}-input 2^{qkr}-output DMC.

The soft-decision demodulator output is passed to the COVQ decoder, where the estimate $\hat{\mathbf{V}}$ is chosen from the COVQ codebook $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{2^{qkr}}\}$ according to the binary index passed from the soft-decision demodulator. Note the notation change, where previously $\mathbf{Y}$ represented the codeword, whereas now it represents the binary index of the codeword $\mathbf{c}_\mathbf{y}$. Also, $\mathbf{X}$ is the binary index to be sent over the channel representing source random variable $\mathbf{V}$.

The main feature of the soft-decision COVQ design is at the receiver. Each signal in the BPSK constellation represents one bit. Since q bits are received for every bit sent, there are $2^{(q-1)kr}$ times as many codewords than there are indices sent. The received signal is a real value; it is put through a scalar quantizer at the receiver with 2^q levels. The quantizer's step-size Δ, or length of quantizer interval, at the

receiver is predetermined according to the channel noise variance. The step-size is numerically selected to maximize the capacity of the equivalent DMC resulting from the concatenation of the modulation, the channel, and the q-bit soft-decision demodulator. Two reasons for choosing the step-size Δ in this manner are typically an increase in channel capacity results in an overall system performance and optimizing the soft-decision quantizer to maximize capacity is simpler than doing so to maximize SDR [25].

The average squared-error distortion per sample used in [1] was the same function in [14] stated in (3.10). The optimal reconstruction vectors $\mathbf{c_y}$ and partition cells S_i were also the same as in (3.11) and (3.12), with the exception of the different size input and output alphabets resulting in more codewords at the receiver than cell indices at the transmitter. The soft-decision COVQ design uses the LBG algorithm with the modified distortion and necessary conditions to find the optimal quantizer.

Note that for the case of BPSK modulation, the channel transition probability matrix is weakly symmetric; i.e. it can be divided along its columns into symmetric arrays. Symmetric arrays have rows and columns that are permutations of each other [16]. This symmetry property allows easy computation of capacity by evaluating $I(X;Y)$, the mutual information of the channel, using a uniform distribution on X.

For channel transition matrices without symmetric properties, closed-form expressions for calculating capacity are not known. Blahut develops an iterative algorithm for calculating the capacity of a channel using the transition matrix [7], which we will take a closer look at in Section 3.4, as it is used in our soft-decision COVQ design.

3.2.2 Hadamard-Based Soft-Decoding for VQ over Noisy Channels

Skoglund has developed a soft decoding scheme for vector quantization over noisy channels based on the Hadamard transform. The difference between 'normal' vector quantizers for noisy channels and Skoglund's scheme is in the decoder. The VQ decoder in [28] interprets the soft channel output. Soft decoders map unquantized (soft) channel output to source vector estimates. These scheme contains estimator-based decoding, in contrast to detector-based decoding, which simply uses a table lookup.

The Hadamard framework allows the source vector estimate to be described by the estimates of the transmitted index's individual bits [28]. The Hadamard matrix comes into play in the expression for representing each encoding region's centroid. Skoglund's Hadamard-based soft decoder, optimal in the minimum mean square error (MMSE) sense, uses *a priori* index probability as well as channel information to estimate the source vector. For further mathematical details, the reader is referred to [28].

A disadvantage of this soft decoding scheme is its complexity. It has a higher complexity compared to Alajaji and Phamdo's table look-up scheme, and obviously over the traditional hard-decision decoding of a 'normal' COVQ. On the other hand, it has a lower memory requirement.

Skoglund's follow-up paper on bit-estimate based decoders focused on improving

flexibility and complexity [30]. The paper provides a structure for the soft decoder using soft minimum mean-square error (MMSE) estimates for the transmitted bits.

3.3 Codebook Initialization Techniques

It is known that simply using iterative improvement algorithms to create codebooks and partitions do not generally yield optimum codes, and may even result in poor performance [18]. By looking at the breakdown of the distortion measure, it becomes evident that codebook initialization techniques are needed for quantization and channel noise resilience. Certain codebook initialization techniques minimize distortion by targeting specific distortion variables.

Recall the overall distortion of a scalar quantizer, in the squared error sense, can be broken down into three terms:

$$\epsilon^2 = \epsilon_q^2 + \epsilon_c^2 + \epsilon_m \tag{3.16}$$

where the third term ϵ_m vanishes when each reconstruction vector $\mathbf{c}_i$ of the quantizer is the centroid of the corresponding encoding cell S_i. Cheng [8] presents the VQ extension of the proof by Totty and Clark that shows the mutual cross-term ϵ_m vanishes when the codewords are picked according to the centroid condition for optimality and distortion is measured in the squared error sense. The other two terms represent the distortion due to quantization (i.e. the error caused by representing source samples by a predetermined finite set) and channel noise (i.e. the error incurred by poor decoding decisions at the receiver).

The two techniques for developing an initial codebook discussed in this section are the splitting algorithm and simulated annealing. The purpose of each technique is to influence the quantizer design to favour a good performance by picking an initial codebook that tackles the two different types of distortion. The splitting algorithm influences the distortion due to quantization ϵ_q^2 by finding the initial codebook for VQ. Simulated annealing attempts to minimize ϵ_c^2, or channel distortion, by finding the best index assignment b.

3.3.1 Splitting Algorithm

One method for initializing a codebook is called the "initial guess by splitting" algorithm presented by Linde, Buzo, and Gray [23]. The splitting algorithm is performed on L-level quantizers with $L = 2^h$, $h = 0, 1, \ldots$ until the initial guess for an N-level quantizer is achieved. This thesis is limited to such quantizers where $N = 2^h$, where h is a positive integer.

The algorithm, as outlined in [23], follows below. Let L represent the number of quantizer levels of the splitting algorithm, while $\hat{\mathcal{C}}_0$ is the initial reproduction alphabet or codebook.

Splitting Algorithm for Codebook Initialization

1. Splitting Algorithm Initialization: Set $L = 1$ and define the alphabet $\hat{\mathcal{C}}_0(1)$ as the centroid of the random vector or the training sequence if a sample distribution is used (as in this thesis).

2. Split: The L vectors $\{\mathbf{y}_i; i = 1, \ldots, L\}$ in the alphabet $\hat{\mathcal{C}}_0(L)$ are "split" into

two close vectors $\mathbf{y}_i+\epsilon$ and $\mathbf{y}_i-\epsilon$, where ϵ is a randomly generated perturbation vector. The result is a new alphabet $\tilde{\mathcal{C}}(L)$ containing $2L$ vectors $\{\mathbf{y}_i+\epsilon, \mathbf{y}_i-\epsilon; i = 1, \ldots, L\}$. Replace L by $2L$.

3. Condition: Is $L = N$? If so, set the codebook $\hat{\mathcal{C}}_0 = \tilde{\mathcal{C}}(L)$ and stop: $\hat{\mathcal{C}}_0$ will be the initial reproduction alphabet for the N-level quantization algorithm. If $L \neq N$, then the generalized Lloyd-Max algorithm should be run for the L-level quantizer with the $\tilde{\mathcal{C}}(L)$ alphabet until a good codebook $\hat{\mathcal{C}}_0(L)$ is produced. Return to step 2.

In the end, once the split algorithm is performed on a training sequence, initial codebooks can be found for quantizers with $1, 2, 4, 8, \ldots, N$ levels.

3.3.2 Simulated Annealing

An algorithm based on simulated annealing can be used to assign binary representations or codewords to the vector quantizer codevectors. Through proper codevector index assignment, the distortion due to channel noise can be minimized, and improvements over randomly selected codewords are shown in [13].

The annealing process was originally used in physics to grow crystals. Materials were melted by increasing the temperature to its melting point, then slowly cooled to allow crystal formation. In [13], the algorithm is also referred to as a Monte Carlo Metropolis algorithm for solving combinatorial optimization problems and also proven successful in designing source and channel codes, as shown in [11].

Let the MSE $D_c(b)$ represent the cost function to be minimized, with index as-

signment function $b = (b(\mathbf{c}_1), b(\mathbf{c}_2), \ldots, b(\mathbf{c}_N))$. Maintaining the physical terminology, the cost function will herein be referred to as the energy of the system and the index assignment function b will be the state. The algorithm follows, as stated in [13].

Simulated Annealing Algorithm

1. Set the initial 'effective temperature' T as $T = T_0$, and randomly choose an initial state b.

2. Next, choose b', a perturbation state of b, according to a random walk and let $\delta D_c = D_c(b') - D_c(b)$.

   (a) If $\delta D_c < 0$, replace b by b', proceed to step 3.

   (b) If $\delta D_c \geq 0$, replace b by b' with probability $\exp\{-\delta D_c/T\}$, proceed to step 3.

3. If the number of energy drops in step 2 exceeds a previously set number $\mathcal{N}$ or if too many perturbations do not result in energy drops, go to step 4; else, go to step 2.

4. Lower temperature by reducing factor α such that the new temperature is αT. If the temperature falls below a freezing point T_f or a stable state is reached, stop - with the final result being 'state b'. Otherwise, go to step 2.

Both the splitting algorithm and simulated annealing are used to develop a promising initial codebook for our soft-decision COVQ scheme. Table 3.1 lists the parameters used in the simulated annealing algorithm, as also chosen in [13].

T_0	10.0
T_f	2.5×10^{-4}
α	0.97
$\mathcal{N}$	200

Table 3.1: Simulated annealing parameters

3.4 Blahut's Algorithm for Channel Capacity

The channel capacity is the maximum rate at which information can be reliably transmitted. Easy computation of the channel capacity is not always possible because the channel transition probability matrix does not always have facilitating properties. Such a problem is encountered in our design of the soft-decision demodulation scheme for COVQ with an M-PAM constellation. Blahut's paper [7] gives an algorithm for computing channel capacity without making assumptions on the size nor properties of the channel transition matrix. By iteratively mapping the set of channel input probability vectors onto itself, the sequence of probability vectors converges to a vector that achieves the capacity of the given channel. The theorems and corollaries from [7] that allow such an algorithm to exist are included in Appendix A.

The algorithm uses the following equation form for the capacity of a channel:

$$C = \max_{p \in \mathbf{P}^n} I(p, Q) = \max_{p \in \mathbf{P}^n} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{\sum_a p_a Q_{k|a}} \tag{3.17}$$

where

$$\mathbf{P}^n = \{p \in \mathbb{R}^n : p_j \geq 0 \, \forall j; \sum_j p_j = 1\}$$

is the set of all possible probability distributions on the channel input and $\{Q_{k|j}\}$ is the channel transition probability matrix. A flow diagram of the capacity computation algorithm is shown in Figure 3.3.

Termination of the algorithm is decided based on two capacity-bound calculations. When the upper and lower bounds of capacity converge within a set threshold, the algorithm stops. These bounds are

$$C \geq \log \sum_j p_j c_j, \text{ and} \tag{3.18}$$

$$C \leq \log(\max_j c_j), \tag{3.19}$$

where c_j is defined as in Theorem A.4.

Independently and earlier than Blahut, Arimoto developed an iterative algorithm that monotonically converged from below to the exact capacity. Arimoto proved similar theorems in [2] that allows the capacity calculation.

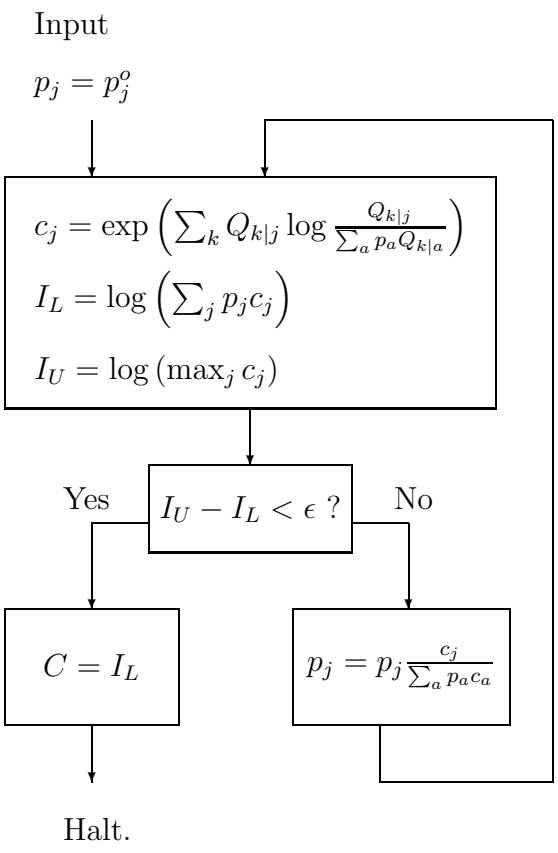

Figure 3.3: The Blahut algorithm for calculating capacity.

Chapter 4

Soft-Decision COVQ with M-PAM Modulation

As was mentioned in Chapter 3, the characteristic component behind the soft-decision scheme by Alajaji and Phamdo is the demodulator at the receiver. Their scheme exploits soft-decision information by having more codewords received than indices sent over the channel: receiving qkr bits per k-dimensional source vector represented (by kr bits), where r is the encoding rate of the COVQ. The channel transition probability matrix expresses this idea mathematically, considering the source dimension k, the COVQ rate r, the q-bit soft-decision decoder, the type of modulation scheme used, and finally, the type of channel over which signals are sent.

The design presented in this chapter extends the work of Alajaji and Phamdo from the binary phase shift keying (BPSK) constellation to any power-of-two number of symmetric constellation signals with varying amplitude in a one-dimensional signal space. This linear modulation technique is otherwise known as M-ary Pulse Am-

plitude Modulation (M-ary PAM), where M represents the number of signals. The soft-decision COVQ using an M-ary PAM constellation is designed for both AWGN and Rayleigh fading channels with the assumption that the statistical properties of the channel are known at the encoder and receiver.

4.1 DMC Channel Model

A block diagram of the soft-decision COVQ system for M-ary PAM signals is shown in Figure 4.1. For each k-dimensional source vector $\mathbf{V} \in \mathbb{R}^k$, the rate r COVQ

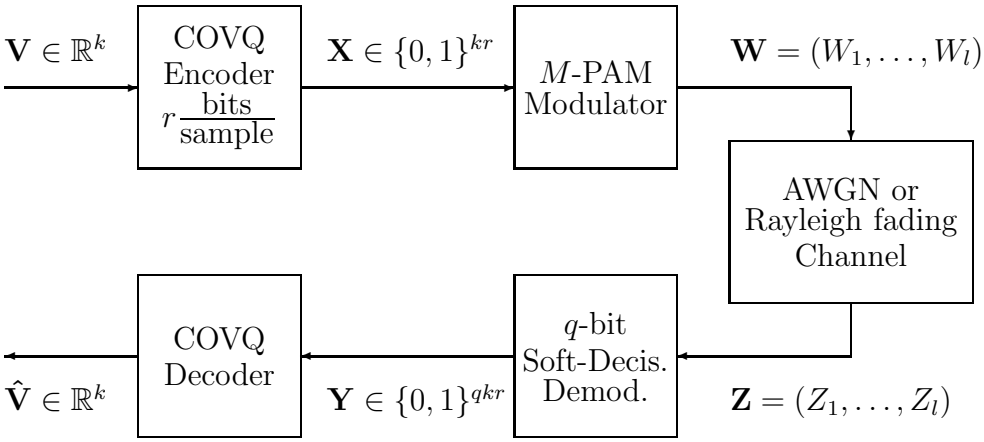

Figure 4.1: An SD-COVQ system with an M-ary PAM constellation.

encoder produces a binary vector $\mathbf{X} \in \{0,1\}^{kr}$. This kr-bit index represents the cell region in which the source vector belongs. Once the kr bits are output from the COVQ encoder, they are M-PAM modulated for channel transmission. When an M-ary PAM constellation is used, each of the M signals in $\{s_0, \ldots, s_{M-1}\}$ represents $\log_2 M$ bits; thus, $kr/\log_2 M$ signals are sent per source vector $\mathbf{V}$. We shall denote

the number of signals per source vector as l such that $l = kr/\log_2 M$. The kr value is assumed to be a multiple of $\log_2 M$; thus, l is always an integer. Thus, $\mathbf{W}$ is the l-dimensional vector $(W_1, \ldots, W_l) \in \{s_0, \ldots, s_{M-1}\}^l$ representing the signals sent per source vector $\mathbf{V}$.

