# Scalar Quantization Algorithms for the Robust Transmission of Correlated Sources Over One- and Two-Way Channels

by

Sayed Saeed Rezazadeh

A thesis submitted to the
School of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
October 2019

# Abstract

The problem of lossy transmission of correlated sources over discrete two-way channels (TWCs) is considered. The objective is to develop a robust low delay and low complexity source-channel coding scheme without using error correction. A simple full-duplex channel optimized scalar quantization (COSQ) scheme that implicitly mitigates TWC interference is designed. Numerical results for sending Gaussian bivariate sources over binary additive-noise TWCs with either additive or multiplicative user interference show that, in terms of signal-to-distortion ratio performance, the proposed full-duplex COSQ scheme compares favourably with half-duplex COSQ. Moreover, our numerical results illustrate that one can achieve significant gain when the propose two-user COSQ is optimally designed for a discrete TWC with additive Markov noise compared to the case when the TWC is fully interleaved. Furthermore, it is demonstrated that correlation between sources can be useful in order to reduce quantization distortion and boost the decoders' reconstruction reliability.

Also, we investigated the effects of feedback in the design of a COSQ over a discrete one-way channel with additive Markov noise by proposing an adaptive COSQ (ACOSQ) where the channel input sequence are adaptively generated based on the received symbols over the feedback link. Numerical results indicate that one can achieve a lower overall distortion by employing feedback in the design of a quantizer for a noisy channel compared to the case where feedback information is not available. Similar to our proposed two-user COSQ, our numerical results demonstrate that one can obtain significant improvement in terms of signal-to-distortion ratio when the ACOSQ scheme is optimally designed for a discrete one-way channel with memory compared to the case where channel is rendered memoryless by interleaving.

# Acknowledgments

I would like to express my appreciation to those who have supported me and had their contribution in making this thesis possible over the course of the last two years at Queen's University.

First of all, I would like to express my gratitude to my supervisors Professor Fady Alajaji and Professor Wai-Yip Geoffrey Chan for their insightful guidance and their precious comments regarding the problems I encountered while completing my degree. There was never a time that I did not feel cared for, thanks to their constant support and guidance.

Life at Queen's University has always been fun and exciting, thanks to the wonderful friends and colleagues I found around here. I deeply wish to thank my colleague and also my best friend Jian-Jia Weng for numerous invaluable discussions we had at Jeffrey Hall about joint source-channel coding.

Last but not least, I would love to sincerely thank my family and specially my parents for their patience, support, and their ever lasting love during my time at Kingston.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Communication System Models

A communication system is designed to reliably transceive the source information between the nodes of a communication network. In practical situations, the source is usually, data, speech, or image signals and is modeled as a random (stochastic) process. In a general communication system sources can be discrete (finite or countable alphabet) or continuous (uncountable alphabet) in value and in time. Continuous-time sources are sampled to form discrete-time sources since in many contexts, processing discrete-time sources is more flexible.

A general point-to-point (or single-user) one-way communication system is depicted in Figure 1.1 representing a separate (tandem) source-channel coding system. Since real-world communication systems often have limited channel bandwidth and storage capacity the role of the source encoder is to compress the source by removing its unnecessary or redundant information. Using an algebraic structure, the channel encoder adds controlled redundancy to outputs of the source encoder, which are vulnerable to channel noise, enabling reliable reproduction of transmitted messages

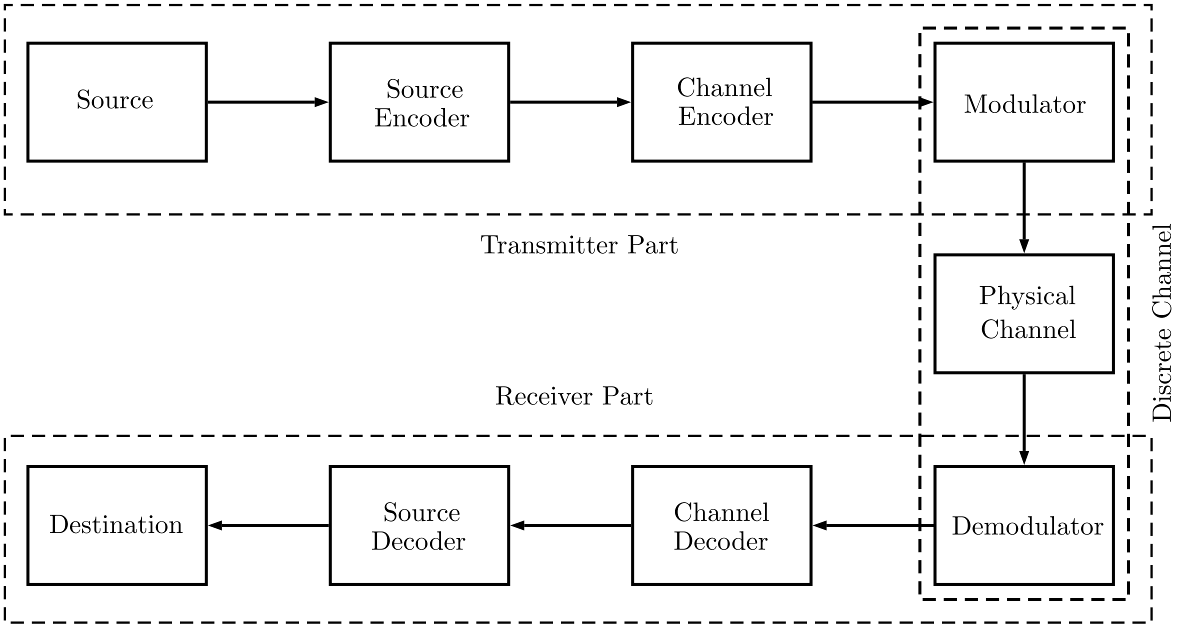


Figure 1.1: Block diagram of a general point-to-point communication system with tandem source-channel coding.

over the channel. The physical channel is a noisy medium, which is only capable of transmitting analog signals and is usually modeled via a sequence of conditional probability distributions of receiving an output given that a specific input was sent. The modulator transforms the channel encoder outputs into waveforms suitable for transmission over the noisy medium. At the receiving part, the demodulator converts the received analog signal to the digital signal to go through the channel and source decoders for estimating the original message produced by the source.

Information theory was first initiated in the late 1940s when Claude Elwood Shannon published his prominent work [1] where he asserted that reliable information transfer with arbitrarily small probability of error is possible provided that the transmission rate is below the channel capacity. He defined the channel capacity as a quantity that depends on the channel statistical characteristics. He further showed that, for a given stochastic source, the minimum distortion-free compression rate is given by the source's intrinsic amount of information, which he later dubbed as "source entropy" and defined in terms of the source statistics. Moreover, Shannon proved that reliable transmission (with asymptotically vanishing probability of error) over a noisy channel is feasible if the source entropy is smaller than the channel capacity. This key theorem, which holds for well behaved source and channel pairs (including memoryless sources and channels), is also known as Shannon's source-channel separation principle. Shannon's separation principle lends its name to its necessary and sufficient conditions for reliable transmissibility which are entirely functions of separable or disentangled information quantities; the source's minimum compression rate and channel capacity without any other parameters depending on both source and channel characteristics. Shannon further provided the lossy counterparts of the aforementioned lossless (distortion-free) source-channel separation principle where the source can be compressed and reproduced within a tolerable distortion threshold [2].

Shannon's separation principle makes the design of a communication system modular and flexible with separate designs for the source and channel encoder/decoder functions without having to sacrifice the system optimality, in terms of reliable transmissibility, by using sourcewords with asymptotically large block lengths in the coding procedures. The requirement of asymptotically large codeword lengths introduces serious decoding latency; making the design inapplicable for systems with complexity and delay constraints such as wireless links. Moreover, it is important to point out that, with exception of certain network topologies [3, 4] where separation is optimal, the separation principle may not hold for multi-user (multi-terminal) systems even under unlimited resources [5, 6], thus performing a joint source-channel coding scheme

(JSCC) is the only viable solution for multi-user designs or system with low complexity and delay tolerance. Figure 1.2 depicts a generic point-to-point communication system using a JSCC scheme. Based on Shannon's separation principle, the tandem source and channel coding design of Figure 1.1 is as good (considering unlimited delay and complexity) as the JSCC scheme of Figure 1.2 in which the joint coding operation depends both on the source and channel statistics.

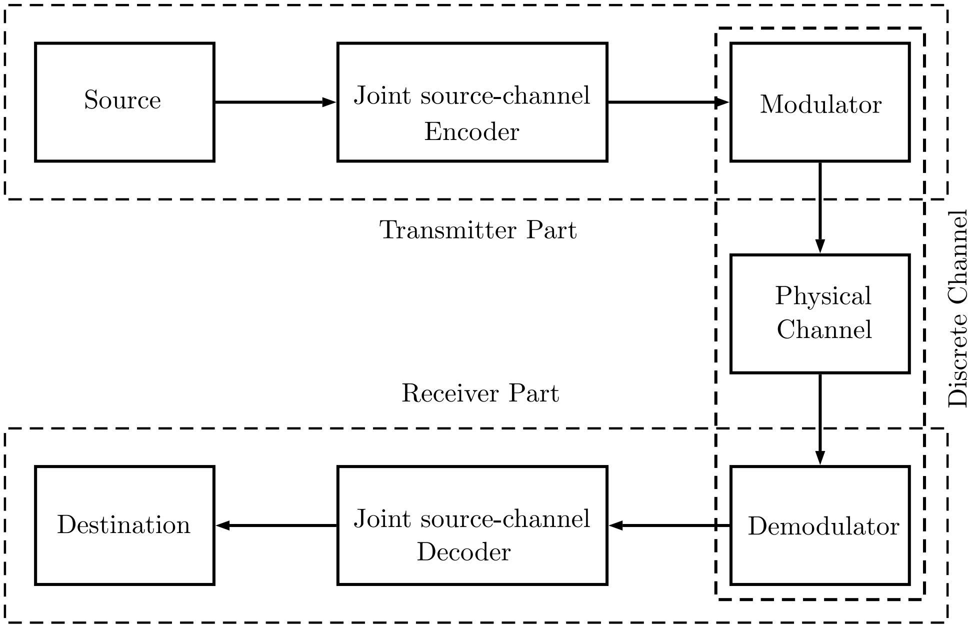


Figure 1.2: Block diagram of a general point-to-point communication system with joint source-channel coding.

## 1.2 Background and Literature Review

To provide an information theoretic rationale for adopting JSCC over tandem coding, Zhong et al. in [7, 8, 9] showed that even in the infinite block length regime where, in terms of reliable transmissibility, separate (tandem) coding is optimal, JSCC schemes can achieve an error exponent[1] that is up to double the error exponent achieved under separate coding. These findings indicate that the same overall probability of error as separate coding is realizable via JSCC schemes while introducing half the (encoding and decoding) delay of separate coding which consequently leads to more than 2-dB power saving when sending binary sources over Gaussian channels. Over the last decades, various efficient JSCC schemes were developed in single-user systems (see [11, Section 4.6] and references therein) which further emphasizes merits of joint versus separate source-channel coding. Kostina et al. in [12], gave an analysis of JSCC by finding new tight bounds for the best achievable lossy JSCC rate; demonstrating the considerable advantage of using JSCC over a separate one in the non-asymptotic regime. In [13], joint and tandem source-channel coding schemes are quantitatively compared on the basis of distortion versus complexity and distortion versus delay. The results of this paper suggest there are complexity and delay thresholds for which the JSCC design is superior compared to the tandem coding in terms of signal to distortion ratio (SDR) values. Also in [14], the problem of JSCC is addressed when variable length codes are used over discrete memoryless channels (DMCs). Channel optimized scalar quantization (COSQ) is a well-known robust lossy JSCC scheme with low complexity and low delay [15] in which the source encoder is a zero-memory scalar quantizer. The performance of a COSQ can be improved with the use of high dimensional channel optimized vector quantizers (COVQs) [16] specially when the source has memory. The performance improvement increases as the source block-length increases. However, the COVQ is constrained by complexity which grows exponentially with the product of rate and source symbols block length. Phamdo et

[1] If the coding block length is allowed to grow without bound, the error exponent (also known as reliability function) of a coding system is defined as the largest rate of exponential decay of its decoding probability of error [10].

al. in [17] proposed a low complexity channel matched tree-structured VQ (TSVQ), which reduces the encoder computational complexity of COVQ by imposing a tree-structure on the codebook. However, the imposed tree-structure causes the design to be sub-optimal [18]. For a complete description of TSVQs the interested reader is referred to [19]. Moreover, Bakus et al. studied in [20] the joint source-channel scalar quantizers in conjunction with turbo-codes; the proposed design showed superior performance compared to conventional COSQ schemes [15] by using a soft reconstruction of samples at the decoders.

It is noteworthy to underscore that the literature on JSCC design for multi-user systems is significant. The transmission media in multi-user systems is usually characterized by multi-input multi-output channels. Specifically in a two-user system setup, the two-way channel (TWC) is a fundamental two-user model that allows both users to transmit information in a full-duplex manner, thus improving the spectral efficiency with respect to one-way systems (i.e., half-duplex transmission) [21]. Moreover, the TWC setup inherently enables users to cooperate by adapting channel inputs to previously received signals, potentially resulting in lower end-to-end errors or distortions. Information-theoretical studies for TWCs were made from different perspectives. In [22], Kaspi investigated the lossy source coding problem over a noiseless TWC, in which only one user can use the channel at each time instant, and established a rate-distortion region. Maor et al. later adopted the interactive protocol proposed in [22] and developed a lossy transmission scheme for noisy TWCs [23]. Moreover, scalar quantize and forward designs for the two-way relay channel were studied in [24] for systems where the two terminals exchange source data with the help of a relay when there is no direct link between the terminals. Achievability and converse results for the lossy transmission of correlated sources over TWCs were derived in [25]. In the same paper, the optimality of scalar coding for TWCs with discrete modulo additive noise as well as additive white Gaussian noise was investigated. Considering that a TWC can be described as two state-dependent one-way channels, the authors in [26] extended the one-way hybrid digital/analog coding scheme of [27] to TWCs. The authors in [28] used multi-resolution quantization and layered JSCC to serve simultaneously several users over the binary erasure broadcast channel. Also, in [29], the proposed single-user JSCC scheme of [30] was adopted for the transmission of two correlated sources over an orthogonal multiple-access channel (MAC). Certain transmission systems might be sensitive to delay. Hence sophisticated data compression and error control coding cannot be afforded. For such communication systems, the best alternative is a source-matched modulation scheme that maps each source samples directly to the modulated symbols. Weng et al in [31] studied a joint source-channel-modulation problem for such systems over non-orthogonal Gaussian MACs (GMACs). Specifically in this paper, the two transmitters, send correlated information using binary-pulse-amplitude modulation (BPAM) where the receiver recovers the messages via joint maximum-a-posteriori decoding in a real time fashion. Also, [32] studied BPAM design for non-uniform sources transmitted over orthogonal GMACs.

## 1.3 Thesis Contribution

In a traditional cellular network, all communications must be directed toward the base station even if the two communicating parties are close enough for a direct communication. This architecture is only suitable for low data rate services such as voice calls and text messages. However, with ever growing demands for high data rate services such as video sharing, gaming, and social networking, this design is no longer suitable for today's cellular networks. Moreover, mobile users are potentially in the range close enough for a direct communication; making the device-to-device communication (D2D) scheme a viable architecture to increase the spectral efficiency of the network [33]. Improving the system throughput, energy efficiency, and delay are among other advantages of the D2D communications.

Devising a two-user system to model the D2D communication is one motivation for considering the problem of this thesis. In this thesis, we extend the results of [15], where a single-user COSQ is designed for a one-way DMC, to a two-user setup where two correlated Gaussian sources are transmitted over a discrete TWC [21]. Our proposed two-user COSQ judiciously mitigates the self-interference caused by simultaneous transmission of both users and exploits the statistical dependency between the sources as receivers' side information.