The signals in the M-PAM constellation $\{s_0, \ldots, s_{M-1}\}$ have an average signal energy given by

$$E[W^2] = \sum_{i=0}^{M-1} p(s_i) {s_i}^2 \triangleq E_s \tag{4.1}$$

where $p(s_i)$ is the probability of signal s_i. The constellation signals have the form

$$s_i = [2i - (M-1)]a \qquad \text{for all } i = 0, \ldots, M-1 \tag{4.2}$$

where

$$a = \sqrt{\frac{M}{\sum_{i=0}^{M-1}(2i - (M-1))^2}} \tag{4.3}$$

is chosen such that $E_s = 1$ if the signals are uniformly distributed.

The M-ary PAM constellations used for $M = 2$, $M = 4$, and $M = 8$ were labeled using the natural ordering. These constellations are shown in Figures 4.2, 4.3, and 4.4, with the signals depicted as $\times$.

The M-PAM modulator, channel, and demodulator produce a 2^{kr}-input, 2^{qkr}-output DMC. Since the noise is i.i.d., the probability of receiving a codeword index, given an encoding index was sent is just the product of l "transmission" probabilities. The "transmission" probabilities correspond to the probability of receiving a particular demodulator index, given a particular M-PAM signal was sent. For example,

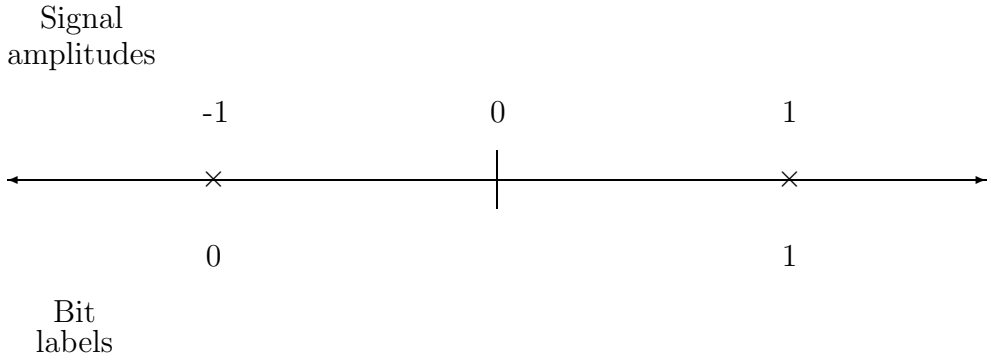

Figure 4.2: Signal constellation bit labels for $M = 2$.

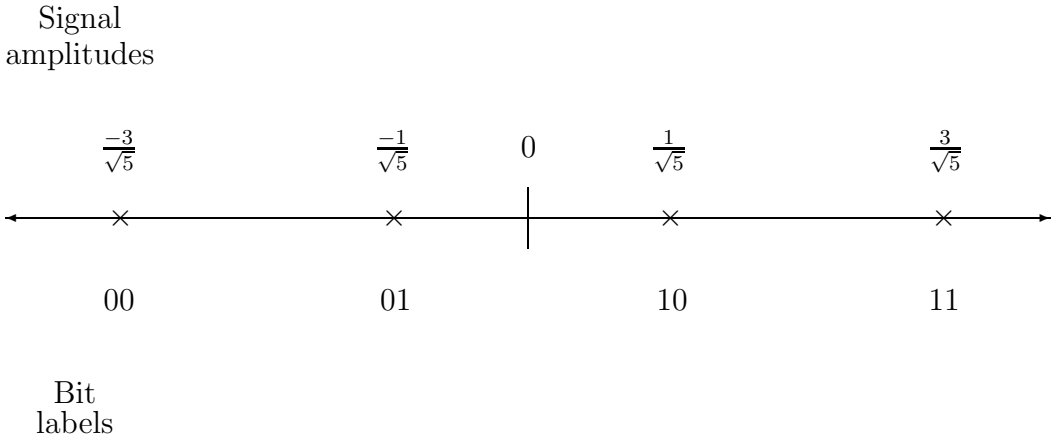

Figure 4.3: Signal constellation bit labels for $M = 4$.

when $M = 4$ and $k = r = q = 2$, $p(11010110|1100) = p(1101|11)p(0110|00)$. Thus, the channel is equivalent to an M-input M^q-output DMC used l times.

The channel transition matrix entries $\pi_{i,j} = \Pr(J = j|I = i)$ are determined by the type of channel and the quantizer thresholds T_j at the receiver, where I is the index of the sent M-PAM signal and J is the index of the received signal's region.

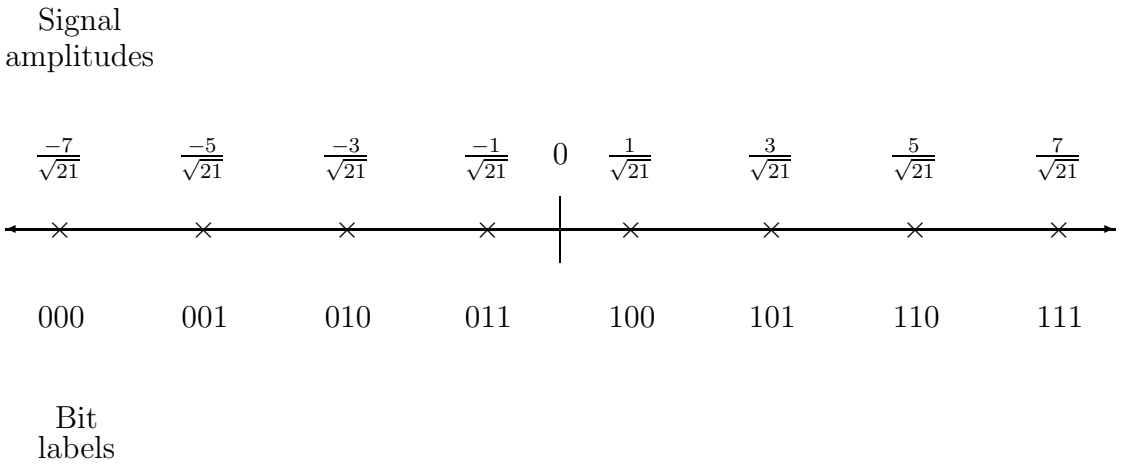

Figure 4.4: Signal constellation bit labels for $M = 8$.

The channel transition matrix entries can be written as

$$\pi_{i,j} \triangleq \Pr(J = j | I = i), \quad i \in \mathcal{I}, j \in \mathcal{J} \tag{4.4}$$

$$= \Pr\{Z \in (T_{j-1}, T_j) | W = s_i\} \tag{4.5}$$

where $W = s_i$ is the sent M-PAM signal, I is the index of the sent M-PAM signal, with $\mathcal{I} = \{0, 1, \ldots, M-1\}$, J is the index of the region in which the received signal lies, with $\mathcal{J} = \{0, 1, \ldots, M^q - 1\}$, T_j are the thresholds defined in Equation (4.7), and finally Z is the received M-PAM signal. The thresholds distinguish the partition cells of the decoder and are dependent on the type of modulation used to send the signals, the number of bits at which the soft-decision demodulator operates (q bits), and the uniform scalar quantizer step-size Δ at the receiver.

At the soft-decision demodulator, the received signal is decoded with the uniform scalar quantizer $\alpha(\cdot)$ with step-size Δ such that

$$\alpha(z) = j, \quad \text{if } z \in (T_{j-1}, T_j) \tag{4.6}$$

where the thresholds of the quantizer cells are defined as

$$T_j = \begin{cases} -\infty, & \text{if } j = -1 \\ (j + 1 - \frac{M^q}{2})\Delta, & \text{if } j = 0, 1, \ldots, M^q - 2 \\ +\infty, & \text{if } j = M^q - 1. \end{cases} \tag{4.7}$$

The thresholds T_j are dependent on the uniform quantizer step-size Δ. At a particular channel noise variance σ^2, the Δ is chosen to maximize the capacity of the channel because i) there is a correlation between an increase in capacity and an increase in the overall system SDR, and ii) optimizing the system in terms of maximizing capacity is easier than in terms of maximizing SDR, as stated in [25]. The transition matrix entries (dependent on the thresholds, thus Δ) are used to calculate the channel capacity. Therefore, the step-size Δ is chosen to yield the maximum capacity of the M-input M^q-output DMC. The capacity is calculated through the iterative algorithm by Blahut, outlined in Section 3.4.

Figure 4.5 illustrates a soft-decision demodulator for a specific example where $M = 4$ signals are in the M-ary PAM constellation, depicted as $\times$. In this example, $\log_2 M = 2$ bits per signal are sent, but $q \log_2 M = 4$ bits are received per M-PAM signal sent, and $k = r = 2$. Thus, the received signal is demodulated into one of 16 cells. The quantizer step-size $\Delta = 0.22$ to provide a capacity of $C = 1.15$ bits/channel use over an AWGN channel with $E_N = -6.0$ dB. The noise power is defined as

$$E_N \triangleq 10 \log_{10} \sigma^2 \text{ (dB)} \tag{4.8}$$

where σ^2 is the channel noise variance.

At the receiver, the elements of the received vector $\mathbf{Z} \in \mathbb{R}^l$ represent the soft

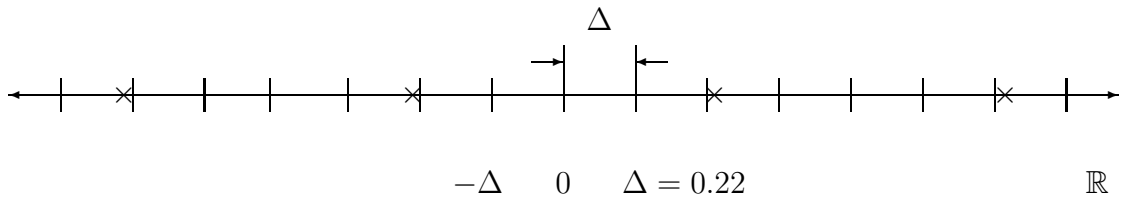

Figure 4.5: A soft-decision demodulator. For $M = 4$, $k = r = q = 2$, and $E_N = -6.0$ dB, $\Delta = 0.22$ over an AWGN channel.

information of each M-ary PAM signal received. The soft-decision demodulator uses the soft information to produce $q \log_2 M$ bits per M-ary PAM signal received. Since $l = kr/\log_2 M$ signals are sent per source vector,

$$\frac{kr}{\log_2 M} \frac{\text{signals}}{\text{source vector}} \times q \log_2 M \frac{\text{bits}}{\text{signal}} = qkr \frac{\text{bits}}{\text{source vector}}. \tag{4.9}$$

Thus, the soft-decision demodulator produces qkr bits per k-dimensional source vector. Technically speaking, each M-ary PAM signal is demodulated into $q \log_2 M$ bits through a uniform scalar quantizer with step-size Δ. Pertaining to each source vector, l signals are received this way, to be concatenated into a qkr-bit word $\mathbf{Y} \in \{0,1\}^{qkr}$ through the soft-decision demodulator. The qkr bits represent an index corresponding to one of 2^{qkr} codewords $\hat{\mathbf{V}} \in \mathbb{R}^k$. This conversion is done through the COVQ decoder using a look-up table or codebook, predetermined through COVQ training.

4.2 SD-COVQ Design

The soft-decision COVQ design (used to determine the optimal partition $\mathcal{P}$ and codebook $\mathcal{C}$) is similar to the iterative algorithm presented in [23], using equations from

[14] and the channel transition matrix Π.

For a real-valued i.i.d. source $\mathcal{V} = \{V_i\}_{i=1}^{\infty}$ with pdf $f(v)$, the COVQ encoder produces kr bits for each k-dimensional sequence. The kr bits are to be transmitted over the 2^{kr} -input, 2^{qkr}-output DMC with transition probability distribution $P(\mathbf{y}|\mathbf{x}) = \prod_{h=1}^{l} \pi_{x_h,y_h}$, where $\mathbf{x} \in \mathcal{I}^l$ and $\mathbf{y} \in \mathcal{J}^l$.

Figure 4.6 illustrates a generic block diagram of a COVQ system. As we recall from Chapter 3, the COVQ encoder and decoder blocks represent encoder and decoder functions.

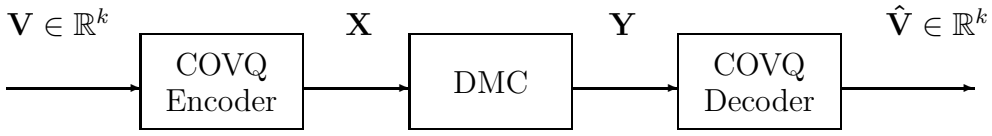

Figure 4.6: A generic block diagram of the COVQ system.

The encoder mapping $\gamma\colon \mathbb{R}^k \mapsto \mathcal{I}^l$ can be described by the partition $\mathcal{P} = \{S_{\mathbf{x}} \subset \mathbb{R}^k\colon \mathbf{x} \in \mathcal{I}^l\}$, which divides $\mathbb{R}^k$ into 2^{kr} subsets since $|\mathcal{I}^l| = 2^{kr}$. The encoder mapping is defined as $\gamma(\mathbf{v}) = \mathbf{x}$ if $\mathbf{v} \in S_{\mathbf{x}}, \mathbf{x} \in \mathcal{I}^l$, where $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ is a block of k successive source samples. The 2^{kr}-input 2^{qkr}-output DMC maps the kr-bit sequence $\mathbf{x}$ to a qkr-bit sequence $\mathbf{y}$ and is described by the channel transition probability matrix $P(\mathbf{y}|\mathbf{x})$. The decoder mapping, denoted as β, is described by the codebook $\mathcal{C} = \{\mathbf{c}_{\mathbf{y}} \in \mathbb{R}^k\colon \mathbf{y} \in \mathcal{J}^l\}$ which is previously obtained through COVQ training using Lloyd's algorithm. The decoder mapping is expressed as $\beta(\mathbf{y}) = \mathbf{c}_{\mathbf{y}}, \mathbf{y} \in \mathcal{J}^l$.

The encoding rate of the COVQ system is r bits per sample and its average

squared-error distortion per sample is given by [14]

$$D = \frac{1}{k} \sum_{\mathbf{x}} \int_{S_{\mathbf{x}}} \left\{ \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \| - \mathbf{c}_{\mathbf{y}} \|^2 \right\} f(\mathbf{v})\, d\mathbf{v} \tag{4.10}$$

where $f(\mathbf{v})$ is the k-dimensional source pdf. The discrete form of this equation, as used in the design of the practical system, is

$$D = \frac{1}{n} \frac{1}{k} \sum_{\mathbf{x}} \sum_{\mathbf{v} \in S_{\mathbf{x}}} \left\{ \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \| \mathbf{v} - \mathbf{c}_{\mathbf{y}} \|^2 \right\} \tag{4.11}$$

where n is the number of training vectors. For a given source, channel, k and r, we wish to minimize D by finding the most suitable $\mathcal{P}$ and $\mathcal{C}$.