In space communications, usually the link from the ground station to the satellite is modeled via a noiseless channel due to the presence of high power at the ground transmitter, whereas the reverse path is represented via a noisy channel. Amanullah et al. proposed in [34] a JSCC scheme in the presence of the noiseless feedback to model such a communication system where the feedback information is used in the design of a scalar quantizer over a discrete memoryless one-way channel. We herein propose a new adaptive COSQ (ACOSQ) where every channel input is generated adaptive to the information received over the feedback link. Our simulation results show that our proposed ACOSQ compares favorably with the scheme proposed by [34] particularly when the channel has memory.

## 1.4 Thesis Overview

In Chapter 2, we give an overview of the digital communication models and describe the channel models (with and without memory) used in this thesis. We describe the discrete one-way channel used with and without feedback and mathematically derive the channel's transition distribution. Also, to model a full-duplex transmission, Shannon's TWC is described. We use the Polya urn contagion model to describe the additive Markovian noise process of one-way and two-way channels with memory. At the end of the chapter, we examine scalar quantizers (SQs) with index mapping as well as channel optimized scalar quantizers (COSQs) as robust, low delay, and low complexity JSCC schemes. In Chapter 3, we propose a two-user COSQ for transmitting two generally correlated sources over a discrete TWC with additive and multiplicative user-interference. In this system, the statistical correlation between the two sources are used as side information at the decoders. We examine how the sources correlation can help improve the overall system performance. We also assess the performance of the proposed two-user COSQ optimally designed for TWCs with memory. It is illustrated that the overall system performance is significantly improved compared to the case where the TWC is fully interleaved and the TWC's memory is ignored in design of the proposed two-user COSQ. In Chapter 4, we discuss another perspective towards JSCC schemes. We consider a discrete one-way channel that in general can have memory accompanied by a noiseless feedback link. The effects of information received over the feedback link in the design of a COSQ are investigated using the adaptive scheme of [34] where the channel input sequences are interactively adapted to the received information over the feedback link. We further propose another adaptive COSQ (ACOSQ) that compares favorably with the adaptive scheme of [34] particularly over discrete channels with high noise correlation. Finally, conclusions and directions for future works are presented in Chapter 5.

# Chapter 2

# Preliminaries

## 2.1 Communication Channel Models

The focus of this thesis is to design a digital communication system. As shown in Figure 1.1, the input data symbols to the modulator and output of the demodulator are discrete values. Hence, the concatenation of modulator, physical channel, and demodulator can be regarded as a discrete channel. A discrete channel can be described with a finite input alphabet $\mathcal{X}$ and a finite output alphabet $\mathcal{Y}$ using a sequence of conditional probability (transition) distributions $\{P(Y_i = y_i|X^i = x^i, Y^{i-1} = y^{i-1})\}_{i=1}^{\infty}$ where $x^i = (x_1, \cdots, x_i) \in \mathcal{X}^i$ is the $i$-tuple channel input and $y^i = (y_1, \cdots, y_i) \in \mathcal{Y}^i$ is the $i$-tuple received output.

It is noteworthy to mention that for the sake of brevity, throughout this thesis we use the following notations interchangeably

$$P(Y = y|X = x) = P_{Y|X}(y|x) = P(y|x),$$

where the capital letters denote random variables with their respective realizations represented by lower case letters.

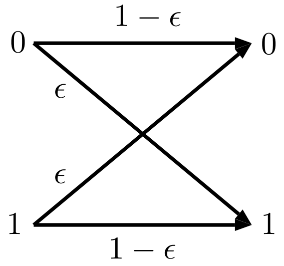


Figure 2.1: A memoryless BSC with cross-over probability $\epsilon$.

### 2.1.1 Discrete One-Way Channel

For a DMC, the following equality holds:

$$P(Y_i = y_i | X^i = x^i, Y^{i-1} = y^{i-1}) = P(Y = y_i | X = x_i) \quad \forall i = 1, 2, \cdots \tag{2.1}$$

In other words, the condition expressed in (2.1) states that for a DMC, the current output only depends on the current input but not on the previous inputs and previous outputs. The memoryless binary symmetric channel (BSC), depicted in Figure 2.1, is a well-known example of a DMC with the following channel transition matrix:

$$\mathbb{Q} \triangleq [P(Y = y|X = x)] = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix} \tag{2.2}$$

where $0 \leq \epsilon < \frac{1}{2}$ is the channel cross-over probability.

#### Discrete one-way channel used without feedback

Furthermore, if a discrete channel is used without feedback then

$$P(X_i = x_i | X^{i-1} = x^{i-1}, Y^{i-1} = y^{i-1}) = P(X_i = x_i | X^{i-1} = x^{i-1}) \quad \forall i \geq 1, \tag{2.3}$$

holds for every $x^i \in \mathcal{X}^i$ and $y^{i-1} \in \mathcal{Y}^{i-1}$, implying that the next channel input is not related to the previous channel outputs given all the previous channel inputs. Therefore, the joint probability of all the channel input and channel output sequences, for $n \geq 1$, is given by

$$\begin{aligned} &P(Y^n = y^n, X^n = x^n) \\ &\quad = \prod_{i=1}^{n} P(Y_i = y_i, X_i = x_i | Y^{i-1} = y^{i-1}, X^{i-1} = x^{i-1}) &(2.4) \\ &\quad = \prod_{i=1}^{n} P(Y_i = y_i | Y^{i-1} = y^{i-1}, X^i = x^i) P(X_i = x_i | Y^{i-1} = y^{i-1}, X^{i-1} = x^{i-1}) &(2.5) \end{aligned}$$

where (2.4) and (2.5) are due to the chain rule. Considering the channel is used without feedback, then using (2.3) in (2.5), we have

$$P(Y^n = y^n, X^n = x^n) = P(X^n = x^n) \prod_{i=1}^{n} P(Y_i = y_i | Y^{i-1} = y^{i-1}, X^i = x^i), \qquad (2.6)$$

which readily leads to

$$P(Y^n = y^n | X^n = x^n) = \prod_{i=1}^{n} P(Y_i = y_i | Y^{i-1} = y^{i-1}, X^i = x^i), \qquad (2.7)$$

provided that $P(X^n = x^n) \neq 0$ holds for any channel input sequence $X^n \in \mathcal{X}^n$. Moreover, if the discrete channel is also memoryless, then applying (2.1) in (2.7) gives

$$P(Y^n = y^n | X^n = x^n) = \prod_{i=1}^{n} P(Y = y_i | X = x_i). \qquad (2.8)$$

It is necessary to mention that (2.8) describes a memoryless channel used *without feedback*. For a clear illustration, consider the following example. If the feedback information is used in the manner that the previous received channel output $y_{i-1}$ is used to choose the next channel input such that $x_i = y_{i-1}$, then $P(Y^n = y^n | X^n = x^n) = 0$ if for some $i = \{2, \cdots, n\}$, $x_i \neq y_{i-1}$, as a result (2.8) will not be satisfied. The use of feedback is prohibited in (2.8) as for a channel with feedback, the current channel input is a function of the previous channel outputs. In other words, a DMC used without feedback is fully describable using a $(|\mathcal{X}| \times |\mathcal{Y}|)$-dimensional transition matrix $\mathbb{Q} = [P(Y = y|X = x)]$ where $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $|\cdot|$ determines the cardinality of the set.

### Discrete one-way channels with memory

In real-world communication systems channel errors often occur in burst rather than independently. To realistically model practical communication channels, we briefly describe one of the most widely used channel models with memory known as the Gilbert-Elliott Channel (GEC) [35]. The GEC is a BSC whose cross-over probability varies over time. As depicted in Figure 2.2, the channel's cross-over probability is driven by a first order Markov process with two states: the *good state* representing a BSC with a low cross-over probability, whereas the *bad state* represents a BSC with a high cross-over probability.

Another channel model that fairly well characterizes the discrete channels with memory is the binary additive Markov noise channel which is described as follows

$$Y_i = X_i \oplus Z_i \quad i = 1, 2, \cdots \tag{2.9}$$

where $\oplus$ is the modulo-2 addition, $X_i$ is the binary channel input, $Y_i$ is the binary channel output at time $i$, and $\{Z_i\}_{i=1}^{\infty}$ is a binary stationary ergodic Markov process

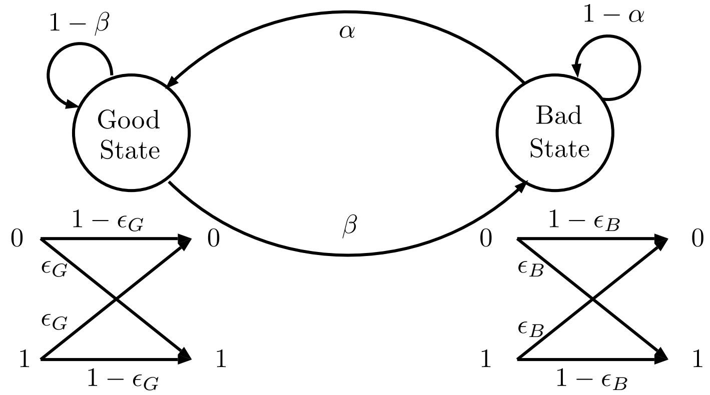


Figure 2.2: The Gilbert-Elliott channel model, where $\alpha$ and $\beta$ are the state transition probabilities and $\epsilon_G$ and $\epsilon_B$ are the cross-over probabilities in the good state and in the bad state, respectively.

of order $M$. The aforementioned Markov noise process is independent of channel input and is produced by a $2^M$ by $2^M$ dimensional transition matrix. In general, $2^M$ independent parameters are required to characterize the Markov process resulting in excessive complexity when the memory order is high.

A more explicit model for channel with memory is proposed in [36] where the additive noise process, which is independent of the channel inputs, is generated using a finite memory contagion urn process of order $M$. According to this model, the noise transition probabilities are given by

$$\begin{aligned} P(Z_i = 1 | Z_{i-1} = z_{i-1}, \ldots, Z_1 = z_1) & \\ &= P(Z_i = 1 | Z_{i-1} = z_{i-1}, \ldots, Z_{i-M} = z_{i-M}) \\ &= P(Z_i = 1 | \sum_{k=i-M}^{i-1} Z_k = \sum_{k=i-M}^{i-1} z_k) \\ &= \frac{\epsilon + \delta\big(\sum_{k=i-M}^{i-1} z_k\big)}{1 + \delta M} \quad \text{for } \; i = 1, 2, \ldots \end{aligned} \tag{2.10}$$

where $P(Z_i = 1) = \epsilon$ and $\delta$ determines the noise correlation via $\rho_z = \dfrac{\delta}{1+\delta}$ where $\rho_z$ is the noise process correlation coefficient. This method, as noted in [36], is fully characterized by three parameters: memory order $M$, channel cross-over probability $\epsilon$, and noise correlation coefficient $\rho_z = \dfrac{\delta}{1+\delta}$.

### 2.1.2 Discrete Two-Way Channels

In a two-way communication system, two users wish to simultaneously exchange source data over a discrete TWC [21]. The block diagram of a general TWC is depicted in Figure 2.3 where $X_1 \in \mathcal{X}$ is the channel input at terminal one such that $Y_2 \in \mathcal{Y}$ is the corresponding channel output at terminal two. Likewise, $X_2 \in \mathcal{X}$ is the channel input at terminal two with $Y_1 \in \mathcal{Y}$ be the corresponding channel output at terminal one. Due to the simultaneous transmission of the two users' signals, the TWC in general exhibits user-interference. In this thesis we consider two types of discrete memoryless TWCs: the binary-additive TWC (BA-TWC) with additive noise and the binary-multiplying TWC (BM-TWC) with additive noise where the latter is an extension of Blackwell's classical binary multiplying channel [21]. The outputs of the BA-TWC with additive noise at time $i$ for $i = 1, 2, \cdots$ can be described as

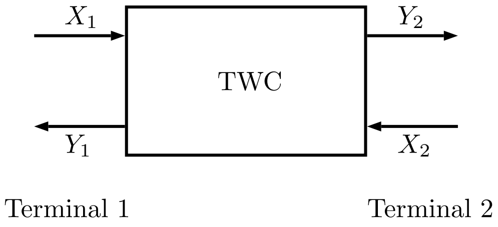


Figure 2.3: A general discrete two-way channel.

modulo-2 sum of its inputs and noise variables:

$$\begin{cases} Y_{1i} = X_{1i} \oplus X_{2i} \oplus Z_{1i}, \\ Y_{2i} = X_{1i} \oplus X_{2i} \oplus Z_{2i}, \end{cases} \tag{2.11}$$

where $Y_{ji}$, $X_{ji}$, and $Z_{ji}$ are the channel outputs, inputs and noise variables at terminal $j$, respectively. The alphabets $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1\}$ are all binary. The noise variables $Z_1$ and $Z_2$, which are assumed to be memoryless in time, independent of each other, and of the channel inputs, have the following distributions:

$$P_{Z_j}(z_j = 1) = \epsilon_j \tag{2.12}$$

where $0 \leq \epsilon_j < 1/2$ for $j = 1, 2$. Similarly, the BM-TWC with additive noise, at time $i$, can be described as:

$$\begin{cases} Y_{1i} = X_{1i}X_{2i} \oplus Z_{1i}, \\ Y_{2i} = X_{1i}X_{2i} \oplus Z_{2i}, \end{cases} \tag{2.13}$$

where the channel outputs, inputs, and noise processes have the same alphabets as described for the BA-TWC with additive noise. The noise variables for (2.13) are identical to those in (2.11) and (2.12).

### Discrete TWCs with memory

In this thesis we also consider TWCs with memory; in particular we consider the BA-TWC with memory and the BM-TWC with memory as described above with the exception that the channels noise processes $\{Z_{1i}\}$ and $\{Z_{2i}\}$ are each generated by the finite memory Markovian contagion urn process of [36]. Based on this model, the noise transition probabilities at terminal $j$ $(j = 1, 2)$ are given by

$$\begin{aligned} & P(Z_{ji} = 1 | Z_{j(i-1)} = z_{j(i-1)}, \cdots, Z_{j1} = z_{j1}) \\ & \qquad = P(Z_{ji} = 1 | Z_{j(i-1)} = z_{j(i-1)}, \cdots, Z_{j(i-M)} = z_{j(i-M)}) \\ & \qquad = P(Z_{ji} = 1 | \sum_{k=i-M}^{i-1} Z_{jk} = \sum_{k=i-M}^{i-1} z_{jk}) \\ & \qquad = \frac{\epsilon_j + \delta_j \big( \sum_{k=i-M}^{i-1} z_{jk} \big)}{1 + \delta_j M} \end{aligned} \tag{2.14}$$

where $P(Z_{ji} = 1) = \epsilon_j$ and the parameter $\delta_j$ determines the noise correlation via $\rho_{z_j} = \dfrac{\delta_j}{1 + \delta_j}$.

## 2.2 Source Coding

In a general communication system, the source needs to be processed (encoded) before being transmitted over the channel. The source information is compressed as much as possible by eliminating its redundancy, reducing the number of bits required for transmitting or storing the source. The source is then represented using a sequence of symbols from a given discrete alphabet and sent over a channel with certain statistical characteristics. The redundant information in a source can stem from the non-uniformity of its marginal probability distribution and the existence of statistical correlation between its successive outputs (i.e., source memory) which the former is quantified by the concept of *source entropy* and the latter by the concept of *source entropy rate.* Shannon in [1] for the first time introduced the concept of *entropy* as a means of measuring the amount of uncertainty in a random variable. A stationary discrete memoryless source (DMS) can modeled as a random variable $X$ with a given alphabet $\mathcal{X} = \{0, 1, \cdots, N-1\}$ whose entropy is defined as:

$$H(X) = -\sum_{x\in\mathcal{X}} P(X=x)\log P(X=x) = -\mathbf{E}_X[\log(P(X))] \tag{2.15}$$

The discrete source with memory; however, is modeled via a stochastic process $\{X_i\}_{i=1}^{\infty}$ whose amount of uncertainty is measured via *entropy rate*, $H_\infty(\mathcal{X})$, defined as

$$H_\infty(\mathcal{X}) = \lim_{n\to\infty}\frac{1}{n}\big(-\mathbf{E}_{X_1,X_2,\ldots,X_n}[\log P(X_1, X_2, \ldots, X_n)]\big). \tag{2.16}$$

It can be shown that the entropy rate, $H(\mathcal{X})$, of a DMS is equal to the entropy of any of its output (e.g., $H(X_1)$). In facts, for a DMS we have

$$H(\mathcal{X}) = H(X_1) = H(X),$$

however, in general for a stationary source with memory (i.e. a stationary Markov source) we have

$$H(\mathcal{X}) \leq H(X),$$

which means that due to the correlation between successive source samples, a Markov source renders less amount of uncertainty than the corresponding DMS with identical marginal distribution.