For a fixed $\mathcal{C}$, the optimal $\mathcal{P} = \{S_{\mathbf{x}}\}$ that minimizes (4.10) is [14]

$$S_{\mathbf{x}} = \left\{ \mathbf{v} \colon \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \| \mathbf{v} - \mathbf{c}_{\mathbf{y}} \|^2 \leq \sum_{\mathbf{y}} P(\mathbf{y}|\tilde{\mathbf{x}}) \| \mathbf{v} - \mathbf{c}_{\mathbf{y}} \|^2 \quad \forall \tilde{\mathbf{x}} \in \mathcal{I}^l \right\} \tag{4.12}$$

for $\mathbf{x} \in \mathcal{X}^{kr}$. The optimal codebook $\mathcal{C} = \{\mathbf{c}_{\mathbf{y}}\}$ for a given partition is [14]

$$\mathbf{c}_{\mathbf{y}} = \frac{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \int_{S_{\mathbf{x}}} \mathbf{v} f(\mathbf{v})\, d\mathbf{v}}{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \int_{S_{\mathbf{x}}} f(\mathbf{v})\, d\mathbf{v}}. \tag{4.13}$$

In discrete form, the codewords are calculated as

$$\mathbf{c}_{\mathbf{y}} = \frac{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \sum_{\mathbf{v} \in S_{\mathbf{x}}} \mathbf{v} \ \Pr(\mathbf{v})}{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \ \Pr(\mathbf{v} \in S_{\mathbf{x}})} \tag{4.14}$$

$$= \frac{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \sum_{\mathbf{v} \in S_{\mathbf{x}}} \mathbf{v}}{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) |S_{\mathbf{x}}|}. \tag{4.15}$$

The two equations (4.12) and (4.15) are used to derive the partition and codebook in the generalized Lloyd-Max algorithm. The above design can be easily extended for the case of sources with memory.

4.3 SD-COVQ with M-PAM Signals over AWGN Channels

For an AWGN Channel, the received signal Z_t is

$$Z_t = W_t + N_t, \quad t = 0, \ldots, l \tag{4.16}$$

where W_t is the sent signal from the M-ary PAM constellation and N_t is the i.i.d. noise process, distributed as $\mathcal{N}(0, \sigma^2)$, and independent of W_t.

The thresholds of the soft-decoder at the receiver take into account which modulation scheme is used and the number of bits at which the soft-decision demodulator operates, which is q. For the AWGN channel, the thresholds are exactly as presented in Equation (4.7). If $\mathcal{I} = \{0, 1, \ldots, M-1\}$ and $\mathcal{J} = \{0, 1, \ldots, M^q - 1\}$, then the channel transition matrix $\mathbf{\Pi}$ is given by

$$\mathbf{\Pi} = [\pi_{i,j}], \quad i \in \mathcal{I}, j \in \mathcal{J}. \tag{4.17}$$

The matrix entries are calculated as follows.

$$\begin{aligned}
\pi_{i,j} &= P(J = j | I = i) \\
&= P(Z > T_{j-1} \text{ and } Z < T_j | W = s_i) \\
&= P(W + N > T_{j-1} \text{ and } W + N < T_j | W = s_i) \\
&= P(N < T_j - s_i) - P(N < T_{j-1} - s_i) \\
&= Q\left((T_{j-1} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((T_j - s_i)\sqrt{\frac{1}{\sigma^2}}\right)
\end{aligned} \tag{4.18}$$

where s_i is the modulated signal sent with index i, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\{-t^2/2\}\, dt$, and σ^2 is the channel noise variance.

We can show that the channel transition matrix is *centrosymmetric*, meaning the elements are symmetric about the center of the matrix such that

$$a_{i,j} = a_{m-1-i,n-1-j} \tag{4.19}$$

where $a_{i,j}$ are the entries of an $m \times n$ matrix indexed $i = 0, \ldots, m-1$ and $j = 0, \ldots, n-1$. The following facts are used in upcoming proofs of centrosymmetric matrices. The signal amplitudes are symmetric in the sense that $s_{M-1-i} = -s_i$ for $i = 0, \ldots, M-1$: from our definition of s_i in Equation (4.2),

$$\begin{aligned} s_{M-1-i} &= [2(M-1-i) - (M-1)]a \\ &= [2M - 2 - 2i - M + 1]a \\ &= [-2i + M - 1]a \\ &= -[2i - (M-1)]a \\ &= -s_i. \end{aligned}$$

Also, the thresholds are symmetric in the sense that $T_{M^q-1-j} = -T_j$ for $j = 0, \ldots, M^q - 1$: from our definition of T_j in Equation (4.7),

$$\begin{aligned} T_{M^q-1-j} &= \left((M^q - 1 - j) + 1 - \frac{M^q}{2}\right)\Delta \\ &= \left(\frac{2M^q}{2} - \frac{M^q}{2} - 1 + 1 - j\right)\Delta \\ &= -\left(j - \frac{M^q}{2}\right)\Delta \\ &= -\left((j-1) + 1 - \frac{M^q}{2}\right)\Delta \\ &= -T_{j-1}. \end{aligned}$$

We next prove the centrosymmetric property for the AWGN channel; i.e., that $\pi_{i,j} = \pi_{M-1-i,M^q-1-j}$, beginning with Equation (4.18).

Lemma 4.3.1. *The channel transition matrix of a soft-decision M-PAM demodulated AWGN channel is centrosymmetric.*

Proof. For the case when $i = 0, \ldots, M-1$ and $j = 1, \ldots, M^q - 2$,

$$\begin{aligned}
\pi_{i,j} &= Q\left((T_{j-1} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((T_j - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= Q\left((-T_{M^q-1-j} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((-T_{M^q-2-j} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&\qquad\qquad \text{since } T_{M^q-1-j} = -T_{j-1} \\
&= Q\left((-T_{M^q-1-j} + s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((-T_{M^q-2-j} + s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) \\
&\qquad\qquad \text{since } s_{M-1-i} = -s_i \\
&= Q\left(-(T_{M^q-1-j} - s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) - Q\left(-(T_{M^q-2-j} - s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) \\
&= Q\left(-(T_{M^q-2-j} - s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) - Q\left(-(T_{M^q-1-j} - s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) \\
&\qquad\qquad \text{since } Q(x_1) - Q(x_2) = Q(-x_2) - Q(-x_1) \\
&= Q\left((T_{(M^q-1-j)-1} - s_{M-i-1})\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((T_{M^q-1-j} - s_{M-i-1})\sqrt{\frac{1}{\sigma^2}}\right) \\
&= \pi_{M-1-i,M^q-1-j}.
\end{aligned}$$

For the case of $i = 0, \ldots, M-1$ and $j = 0$, using Equation (4.7),

$$\begin{aligned}
\pi_{i,0} &= Q\left((T_{-1} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((T_0 - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= 1 - Q\left((T_0 - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= Q\left(-(T_0 - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } 1 - Q(x) = Q(-x) \\
&= Q\left(-(-T_{M^q-2-j} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } T_{M^q-1-j} = -T_{j-1} \\
&= Q\left(-(-T_{M^q-2-j} + s_{M-i-1})\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } s_{M-1-i} = -s_i \\
&= Q\left((T_{M^q-2-j} - s_{M-i-1})\sqrt{\frac{1}{\sigma^2}}\right) \\
&= \pi_{M-1-i,M^q-1}.
\end{aligned}$$

Similarly, for $i = 0, \ldots, M-1$ and $j = M^q - 1$:

$$\begin{aligned}
\pi_{i,M^q-1} &= Q\left((T_{M^q-2} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) - Q\left((T_{M^q-1} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= Q\left((T_{M^q-2} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= 1 - Q\left(-(T_{M^q-2} - s_i)\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } 1 - Q(x) = Q(-x) \\
&= 1 - Q\left((-T_{M^q-2} + s_i)\sqrt{\frac{1}{\sigma^2}}\right) \\
&= 1 - Q\left((T_0 + s_i)\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } T_{M^q-1-j} = -T_{j-1} \\
&= 1 - Q\left((T_0 - s_{M-1-i})\sqrt{\frac{1}{\sigma^2}}\right) \text{ since } s_{M-1-i} = -s_i \\
&= \pi_{M-1-i,0}.
\end{aligned}$$

□

4.4 SD-COVQ with M-PAM Signals over Rayleigh Fading Channels

Recall the construction of a Rayleigh fading channel as

$$Z_t = A_t W_t + N_t, \quad t = 0, \ldots, l \tag{4.20}$$

where A_t denotes the attenuation factor and N_t is the noise process, an i.i.d. Gaussian random process with zero-mean and σ^2 variance; N_t, A_t, and W_t are all independent of each other. The attenuation factor A_t itself is also i.i.d. with Rayleigh pdf

$$f_A(a) = \begin{cases} 2ae^{-a^2}, & \text{if } a > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4.21}$$

The thresholds of the quantizer follow from Equation (4.7). If $\mathcal{I} = \{0, 1, \ldots, M-1\}$ and $\mathcal{J} = \{0, 1, \ldots, M^q - 1\}$, then the channel transition matrix $\mathbf{\Pi}$ is given by

$$\mathbf{\Pi} = [\pi_{i,j}], \quad i \in \mathcal{I}, j \in \mathcal{J}. \tag{4.22}$$

The following is a breakdown of the channel transition matrix entries.

First, for $s_i > 0$, we have

$$\begin{aligned}
\pi_{i,j} &= P(J = j | I = i) \\
&= P(Z > T_{j-1} \text{ and } Z < T_j | W = s_i) \\
&= E_A[P(A \cdot W + N > T_{j-1} \text{ and } A \cdot W + N < T_j | W = s_i)] \\
&= E_A[P(N < T_j - A \cdot W | W = s_i) - P(N < T_{j-1} - A \cdot W | W = s_i)] \\
&= E_A[P(N < T_j - A \cdot s_i) - P(N < T_{j-1} - A \cdot s_i)] \\
&= E_A[P(N < T_j - A \cdot s_i)] - E_A[P(N < T_{j-1} - A \cdot s_i)] \qquad (4.23) \\
&= E_A\left[P\left(\frac{N}{s_i} < \frac{T_j}{s_i} - A\right)\right] - E_A\left[P\left(\frac{N}{s_i} < \frac{T_{j-1}}{s_i} - A\right)\right] \qquad (4.24)
\end{aligned}$$

where s_i is the modulated signal sent with index i. From [32],

$$\begin{aligned}
E_A[\Pr\{N \le z - A\}] = 1 - \frac{1}{2}\operatorname{erfc}\left(\frac{z}{\sqrt{2\sigma^2}}\right) - \frac{1}{\sqrt{2\sigma^2+1}} \\
\times \left[1 - \frac{1}{2}\operatorname{erfc}\left(\frac{z}{\sqrt{2(2\sigma^2+1)\sigma^2}}\right)\right] e^{-z^2/(2\sigma^2+1)} \qquad (4.25)
\end{aligned}$$

where $\operatorname{erfc}(x) \triangleq (2/\sqrt{\pi}) \int_x^\infty e^{-t^2}\, dt$. Since $Q(x) = \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$,

$$\begin{aligned}
\Lambda(z,\sigma) \triangleq E_A[\Pr\{N \le z - A\}] = 1 - Q\left(\frac{z}{\sqrt{\sigma^2}}\right) - \frac{1}{\sqrt{2\sigma^2+1}} \\
\times \left[1 - Q\left(\frac{z}{\sqrt{(2\sigma^2+1)\sigma^2}}\right)\right] e^{-z^2/(2\sigma^2+1)}. \qquad (4.26)
\end{aligned}$$

Thus, the matrix entries can be written as

$$\pi_{i,j} = \Lambda\left(\frac{T_j}{s_i}, \frac{\sigma}{s_i}\right) - \Lambda\left(\frac{T_{j-1}}{s_i}, \frac{\sigma}{s_i}\right), \text{ for } s_i > 0. \qquad (4.27)$$

To get the expression for $\pi_{i,j}$ when $s_i < 0$, beginning with Equation (4.23), we obtain

$$\begin{aligned}
\pi_{ij} &= E_A[P(N < T_j - A \cdot s_i)] - E_A[P(N < T_{j-1} - A \cdot s_i)] \\
&= E_A\left[P\left(\frac{N}{s_i} > \frac{T_j}{s_i} - A\right)\right] - E_A\left[P\left(\frac{N}{s_i} > \frac{T_{j-1}}{s_i} - A\right)\right] \\
&= E_A\left[1 - P\left(\frac{N}{s_i} < \frac{T_j}{s_i} - A\right)\right] - E_A\left[1 - P\left(\frac{N}{s_i} < \frac{T_{j-1}}{s_i} - A\right)\right] \\
&= E_A\left[P\left(\frac{N}{s_i} < \frac{T_{j-1}}{s_i} - A\right)\right] - E_A\left[P\left(\frac{N}{s_i} < \frac{T_j}{s_i} - A\right)\right] \\
&= \Lambda\left(\frac{T_{j-1}}{s_i}, \frac{\sigma}{s_i}\right) - \Lambda\left(\frac{T_j}{s_i}, \frac{\sigma}{s_i}\right), \ s_i < 0. \qquad (4.28)
\end{aligned}$$

We next prove the centrosymmetry property of $\pi_{i,j}$ for the Rayleigh fading case. Since we will be using the fact that $s_{M-1-i} = -s_i$, we will start with the expression for negative s_i and end at the expression for positive s_i. Thus, we will prove centrosymmetry for $i = 0, \ldots, \frac{M}{2} - 1$. A similar proof can be done for $i = \frac{M}{2}, \ldots, M-1$.

Lemma 4.4.1. *The channel transition matrix of a soft-decision M-PAM demodulated Rayleigh fading channel is centrosymmetric.*

Proof. Note, since $T_{M^q-1-j} = -T_{j-1}$ and $s_{M-1-i} = -s_i$,

$$\frac{T_{j-1}}{s_i} = \frac{T_{M^q-1-j}}{s_{M-1-i}} \qquad (4.29)$$

and

$$\left(\frac{\sigma}{s_i}\right)^2 = \left(\frac{\sigma}{s_{M-1-i}}\right)^2. \qquad (4.30)$$

For the case when $i = 0, \ldots, \frac{M}{2} - 1$ and $j = 1, \ldots, M^q - 2$ and using Equation (4.28),

$$
\begin{aligned}
\pi_{i,j} &= \Lambda\left(\frac{T_{j-1}}{s_i}, \frac{\sigma}{s_i}\right) - \Lambda\left(\frac{T_j}{s_i}, \frac{\sigma}{s_i}\right) \\
&= 1 - Q\left(\frac{T_{j-1}/s_i}{\sqrt{(\sigma/s_i)^2}}\right) - \frac{1}{\sqrt{2(\sigma/s_i)^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_{j-1}/s_i}{\sqrt{(2(\sigma/s_i)^2 + 1)(\sigma/s_i)^2}}\right)\right] e^{-(T_{j-1}/s_i)^2/[(\sigma/s_i)^2+1]} \\
&- 1 + Q\left(\frac{T_j/s_i}{\sqrt{(\sigma/s_i)^2}}\right) + \frac{1}{\sqrt{2(\sigma/s_i)^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_j/s_i}{\sqrt{(2(\sigma/s_i)^2 + 1)(\sigma/s_i)^2}}\right)\right] e^{-(T_j/s_i)^2/[(\sigma/s_i)^2+1]} \\
&= 1 - Q\left(\frac{T_{M^q-1-j}/s_{M-1-i}}{\sqrt{(\sigma/s_{M-1-i})^2}}\right) - \frac{1}{\sqrt{2(\sigma/s_{M-1-i})^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_{M^q-1-j}/s_{M-1-i}}{\sqrt{(2(\sigma/s_{M-1-i})^2 + 1)(\sigma/s_{M-1-i})^2}}\right)\right] \\
&\times e^{-(T_{M^q-1-j}/s_{M-1-i})^2/[(\sigma/s_{M-1-i})^2+1]} \\
&- 1 + Q\left(\frac{T_{M^q-2-j}/s_{M-1-i}}{\sqrt{(\sigma/s_{M-1-i})^2}}\right) + \frac{1}{\sqrt{2(\sigma/s_{M-1-i})^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_{M^q-2-j}/s_{M-1-i}}{\sqrt{(2(\sigma/s_{M-1-i})^2 + 1)(\sigma/s_{M-1-i})^2}}\right)\right] \\
&\times e^{-(T_{M^q-2-j}/s_{M-1-i})^2/(\sigma/[s_{M-1-i})^2+1]} \\
&= \Lambda\left(\frac{T_{M^q-1-j}}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) - \Lambda\left(\frac{T_{M^q-2-j}}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) \\
&= \pi_{M-1-i, M^q-1-j},
\end{aligned}
$$

where the last equality follows from the expression of the transition probability for a positive s_i in Equation (4.27) since $s_{M-1-i} > 0$.

For the case where $i = 0, \ldots, \frac{M}{2} - 1$ and $j = 0$, since $s_i < 0$, we have

$$\begin{aligned}
\pi_{i,0} &= \Lambda\left(\frac{T_{-1}}{s_i}, \frac{\sigma}{s_i}\right) - \Lambda\left(\frac{T_0}{s_i}, \frac{\sigma}{s_i}\right) \\
&= 1 - 1 + Q\left(\frac{T_0/s_i}{\sqrt{(\sigma/s_i)^2}}\right) + \frac{1}{\sqrt{2(\sigma/s_i)^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_0/s_i}{\sqrt{(2(\sigma/s_i)^2 + 1)(\sigma/s_i)^2}}\right)\right] e^{-(T_0/s_i)^2/[(\sigma/s_i)^2+1]} \\
&= 1 - 1 + Q\left(\frac{T_{M^q-2}/s_{M-1-i}}{\sqrt{(\sigma/s_{M-1-i})^2}}\right) + \frac{1}{\sqrt{2(\sigma/s_{M-1-i})^2 + 1}} \\
&\times \left[1 - Q\left(\frac{T_{M^q-2}/s_{M-1-i}}{\sqrt{(2(\sigma/s_{M-1-i})^2 + 1)(\sigma/s_{M-1-i})^2}}\right)\right] \\
&\times e^{-(T_{M^q-2}/s_{M-1-i})^2/[(\sigma/s_{M-1-i})^2+1]} \\
&= \Lambda\left(\frac{T_{M^q-1}}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) - \Lambda\left(\frac{T_{M^q-2}}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) \\
&= \pi_{M-1-i,M^q-1},
\end{aligned}$$

where, again, the last equality follows from the transition probability expression for a positive s_i since $s_{M-1-i} > 0$.

For the case where $i = 0, \ldots, \frac{M}{2} - 1$ and $j = M^q - 1$, since $s_i < 0$, we have

$$\begin{aligned}
\pi_{i,M^q-1} &= \Lambda\left(\frac{T_{M^q-2}}{s_i}, \frac{\sigma}{s_i}\right) - \Lambda\left(\frac{T_{M^q-1}}{s_i}, \frac{\sigma}{s_i}\right) \\
&= 1 - Q\left(\frac{T_{M^q-2}/s_i}{\sqrt{(\sigma/s_i)^2}}\right) - \frac{1}{\sqrt{2(\sigma/s_i)^2+1}} \\
&\times \left[1 - Q\left(\frac{T_{M^q-2}/s_i}{\sqrt{(2(\sigma/s_i)^2+1)(\sigma/s_i)^2}}\right)\right] e^{-(T_{M^q-2}/s_i)^2/[(\sigma/s_i)^2+1]} \\
&= 1 - Q\left(\frac{T_0/s_i}{\sqrt{(\sigma/s_{M-1-i})^2}}\right) - \frac{1}{\sqrt{2(\sigma/s_{M-1-i})^2+1}} \\
&\times \left[1 - Q\left(\frac{T_0/s_{M-1-i}}{\sqrt{(2(\sigma/s_{M-1-i})^2+1)(\sigma/s_{M-1-i})^2}}\right)\right] \\
&\times e^{-(T_0/s_{M-1-i})^2/[(\sigma/s_{M-1-i})^2+1]} \\
&= \Lambda\left(\frac{T_0}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) - \Lambda\left(\frac{T_{-1}}{s_{M-1-i}}, \frac{\sigma}{s_{M-1-i}}\right) \\
&= \pi_{M-1-i,0},
\end{aligned}$$

where the last line is the transition probability for a positive s_i since $s_{M-1-i} > 0$. Similar proofs hold for $i = \frac{M}{2}, \ldots, M-1$ where $s_i > 0$. □

Chapter 5

Numerical Results and Discussion

The soft-decision COVQ (SD-COVQ) system design presented in the previous chapter was implemented over AWGN and Rayleigh fading channels. In this chapter, we will present the numerical results and discuss them. Various values of the source vector dimension k, COVQ rate r, number of modulation signals M, and soft-decision decoder rate q were combined to form COVQ systems for analysis. Both sources with and without memory were tested. The performance of the M-ary PAM modulated SD-COVQ systems in this thesis were measured in overall source signal-to-distortion ratio (SDR).

5.1 Design Parameters

This section discusses the parameters used in the design of the M-ary PAM modulated SD-COVQ system. For a particular M, k, r, and q, the quantizer at the receiver is determined by selecting the step-size Δ which maximizes the capacity at specified

noise variances. Recall that Δ is chosen this way because increases in capacity typically yield increases in SDR and it is easier than finding Δ under the maximum SDR criterion [25]. The capacity is calculated using Blahut's algorithm, described in Chapter 3. Once Δ is determined, the channel transition probability matrix is computed using Equations (4.18) or (4.27). The transition matrix, along with other variables, is used in the codebook training process in computing the overall distortion function to be minimized.

At each specified noise variance σ^2, the codebooks are trained using $100,000$ k-dimensional vectors, using a memoryless Gaussian source, or a Gauss-Markov source with correlation coefficient $\rho = 0.9$. The first codebook is initialized using the splitting algorithm described in Chapter 3 until 2^{qkr} codewords are computed with just the knowledge of the training source vectors.

Next, simulated annealing [13] is performed on the codebook to determine a good decoder index-codeword pairing. Simulated annealing perturbs the index-codeword assignment by interchanging a random index pair in order to minimize the cost function, which is the overall distortion in Equation (4.10). According to [4], the cost function can be further reduced to

$$\sum_{i=0}^{2^{kr}-1} p(i) \sum_{j=0}^{2^{qkr}-1} p(j|i) \langle \mathbf{c}_{b(j)}, (\mathbf{c}_{b(j)} - 2\mathbf{m}_i) \rangle \tag{5.1}$$

where 2^{kr} is the number of encoding regions, 2^{qkr} is the number of codewords at the receiver, $p(j|i)$ is the probability of receiving index j given i was sent, $\mathbf{m}_i$ is the centroid of encoding region S_i, $\mathbf{c}_j$ is the j-th codeword at the receiver, $\langle \mathbf{c}, \mathbf{m} \rangle = \sum_i c_i m_i$ is the inner product function and finally, b: $\{0, \ldots, 2^{qkr} - 1\} \rightarrow \{0, \ldots, 2^{qkr} - 1\}$ is

the one-to-one decoder index mapping to be altered in simulated annealing. The cost function in Equation (5.1) is used during simulated annealing prior to codebook training.

Once simulated annealing is applied to the initial codebook of the lowest noise power, a method similar to that used in [14] is used to train soft-decision COVQ codebooks at specified noise powers. In [14], Farvardin and Vaishampayan use the idea of passing trained codebooks for channels with small to large p_e, where p_e is the crossover probability of a binary symmetric channel. This method is also used for training codebooks in [4], except with high-to-low-to-high channel SNRs. Our codebook initialization begins with the lowest specified noise power E_N; codebooks are trained using the generalized Lloyd-Max algorithm and the resulting codebook is passed to act as initial codebooks for channels with higher noise variances. The process is repeated with descending noise power. At each noise variance, the generalized Lloyd-Max algorithm terminates once the relative distortion falls below a certain threshold ϵ, chosen to be $\epsilon = 10^{-3}$ in our simulations. The performance of the system is measured in terms of source SDR, with numerical results obtained from the training set.

5.2 Experimental Results

The capacity of the SD-COVQ system is computed using the channel transition probability matrix. An observed centrosymmetric transition matrix is given below with

approximate entries for a 4×16 DMC with AWGN, where $M = 4$ and $q = 2$ at $E_N = -6.0$ dB.

$$\mathbf{\Pi} = \begin{pmatrix} 0.4 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.1 & 0.1 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.1 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.1 & 0.1 & 0.2 & 0.2 & 0.4 \end{pmatrix}. \tag{5.2}$$

5.2.1 Results for AWGN Channels

We list the capacity for various AWGN soft-decision COVQ system parameters at specified noise powers in Tables 5.1 to 5.3. Since the capacity is only dependent on the $M \times M^q$ transition matrix, the capacity values for a given number of modulated signals M are the same for all acceptable k and r. For all q, the capacity approaches $\log_2 M$ bits per channel use as the channel noise variance σ^2 decreases; this is expected since each signal sends $\log_2 M$ bits at each channel use. Tables 5.1 to 5.3 show that most of the capacity gain occurs in going from $q = 1$ to $q = 2$. Only a small increase in capacity is seen from $q = 2$ to $q = 3$. As q increases, the general trend of improvement in channel capacity suggests a similar trend for the source SDR of the soft-decision COVQ system.

In the BPSK case where $M = 2$, the transition matrix of the DMC is weakly symmetric [1]; thus, the maximum capacity is achieved by a uniform distribution on the DMC's input. In the M-ary PAM case where $M = 2^h$ with h as a positive integer and $h > 1$, the absence of these symmetric properties implies a different probability

distribution on the channel input vector. The input probabilities for $M = 8$ for each noise variance σ^2 are presented in Tables 5.4 to 5.6. Again, these input probabilities maximize the capacity of the DMC for a particular noise power E_N, q, and Δ. For all q, the capacity-maximizing input distribution appears to favour the outer-most signals when the channel is very noisy. For hard-decision decoding, i.e. when $q = 1$, the input probabilities approach a uniform distribution for small noise powers. Similar observations can be made for the $M = 4$ case.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.206	0.267	1.281	0.284	0.723
2.0	0.252	0.322	1.124	0.341	0.633
1.0	0.306	0.385	0.985	0.406	0.553
0.0	0.369	0.455	0.860	0.477	0.482
-1.0	0.440	0.532	0.750	0.554	0.419
-2.0	0.518	0.612	0.652	0.634	0.363
-3.0	0.602	0.693	0.566	0.713	0.314
-4.0	0.687	0.770	0.491	0.788	0.272
-6.0	0.842	0.898	0.366	0.908	0.201
-8.0	0.947	0.971	0.272	0.975	0.147
-10.0	0.991	0.996	0.201	0.997	0.107
-12.0	0.999	1.000	0.195	1.000	0.065
-14.0	1.000	1.000	0.062	1.000	0.062
-16.0	1.000	1.000	0.005	1.000	0.002

Table 5.1: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for an AWGN channel with $M = 2$.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.409	0.443	0.392	0.445	0.122
2.0	0.487	0.519	0.342	0.521	0.107
1.0	0.567	0.598	0.299	0.600	0.094
0.0	0.646	0.679	0.261	0.682	0.083
-1.0	0.723	0.758	0.229	0.760	0.074
-2.0	0.794	0.829	0.205	0.831	0.079
-3.0	0.856	0.895	0.236	0.899	0.073
-4.0	0.916	0.971	0.237	0.976	0.071
-6.0	1.050	1.149	0.219	1.157	0.065
-8.0	1.216	1.356	0.198	1.367	0.056
-10.0	1.427	1.576	0.179	1.590	0.049
-12.0	1.660	1.777	0.164	1.791	0.043
-14.0	1.856	1.916	0.152	1.926	0.039
-16.0	1.964	1.981	0.147	1.985	0.037
-18.0	1.996	1.998	0.148	1.999	0.034
-20.0	2.000	2.000	0.149	2.000	0.030

Table 5.2: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for an AWGN channel with $M = 4$.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.513	0.530	0.120	0.530	0.021
2.0	0.596	0.609	0.106	0.609	0.018
1.0	0.678	0.690	0.094	0.690	0.015
0.0	0.756	0.767	0.086	0.767	0.013
-1.0	0.826	0.840	0.088	0.840	0.011
-2.0	0.891	0.913	0.081	0.913	0.011
-3.0	0.976	0.996	0.078	0.997	0.012
-4.0	1.064	1.088	0.072	1.089	0.011
-6.0	1.252	1.284	0.060	1.285	0.010
-8.0	1.437	1.492	0.060	1.493	0.008
-10.0	1.642	1.720	0.054	1.721	0.008
-12.0	1.834	1.956	0.055	1.958	0.008
-14.0	2.042	2.215	0.053	2.218	0.007
-16.0	2.293	2.483	0.050	2.488	0.007
-18.0	2.570	2.726	0.048	2.732	0.007
-20.0	2.809	2.896	0.046	2.901	0.006
-22.0	2.948	2.976	0.045	2.979	0.006

Table 5.3: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for an AWGN channel with $M = 8$.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	$p(s_5)$	$p(s_6)$	$p(s_7)$	$p(s_8)$
3.0	0.474	0.023	0.003	0.001	0.001	0.003	0.023	0.474
2.0	0.475	0.021	0.002	0.001	0.001	0.002	0.021	0.475
1.0	0.473	0.022	0.003	0.001	0.001	0.003	0.022	0.473
0.0	0.470	0.024	0.004	0.002	0.002	0.004	0.024	0.470
-1.0	0.460	0.027	0.007	0.006	0.006	0.007	0.027	0.460
-2.0	0.432	0.033	0.016	0.019	0.019	0.016	0.033	0.432
-3.0	0.417	0.003	0.007	0.073	0.073	0.007	0.003	0.417
-4.0	0.387	0.004	0.009	0.100	0.100	0.009	0.004	0.387
-6.0	0.346	0.005	0.019	0.130	0.130	0.019	0.005	0.346
-8.0	0.296	0.021	0.066	0.117	0.117	0.066	0.021	0.296
-10.0	0.267	0.009	0.130	0.094	0.094	0.130	0.009	0.267
-12.0	0.228	0.043	0.129	0.100	0.100	0.129	0.043	0.228
-14.0	0.188	0.086	0.116	0.110	0.110	0.116	0.086	0.188
-16.0	0.161	0.108	0.116	0.115	0.115	0.116	0.108	0.161
-18.0	0.144	0.118	0.119	0.119	0.119	0.119	0.118	0.144
-20.0	0.132	0.123	0.123	0.123	0.123	0.123	0.123	0.132
-22.0	0.127	0.124	0.124	0.124	0.124	0.124	0.124	0.127

Table 5.4: Capacity-maximizing input probabilities of an 8×8 DMC with AWGN for $q = 1$.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	$p(s_5)$	$p(s_6)$	$p(s_7)$	$p(s_8)$
3.0	0.475	0.022	0.002	0.000	0.000	0.002	0.022	0.475
2.0	0.476	0.022	0.002	0.000	0.000	0.002	0.022	0.476
1.0	0.480	0.019	0.001	0.000	0.000	0.001	0.019	0.480
0.0	0.481	0.018	0.001	0.000	0.000	0.001	0.018	0.481
-1.0	0.492	0.000	0.000	0.008	0.008	0.000	0.000	0.492
-2.0	0.488	0.000	0.000	0.012	0.012	0.000	0.000	0.488
-3.0	0.489	0.000	0.000	0.011	0.011	0.000	0.000	0.489
-4.0	0.491	0.000	0.000	0.009	0.009	0.000	0.000	0.491
-6.0	0.368	0.121	0.011	0.000	0.000	0.011	0.121	0.368
-8.0	0.259	0.223	0.018	0.000	0.000	0.018	0.223	0.259
-10.0	0.222	0.253	0.025	0.000	0.000	0.025	0.253	0.222
-12.0	0.208	0.203	0.089	0.000	0.000	0.089	0.203	0.208
-14.0	0.203	0.000	0.297	0.000	0.000	0.297	0.000	0.203
-16.0	0.201	0.000	0.299	0.000	0.000	0.299	0.000	0.201
-18.0	0.202	0.000	0.298	0.000	0.000	0.298	0.000	0.202
-20.0	0.201	0.000	0.297	0.002	0.002	0.297	0.000	0.201
-22.0	0.199	0.000	0.266	0.035	0.035	0.266	0.000	0.199

Table 5.5: Capacity-maximizing input probabilities of an 8×8^2 DMC with AWGN for $q = 2$.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	$p(s_5)$	$p(s_6)$	$p(s_7)$	$p(s_8)$
3.0	0.475	0.022	0.002	0.000	0.000	0.002	0.022	0.475
2.0	0.476	0.022	0.002	0.000	0.000	0.002	0.022	0.476
1.0	0.480	0.019	0.001	0.000	0.000	0.001	0.019	0.480
0.0	0.481	0.018	0.001	0.000	0.000	0.001	0.018	0.481
-1.0	0.491	0.000	0.000	0.009	0.009	0.000	0.000	0.491
-2.0	0.487	0.000	0.000	0.013	0.013	0.000	0.000	0.487
-3.0	0.489	0.000	0.000	0.011	0.011	0.000	0.000	0.489
-4.0	0.491	0.000	0.000	0.009	0.009	0.000	0.000	0.491
-6.0	0.371	0.120	0.009	0.000	0.000	0.009	0.120	0.371
-8.0	0.259	0.223	0.018	0.000	0.000	0.018	0.223	0.259
-10.0	0.222	0.253	0.025	0.000	0.000	0.025	0.253	0.222
-12.0	0.208	0.204	0.088	0.000	0.000	0.088	0.204	0.208
-14.0	0.203	0.000	0.297	0.000	0.000	0.297	0.000	0.203
-16.0	0.201	0.000	0.299	0.000	0.000	0.299	0.000	0.201
-18.0	0.202	0.000	0.298	0.000	0.000	0.298	0.000	0.202
-20.0	0.201	0.000	0.297	0.002	0.002	0.297	0.000	0.201
-22.0	0.199	0.000	0.266	0.035	0.035	0.266	0.000	0.199

Table 5.6: Capacity-maximizing input probabilities of an 8×8^3 DMC with AWGN for $q = 3$.