Based on Shannon's block source coding theorem, the entropy rate of a stationary ergodic source establishes the minimum compression rate, in terms of the number of bits per source sample, for achieving arbitrary small probability of error. On the other hand, $\log_2|\mathcal{X}|$ bits per source sample is required if one uses a uniquely decodable block source coding in the sense that $|\mathcal{X}|^n$ codewords are used to encode every sourceword with blocklength $n$. The *total block source coding redundancy* ($\rho_t$) is given by (2.17) which states the amount of reduction in coding rate one can achieve via using the former approach, which is an asymptotically lossless block source coding, versus the latter scheme which is a fixed-length naive lossless bit representation of the source.

$$\rho_t = \log_2 |\mathcal{X}| - H(\mathcal{X}) \tag{2.17}$$

For a memoryless or independent and identically distributed (i.i.d) source with uniform distribution, the entropy rate equals to $\log_2 |\mathcal{X}|$. This means that $\rho_t = 0$ and the source is incompressible or in other words, there are no superfluous information that can be discarded via source coding. However, if the source has memory or has a non-uniform distribution, then its total redundancy can decomposed into two parts:

$$\begin{aligned} \rho_t &= \rho_M + \rho_D \\ \rho_M &= H(X_1) - H(\mathcal{X}) \\ \rho_D &= \log_2 |\mathcal{X}| - H(X_1) \end{aligned}$$

where $\rho_M$ and $\rho_D$ denote the redundancy due to the source memory and due to its non-uniform distribution, respectively.

In general, data compression schemes describe different methods of representing source information, for a reliable transmission over a noisy channel, with reasonably small coding rate. These representation methods can be lossless such that all the source redundancy is removed after compressing the source while the data is still fully retrievable. In other words, the reconstructed source after decompression is identical (or asymptotically identical with arbitrary small probability of error) to the original source. Based on Shannon's lossless source coding theorem [1], for sufficiently large source block length $n$ and a stationary ergodic source $\{X_i\}_{i=1}^{\infty}$, lossless fixed-to-variable length source coding is feasible with the coding rate arbitrarily close to the source entropy rate $H_{\infty}(\mathcal{X})$; conversely, the probability of error is bounded away from zero (i.e., the probability of error cannot asymptotically vanish) if the source coding rate $r$ is less than the entropy rate.

The entropy rate of a continuous source is theoretically infinite which indicates that infinite precision is required to represent the source without introducing any distortion or loss. Hence lossless reproduction of a continuous source is impossible using a finite-rate code, leaving lossy source coding as the only practical solution. In a lossy source coding regime, the reconstructed source is allowed to deviate from the original data within an acceptable distortion threshold. Now the question is how to find the best representation of a source for a given coding rate. To that end, consider representing a single sample drawn from a continuous source. Let $U_0 \in \mathbb{R}$ denote the random variable and let it be represented by $\hat{U}_0$. If we are given a total of $r$ bits to represent the source sample $U_0$, then there are a total of $2^r$ possible values for $\hat{U}_0$. Finding the optimum set of values for $\hat{U}_0$ and the associated regions represented with each of the $2^r$ possible values of $\hat{U}_0$ is the main goal of the process called *quantization* with which the analog source symbols are mapped to discrete (digital) symbols from a finite alphabet at the cost of introducing some distortion with respect to the original source.

### 2.2.1 Scalar Quantization (SQ)

In general, the continuous domain of an analog signal is partitioned into finite number of regions such that all members of each region are represented by a single value called output level or reproduction codeword. The set of all possible reproduction codewords is called the codebook. For a continuous source alphabet $\mathbb{R}$, an $N$-level SQ is a mapping

$$Q : \mathbb{R} \to \mathcal{C}$$

where $\mathcal{C} = \{c_i\}_{i=1}^{N} \subset \mathbb{R}$ is the codebook of $Q$ and $c_i$'s are the quantization codewords. The quantizer operates by partitioning the source alphabet $\mathbb{R}$ into $N$ distinct regions, $\mathcal{S}_i$, called *Voronoi* or *Dirichlet* partitions such that

$$\mathcal{P} = \Big\{ \mathcal{S}_i : \mathcal{S}_i \cap \mathcal{S}_j = \emptyset,\ \forall j \neq i, j = \{1, 2, \ldots, N\},\ \bigcup_{i=1}^{N} \mathcal{S}_i = \mathbb{R} \Big\},$$

for $i = 1, 2, \ldots, N$. Usually in practice the SQ is described as an encoder-decoder pair (Figure 2.4) as

$$\begin{gathered} \text{Encoder } \mathcal{E} : \mathbb{R} \to \{1, 2, \ldots, N\} \\ \mathcal{E}(u) = i \ \text{ if and only if } \ u \in \mathcal{S}_i \\ \text{Decoder } \mathcal{D} : \{1, 2, \ldots, N\} \to \mathcal{C} \\ \mathcal{D}(i) = c_i, \end{gathered}$$

then the overall quantizer is described as

$$Q(u) = \mathcal{D}(\mathcal{E}(u)) = c_i \ \text{ if and only if } u \in \mathcal{S}_i.$$

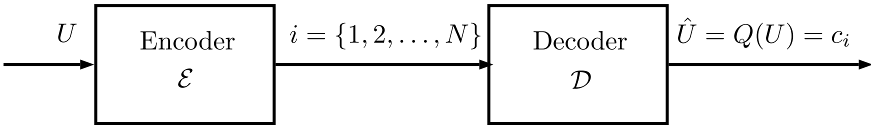


Figure 2.4: Representing an SQ as an encoder/decoder pair.

Assuming a general distortion measure $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$, which is a mapping from the set of source alphabet-reproduction alphabet pairs into a set of non-negative real numbers, the quality of a quantizer, $Q$, is measured via the total expected amount of distortion defined as:

$$D = E[d(U, \hat{U})] \tag{2.18}$$

where $\hat{U} = Q(U)$. There are many different choices for the distortion measure; however, one of the most common distortion measures for continuous alphabet sources is the following squared-error distortion which is exclusively used throughout this thesis:

$$d(u, \hat{u}) = (u - \hat{u})^2. \tag{2.19}$$

The end-to-end distortion can then be calculated as follows:

$$\begin{aligned} D &= E[d(U, Q(\hat{U}))] && (2.20)\\ &= \sum_{i=1}^{N} E[(U - c_i)^2 | U \in \mathcal{S}_i] P(U \in \mathcal{S}_i) && (2.21)\\ &= \sum_{i=1}^{N} \int_{\mathcal{S}_i} f_U(u)(u - c_i)^2 \mathrm{d}u && (2.22) \end{aligned}$$

For the design of an optimal SQ, the Voronoi regions, $\mathcal{P}$, and the reconstruction codebook, $\mathcal{C}$, must satisfy the following two necessary conditions [19]:

- *Nearest neighbor condition (NNC)*: Among all $N$-level SQs with a fixed codebook, $\mathcal{C} = \{c_i\}_{i=1}^{N} \subset \mathbb{R}$, the SQ with quantization cells $\mathcal{S}_i$ satisfying

$$\mathcal{S}_i \subset \Big\{u \in \mathbb{R} : d(u, c_i) \leq d(u, c_j)\ \forall j \neq i, j \in \{1, 2, \ldots, N\}\Big\},\ \forall i \in \{1, 2, \ldots, N\} \tag{2.23}$$

is optimal, that is,

$$Q(u) = c_i \text{ if and only if } d(u, c_i) \leq d(u, c_j)\ \forall j \neq i. \tag{2.24}$$

Thus, for a given decoder (codebook), the encoder (quantization cells) is a minimum distortion or nearest neighbor mapping and hence

$$Q(u) = \arg\min_{c_j\ \in\ \mathcal{C}} d(u, c_j). \tag{2.25}$$

- *Centroid condition (CC)*: The optimal reconstruction codewords should minimize the expected distortion given their respective assignment regions [6]. In other words, among all $N$-level SQs with a fixed partition set, $\mathcal{P}$, the quantizer, $Q$, with reproduction codewords given by

$$c_i = \arg\min_{\hat{u}\ \in\ \mathbb{R}} E(d(U, \hat{u})|U \in \mathcal{S}_i) \tag{2.26}$$

is optimal where $c_i$ is called the centroids of the cell $\mathcal{S}_i$. Moreover, for the squared-error distortion measure, the centroid of a given region is its center mass i.e.,

$$c_i = E(U|U \in \mathcal{S}_i), \quad \text{for } i \in \{1, 2, \ldots, N\}. \tag{2.27}$$

Based on the optimality conditions in (2.23) and (2.27) a simple algorithm can be developed to design a "good" quantizer: starting from an initial set of reconstruction codewords, find the optimal set of quantization cells using (2.23), which are the nearest neighbor regions with respect to the distortion measure. In the next step, for the given updated regions, find the optimal reconstruction codewords (which are the centroids of their corresponding regions under the squared-error distortion measure) using (2.27); then repeat the iteration for this new set of codewords. Successive application of (2.23) and (2.27) forms a non-increasing sequence of average distortion values and since the sequence is bounded below by zero, it must converge after a finite number of iterations. However, the distortion might not converge to the global optimal solution and only local optimality is guaranteed. As mentioned earlier, (2.23) and (2.27) are both necessary and sufficient conditions when the codebook, $\mathcal{C}$, and nearest neighbor regions, $\mathcal{P}$, are fixed respectively; however, the ultimate solution given by the iterative algorithm does not satisfy the overall system's optimality condition since for the design of one element of the system, the other component is fixed. This algorithm is called *Lloyd algorithm* [37] whose details are also depicted by Algorithm 1. Note that if the source has a log-concave distribution, then it is shown in [38] that there exits a unique globally optimal quantizer $Q^*$ and the Lloyd's algorithm may be used to find $Q^*$. Furthermore, a quantizer that takes $k > 1$ source samples at the time and outputs $2^{kr}$ quantized symbols is called vector quantizer (VQ) which is designed using a generalized Lloyd algorithm [39] also known as *Linde-Buzo-Gray vector quantizer.*

One important aspect of the Lloyd algorithm is the choice of initial codebook. Several options for choosing the initial codebook have been proposed in the literature.

**Algorithm 1:** The Lloyd algorithm for SQ design

```
**Input**: Source pdf $f$, initial codebook $\mathcal{C}^{(0)}$, and the stopping threshold $T$
**Output**: Voronoi regions, $\mathcal{P}^{(m)}$, reconstruction codebook, $\mathcal{C}^{(m+1)}$, and the
  ultimate distortion value $D^{(m+1)}$
$D^{(0)} \leftarrow \infty$
$D^{(1)} \leftarrow 0$
m $\leftarrow$ 0
**while** $\frac{D^{(m)}-D^{(m+1)}}{D^{(m)}} > T$ **do**
    $\mathcal{S}_i^{(m)} \leftarrow \{u : d(u, c_i^{(m)}) \leq d(u, c_j^{(m)}),\ \forall j \neq i\},\ \forall i \in \{1, 2, \ldots, N\}$
    $c_i^{(m+1)} \leftarrow E(U|U \in \mathcal{S}_i^{(m)})$
    $D^{(m+1)} \leftarrow E[(U - \hat{U})^2]$
    $m \leftarrow m + 1$
**end**
```

The MATLAB scripts corresponding to this algorithm are available at `https://github.com/Saeed-Rezazadeh/COSQ.git`.

One naive method is to draw $N$ codewords from the source distribution and use them as the initial codebook. The other widely used approach for generating the initial codebook is the so-called splitting algorithm [39]. In this method, a training sequence, $T_w$, is generated from the source distribution. The first codeword, $c_0$, is the centroid of the entire training sequence i.e.,

$$c_0 = \frac{1}{W} \sum_{i=1}^{W} t_i$$

where $T_w = \{t_i\}_{i=1}^{W}$ is the training sequence of size $W$. This codeword is then split into two points (by perturbing with a small value $\alpha$) for which a two-level Lloyd quantizer is designed. In other words, the two codewords are

$$c_1 = c_0 + \alpha$$

$$c_2 = c_0 - \alpha.$$

which are used to initialize a two-level Lloyd quantizer. The splitting process is again applied to the ultimate codewords of the two-level quantizer; creating four codewords to initialize a four-level Lloyd quantizer. This process is continued until the initial $N$ codewords for an $N$-level quantizer is generated.

Note that to store a continuous source in a storage medium, it is not necessary to use the actual quantized values but rather in practice, the $r$-bit ($r = \log_2 N$) binary representation of the encoder's output, $i$, is used to store the quantized source.

Moreover, a quantizer can be viewed from another perspective as depicted in Figure 2.5 where there is a noisy channel between the encoder-decoder pair. Assuming that the receiver already knows the quantizer codebook $\mathcal{C}$, it is enough to transmit the the indices of the quantization cells, $i$, over the channel to retrieve the reconstruction codewords, $c_j$, at the receiver. For a special case of a noiseless channel we have $i = j$ where $j \in \{1, 2, \ldots, N\}$ is the corresponding channel output. In a real-world communication system, the $r$-bit binary representation of the quantization cells indices, $i$, are sent over the channel. For a noiseless channel, indices can be arbitrarily assigned to the reproduction codewords as long as the same assignment is used for quantization regions. As one can realize from (2.22), the end-to-end distortion is independent of the index assignment since different index assignments results in relabeling the quantization cells that changes the order of integrals over the quantization regions. However, when the channel is noisy, the overall distortion is not only due to quantization, but it is also due to channel noise. As a result, for the case of a noisy channel, index assignment plays a vital role in overall system distortion. Therefore, the design of a SQ "optimized" for a particular channel is worth investigating.

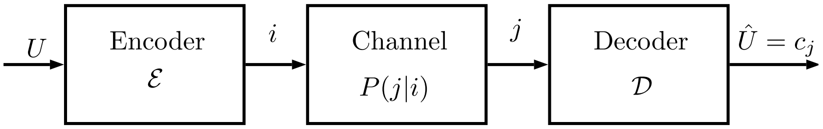


Figure 2.5: The block diagram of an SQ in a communication system where $P(j|i)$ is the channel transition probability.

### 2.2.2 Channel Optimized Scalar Quantizer (COSQ)

As mentioned earlier, channel noise plays a significant role in system performance; therefore, it is not efficient to transmit the quantizer indices over the channel directly. At the cost of increasing the system complexity, a channel coding module is added whose aim is to make the SQ robust to channel errors. In [40], a modified version of simulated annealing (SA) is used for devising an algorithm to assign encoding regions indices to SQ codewords. The end-to-end distortion of an SQ that is robust to channel noise (i.e., SQ with index assignment) can be written as:

$$\begin{aligned}
D &= \mathbf{E}[d(U,\hat{U})] \\
&= \sum_{\mathbf{x}\in\mathcal{X}^r}\sum_{\mathbf{y}\in\mathcal{Y}^r} \mathbf{E}[d(U,\hat{U})|U\in\mathcal{S}_{\mathbf{x}},\hat{U}=c_{\mathbf{y}}]P(\hat{U}=c_{\mathbf{y}}|U\in\mathcal{S}_{\mathbf{x}})P(U\in\mathcal{S}_{\mathbf{x}}) \\
&= \sum_{\mathbf{x}\in\mathcal{X}^r}\sum_{\mathbf{y}\in\mathcal{Y}^r} P(\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x}) \int\limits_{\mathcal{S}_{\mathbf{x}}} f_U(u)d(u,c_{\mathbf{y}})\mathrm{d}u \qquad (2.28)
\end{aligned}$$

where $P(\hat{U} = c_{\mathbf{y}}|U \in \mathcal{S}_{\mathbf{x}}) = P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ is probability of receiving $\mathbf{y} \in \mathcal{Y}^r$ when[1] $\mathbf{x} \in \mathcal{X}^r$ is sent. Intuitively, one can conjecture that, compared to an SQ, the new source of distortion is the channel noise. In fact it is shown in [40], under squared-error distortion measure, if $c_{\mathbf{x}}$ represents the centroids of the encoding region

[1] Note that throughout this thesis, boldface letters denote a vector of a given length.

$\mathcal{S}_{\mathbf{x}}$, then the overall distortion can be rewritten as:

$$\begin{aligned} D &= D_q + D_c \\ &= \sum_{\mathbf{x}\in\mathcal{X}^r} \int_{\mathcal{S}_{\mathbf{x}}} f_U(u)(u-c_{\mathbf{x}})^2 \mathrm{d}u + \sum_{i=1}^{N} \sum_{\mathbf{y}\in\mathcal{Y}^r} P(U\in\mathcal{S}_i)P(\mathbf{y}|b(i))(c_{b(i)}-c_{\mathbf{y}})^2 \end{aligned} \tag{2.29}$$

where the first term, $D_q$, is the distortion caused by the quantizer while the second term, $D_c$, can be interpreted as the channel distortion, and

$$b : \{1, 2, \ldots, N\} \ \rightarrow \ \mathcal{X}^r,$$

is a 1-1 mapping such that $b(i)$, for $i = \{1, 2, \ldots, N\}$, is an $r$-bit binary index assigned to $c_i$ which is used for subsequent transmission. Now the problem of designing the system boils down to using the standard SQ designed for a noiseless channel and minimizing $D_c$ over the index mapper $b(\cdot)$. The proposed method in [40] uses the SA algorithm as a randomized stochastic relaxation method that tries to stochastically avoid local minima in its search for the global minimum of the channel distortion. Note that in this algorithm, every index assignment $\mathbf{b} = (b(1), \ldots, b(N))$ is the state vector of the system and the resulting channel distortion $D_c$, given by (2.29), is the objective function to be minimized. The stochastic perturbation scheme used in SA to choose the next system state, $\mathbf{b}'$, is as follows. Two elements of the current state vector $\mathbf{b}$ are picked at random and are interchanged. This perturbation scheme makes it

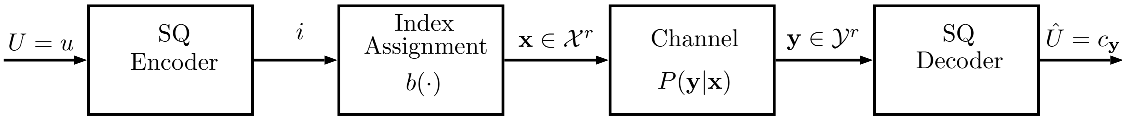

$U = u$
SQ Encoder
$i$
Index Assignment $b(\cdot)$
$\mathbf{x} \in \mathcal{X}^r$
Channel $P(\mathbf{y}|\mathbf{x})$
$\mathbf{y} \in \mathcal{Y}^r$
SQ Decoder
$\hat{U} = c_{\mathbf{y}}$

Figure 2.6: The block diagram of the coding system with index assignment.

possible to move from any states to any states within a finite number of perturbations. Details of SA is described in Algorithm 2. The exact same set of parameters as those used in [40] are used in this thesis which are listed in Table 2.1. It is necessary to note that the cooling schedule is such that the temperature, $T$, is reduced by a factor $\alpha = 0.97$ either after five non-consecutive drops in channel distortion, $D_c$, ($N_{success} = 5$) or after 200 perturbation ($N_{cut} = 200$); whichever occurs earlier. Also, the algorithm is terminated after 50000 consecutive perturbation ($N_{failure} = 50000$) that do not result in a drop in channel distortion or either the temperature, $T$, is below the freezing threshold ($T_f = 2.5 \times 10^{-4}$). As mentioned earlier, the final goal of the SA is to minimize the system energy function or in other words the channel distortion $D_c$. Since $\Delta D_c \leq 0$ indicates a drop in channel distortion, the corresponding system state vector is accepted. Although, $\Delta D_c > 0$ corresponds to an increase in channel distortion, the corresponding perturbation is still accepted with a probability that decreases as the system effective temperature decreases which means that in the beginning, almost all perturbations are accepted allowing the algorithm to escape from local minimums when the temperature is high. However, as the temperature is reduced, the perturbations causing increase in channel distortion are accepted with small probabilities; hoping, once the system is completely cooled down, the system state falls in the global minimum. It can be shown that with a sufficiently slow cooling

Table 2.1: Simulated annealing parameters.

| | |
|---|---|
| $T_0$ | 10 |
| $T_f$ | $2.5 \times 10^{-4}$ |
| $\alpha$ | 0.97 |
| $N_{failure}$ | 50000 |
| $N_{success}$ | 5 |
| $N_{cut}$ | 200 |

**Algorithm 2:** Simulated annealing algorithm for index assignment

```
Input: Initial temperature $T_0$, cooling factor $\alpha$, initial system state $\mathbf{b}$,
  freezing temperature $T_f$, maximum number of failed trials $N_{failure}$ ,
  maximum number of successful trial $N_{success}$, total of number of trials $N_{cut}$
Output:$\mathbf{b}$
$T \leftarrow T_0$
$count \leftarrow 0$
$count_{success} \leftarrow 0$
$count_{failure} \leftarrow 0$
while $T > T_f$ and $count_{failure} < N_{failure}$ do
    Choose the next state $\mathbf{b}'$ by randomly perturbing $\mathbf{b}$
    $\Delta D_c \leftarrow D(\mathbf{b}') - D(\mathbf{b})$
    if $\Delta D_c \leq 0$ then
        $\mathbf{b} \leftarrow \mathbf{b}'$
        $count_{success} \leftarrow count_{success} + 1$
        $count_{failure} \leftarrow 0$
    end
    else if with probability $e^{-\Delta D_c/T}$ , do then
        $\mathbf{b} \leftarrow \mathbf{b}'$
        $count_{failure} \leftarrow count_{failure} + 1$
    end
    else
        $count_{failure} \leftarrow count_{failure} + 1$
    end
    if $count \geq N_{cut}$  or  $count_{success} \geq N_{success}$ then
        $T \leftarrow \alpha T$
        $count \leftarrow 0$
        $count_{success} \leftarrow 0$
    end
    $count \leftarrow count + 1$
end
```

The MATLAB scripts corresponding to this algorithm are available at `https://github.com/Saeed-Rezazadeh/Simulated-Annealing.git`.

scheme and a proper perturbation method, the SA algorithm converges to the global minimum of the system in probability. It is shown in [41] and [42] that using a cooling schedule described by $T_k = {}^{c}/_{\log(k+1)}$ insures convergence in probability to the global minimum.

Note that a better design is the so-called channel optimized scalar quantizer (COSQ) where the aim is to minimize the overall system distortion $D$, given by (2.28), over both the partition set, $\mathcal{P}$, and the codebook $\mathcal{C}$. Farvardin et al. in [15] designed a joint source-channel optimized scalar quantizer whose encoder, $\mathcal{E}$ as depicted in Figure 2.6, takes the source sample $U$ from a continuous alphabet as input and outputs the channel input $\mathbf{x} \in \mathcal{X}^r$. Similar to the design of an $N$-level (i.e., $r$-bit) SQ, the encoding regions and the reconstruction codebook must satisfy the following optimality conditions to minimize the overall distortion given by (2.28):

- *Generalized NNC*: Rewriting the distortion function in (2.28) as

$$D = \sum_{\mathbf{x} \in \mathcal{X}^r} \int_{\mathcal{S}_{\mathbf{x}}} f_U(u) \Big\{ \sum_{\mathbf{y} \in \mathcal{Y}^r} P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})(u - c_{\mathbf{y}})^2 \Big\} \mathrm{d}u \tag{2.30}$$

  yields that the problem of minimizing the average distortion is similar to the design of an SQ with the following modified distortion measure

$$d'(u, \mathbf{x}) \triangleq \sum_{\mathbf{y} \in \mathcal{Y}^r} P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x})(u - c_{\mathbf{y}})^2. \tag{2.31}$$

  Specifically, for a fixed codebook $\mathcal{C}$, the optimal partition is such that

$$\mathcal{S}_{\mathbf{x}} = \Big\{ u : d'(u, \mathbf{x}) \leq d'(u, \hat{\mathbf{x}}),\ \forall \hat{\mathbf{x}} \neq \mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}^r \Big\},\ \mathbf{x} \in \mathcal{X}^r. \tag{2.32}$$

- *Generalized CC*: For a fixed partition set, $\mathcal{P}$, the optimal codebook, $\mathcal{C}$, is given by

$$c_{\mathbf{y}} = \arg\min_{\hat{u} \in \mathbb{R}} \mathbf{E}[d(U, \hat{u}) | \mathbf{Y} = \mathbf{y}], \quad \forall \mathbf{y} \in \mathcal{Y}^r \tag{2.33}$$

and in case of the squared-error distortion measure we have,

$$c_{\mathbf{y}} = E(U|\mathbf{Y} = \mathbf{y}), \quad \mathbf{y} \in \mathcal{Y}^r. \tag{2.34}$$

Therefore, a straightforward extension of the method described in Algorithm 1 can be used to optimize the partition set $\mathcal{P}$ and the codebook $\mathcal{C}$ by replacing the NNC and CC conditions in Algorithm 1 with (2.32) and (2.33), respectively. Similar to the scalar quantizer described in Section 2.2.1, the initial codebook plays a significant role in design of an optimal COSQ. An SQ (considering a noiseless channel) is designed using the splitting algorithm to choose the initial codebook. The $r$-bit binary indices are assigned to the codewords of the SQ using the SA algorithm. The resulting codebook is then used as the initial codebook to design a $2^r$-level COSQ for a discrete channel with the lowest cross-over probability $\epsilon$. After the training is complete, we slightly increase the channel noise level and train the COSQ again, setting the previously trained codebook for small $\epsilon$ as the initial state of the system with new $\epsilon$. This process is continued until the desired channel's cross-over probability is reached.

# Chapter 3

# Channel Optimized Scalar Quantizers for TWCs

To the best of our knowledge, the problem of an $N$-level COSQ design over TWCs has not been studied[1]. In this chapter, we extend the optimality conditions of [15] to TWCs. Note that these channels need not be memoryless; in general they can have memory. Our proposed scheme judiciously mitigates the self-interference caused by either users, also the statistical dependency between the users is utilized as receivers' side information. In other words, the correlated source at each terminal is treated as side information for the source at the other terminal. As a first step, we consider a restricted TWC that does not adapt the channel inputs to prior received outputs.

## 3.1 The COSQ Design for Discrete TWCs

Consider a two-way communication system where two users wish to simultaneously exchange source data over a discrete TWC. The block diagram of the proposed system is depicted in Figure 3.1. At terminal $j$, the input source to the COSQ encoder is a real-valued memoryless process $\{U_{j,i}\}_{i=1}^{\infty}$ for $j = 1, 2$. We assume that at each time instant $i$, the source samples $U_{1,i}$ and $U_{2,i}$ are correlated in general. For the sake

[1] The system proposed in this chapter was previously presented in part in [43].

of convenience, as we deal with time-memoryless sources, we drop the time index $i$ and write $U_{j,i}$ as $U_j, j = 1, 2$. The corresponding COSQ encoder at terminal $j$ is a mapping $\mathcal{E}_j$ that takes a source realization $u_j \in \mathbb{R}$ and outputs an index $r$-tuple $\mathbf{x}_j = (x_{j1}, \ldots, x_{jr}) \in \mathcal{X}^r$ where $\mathcal{X}$ is the channel input alphabet such that

$$\mathcal{E}_j(u_j) = \mathbf{x}_j, \text{ if } u_j \in \mathcal{S}_{\mathbf{x}_j} \tag{3.1}$$

where $\mathcal{P}_j = \{\mathcal{S}_{\mathbf{x}_j} : \bigcup_{\mathbf{x}_j \in \mathcal{X}^r} \mathcal{S}_{\mathbf{x}_j} = \mathbb{R},\ \mathcal{S}_{\mathbf{x}_j} \cap \mathcal{S}_{\hat{\mathbf{x}}_j} = \emptyset,\ \forall \hat{\mathbf{x}}_j \neq \mathbf{x}_j \in \mathcal{X}^r\}$ is a partition of $\mathbb{R}$ and $r$ is the coding rate given by:

$$r = \log_2\ N \ \text{ bits/source symbol} \tag{3.2}$$

where $N$ is the total number of quantization indices (i.e., quantization levels). The $r$-tuples $\mathbf{x}_j \in \mathcal{X}^r$ are then transmitted via $r$ uses of a discrete TWC (used without adaptation) with the transition distribution $P_{\mathbf{Y}_1,\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}$. If such a channel is also memoryless then we have that

$$P_{\mathbf{Y}_1,\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2) = \prod_{i=1}^{r} P_{Y_1,Y_2|X_1,X_2}(y_{1i}, y_{2i}|x_{1i}, x_{2i})$$

where for $j = 1, 2$, $\mathbf{y}_j = (y_{j1}, \ldots, y_{jr}) \in \mathcal{Y}^r$ is the received sequence at terminal $j$ and $\mathcal{Y}$ is the channel output alphabet. Note that for both the BA-TWC and the BM-TWC with additive Markov noise (as described in Chapter 2), we have:

$$\begin{aligned} P(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2) &= P(\mathbf{Z}_1 = \mathbf{z}_1, \mathbf{Z}_2 = \mathbf{z}_2) \\ &= P(\mathbf{Z}_1 = \mathbf{z}_1)P(\mathbf{Z}_2 = \mathbf{z}_2) \end{aligned} \tag{3.3}$$

where $\{Z_{1i}\}_{i=1}^{\infty}$ and $\{Z_{2i}\}_{i=1}^{\infty}$ are assumed to be independent of each other,

$$\begin{aligned} P(\mathbf{Z}_j = \mathbf{z}_j) &= \prod_{i=1}^{r} P(Z_{ji} = z_{ji} | Z_{j(i-1)} = z_{j(i-1)}, \ldots, Z_{j(i-M)} = z_{j(i-M)}) \\ &= L \prod_{i=M+1}^{r} \Big[ \frac{\epsilon_j + \delta_j \sum\limits_{k=i-M}^{i-1} z_{jk}}{1 + M\delta_j} \Big]^{z_{ji}} \Big[ \frac{1 - \epsilon_j + \delta_j (M - \sum\limits_{k=i-M}^{i-1} z_{jk})}{1 + M\delta_j} \Big]^{1-z_{ji}}, \quad (3.4) \end{aligned}$$

$$L \triangleq \frac{\prod_{i=0}^{q_j-1}(\epsilon_j + i\delta_j) \prod_{h=0}^{M-1-q_j}(1 - \epsilon_j + h\delta_j)}{\prod_{l=1}^{M-1}(1 + l\delta_j)}, \tag{3.5}$$