In Figures 5.1 to 5.5, we measure the performance of the soft-decision COVQ systems in terms of source SDR versus channel SNR, where

$$\text{SNR} \triangleq 10 \log_{10} \frac{E_s}{\sigma^2} \text{ (dB)} \tag{5.3}$$

and we compare it to the hard-decision demodulation case where $q = 1$. The results are plotted for $q = 1, 2$, and 3. Performance improves as q increases; a majority of the performance gain is achieved with $q = 2$, as foreshadowed by the capacity gains listed in Tables 5.1 to 5.3.

Since a majority of SDR gain was seen between $q = 1$ and $q = 2$, we show the maximum SDR gains due to increasing q from $q = 1$ and $q = 2$ in Table 5.7. In this table, we show at which channel SNR the maximum gain is achieved for various combinations of constellation size M, dimension k, and COVQ rate r, as inferred from Figures 5.1 to 5.5. The relative SDR gain from $q = 1$ to $q = 2$, for all $M = 4$ cases, was around 10% near a channel SNR of 8 dB; for $M = 8$ case, the relative SDR gain was almost 7% at around 14 dB. For all M over an AWGN channel, minimal gain is observed between $q = 2$ and $q = 3$ and at high SNR (> 14 dB), the SDR curves converge for all q.

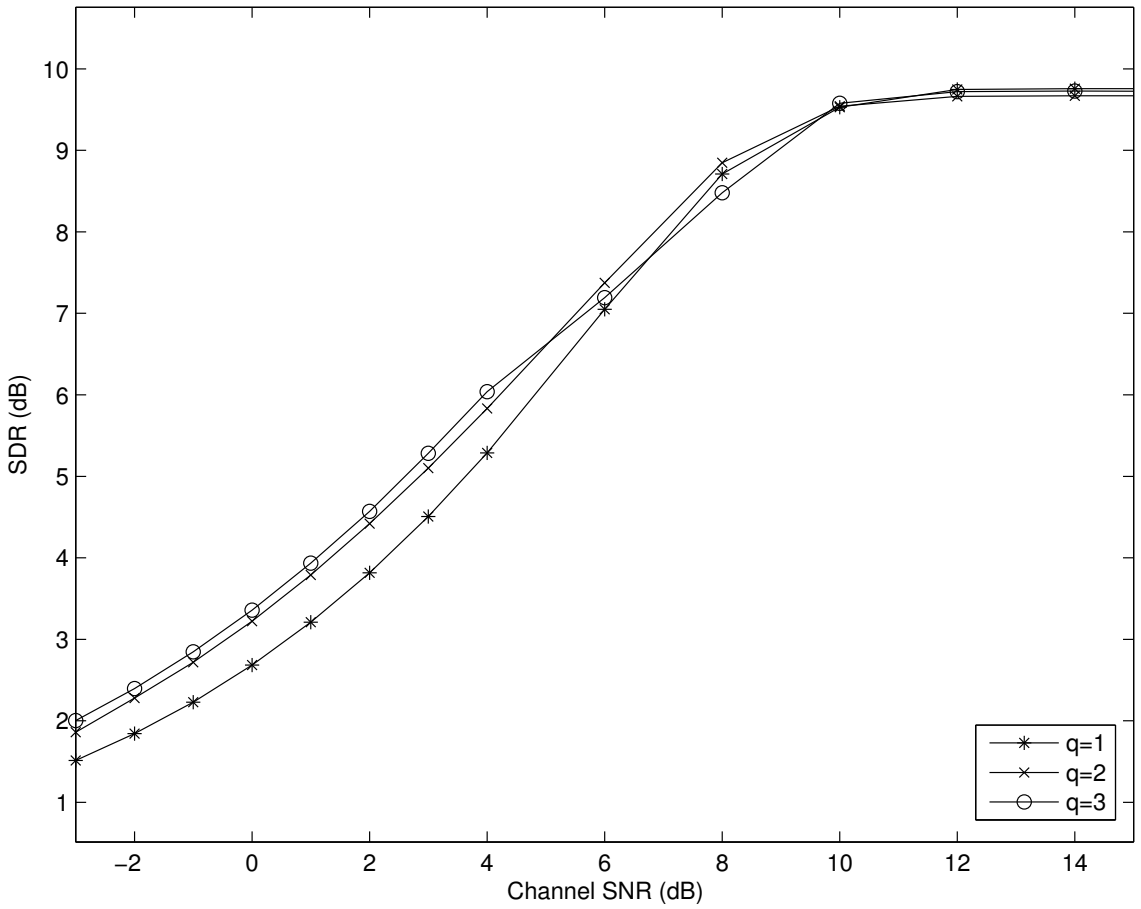

Figure 5.1: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 2$, $k = r = 2$, over an AWGN channel with a memoryless Gaussian source.

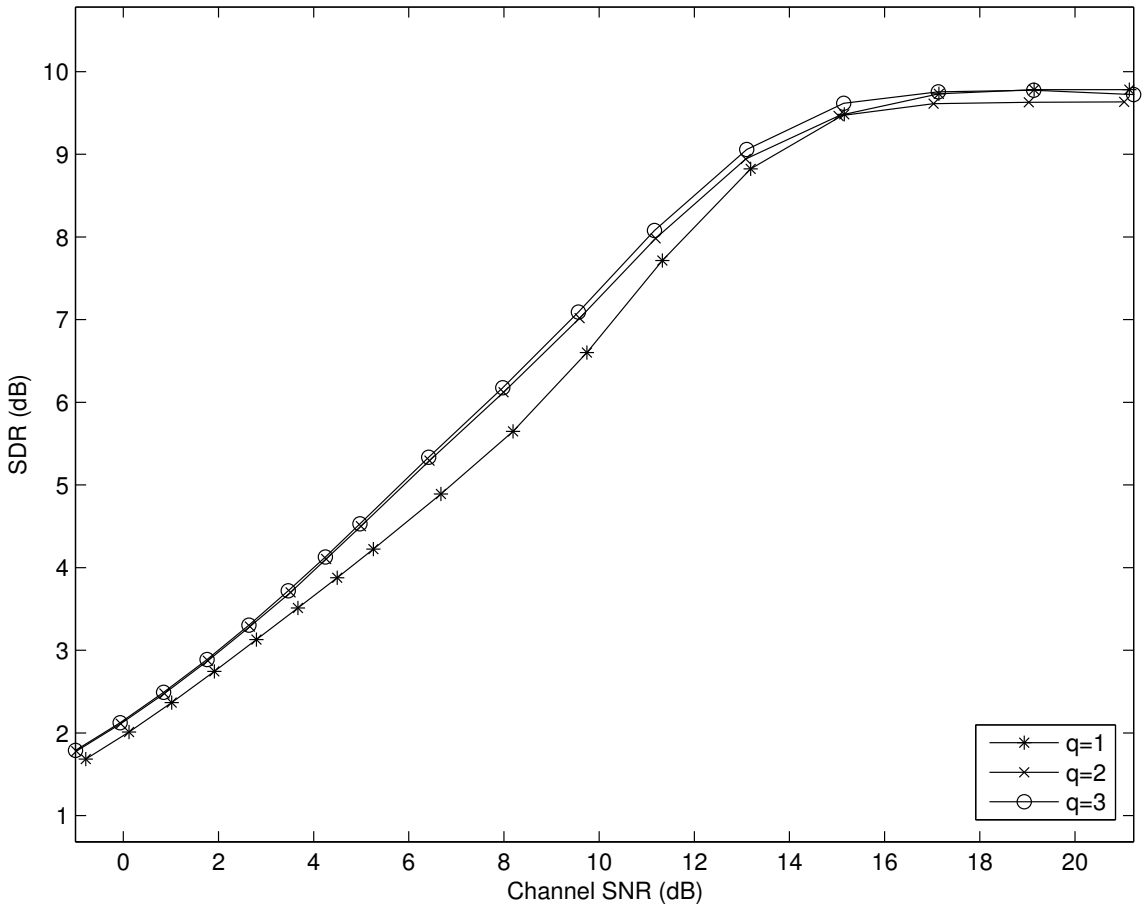

Figure 5.2: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = r = 2$, over an AWGN channel with a memoryless Gaussian source.

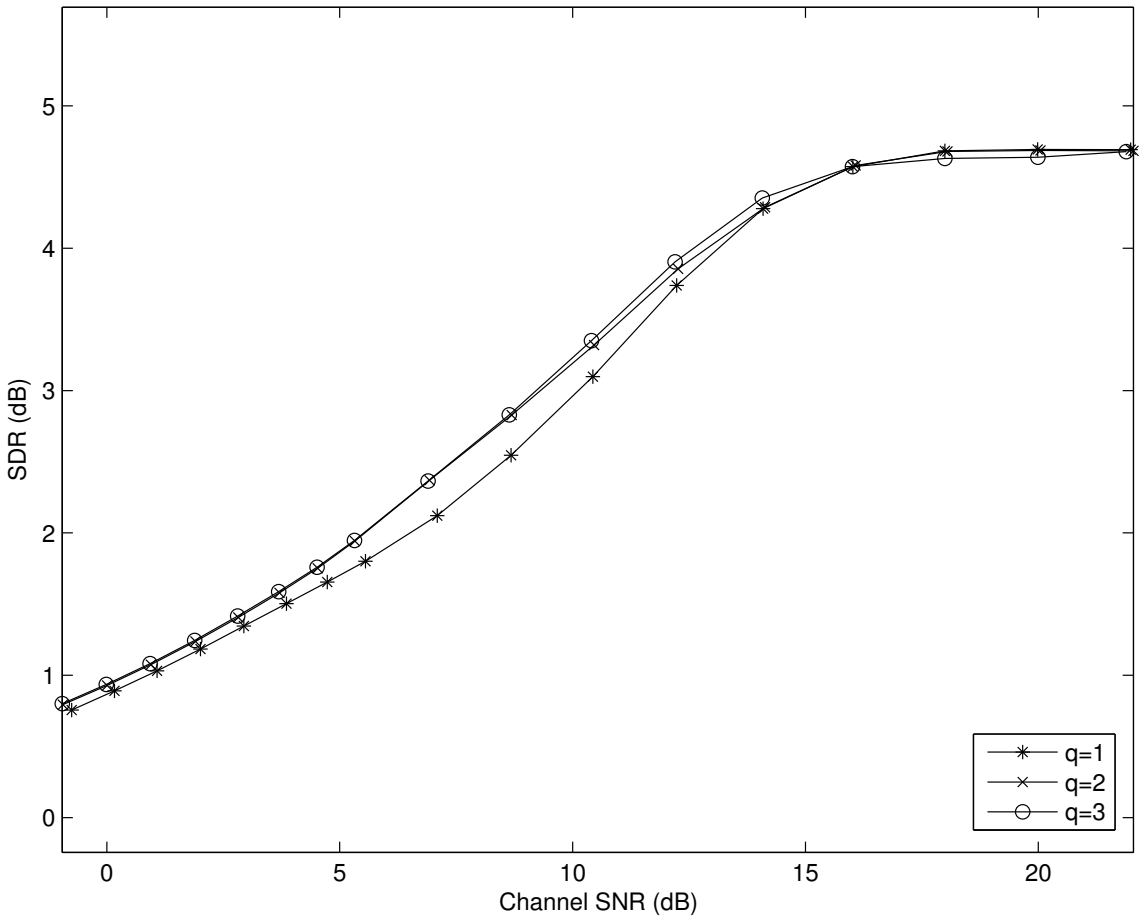

Figure 5.3: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = 4, r = 1$, over an AWGN channel with a memoryless Gaussian source.

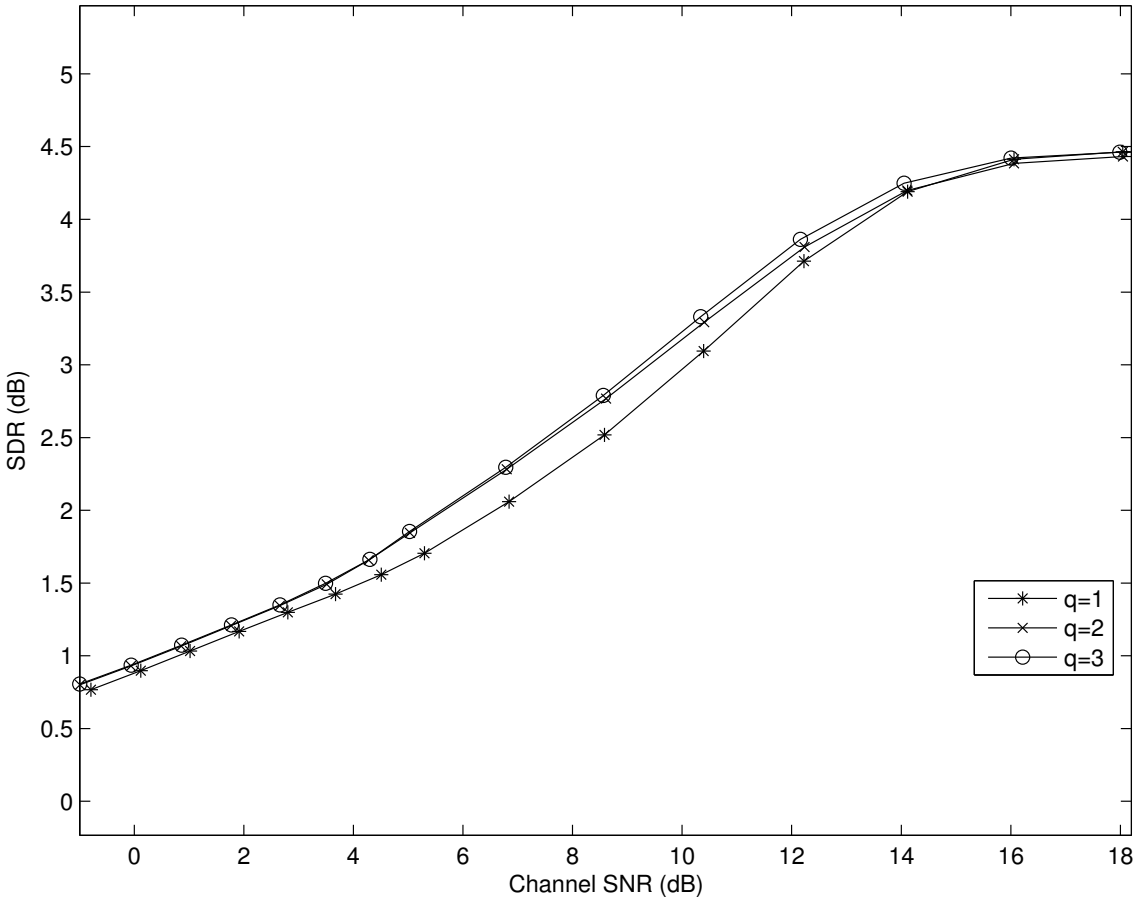

Figure 5.4: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = 2, r = 1$, over an AWGN channel with a memoryless Gaussian source.