$q_j = \sum_{i=1}^{M} z_{ji}$, $z_{ji} = y_{ji} \oplus x_{1i} \oplus x_{2i}$ for additive, and $z_{ji} = y_{ji} \oplus x_{1i}x_{2i}$ for multiplicative user-interference. Finally, the corresponding decoder at terminal $j$ is a mapping $\mathcal{D}_j : \mathcal{Y}^r \times \mathbb{R} \to \mathbb{R}$ that maps the received $r$-tuple $\mathbf{y}_j$ to the output levels of a quantizer codebook using the statistical dependency between sources at the two terminals as side information:

$$\mathcal{D}_j(\mathbf{y}_j, u_j) = c_{\mathbf{y}_j, u_j}, \quad c_{\mathbf{y}_j, u_j} \in \mathbb{R},\ \mathbf{y}_j \in \mathcal{Y}^r,\ u_j \in \mathbb{R}. \tag{3.6}$$

The proposed COSQ over the discrete TWC, aims to select the codebooks

$$\mathcal{C}_j \ = \ \{c_{\mathbf{y}_j, u_j} : \mathbf{y}_j \in \mathcal{Y}^r, u_j \in \mathbb{R}\}$$

and the partition sets $\mathcal{P}_j = \{\mathcal{S}_{\mathbf{x}_j} : \mathbf{x}_j \in \mathcal{X}^r\}$ to minimize the overall average mean-square error (MSE) given by:

$$D = E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2] \tag{3.7}$$

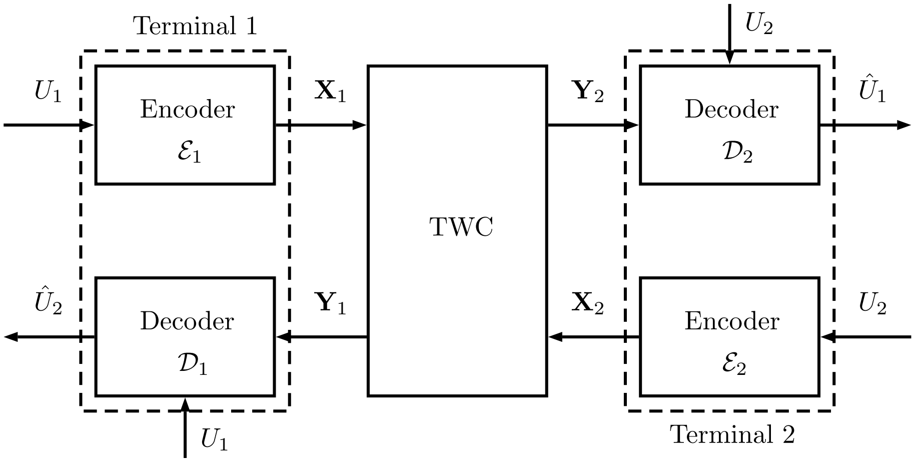


Figure 3.1: The block diagram of the COSQ TWC system.

where $\hat{U}_j$ is the reconstruction of the source $U_j, j = 1, 2$. Hence, the overall MSE is described as:

$$D = \sum_{\mathbf{x}_1 \in \mathcal{X}^r} \sum_{\mathbf{x}_2 \in \mathcal{X}^r} \int_{\mathcal{S}_{\mathbf{x}_1}} \int_{\mathcal{S}_{\mathbf{x}_2}} f_{U_1,U_2}(u_1, u_2)$$
$$\times \sum_{\mathbf{y}_1 \in \mathcal{Y}^r} \sum_{\mathbf{y}_2 \in \mathcal{Y}^r} P(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}_1, \mathbf{x}_2) \Big\{ (u_1 - c_{\mathbf{y}_2, u_2})^2 + (u_2 - c_{\mathbf{y}_1, u_1})^2 \Big\} \mathrm{d}u_1 \mathrm{d}u_2 \quad (3.8)$$

where $f_{U_1,U_2}$ is the joint probability density function (pdf) of the two sources. The performance of this system is usually measured via the average distortion $D$. For a given source, channel, and fixed rate $r$, we wish to find the optimal functions $\mathcal{E}_j$ and $\mathcal{D}_j$ for $j = 1, 2$ to minimize the overall MSE given by (3.8). The necessary conditions for optimal encoders and decoders are next determined. The optimal encoder for each user is derived given fixed decoders and the encoder of the other user. Similarly, the optimal decoder for each user is obtained given fixed encoders. Assuming known and fixed decoders ($\mathcal{D}_1$ and $\mathcal{D}_2$) and the encoder of user two ($\mathcal{E}_2$), we seek the best encoding function for user one (i.e., $\mathcal{E}_1$). Rewriting the overall MSE in (3.7) as:

$$D = \int_{-\infty}^{\infty} f_{U_1}(u_1) \times E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2 | U_1 = u_1] \mathrm{d}u_1, \tag{3.9}$$

we note by (3.1) that

$$E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2 | U_1 = u_1] = E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2 | \mathbf{X}_1 = \mathbf{x}_1, U_1 = u_1]. \tag{3.10}$$

To minimize $D$, it is sufficient to find a mapping $\mathcal{E}_1$ that minimizes the modified distortion measure defined as:

$$d_1(u_1, \mathbf{x}_1) \triangleq E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2 | \mathbf{X}_1 = \mathbf{x}_1, U_1 = u_1].$$

Therefore, given fixed codebooks $\mathcal{C}_1$, $\mathcal{C}_2$, and the partition set, $\mathcal{P}_2$, for user two, $D$ is minimized provided that the partition $\mathcal{P}_1$ satisfies:

$$\mathcal{S}_{\mathbf{x}_1} = \Big\{ u_1 : d_1(u_1, \mathbf{x}_1) \leq d_1(u_1, \hat{\mathbf{x}}_1), \ \ \forall \hat{\mathbf{x}}_1 \neq \mathbf{x}_1, \ \hat{\mathbf{x}}_1 \in \mathcal{X}^r \Big\}, \ \ \mathbf{x}_1 \in \mathcal{X}^r \tag{3.11a}$$

where

$$d_1(u_1, \mathbf{x}_1) = \sum_{\mathbf{x}_2 \in \mathcal{X}^r} \sum_{\mathbf{y}_1 \in \mathcal{Y}^r} \sum_{\mathbf{y}_2 \in \mathcal{Y}^r} P(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}_1, \mathbf{x}_2) \int_{\mathcal{S}_{\mathbf{x}_2}} f(u_2 | u_1) \Big\{ (u_1 - c_{\mathbf{y}_2, u_2})^2 + (u_2 - c_{\mathbf{y}_1, u_1})^2 \Big\} \mathrm{d}u_2. \tag{3.11b}$$

Similarly, given fixed codebooks $\mathcal{C}_1$, $\mathcal{C}_2$, and the partition set, $\mathcal{P}_1$, for user one, the average distortion $D$ is minimized if the partition $\mathcal{P}_2$ satisfies:

$$\mathcal{S}_{\mathbf{x}_2} = \Big\{u_2 : d_2(u_2, \mathbf{x}_2) \leq d_2(u_2, \hat{\mathbf{x}}_2), \;\; \forall \hat{\mathbf{x}}_2 \neq \mathbf{x}_2, \; \hat{\mathbf{x}}_2 \in \mathcal{X}^r\Big\}, \;\; \mathbf{x}_2 \in \mathcal{X}^r \tag{3.11c}$$

where

$$\begin{aligned} d_2(u_2, \mathbf{x}_2) &\triangleq E\{(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2 | \mathbf{X}_2 = \mathbf{x}_2, U_2 = u_2\} \\ &= \sum_{\mathbf{x}_1 \in \mathcal{X}^r} \sum_{\mathbf{y}_1 \in \mathcal{Y}^r} \sum_{\mathbf{y}_2 \in \mathcal{Y}^r} P_{\mathbf{Y}_1, \mathbf{Y}_2 | \mathbf{X}_1, \mathbf{X}_2}(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}_1, \mathbf{x}_2) \\ &\times \int\limits_{\mathcal{S}_{\mathbf{x}_1}} f_{U_1|U_2}(u_1|u_2) \Big\{(u_1 - c_{\mathbf{y}_2, u_2})^2 + (u_2 - c_{\mathbf{y}_1, u_1})^2\Big\} \mathrm{d}u_1. \end{aligned} \tag{3.11d}$$

It can be shown that the optimal decoders for fixed and known encoders are the conditional expectation of the source, given the channel outputs and the locally observed source samples, i.e.,

$$\begin{aligned} \hat{u}_1 &= c_{\mathbf{y}_2, u_2} \\ &= E(U_1 | \mathbf{Y}_2 = \mathbf{y}_2, U_2 = u_2) \\ &= \frac{\displaystyle\sum_{\mathbf{x}_1 \in \mathcal{X}^r} P_{\mathbf{Y}_2 | \mathbf{X}_1, \mathbf{X}_2}(\mathbf{y}_2 | \mathbf{x}_1, \mathcal{E}_2(u_2)) \int\limits_{\mathcal{S}_{\mathbf{x}_1}} u_1 f_{U_1|U_2}(u_1|u_2) \mathrm{d}u_1}{\displaystyle\sum_{\mathbf{x}_1 \in \mathcal{X}^r} P_{\mathbf{Y}_2 | \mathbf{X}_1, \mathbf{X}_2}(\mathbf{y}_2 | \mathbf{x}_1, \mathcal{E}_2(u_2)) \int\limits_{\mathcal{S}_{\mathbf{x}_1}} f_{U_1|U_2}(u_1|u_2) \mathrm{d}u_1} \end{aligned} \tag{3.12a}$$

and

$$\begin{aligned}\hat{u}_2 &= c_{\mathbf{y}_1,u_1}\\ &= E(U_2|\mathbf{Y}_1=\mathbf{y}_1,U_1=u_1)\\ &= \frac{\sum\limits_{\mathbf{x}_2\in\mathcal{X}^r} P_{\mathbf{Y}_1|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1|\mathcal{E}_1(u_1),\mathbf{x}_2)\int\limits_{\mathcal{S}_{\mathbf{x}_2}} u_2 f_{U_2|U_1}(u_2|u_1)\mathrm{d}u_2}{\sum\limits_{\mathbf{x}_2\in\mathcal{X}^r} P_{\mathbf{Y}_1|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1|\mathcal{E}_1(u_1),\mathbf{x}_2)\int\limits_{\mathcal{S}_{\mathbf{x}_2}} f_{U_2|U_1}(u_2|u_1)\mathrm{d}u_2}\end{aligned} \tag{3.12b}$$

where the marginal channel distributions are given by:

$$P_{\mathbf{Y}_1|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1|\mathcal{E}_1(u_1),\mathbf{x}_2)=\sum_{\mathbf{y}_2\in\mathcal{Y}^r} P_{\mathbf{Y}_1,\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1,\mathbf{y}_2|\mathcal{E}_1(u_1),\mathbf{x}_2),$$

and

$$P_{\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_2|\mathbf{x}_1,\mathcal{E}_2(u_2))=\sum_{\mathbf{y}_1\in\mathcal{Y}^r} P_{\mathbf{Y}_1,\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1,\mathbf{y}_2|\mathbf{x}_1,\mathcal{E}_2(u_2)).$$

Given the structure of decoders in (3.12), optimizing the decoders cannot increase the overall distortion. However, optimizing the encoder at terminal $j$ for $j=1,2$ requires a fixed encoder at the other terminal (see (3.11)). This inherent entanglement in the structure of the two encoders may increase the distortion value from one iteration to the next during the optimization procedure. Although there is no guarantee to have monotonically decreasing distortion values as a function of iteration index, the distortion, for the cases we studied in this paper, did not increase and the algorithm converged. The proposed iterative COSQ algorithm is as follows:

1. Input the joint source pdf $f_{U_1,U_2}$, the initial partition sets $\mathcal{P}_1^{(0)}$, $\mathcal{P}_2^{(0)}$, and a stopping threshold $T$.

Set $m \leftarrow 0$ and $D_0 \leftarrow \infty$.

2. - Given the partitions $\mathcal{P}_1^{(m)}$ and $\mathcal{P}_2^{(m)}$, use (3.12) to get the optimal codebooks $\mathcal{C}_1^{(m)}$ and $\mathcal{C}_2^{(m)}$.
   - Given $\mathcal{C}_1^{(m)}$ and $\mathcal{C}_2^{(m)}$, use (3.11a) for a fixed $\mathcal{P}_2^{(m)}$ to find the optimal partition $\mathcal{P}_1^{(m+1)}$. Similarly, given $\mathcal{C}_1^{(m)}$ and $\mathcal{C}_2^{(m)}$, use (3.11c) for an updated and fixed $\mathcal{P}_1^{(m+1)}$ to find the optimal partition $\mathcal{P}_2^{(m+1)}$.

3. Update the distortion $D_{m+1}$ associated with the partitions $\mathcal{P}_1^{(m+1)}$, $\mathcal{P}_2^{(m+1)}$, and codebooks $\mathcal{C}_1^{(m)}$ and $\mathcal{C}_2^{(m)}$.
   - If $\frac{|D_m - D_{m+1}|}{D_{m+1}} \leq T$, stop and output $\mathcal{C}_1^{(m)}$, $\mathcal{C}_2^{(m)}$, $\mathcal{P}_1^{(m+1)}$, and $\mathcal{P}_2^{(m+1)}$.
   - Otherwise, $m \leftarrow m + 1$ and go to step (2).

Algorithm 3 shows the design procedure of the proposed two-user COSQ. This design assumes the joint source probability distribution is known a priori; however, in real-world communication systems only samples from the two sources are available and hence we use training sequences to implement the proposed design. The proposed two-user COSQ with the training set is further described in Section A.4. We also note that the proposed COSQ design implicitly optimizes the mapping from the quantization indices to the channel inputs as the encoders at each terminal directly map source symbols to channel inputs.

## 3.2 Numerical Results

In this section we present numerical results for two types of discrete TWCs with and without memory: the binary-additive TWC (BA-TWC) with additive noise and

**Algorithm 3:** The proposed two-user COSQ for a general discrete TWC.

```
**Input**: Source pdf $f_{U_1,U_2}$, initial partition sets $\mathcal{P}_1^{(0)}$ and $\mathcal{P}_2^{(0)}$, and the stopping threshold $T$
**Output**: Partition sets $\mathcal{P}_1^{(m+1)}$ and $\mathcal{P}_2^{(m+1)}$, reconstruction codebooks, $\mathcal{C}_1^{(m)}$ and $\mathcal{C}_2^{(m)}$, and the ultimate distortion value $D^{(m+1)}$
$D^{(0)} \leftarrow \infty$
$D^{(1)} \leftarrow 0$
m $\leftarrow 0$
while $\frac{D^{(m)}-D^{(m+1)}}{D^{(m)}} > T$ do
    $c^{(m)}_{\mathbf{y}_1,u_1} \leftarrow E[U_2|\mathbf{Y}_1 = \mathbf{y}_1, U_1 = u_1]$
    $c^{(m)}_{\mathbf{y}_2,u_2} \leftarrow E[U_1|\mathbf{Y}_2 = \mathbf{y}_2, U_2 = u_2]$
    $\mathcal{S}^{(m+1)}_{\mathbf{x}_1} \leftarrow \{u : d(u_1,\mathbf{x}_1) \leq d(u_1,\hat{\mathbf{x}}_1),\ \forall\hat{\mathbf{x}}_1 \neq \mathbf{x}_1,\ \hat{\mathbf{x}}_1 \in \mathcal{X}^r\},\ \forall\mathbf{x}_1 \in \mathcal{X}^r$
    $\mathcal{S}^{(m+1)}_{\mathbf{x}_2} \leftarrow \{u : d(u_2,\mathbf{x}_2) \leq d(u_2,\hat{\mathbf{x}}_2),\ \forall\hat{\mathbf{x}}_2 \neq \mathbf{x}_2,\ \hat{\mathbf{x}}_2 \in \mathcal{X}^r\},\ \forall\mathbf{x}_2 \in \mathcal{X}^r$
    $D^{(m+1)} \leftarrow E[(U_1 - \hat{U}_1)^2 + (U_2 - \hat{U}_2)^2]$
    $m \leftarrow m + 1$
end
```

The MATLAB scripts corresponding to this algorithm are available at `https://github.com/Saeed-Rezazadeh/TWC-COSQ.git`.

the binary-multiplying TWC (BM-TWC) with additive noise as described in Section 2.1.2. The two sources outputs are drawn from a bi-variate Gaussian source with the covariance matrix

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

where $\rho$ is the correlation coefficient of the two sources which means that each source is a zero-mean and unite-variance (i.e. $\sigma_j^2 = 1$) Gaussian source. Before assessing the performance of our (full-duplex) COSQ system, we first describe its standard half-duplex counterpart.

### 3.2.1 Half-duplex COSQ System

So far, we have proposed a COSQ design where the two users exchange sources simultaneously which we refer to as the full-duplex system. We also propose another design such that when one terminal is using the TWC, the other user transmits a constant symbol. We refer to the latter approach as the half-duplex system. The block diagram of the half-duplex design is depicted in Figure 3.2. Considering a discrete TWC with arbitrary user-interference, the optimal symbol, in terms of maximal one-way information transfer, to be constantly applied by user two while user one is transmitting its source symbols is obtained by:

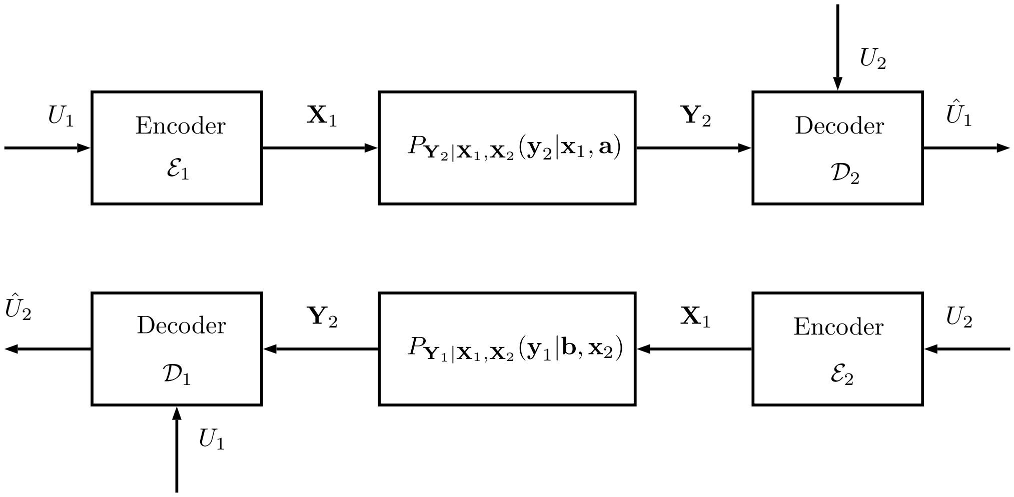


Figure 3.2: The block diagram of the half-duplex COSQ system where $\mathbf{a} = (a^*, \ldots, a^*) \in \mathcal{X}^r$ and $\mathbf{b} = (b^*, \ldots, b^*) \in \mathcal{X}^r$.

$$a^* = \arg\max_{a\in\mathcal{X}} \max_{P_{X_1|X_2}(x_1|a)} I(X_1; Y_2|X_2 = a) \tag{3.13}$$

where $P_{X_1|X_2}(x_1|a)$ is the distribution of the channel inputs generated by user one given that the other user is transmitting $X_2 = a$, and $I(X_1; Y_2|X_2 = a)$ is the conditional mutual information between $X_1$ and $Y_2$ given $X_2 = a$. As a result, the $r$-tuple that is transmitted by user two for $r$ consecutive uses of the TWC is given as: $\mathbf{a} \triangleq (a^*, \ldots, a^*) \in \mathcal{X}^r$. Similarly, the optimal value that user one sends in the half-duplex mode is given by:

$$b^* = \arg\max_{b \in \mathcal{X}} \max_{P_{X_2|X_1}(x_2|b)} I(X_2; Y_1|X_1 = b), \tag{3.14}$$

and the optimal $r$-tuple for user one is: $\mathbf{b} \triangleq (b^*, \ldots, b^*)$. The proposed half-duplex design can be treated as two independent COSQ systems for two one-way channels with side information at the decoders. We adopt the iterative algorithm along with the optimality conditions proposed in [15] and also described in Section 2.2.2 to design the optimal quantizers for half-duplex schemes.

Moreover, for the BA-TWC with additive noise, we choose the partition sets obtained from the half-duplex design as the initial partition sets for the full-duplex scheme with the same source and channel parameters and with the same encoding rate. However, for the BM-TWC the ultimate partition sets from the half-duplex scheme are used to initialize the full-duplex design with twice the quantization rate and with the same channel parameters. For every source sample, the $r/2$-tuple partition index of the half-duplex design is left-padded by all-one $r/2$-tuples for user one. The initial partition indices of the user two in the full-duplex design is obtained in the same way; however, by right-padding the partition indices of the half-duplex scheme with all-one $r/2$-tuples. This method of initialization provides each users symbols with protection against the other user's transmission.

### 3.2.2 Results and Discussion

**Two-user COSQ over discrete memoryless TWCs**

Tables 3.1 and 3.2 show the COSQ performance results for full-duplex designs compared to the corresponding half-duplex schemes for memoryless additive-noise TWCs with additive and multiplicative user interference, respectively. The performance results are in terms of signal-to-distortion ratio (SDR) which in general is defined as: $\text{SDR} = 10 \times \log_{10} \frac{\sigma_1^2+\sigma_2^2}{D}$ (in dB); specifically in this thesis we considered zero-mean unit-variance Gaussian sources (i.e., $\sigma_1^2 = \sigma_2^2 = 1$).

In Table 3.1, we also include the SDR optimum performance theoretically achievable (OPTA) using the complete JSCC theorem in [26, Theorem 3] (by calculating the sum of the users distortions) which readily applies for the case of correlated Gaussian sources sent over the discrete memoryless BA-TWC with additive noise. However in Table 3.2, since there is no complete JSCC theorem for the discrete memoryless BM-TWC (as the exact determination of its capacity region is still an open problem even in the absence of additive noise), we include an upper bound on the system's OPTA using the converse result in [25, Lemma 2][2]. The exact/upper bound OPTA values are meant to show the best performance potentially realizable if one were to employ powerful source-channel codes with unlimited delay and complexity. Naturally, since our low-delay COSQ schemes are scalar, their performance are considerably below the OPTA bound; this gap can however be reduced with the use of high-dimensional COVQs in conjunction with capacity-achieving channel codes.

Setting the quantization rate in the full-duplex system as twice the quantization rate in the half-duplex system makes the number of source symbols transmitted per

[2] Further details on how to compute the OPTA values are provided in Appendix.A.

the total number of TWC uses equal in both full-duplex and half-duplex systems. In this set-up, the full-duplex scheme always outperforms the corresponding half-duplex design for both additive and multiplicative user interferences.

Table 3.1: SDR (in dB) performance results of the full-duplex compared to the corresponding half-duplex design for a discrete memoryless BA-TWC with additive noise. OPTA values are also included.

| Correlation | | | $\epsilon_1 = 0$ | $\epsilon_1 = 0.005$ | $\epsilon_1 = 0.01$ | $\epsilon_1 = 0.05$ |
|---|---|---|---|---|---|---|
| coefficient | r | | $\epsilon_2 = 0$ | $\epsilon_2 = 0.01$ | $\epsilon_2 = 0.05$ | $\epsilon_2 = 0.10$ |
| $\rho = 0$ | 1 | half-duplex | 4.40 | 4.17 | 3.60 | 2.69 |
| | 2 | full-duplex | 9.31 | 8.18 | 6.33 | 4.34 |
| | 2 | OPTA | 12.04 | 11.27 | 9.65 | 7.35 |
| | 2 | half-duplex | 9.31 | 8.18 | 6.33 | 4.34 |
| | 4 | full-duplex | 20.24 | 13.18 | 9.57 | 6.47 |
| | 4 | OPTA | 24.08 | 22.54 | 18.99 | 14.45 |
| $\rho = 0.5$ | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | 2 | full-duplex | 9.53 | 8.47 | 6.74 | 4.88 |
| | 2 | OPTA | 13.29 | 12.52 | 10.90 | 8.60 |
| | 2 | half-duplex | 9.53 | 8.47 | 6.74 | 4.88 |
| | 4 | full-duplex | 20.31 | 13.65 | 10.01 | 7.10 |
| | 4 | OPTA | 25.33 | 23.79 | 20.23 | 15.70 |
| $\rho = 0.9$ | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.09 |
| | 2 | full-duplex | 11.63 | 11.06 | 10.25 | 9.26 |
| | 2 | OPTA | 19.25 | 18.48 | 16.86 | 14.56 |
| | 2 | half-duplex | 11.63 | 11.06 | 10.25 | 9.26 |
| | 4 | full-duplex | 20.68 | 16.76 | 14.03 | 11.65 |
| | 4 | OPTA | 31.29 | 29.75 | 26.20 | 21.66 |

For the BA-TWC with additive noise, due to the channel structure, it is feasible to perfectly "cancel" channel interference by subtracting the locally transmitted symbol from the channel outputs. However, in our decoding scheme we do not attempt to explicitly cancel the effect of self-interference from the channel outputs since one cannot undo the effects of self-interference in all TWCs such as the BM-TWC. Rather, we use the known self-interference, as shown in (3.12), to condition the decoding of

Table 3.2: SDR (in dB) performance results of the full-duplex compared to the corresponding half-duplex design for a discrete memoryless BM-TWC with additive noise. OPTA upper bounds values are also included.

| Correlation coefficient | r | | $\epsilon_1 = 0$<br>$\epsilon_2 = 0$ | $\epsilon_1 = 0.005$<br>$\epsilon_2 = 0.01$ | $\epsilon_1 = 0.01$<br>$\epsilon_2 = 0.05$ | $\epsilon_1 = 0.05$<br>$\epsilon_2 = 0.10$ |
|---|---|---|---|---|---|---|
| $\rho = 0$ | 1 | half-duplex | 4.40 | 4.17 | 3.60 | 2.69 |
| | 2 | full-duplex | 4.40 | 4.18 | 3.60 | 2.69 |
| | 2 | OPTA upper bound | 8.35 | 7.81 | 6.71 | 5.10 |
| | 2 | half-duplex | 9.31 | 8.18 | 6.33 | 4.34 |
| | 4 | full-duplex | 9.31 | 8.19 | 6.34 | 4.34 |
| | 4 | OPTA upper bound | 16.71 | 15.61 | 13.37 | 10.13 |
| $\rho = 0.5$ | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | 2 | full-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | 2 | OPTA upper bound | 9.60 | 9.05 | 7.96 | 6.34 |
| | 2 | half-duplex | 9.53 | 8.47 | 6.74 | 4.88 |
| | 4 | full-duplex | 9.53 | 8.47 | 6.74 | 4.88 |
| | 4 | OPTA upper bound | 17.95 | 16.84 | 14.62 | 11.35 |
| $\rho = 0.9$ | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.09 |
| | 2 | full-duplex | 9.75 | 9.52 | 9.30 | 8.81 |
| | 2 | OPTA upper bound | 15.55 | 15.01 | 13.93 | 12.31 |
| | 2 | half-duplex | 11.63 | 11.06 | 10.24 | 9.26 |
| | 4 | full-duplex | 12.30 | 12.20 | 11.07 | 10.10 |
| | 4 | OPTA upper bound | 23.90 | 22.83 | 20.59 | 17.34 |

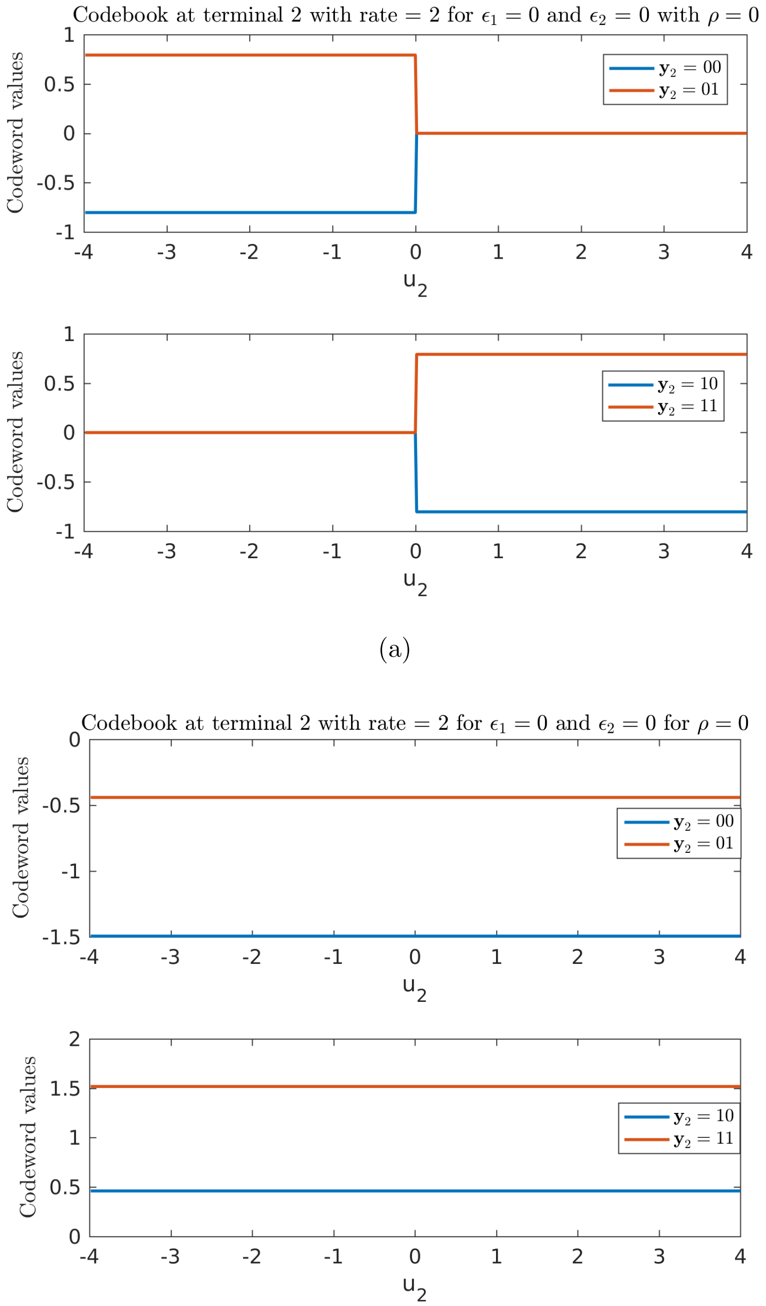
channel outputs.

Considering (3.12a), for instance at terminal two, one can observe that the side-information $U_2 = u_2$ not only does provide prior information $f_{U_1|U_2=u_2}$ for estimating $U_1$ but also determines the right channel state for decoding in the presence of self-interference, e.g., $P_{\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_2|\mathbf{x}_1, \mathcal{E}_2(u_2))$. For a noiseless BM-TWC, the codebook constellations of full-duplex and its corresponding half-duplex design at terminal two with uncorrelated sources ($\rho = 0$) and quantization rate $r = 2$ (bits/source sample) are depicted in Figure 3.3. The abrupt changes in the codebook values for the full-duplex design (Figure 3.3(a)) illustrate how the decoder tries to mitigate interference;

however, for the corresponding half-duplex setup (Figure 3.3(b)), where there is no interference as the local encoder transmits an all-one 2-tuple over the TWC, the codeword values are constant over the entire support of the locally observed source $U_2$. The same interpretation also applies to the codewords for a 2-bit COSQ over a noiseless BA-TWC with uncorrelated sources (i.e., $\rho = 0$). As shown in Figure 3.4, the codeword values have the same absolute values but with opposite signs. More specifically, the codewords corresponding to $\mathbf{y}_2 = (00)_2$ and $\mathbf{y}_2 = (01)_2$ have[3] the exact same values but with opposite signs compared to the cases when channel output $\mathbf{y}_2 = (11)_2$ and $\mathbf{y}_2 = (10)_2$ at terminal two, respectively. This means that the decoder at terminal two is trying to avoid the self-interference caused by its own transmission by using opposite signs for the codeword values, accordingly.