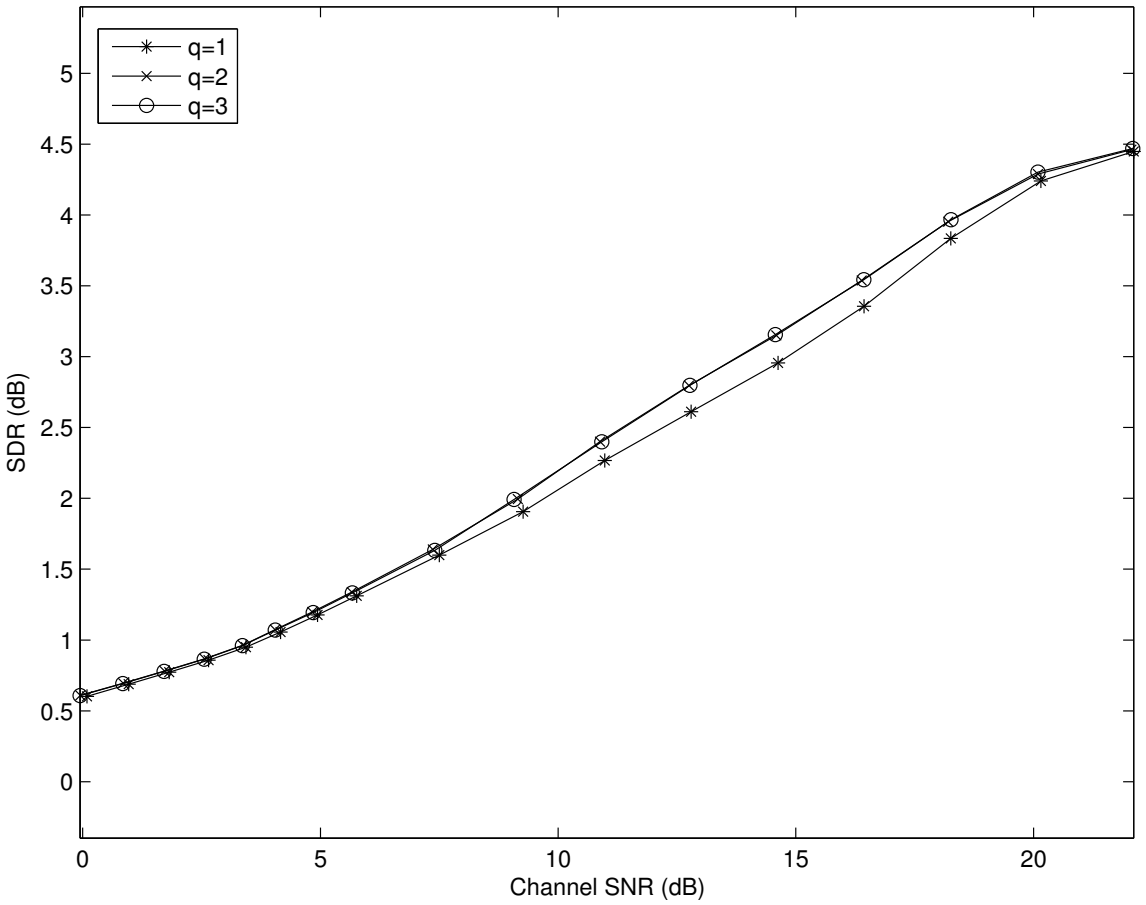

Figure 5.5: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 8$, $k = 3, r = 1$, over an AWGN channel with a memoryless Gaussian source.

M	k	r	SNR (dB)	SDR gain (dB)
2	2	2	2	0.60
4	2	2	8	0.48
4	4	1	8	0.28
4	2	1	8	0.25
8	3	1	14	0.20

Table 5.7: Approximate maximum SDR gains due to increasing q from $q = 1$ and $q = 2$ for AWGN channels and memoryless Gaussian sources.

In Figures 5.4 and 5.6, we compare the performances of soft-decision COVQ between memoryless Gaussian sources and Gauss-Markov sources with correlation coefficient $\rho = 0.9$. Over an AWGN channel, we use $k = 2$ and $r = 1$ bits per source sample, with $M = 4$ constellation signals. In the memoryless source case, the most gain was evident at mid-to-high SNRs, with best improvement for $q = 2$ over hard-decision demodulation at approximately 8.5 dB with a gain of 0.25 dB. For Gauss-Markov sources with $\rho = 0.9$, larger gains were observed over the same SNR region. The largest gains from $q = 1$ to $q = 2$ occur at approximately 8 dB with a gain of 0.42 dB. It was no surprise that the soft-decision COVQ system showed an improvement on SDR for sources with high redundancy in the form of memory, as a similar result was seen in [1] and [25].

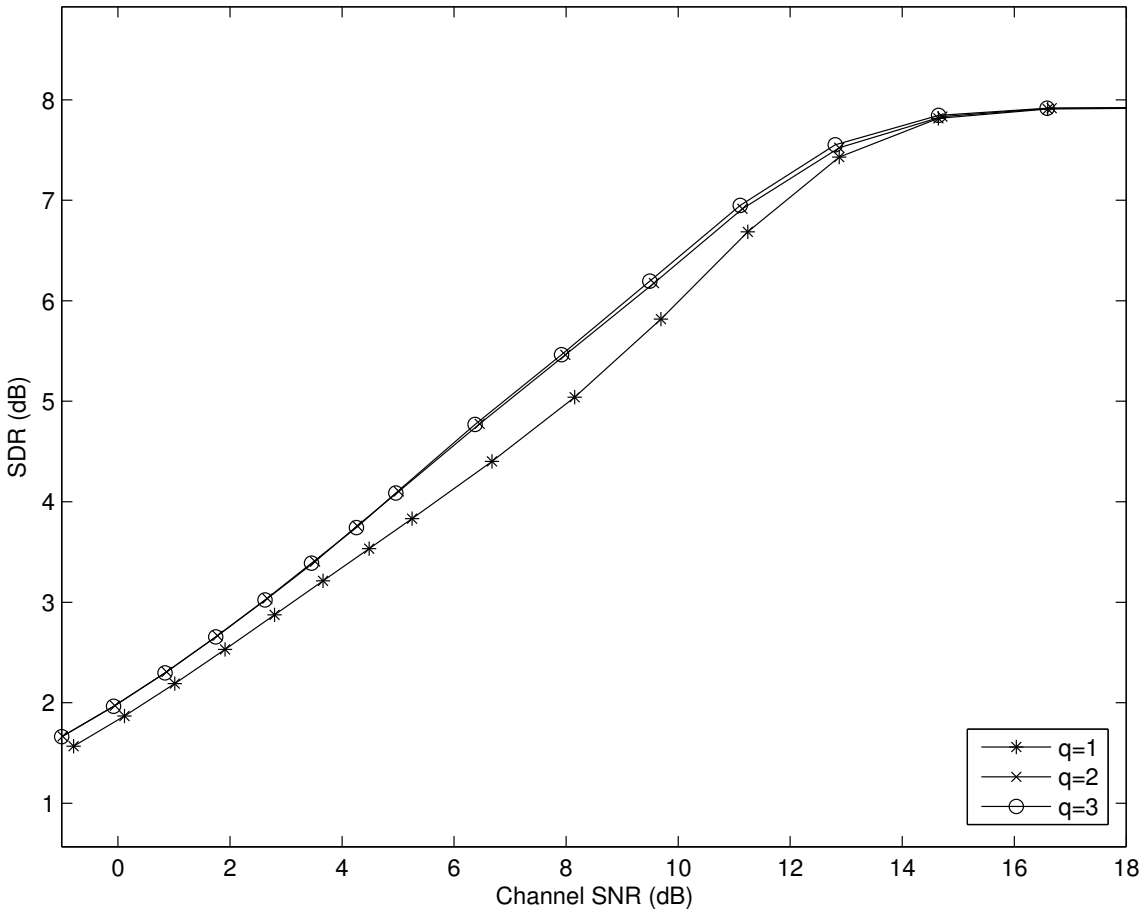

Figure 5.6: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = 2, r = 1$, over an AWGN channel with a Gaussian-Markov source, correlation coefficient $\rho = 0.9$.

5.2.2 Results for Rayleigh Fading Channels

Similar capacity increases hold for Rayleigh fading channels and are shown in Tables 5.9 to 5.11 for a varying number of constellation signals M and soft-decision received bits q. Capacity improves over all q and approaches $\log_2 M$ bits per channel use as SNR increases, but with slower convergence than the AWGN channel.

Confirming trends suggested by the capacity increases, the SDR gains over Rayleigh fading channels also show improvement as q increases. As with capacity gains seen in Tables 5.9 to 5.11, most of the SDR gain is achieved by $q = 2$. Table 5.8 shows the maximum SDR gain, the SNR at which the gain occurs, for various parameters of M, k, and r for Rayleigh fading channels, as seen in Figures 5.7 to 5.11. Thus, for all $M = 4$ cases, the approximate relative SDR gain between $q = 1$ and $q = 2$ is 10% occurring near 12 dB. For the $M = 8$ case shown in Figure 5.11 when $k = 3, r = 1$, the largest SDR gain of 0.11 dB occurs at 18 dB, with a relative SDR gain of 6%.

M	k	r	SNR (dB)	SDR gain (dB)
2	2	2	6	0.65
4	2	2	12	0.45
4	4	1	12	0.19
4	2	1	12	0.20
8	3	1	18	0.11

Table 5.8: Approximate maximum SDR gains due to increasing q from $q = 1$ and $q = 2$ for Rayleigh fading channels and memoryless Gaussian sources.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.150	0.198	1.393	0.213	0.789
2.0	0.180	0.237	1.244	0.254	0.703
1.0	0.215	0.280	1.115	0.298	0.628
0.0	0.256	0.329	0.995	0.349	0.560
-1.0	0.300	0.381	0.892	0.403	0.500
-2.0	0.347	0.435	0.803	0.458	0.449
-3.0	0.399	0.492	0.721	0.516	0.401
-4.0	0.451	0.548	0.651	0.572	0.361
-6.0	0.557	0.656	0.531	0.678	0.292
-8.0	0.656	0.749	0.436	0.769	0.238
-10.0	0.742	0.824	0.360	0.840	0.194
-12.0	0.811	0.880	0.297	0.892	0.158
-14.0	0.865	0.920	0.245	0.929	0.128
-16.0	0.906	0.947	0.202	0.954	0.104
-18.0	0.935	0.966	0.166	0.970	0.085
-20.0	0.955	0.978	0.136	0.981	0.069
-22.0	0.969	0.986	0.111	0.988	0.055

Table 5.9: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for a Rayleigh fading channel with $M = 2$.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.289	0.324	0.453	0.326	0.140
2.0	0.342	0.376	0.406	0.379	0.126
1.0	0.397	0.431	0.366	0.434	0.114
0.0	0.457	0.491	0.327	0.494	0.102
-1.0	0.514	0.550	0.295	0.552	0.092
-2.0	0.569	0.606	0.267	0.608	0.084
-3.0	0.622	0.661	0.244	0.664	0.077
-4.0	0.670	0.712	0.246	0.716	0.081
-6.0	0.759	0.822	0.244	0.828	0.073
-8.0	0.845	0.936	0.224	0.943	0.065
-10.0	0.932	1.046	0.206	1.055	0.058
-12.0	1.018	1.147	0.191	1.158	0.053
-14.0	1.099	1.232	0.179	1.245	0.051
-16.0	1.169	1.300	0.171	1.314	0.048
-18.0	1.227	1.351	0.166	1.365	0.046
-20.0	1.271	1.387	0.163	1.402	0.045
-22.0	1.305	1.412	0.162	1.427	0.044

Table 5.10: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for a Rayleigh fading channel with $M = 4$.

E_N (dB)	$q = 1$	$q = 2$		$q = 3$	
	C	C	Δ	C	Δ
3.0	0.364	0.385	0.143	0.385	0.024
2.0	0.423	0.441	0.130	0.441	0.021
1.0	0.481	0.498	0.119	0.498	0.019
0.0	0.542	0.556	0.111	0.557	0.016
-1.0	0.600	0.613	0.115	0.613	0.015
-2.0	0.653	0.670	0.085	0.671	0.013
-3.0	0.706	0.727	0.089	0.728	0.012
-4.0	0.768	0.785	0.080	0.786	0.012
-6.0	0.884	0.907	0.066	0.908	0.010
-8.0	0.998	1.029	0.055	1.030	0.009
-10.0	1.103	1.146	0.047	1.147	0.007
-12.0	1.197	1.256	0.041	1.257	0.008
-14.0	1.281	1.360	0.054	1.362	0.008
-16.0	1.359	1.457	0.053	1.459	0.008
-18.0	1.434	1.543	0.051	1.545	0.008
-20.0	1.503	1.615	0.050	1.619	0.008
-22.0	1.564	1.672	0.049	1.677	0.008

Table 5.11: Capacity in bits per channel use and capacity-maximizing step-size Δ of an $M \times M^q$ DMC for a Rayleigh fading channel with $M = 8$.