Moreover, the encoding criterion described in (3.11a), specifies a set $\mathcal{S}_{\mathbf{x}_1}$ on the real line whose members, if mapped to the $\mathbf{x}_1^{th}$ channel input compared to any other channel inputs $\hat{\mathbf{x}}_1 \in \mathcal{X}^r$ at terminal one, will result in a lower MSE. This criterion attempts to form $\mathcal{P}_1$ as a partition of $\mathbb{R}$ such that the self-interference is avoided by taking into account all possible interference induced by user two, i.e., by averaging over all possible values of $\mathbf{x}_2 \in \mathcal{X}^r$ in addition to combating channel noise. The encoding (3.11c) and decoding (3.12b) functions attempt to mitigate self-interference in a similar manner. For a noiseless BM-TWC, the quantization cells of the full-duplex design as well as its corresponding half-duplex scheme with uncorrelated sources ($\rho = 0$) and quantization rate $r = 2$ (bits/source sample) are shown in Figure 3.5. To combat interference, the full-duplex design (Figure 3.5(a)) does not use all partition

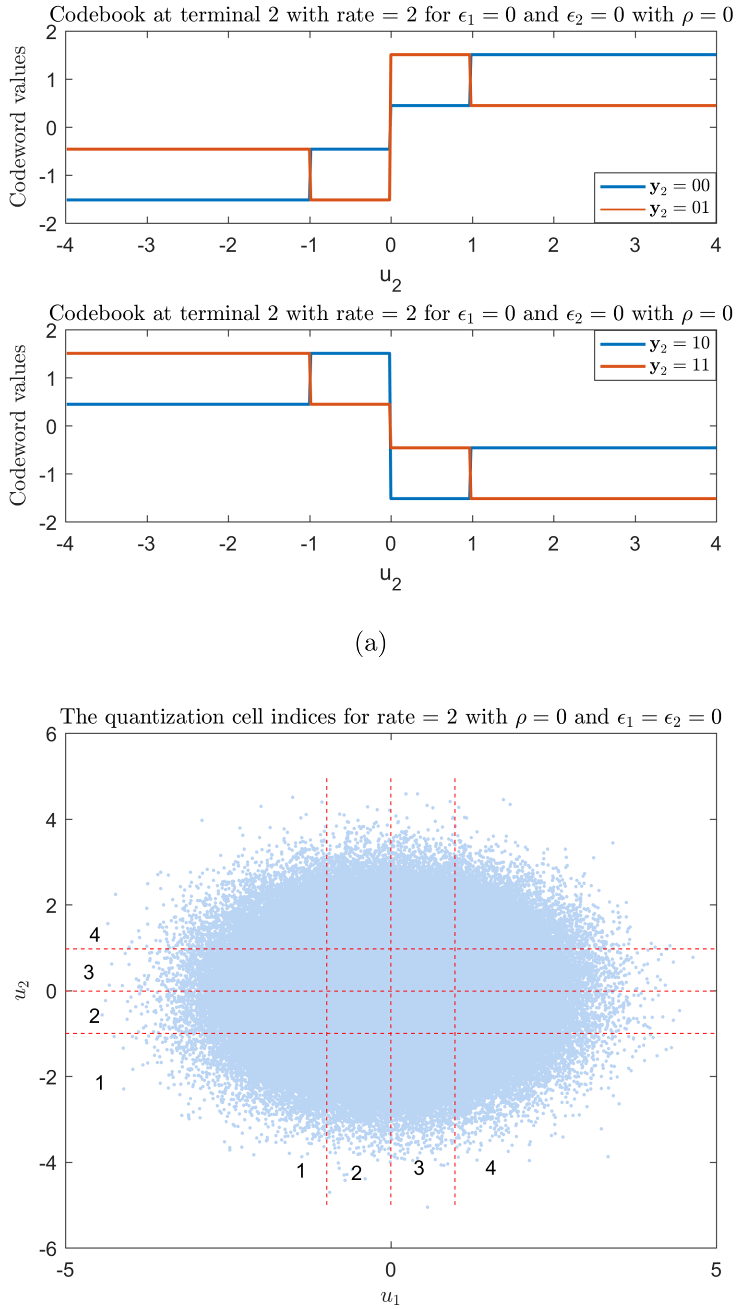
[3] The lower-script for the term $(00)_2$ describes the base with which the sequence inside the parentheses must be computed. For the example of interest, we are dealing with base-2 (i.e., binary digits).



(a)

(b)

Figure 3.3: Codebook constellations at terminal two for full-duplex (Figure 3.3(a)) and half-duplex (Figure 3.3(b)) schemes over a noiseless BM-TWC with uncorrelated sources ($\rho = 0$) and 2-bit quantizers.

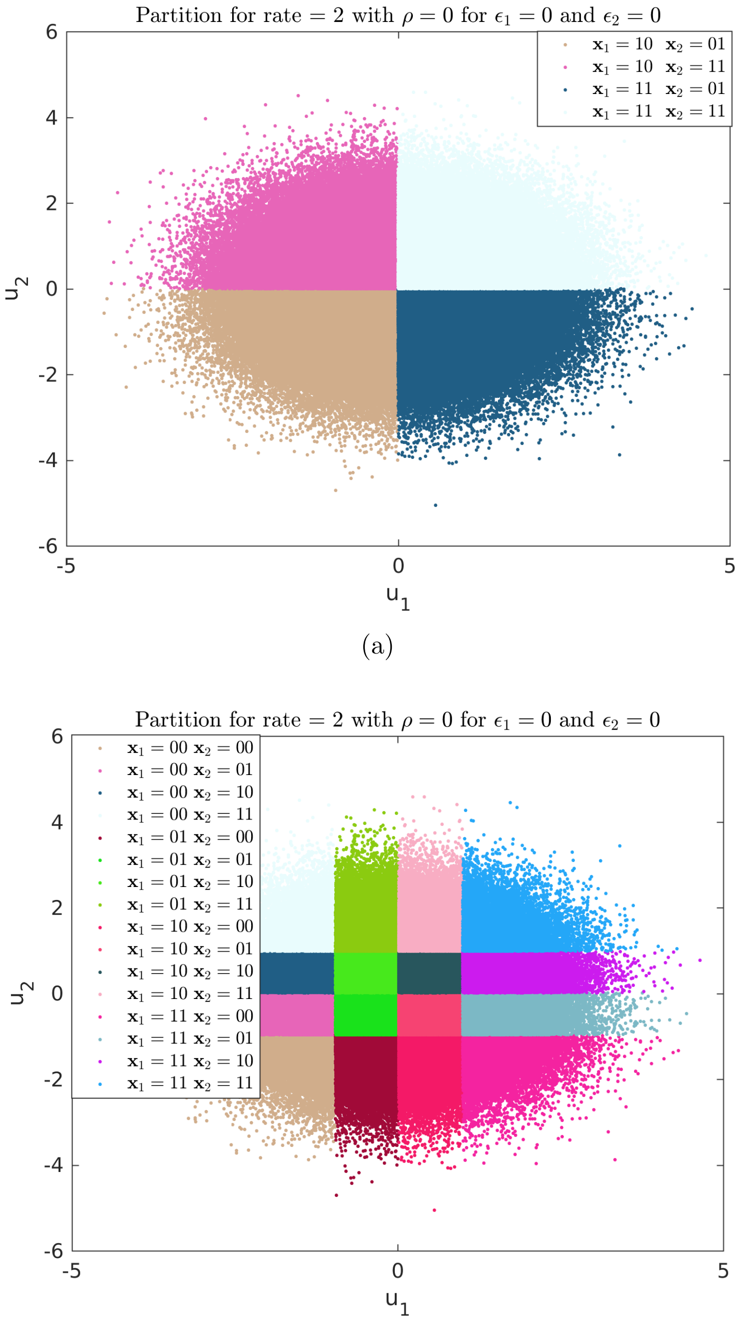


(a)

(b)

Figure 3.4: Codebook constellations (Figure 3.4(a)) at terminal two and the quantization cells (Figure 3.4(b)) for a noiseless BA-TWC with uncorrelated sources ($\rho = 0$) and 2-bit quantizers.

indices, whereas, in the half-duplex setup (Figure 3.5(b)), where interference is perfectly avoided, the encoders can use all possible quantization indices for transmission. Unlike for a BM-TWC where some of the partition indices are avoided, for the BA-TWC where interference can be perfectly *canceled*, the quantizer does not trade off the quantization accuracy to avoid interference, but rather all possible indices are used for transmission (Figure 3.4(b)). Moreover, for a single-user COSQ it is shown in [15] that the NNC is linear in $U = u$ implying that the quantization cells must be intervals and a close form expression is also derived to compute the boundaries of these cells for the squared error distortion measure. However, for a two-user system, (3.11) is not linear in $U_j = u_j$ and $\mathcal{S}_{\mathbf{x}_j}$ will not in general be an interval as depicted in Figure 3.6. Therefore, $\mathcal{S}_{\mathbf{x}_j}$ is designed according to (3.11) by numerically evaluating the modified distortion measure given by (3.11b) and (3.11d). Figure 3.6, represents the samples of the two highly correlated sources $U_j = u_j$ (i.e., $\rho = 0.99$) in blue dots and the boundaries of the quantization cells are shown for a 2-bit quantizer with red dashed lines. The decimal representation of the quantization cells indices (i.e., $\mathbf{x}_j \in \mathcal{X}^r$ which are transmitted over a discrete noiseless BA-TWC) are shown along the two axes. As it can be seen in Figure 3.6, some of the indices are used to annotate more than one quantization cell. For example the index $\mathbf{x}_1 = (10)_2 = 2$ is used for six separated regions, therefore, it is not possible to distinguish where $U_1 = u_1$ belongs to. However, if $\mathbf{x}_2 = (11)_2 = 3$, then it is highly probable that approximately $U_1 \in [2.1, 2.37]$ (i.e., the first region indexed by $\mathbf{x}_1 = (10)_2 = 2$ from the right in Figure 3.6). In this way distributed coding is used to decrease the quantization distortion. In other words, for correlated sources, the sequence transmitted by each user bears extra information about which region the source samples of the other



(a)

(b)

Figure 3.5: Quantization cells for full-duplex (Figure 3.5(a)) and half-duplex (Figure 3.5(b)) schemes over a noiseless BM-TWC with uncorrelated sources ($\rho = 0$) and 2-bit quantizers.

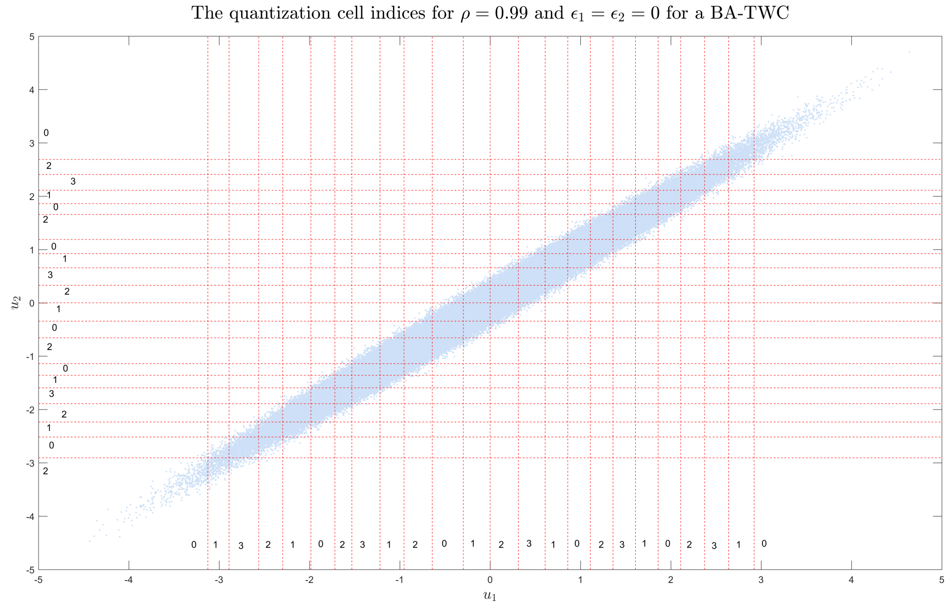
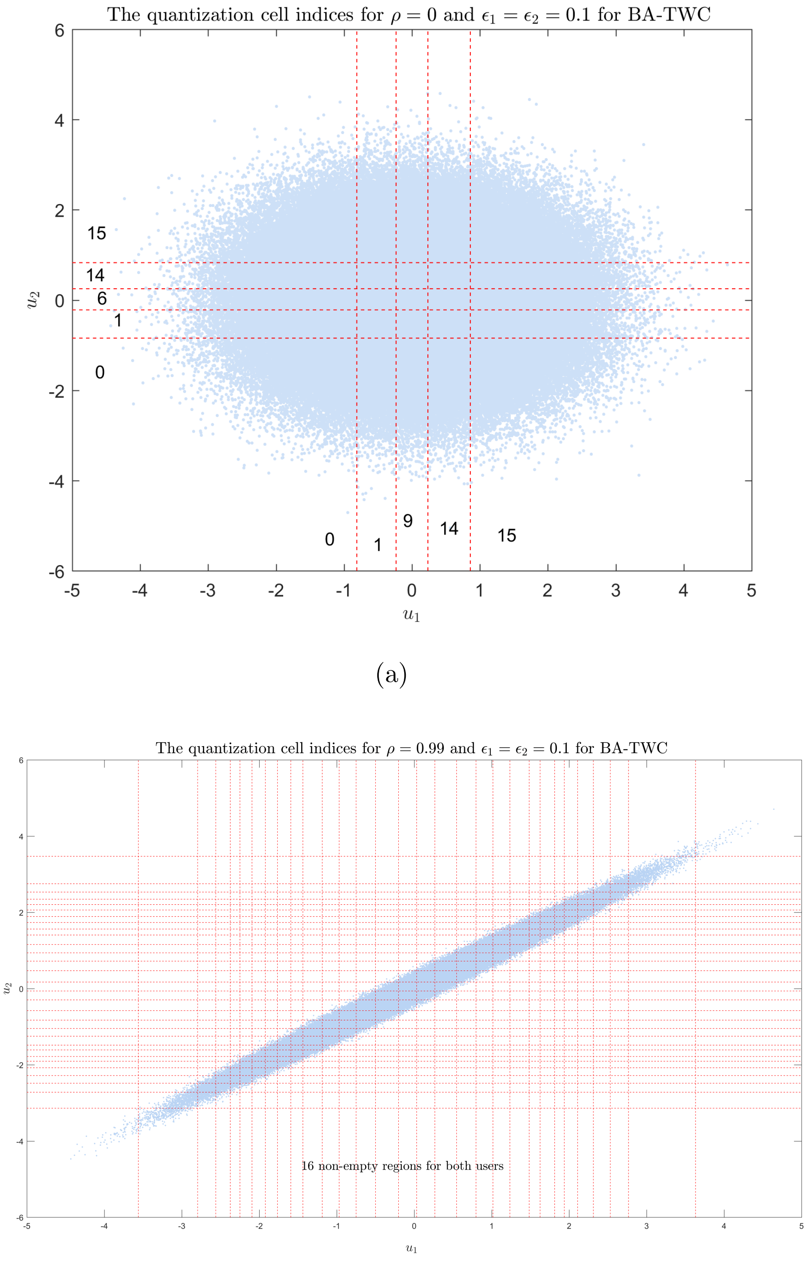


Figure 3.6: Encoding structure (quantization cells) for a 2-bit two-user COSQ over a BA-TWC with $\epsilon_1 = \epsilon_2 = 0$ and $\rho = 0.99$

user reside in with high probability which leads to improved performance in terms of SDR values.