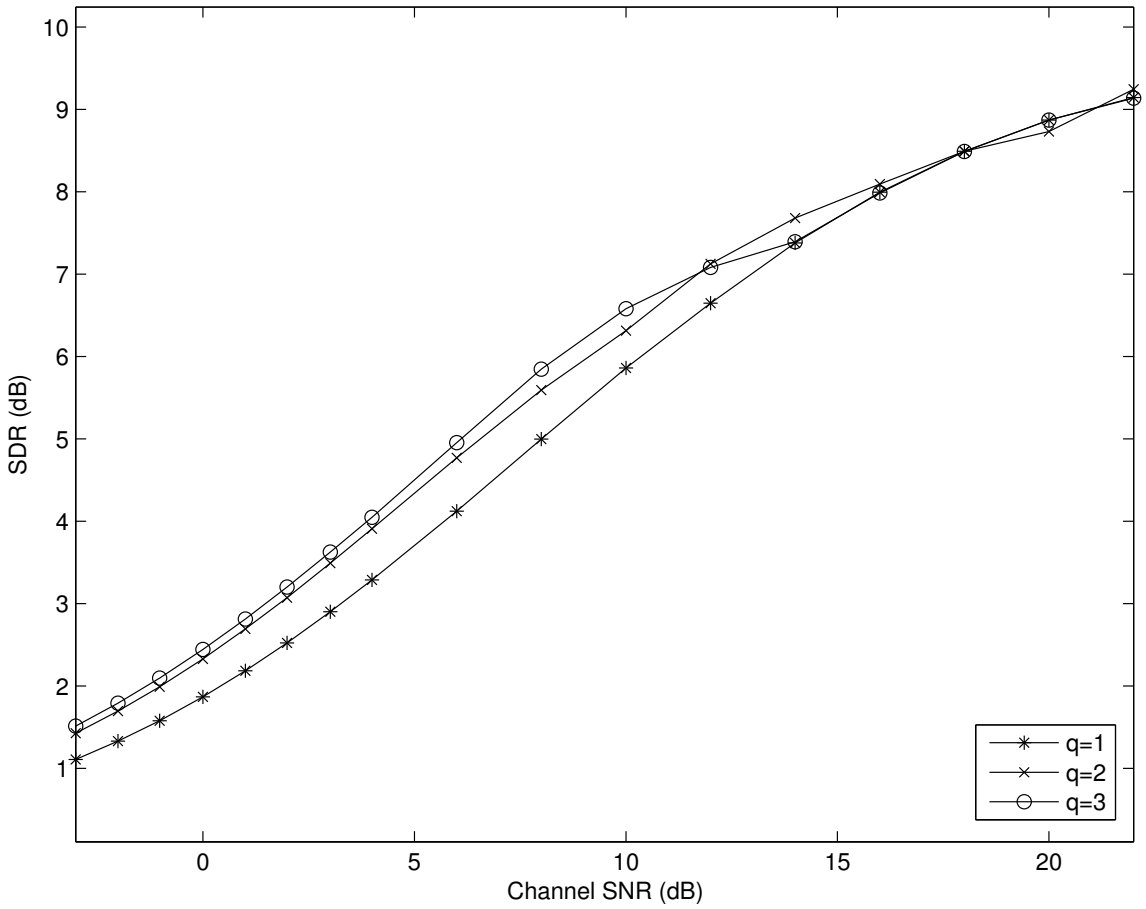

Figure 5.7: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 2$, $k = r = 2$, over a Rayleigh fading channel with a memoryless Gaussian source.

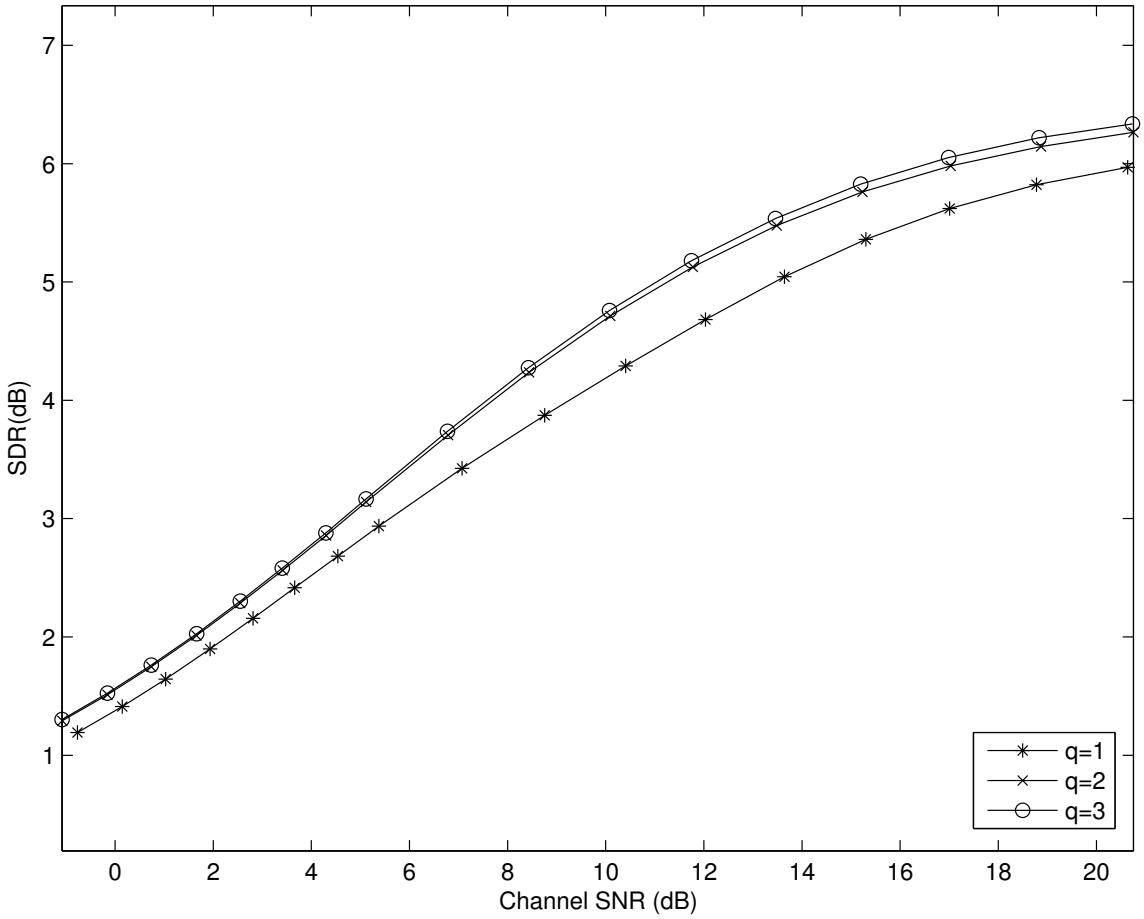

Figure 5.8: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = r = 2$, over a Rayleigh fading channel with a memoryless Gaussian source.

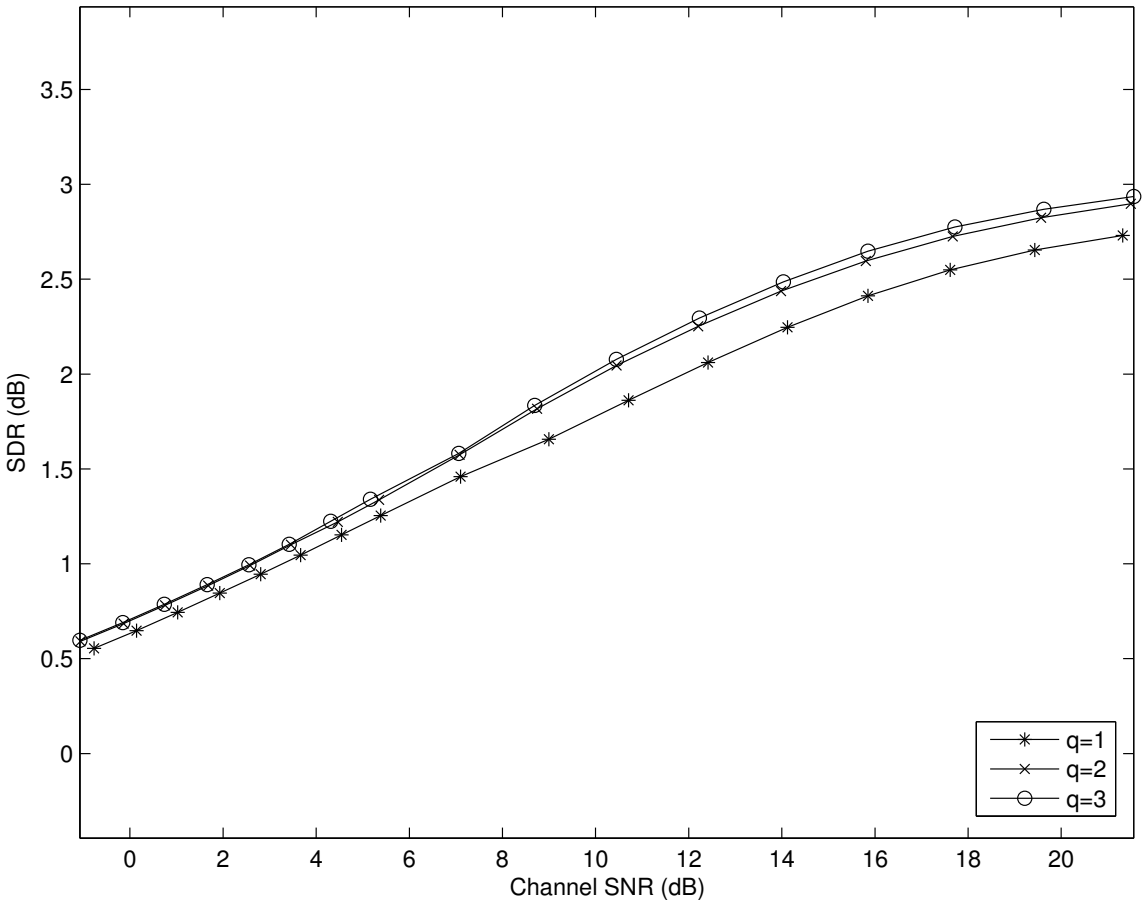

Figure 5.9: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = 4, r = 1$, over a Rayleigh fading channel with a memoryless Gaussian source.

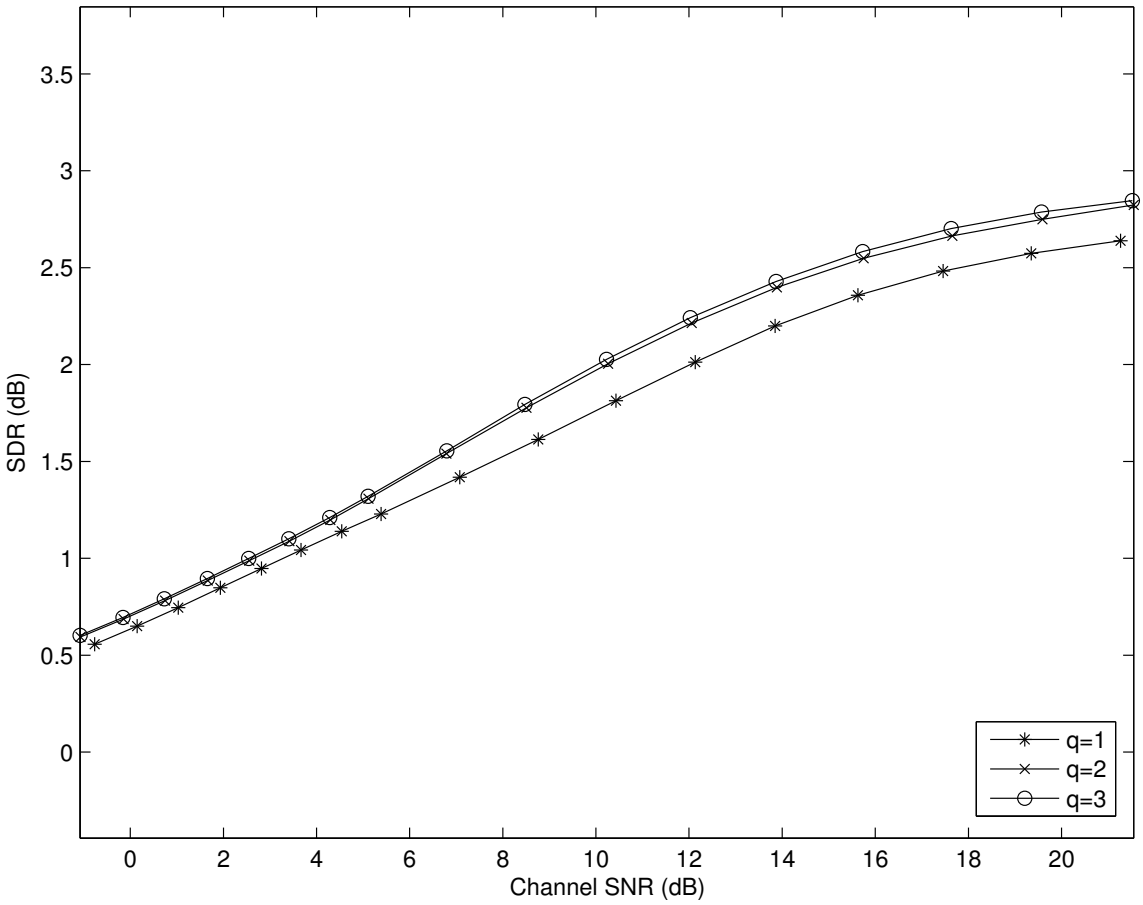

Figure 5.10: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 4$, $k = 2, r = 1$, over a Rayleigh fading channel with a memoryless Gaussian source.

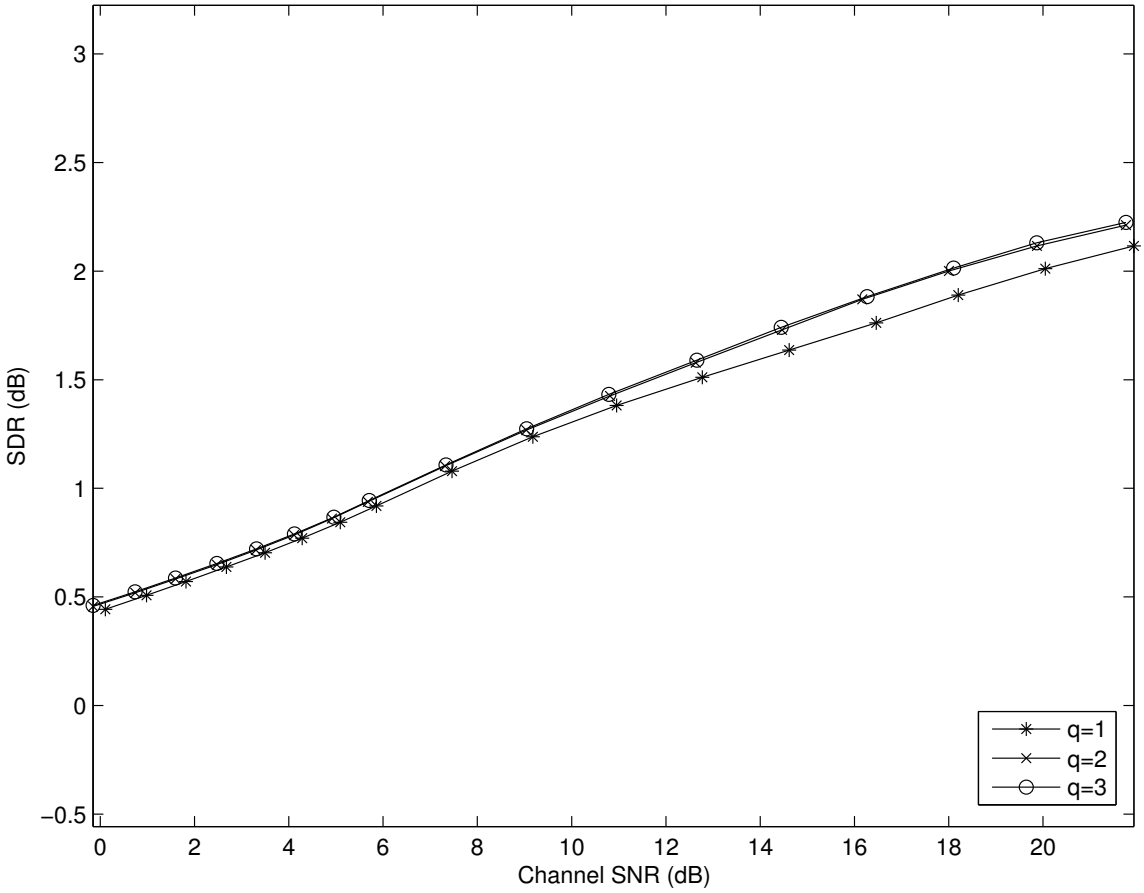

Figure 5.11: Performances using SD-COVQ with $q = 1, 2, 3$, and $M = 8$, $k = 3, r = 1$, over a Rayleigh fading channel with a memoryless Gaussian source.

5.3 Channel SNR Calculation

As previously stated, the plots throughout this chapter display the source SDR versus the channel SNR. The channel SNR is dependent on the average signal energy E_s. The average signal energy was computed by measuring the M-PAM signal probabilities $p(s_i)$ empirically during training. Once the encoding region index was chosen for each source vector, the M-PAM signals used to transmit the index were counted to calculate $p(s_i)$, with signals labeled according to Figures 4.2 to 4.4. Next, the signal probabilities were used to compute the signal energy in Equation (4.1), and consecutively the channel SNR using Equation (5.3).

In Tables 5.12 to 5.14, for all q we list the signal probabilities $p(s_i)$, the average signal energy E_s, and the channel SNR, for a memoryless source over an AWGN channel, with $M = 4, k = 4, r = 1$ for each E_N. We notice that for all q, the signal probabilities approach a uniform distribution as the SNR increases. The average signal energies are well above one for high noise powers.

For a memoryless source sent over an AWGN channel and when $M = 4, k = r = 2$, for all q the average signal energy is above one for high noise powers, is approximately unit energy at around a channel SNR of 8 dB, and drops below one for low noise powers. The same pattern holds true when $M = 4, k = 2, r = 1$, except the average signal energy drops below one between channel SNRs 12 dB and 14 dB. For the $M = 8$ case where $k = 3, r = 1$, the actual average signal energy never drops below one, but does gradually decrease as SNR increases.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	E_s	SNR (dB)
3.0	0.463	0.038	0.039	0.460	1.677	-0.76
2.0	0.453	0.048	0.049	0.450	1.645	0.16
1.0	0.444	0.058	0.058	0.440	1.614	1.08
0.0	0.436	0.066	0.066	0.432	1.589	2.01
-1.0	0.427	0.074	0.073	0.425	1.564	2.94
-2.0	0.418	0.083	0.083	0.416	1.533	3.86
-3.0	0.404	0.097	0.097	0.402	1.490	4.73
-4.0	0.385	0.116	0.116	0.384	1.429	5.55
-6.0	0.340	0.161	0.160	0.339	1.287	7.09
-8.0	0.303	0.194	0.200	0.303	1.170	8.68
-10.0	0.282	0.213	0.220	0.284	1.106	10.44
-12.0	0.266	0.231	0.235	0.268	1.055	12.23
-14.0	0.255	0.243	0.244	0.258	1.021	14.09
-16.0	0.248	0.248	0.250	0.254	1.004	16.02
-18.0	0.247	0.249	0.252	0.252	0.998	17.99
-20.0	0.247	0.249	0.253	0.251	0.997	19.99
-22.0	0.247	0.249	0.254	0.251	0.996	21.98

Table 5.12: Signal probabilities, average signal energies, and SNR for $q = 1$ and $M = 4, k = 4, r = 1$ over an AWGN channel with a memoryless source.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	E_s	SNR (dB)
3.0	0.439	0.060	0.061	0.440	1.606	-0.94
2.0	0.432	0.066	0.067	0.434	1.586	0.00
1.0	0.426	0.073	0.073	0.428	1.566	0.95
0.0	0.421	0.078	0.079	0.422	1.549	1.90
-1.0	0.412	0.087	0.087	0.414	1.521	2.82
-2.0	0.400	0.099	0.100	0.401	1.481	3.71
-3.0	0.382	0.118	0.119	0.382	1.422	4.53
-4.0	0.361	0.139	0.139	0.361	1.356	5.32
-6.0	0.326	0.179	0.172	0.323	1.238	6.93
-8.0	0.304	0.200	0.192	0.304	1.173	8.69
-10.0	0.284	0.219	0.212	0.286	1.112	10.46
-12.0	0.268	0.235	0.226	0.272	1.063	12.27
-14.0	0.258	0.246	0.235	0.262	1.031	14.13
-16.0	0.254	0.250	0.239	0.258	1.018	16.08
-18.0	0.253	0.252	0.241	0.255	1.012	18.05
-20.0	0.253	0.252	0.241	0.254	1.011	20.05
-22.0	0.253	0.252	0.242	0.253	1.010	22.04

Table 5.13: Signal probabilities, average signal energies, and SNR for $q = 2$ and $M = 4, k = 4, r = 1$ over an AWGN channel with a memoryless source.

E_N (dB)	$p(s_1)$	$p(s_2)$	$p(s_3)$	$p(s_4)$	E_s	SNR (dB)
3.0	0.437	0.063	0.062	0.438	1.600	-0.96
2.0	0.431	0.069	0.068	0.432	1.580	-0.01
1.0	0.425	0.075	0.075	0.425	1.560	0.93
0.0	0.419	0.081	0.080	0.420	1.543	1.88
-1.0	0.411	0.090	0.087	0.412	1.517	2.81
-2.0	0.398	0.102	0.101	0.399	1.475	3.69
-3.0	0.380	0.120	0.119	0.381	1.418	4.52
-4.0	0.359	0.140	0.139	0.362	1.354	5.32
-6.0	0.323	0.182	0.174	0.321	1.230	6.90
-8.0	0.301	0.207	0.194	0.299	1.159	8.64
-10.0	0.283	0.224	0.215	0.279	1.098	10.41
-12.0	0.267	0.236	0.234	0.262	1.047	12.20
-14.0	0.259	0.243	0.246	0.252	1.017	14.07
-16.0	0.255	0.247	0.252	0.246	1.003	16.01
-18.0	0.254	0.248	0.252	0.245	0.999	18.00
-20.0	0.254	0.249	0.253	0.245	0.998	19.99
-22.0	0.234	0.259	0.257	0.250	0.974	21.89

Table 5.14: Signal probabilities, average signal energies, and SNR for $q = 3$ and $M = 4, k = 4, r = 1$ over an AWGN channel with a memoryless source.

As q approaches infinity, the soft-decision COVQ decoder converges to the optimal soft decoder (i.e., the one which minimizes mean-square error), as noted in [6]. Cross-over occurences of SDR curves for varying q seen in the figures suggest numerical instability from simulations.

The advantage of the SD-COVQ system is low complexity, relative to other soft-decision schemes; hence, there is minimal delay. Assuming that the channel conditions are known, the channel transition probability matrix is needed at the encoder, while the codeword look-up table is required at the decoder. The codebook is trained before transmission; thus, there is minimal decoding complexity at the receiver. Consequently, the system's disadvantage is its need for storage. The transition matrix is a $2^{kr} \times 2^{qkr}$ matrix, whilst the look-up table consists of 2^{qkr} k-dimensional vectors. Since the decoding complexity is linear with q, and the storage increases exponentially with q, our results show that there is not enough justification to increase q higher than 2, as there was no significant gain beyond $q > 2$.

One possible improvement for showing more SDR gain is modifying the uniform scalar quantizer in the soft-decision demodulator to better represent the noise endured by sent signals. Another possible improvement would be to iteratively calculate the signal probabilities and average signal energy, in order to adjust the signal constellation accordingly.

Chapter 6

Conclusions

6.1 Summary

In Chapter 2, we presented typical source and channel models, introduced information theoretic concepts, and stated the necessary conditions for optimality for VQ using the MSE distortion measure. In Chapter 3, we presented algorithms used to combine data compression and channel noise protection, namely COVQ. We introduced background work on soft-decision COVQ and summarized algorithms used in the initialization of our soft-decision COVQ system.

We proposed a soft-decision COVQ system for M-ary PAM modulated channels in Chapter 4. More specifically, we stated the details of the system, including the channel transition probability matrix for both AWGN and Rayleigh fading channels, and the soft-decision demodulator at the receiver.

In Chapter 5, we tabulated and plotted the numerical results of several scenarios for the number of constellation signals M, dimension of source vector k, COVQ

rate r, soft-decision bits q, and Gauss-Markov sources with correlation coefficients $\rho = 0.0, 0.9$. Trends seen in the increase in capacity were noticed in the SDR gain. A majority of the gain over hard-decision decoding was achieved by $q = 2$, and insignificant gain was achieved between $q = 2$ and $q = 3$. Since the system computes the codebook prior to transmission, encoding and decoding complexity is low, but storage requirements are high. Also, since complexity is exponential with q, $q = 2$ is the most beneficial soft-decision COVQ system for SDR gain, since minimal gain was seen between $q = 2$ and $q = 3$.

6.2 Future Work

Future work with soft-decision COVQ includes using the one-dimensional constellation system as a basis for soft-decision COVQ over two-dimensional constellations, such as quadrature amplitude modulation (QAM). Other possible work could encompass testing other codebook initialization techniques, including encoder index assignments and codeword index mappings. Furthermore, the uniform scalar quantizer at the soft-decision demodulator could be modified to account for channel noise endured by the signals. The system's resilience to channel mismatch could also be investigated and is expected to be fairly robust, as channel-optimized quantization schemes have shown robustness to channel mismatch [6].

Bibliography

[1] F. Alajaji and N. Phamdo, "Soft-decision COVQ for Rayleigh-fading channels," *IEEE Commun. Lett.*, vol. 2, pp. 162-164, June 1998.

[2] S. Arimoto, " An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 14-20, Jan. 1972.

[3] E. Ayanoglu and R. M. Gray, "The design of joint source and channel trellis waveform coders," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 855-865, Nov. 1987.

[4] F. Behnamfar, F. Alajaji, and T. Linder, "Transmission of continuous-alphabet sources over MIMO channels," Proceedings of the *2004 International Symposium on Information Theory and its Applications*, Parma, Italy, Oct. 2004.

[5] F. Behnamfar, F. Alajaji, and T. Linder, "Image transmission over the Polya channel via channel-optimized quantization," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 728-733, Feb. 2005.

[6] F. Behnamfar, F. Alajaji, and T. Linder, "Channel-optimized quantization with soft-decision demodulation for MIMO channels," to appear in *IEEE Transactions on Signal Processing*.

[7] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 460-473, July 1972.

[8] J. Cheng, *Channel Optimized Quantization of Images over Binary Channels with Memory*, M. Sc. Eng. Thesis, Department of Mathematics and Statistics, Queen's University, 1997.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.

[10] J. G. Dunham and R. M. Gray, "Joint source and channel trellis encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 516-519, July 1981.

[11] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei,"Using simulated annealing to design good codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 116-123, Jan. 1987.

[12] 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

[13] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. 36, pp. 799-809, July 1990.

[14] 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.

[15] P. E. Fleischer, "Sufficient conditions for achieving minimum distortion in a quantizer," *IEEE Int. Conv. Rec., Part 1*, pp. 104-111, 1964.

[16] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[17] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[18] R. M. Gray, "Vector quantization," *IEEE Acoust. Speech Signal Processing Mag.*, vol. 1, pp. 4-29, Apr. 1984.

[19] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electronics and Engineering in Japan*, vol. 67-B, no. 4, pp. 39-47, 1984.

[20] A. J. Kurtenbach and P. A. Wintz, "Quantizing for noisy channels," *IEEE Trans. Commun. Technol.*, vol. COM-17, pp. 291-302, Apr. 1969.

[21] F. H. Liu, P. Ho, and V. Cuperman, "Joint source and channel coding using a non-linear receiver," in *Proc. IEEE Int. Conf. Communications* (Geneva, Switzerland, 1993), pp. 1502-1507.

[22] S. P. Lloyd, "Least squares quantization in PCM," Unpublished Bell Laboratories Technical Note. Portions presented at the Institute of Mathematical Statistics Meeting, Atlantic City, NJ, Sept. 1957.

[23] Y. Linde, A. Buzo, and R. M. Gray,"An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[24] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7-12, March 1960.

[25] N. Phamdo and F. Alajaji, "Soft-decision demodulation design for COVQ over white, colored, and ISI Gaussian channels," *IEEE Trans. Commun.*, vol. 48, pp. 1499-1506, Sept. 2000.

[26] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423 and 623-656, 1948.

[27] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pp. 142-163, Mar. 1959.

[28] M. Skoglund and P. Hedelin, "Hadamard-based soft decoding for vector quantization over noisy channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 515-532, Mar. 1999.

[29] M. Skoglund, "Soft decoding for vector quantization over noisy channels with memory," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1293-1307, May 1999.

[30] M. Skoglund, "Bit-estimate based decoding for vector quantization over noisy channels with intersymbol interference," *IEEE Trans. Commun.*, vol. 48, pp. 1309-1317, Aug. 2000.

[31] G. L. Stüber, *Principles of Mobile Communication*. Norwell, MA: Kluwer, 2001.

[32] G. Taricco, "On the capacity of the binary input Gaussian and Rayleigh fading channels," *Eur. Trans. Telecommun.*, vol. 7, no. 2, pp. 201-208, Mar.-Apr. 1996.

[33] R. E. Totty and G. C. Clark, "Reconstruction error in waveform transmission," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 336-338, Apr. 1967.

[34] V. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1230-1248, July 1992.

Appendix A

Capacity Algorithm Background

The following results presented by Blahut in [7] are the basis for the iterative algorithm which computes the channel capacity (outlined in Chapter 3). Arimoto also developed theorems allowing the monotonic convergence to the exact capacity in [2]. We shall begin by stating definitions, using notation from [7].

A discrete channel is described by its probability transition matrix $Q = [Q_{k|j}]$ (not necessarily square), where k is the k-th output letter received, and j is the j-th input letter transmitted. The capacity of a channel is

$$C = \max_{p \in \mathbf{P}^n} I(p, Q) = \max_{p \in \mathbf{P}^n} \sum_j \sum_k p_j Q_{k|j} \log \frac{Q_{k|j}}{\sum_a p_a Q_{k|a}}, \tag{A.1}$$

where

$$\mathbf{P}^n = \{p \in \mathbb{R}^n \colon p_j \geq 0 \, \forall j; \sum_j p_j = 1\}$$

is the set of all channel input probability distributions. $I(p, Q)$ is the mutual information between the channel input and output.

Theorem A.1. *Suppose the channel transition matrix Q has dimension $n \times m$. For*

any $m \times n$ transition matrix P, let

$$J(p, Q, P) = \sum_j \sum_k p_j Q_{k|j} \log \frac{P_{j|k}}{p_j}.$$

Then the following is true.

a) $C = \max_p \max_P J(p, Q, P)$.

b) For fixed p, $J(p, Q, P)$ is maximized by

$$P_{j|k} = \frac{p_j Q_{k|j}}{\sum_a p_a Q_{k|a}}.$$

c) For fixed P, $J(p, Q, P)$ is maximized by

$$p_j = \frac{\exp(\sum_k Q_{k|j} \log P_{j|k})}{\sum_a \exp(\sum_k Q_{k|a} \log P_{a|k})}.$$

From Theorem A.1, we see that combining the last two parts, we find the following corollary.

Corollary A.2. *If p achieves capacity, then*

$$p_j = \frac{p_j \exp \sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_a p_a Q_{k|a}}}{\sum_b p_b \exp \sum_k Q_{k|b} \log \frac{Q_{k|b}}{\sum_a p_a Q_{k|a}}}.$$

This means a new input probability vector p can be generated from any older vector. The following corollary comes from Theorem A.1, by substituting the last part into the first.

Corollary A.3.

$$C = \max_P \log \sum_j \exp(\sum_k Q_{k|j} \log P_{j|k}).$$

Finally, the following theorem uses the previous facts and forms the basis of the capacity computation algorithm.

Theorem A.4. *For any $p \in \mathbf{P}^n$, let*

$$c_j(p) = \exp \sum_k Q_{k|j} \log \frac{Q_{k|j}}{\sum_a p_a Q_{k|a}}.$$

Then, if p^o is any element of $\mathbf{P}^n$ with all components strictly positive, the sequence of probability vectors defined by

$$p_j^{r+1} = p_j^r \frac{c_j^r}{\sum_a p_a^r c_a^r}$$

is such that $I(p^r, Q) \to C$ as $r \to \infty$.

The proofs of the previous theorems and corollaries are detailed in [7].