As shown in Figure 3.7(a), for a 4-bit COSQ over a discrete memoryless BA-TWC with $\epsilon_1 = \epsilon_2 = 0.1$, and uncorrelated sources, the quantizer avoids certain partition indices to make the design robust against channel noise. The same behavior is also observed for a single-user COSQ in [15]. The useless indices are those that are never optimal according to (3.1). However, for correlated sources (Figure 3.7(b)) the locally observable source samples used as side information at the decoders boost the decoders' reconstruction reliability. Hence, the quantizer uses all possible channel inputs (i.e., quantization indices) for transmitting the two sources and does not sacrifice the quantization accuracy for less sensitivity to the channel noise which leads to significantly improved SDR performance over the case of uncorrelated sources.



(a)

(b)

Figure 3.7: Quantization cells for a 4-bit COSQ over a memoryless BA-TWC with uncorrelated sources (Figure 3.7(a)) and correlated sources $\rho = 0.99$ (Figure 3.7(b)) where $\epsilon_1 = \epsilon_2 = 0.10$.

### Two-user COSQ over discrete TWCs with memory

Most of the previous works on JSCC designs other than a few exceptions such as for [44] and [45] considered memoryless channels in their designs, disregarding the fact that real-world communication channels often have memory. In this section we investigate the performance of proposed two-user COSQ over TWCs with additive and multiplicative user-interferences where the noise processes in both directions of transmission have memory. As described in Section 2.1.2, we extend the finite memory Polya contagion urn process in [36] to model the memory of the noise process in a real-world two-user communication channel. The channel transition probability for a discrete TWC with memory is given by (3.3). The parameter $\delta_j$ for $j = 1, 2$, determines the noise correlation for the corresponding noise process $\{Z_{ji}\}_{i=1}^{\infty}$. In our simulation results we assumed $\delta_1 = \delta_2 = \delta$. Tables 3.3 and 3.4 show the performance results, in terms of SDR values, for full-duplex designs versus the corresponding half-duplex scheme as described in Section 3.2.1, for additive Markov noise TWCs with additive and multiplicative user-interference, respectively.

Note that the memory in the channel can be exploited within $r$ uses of the two-user channel. Tables 3.3 and 3.4 exhibit that, for a given $M$, the effect of intra-block memory of the TWC with both additive and multiplicative user-interference becomes more significant as the coding rate $r$ grows large. This is essentially due to the fact that the intra-block memory increases as the channel input block length (i.e., $r$) increases. The proposed two-user COSQ can properly exploit the increase in intra-block memory of the channel as long as $r > M$.

Interleaving is a traditional technique used for handling channels with memory.

Assume that we are given a discrete TWC with memory and also consider the following two scenarios. For the first scenario, suppose we perfectly know how the channel memory is characterized (such as a Markov TWC) versus the case when we know nothing about channel memory. For the latter scenario, the best approach is to use interleaving that renders the channel memoryless and then design a two-user COSQ for a discrete memoryless TWC with the same noise parameters, $\epsilon_j$, as the Markov TWC. The designed two-user COSQ is then used over an interleaved channel which is a combination of the interleaver, TWC with additive Markov noise, and the de-interleaver. It is assumed that the length of interleaving is sufficiently large such that the Markov TWC along with the interleaver and de-interleaver is perfectly equivalent to a discrete memoryless TWC. Note that the perfectly interleaved system corresponds to the two-user COSQ designed for a memoryless channel (i.e., $\delta = 0$). On the other hand, if we perfectly know how to model the memory of a TWC, a good approach is use a two-user COSQ optimally designed for the TWC whose memory characteristics are perfectly known in advanced.

The SDR values given in Tables 3.1 and 3.2 represent COSQs designed for a discrete memoryless TWCs, whereas the SDR values in Tables 3.3 and 3.4 represent the performance of the two-user COSQ optimally designed for a TWC with additive Markov noise. Comparing the SDR performances in Table 3.1 with SDR values in Table 3.3 reveals that for the additive user-interference the largest gain one can achieve by exploiting the channel memory compared to the case where the TWC is fully interleaved is 3.98 dB which occurs for $r = 4$, $\epsilon_1 = 0.05$, $\epsilon_2 = 0.1$, $\rho = 0.5$, and $\delta = 10$. Likewise, for the BM-TWC with additive Markov noise, the largest gain is 3.12 dB which occurs for $r = 4$, $\epsilon_1 = 0.05$, $\epsilon_2 = 0.1$, $\rho = 0$, and $\delta = 10$. Note that in our experimental results, we considered $M = 1$. Therefore, in order to obtain improvements due exploiting the TWC's memory, the encoding rate must be $r > 1$, as a result the performance of the half-duplex schemes for $r = 1$ is independent of the value of $\delta$. Moreover, as mentioned earlier, the effects of TWC's memory becomes more significant as the encoding rate increases. Although, there are slight improvements for full-duplex schemes with $r = 2$, the largest gains occur with $r = 4$ for TWCs with both additive and multiplicative user-interferences.

Table 3.3: SDR (in dB) performance results of the full-duplex compared to the corresponding half-duplex design for a discrete Markov BA-TWC with memory order $M = 1$. OPTA values are also included.

| Correlation | | | | $\epsilon_1 = 0$ | $\epsilon_1 = 0.005$ | $\epsilon_1 = 0.01$ | $\epsilon_1 = 0.05$ |
|---|---|---|---|---|---|---|---|
| coefficient | $\delta$ | r | | $\epsilon_2 = 0$ | $\epsilon_2 = 0.01$ | $\epsilon_2 = 0.05$ | $\epsilon_2 = 0.10$ |
| $\rho = 0$ | 5 | 1 | half-duplex | 4.40 | 4.18 | 3.60 | 2.69 |
| | | 2 | full-duplex | 9.31 | 8.38 | 6.51 | 4.34 |
| | | 2 | OPTA | 12.04 | 11.81 | 11.28 | 10.40 |
| | 10 | 1 | half-duplex | 4.40 | 4.18 | 3.60 | 2.69 |
| | | 2 | full-duplex | 9.31 | 8.41 | 6.55 | 4.37 |
| | | 2 | OPTA | 12.04 | 11.90 | 11.57 | 11.01 |
| | 5 | 2 | half-duplex | 9.31 | 8.38 | 6.51 | 4.34 |
| | | 4 | full-duplex | 20.28 | 14.08 | 9.62 | 6.80 |
| | | 4 | OPTA | 24.08 | 23.63 | 22.52 | 20.75 |
| | 10 | 2 | half-duplex | 9.31 | 8.41 | 6.55 | 4.37 |
| | | 4 | full-duplex | 20.28 | 16.68 | 13.34 | 10.25 |
| | | 4 | OPTA | 24.08 | 23.80 | 23.13 | 22.00 |
| $\rho = 0.5$ | 5 | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | | 2 | full-duplex | 9.53 | 8.68 | 7.12 | 5.31 |
| | | 2 | OPTA | 13.29 | 13.06 | 12.53 | 11.65 |
| | 10 | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | | 2 | full-duplex | 9.53 | 8.70 | 7.19 | 5.41 |
| | | 2 | OPTA | 13.29 | 13.15 | 12.82 | 12.26 |
| | 5 | 2 | half-duplex | 9.53 | 8.68 | 7.12 | 5.31 |
| | | 4 | full-duplex | 20.31 | 14.28 | 11.92 | 9.45 |
| | | 4 | OPTA | 25.33 | 24.88 | 23.77 | 22.00 |
| | 10 | 2 | half-duplex | 9.53 | 8.70 | 7.19 | 5.41 |
| | | 4 | full-duplex | 20.31 | 16.25 | 13.64 | 11.08 |
| | | 4 | OPTA | 25.33 | 25.05 | 24.38 | 23.25 |
| $\rho = 0.9$ | 5 | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.10 |
| | | 2 | full-duplex | 11.63 | 11.35 | 10.85 | 10.13 |
| | | 2 | OPTA | 19.25 | 19.03 | 18.49 | 17.61 |
| | 10 | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.10 |
| | | 2 | full-duplex | 11.63 | 11.39 | 10.94 | 10.30 |
| | | 2 | OPTA | 19.25 | 19.11 | 18.78 | 18.22 |
| | 5 | 2 | half-duplex | 11.63 | 11.35 | 10.85 | 10.13 |
| | | 4 | full-duplex | 20.64 | 18.40 | 16.31 | 14.07 |
| | | 4 | OPTA | 31.29 | 30.84 | 29.74 | 27.97 |
| | 10 | 2 | half-duplex | 11.63 | 11.39 | 10.94 | 10.30 |
| | | 4 | full-duplex | 20.66 | 18.67 | 16.70 | 14.58 |
| | | 4 | OPTA | 31.29 | 31.02 | 30.34 | 29.22 |

Table 3.4: SDR (in dB) performance results of the full-duplex compared to the corresponding half-duplex design for a discrete Markov BM-TWC with memory order $M = 1$.

| Correlation | | | | $\epsilon_1 = 0$ | $\epsilon_1 = 0.005$ | $\epsilon_1 = 0.01$ | $\epsilon_1 = 0.05$ |
|---|---|---|---|---|---|---|---|
| coefficient | $\delta$ | r | | $\epsilon_2 = 0$ | $\epsilon_2 = 0.01$ | $\epsilon_2 = 0.05$ | $\epsilon_2 = 0.10$ |
| $\rho = 0$ | 5 | 1 | half-duplex | 4.40 | 4.18 | 3.60 | 2.69 |
| | | 2 | full-duplex | 4.40 | 4.33 | 4.14 | 3.80 |
| | 10 | 1 | half-duplex | 4.40 | 4.18 | 3.60 | 2.69 |
| | | 2 | full-duplex | 4.40 | 4.36 | 4.25 | 4.05 |
| | 5 | 2 | half-duplex | 9.31 | 8.38 | 6.51 | 4.34 |
| | | 4 | full-duplex | 9.31 | 8.91 | 8.01 | 6.54 |
| | 10 | 2 | half-duplex | 9.31 | 8.41 | 6.55 | 4.37 |
| | | 4 | full-duplex | 9.31 | 9.07 | 8.50 | 7.46 |
| $\rho = 0.5$ | 5 | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | | 2 | full-duplex | 4.92 | 4.86 | 4.69 | 4.40 |
| | 10 | 1 | half-duplex | 4.92 | 4.72 | 4.21 | 3.42 |
| | | 2 | full-duplex | 4.92 | 4.88 | 4.79 | 4.61 |
| | 5 | 2 | half-duplex | 9.53 | 8.68 | 7.12 | 5.31 |
| | | 4 | full-duplex | 9.53 | 9.18 | 8.41 | 7.14 |
| | 10 | 2 | half-duplex | 9.53 | 8.70 | 7.19 | 5.41 |
| | | 4 | full-duplex | 9.53 | 9.32 | 8.82 | 7.92 |
| $\rho = 0.9$ | 5 | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.10 |
| | | 2 | full-duplex | 9.50 | 9.40 | 9.17 | 8.90 |
| | 10 | 1 | half-duplex | 8.80 | 8.67 | 8.44 | 8.10 |
| | | 2 | full-duplex | 9.46 | 9.39 | 9.20 | 9.00 |
| | 5 | 2 | half-duplex | 11.63 | 11.35 | 10.85 | 10.13 |
| | | 4 | full-duplex | 12.32 | 12.12 | 11.73 | 11.12 |
| | 10 | 2 | half-duplex | 11.63 | 11.39 | 10.94 | 10.30 |
| | | 4 | full-duplex | 12.32 | 12.20 | 11.94 | 11.52 |

# Chapter 4

# Channel Optimized Scalar Quantizers for One-Way Channels With Feedback

Consider a discrete-time, finite alphabet, single-user channel with additive random noise process which in general can have memory. The Gilbert-Elliot burst-noise channel as well as the Polya-contagion channel are examples of discrete single-user channels with memory with their descriptions explained in Section 2.1.1. In this chapter, we consider a discrete single-user channel that is accompanied with a noiseless feedback link. Shannon in [46] for the first time investigated the effect of feedback on the capacity of DMCs from an information theoretic point of view showing that feedback does not increase capacity for DMCs. Furthermore, the use of feedback does not increase the capacity for discrete additive noise channels with memory [47] nor for continuous alphabet channels with additive white Gaussian noise; however, the system's distortion can be improved using the information from a feedback channel. Amanullah et al. in [34] showed that feedback information can be used to design a COSQ over a noisy DMC with lower overall distortion compared to the case where feedback is not available. In this chapter, we additionally investigate the performance of [34] over a discrete channel with memory. We also propose an adaptive single-user COSQ which improves the scheme proposed by Amanullah in [34] particularly when the communication channel has memory.

## 4.1 Single-User Adaptive COSQ With Feedback

For the adaptive scheme described in [34], a continuous alphabet source is encoded using an $N$-level quantizer within multiple steps. In this scheme, the information received over the feedback link is used dynamically at every step to generate the next channel input. Consider that $r$ bits with $r = \log_2 N$ are divided into $L$ groups such that $r = r_1 + r_2 + \cdots + r_L$. At every step $k$, for $k = 1, 2, \ldots, L$, $r_k$ bits are used to quantize (encode) and transmit the source adaptively vis-a-vis the received symbols over the feedback link. More specifically, in the first step, the source is encoded using $r_1$ bits and then the $r_1$-tuple channel input, $X^{r_1} = x^{r_1} \in \mathcal{X}^{r_1}$, is transmitted over the channel. After receiving the corresponding sequence $Y^{r_1} = y^{r_1} \in \mathcal{Y}^{r_1}$, due to the presence of feedback, the encoder knows what codeword was received at the receiver. Therefore, in the second encoding step, $r_2$ bits are generated and used to encode the source based on the particular sequence received in the previous step. This process is continued until the entire $r$ bits are transmitted. The block diagram of the adaptive COSQ (ACOSQ) is depicted in Figure 4.1 where the encoder and decoder functions

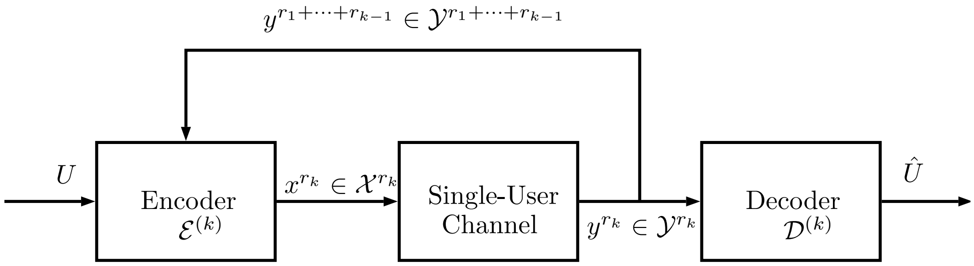

$y^{r_1+\cdots+r_{k-1}} \in \mathcal{Y}^{r_1+\cdots+r_{k-1}}$
$U$
Encoder $\mathcal{E}^{(k)}$
$x^{r_k} \in \mathcal{X}^{r_k}$
Single-User Channel
$y^{r_k} \in \mathcal{Y}^{r_k}$
Decoder $\mathcal{D}^{(k)}$
$\hat{U}$

Figure 4.1: The block diagram of the single-user ACOSQ design for $k = 1, 2, \dots, L$.

at step $k$ are described as

$$\begin{aligned} \text{Encoder: } \mathcal{E}^{(k)} &: \mathbb{R} \times \mathcal{Y}^{r_1+\cdots+r_{k-1}} \to \mathcal{X}^{r_k} \\ \mathcal{E}^{(k)}(u, y^{r_1}, \dots, y^{r_{k-1}}) &= x^{r_k} \\ \text{Decoder: } \mathcal{D}^{(k)} &: \mathcal{Y}^{r_1+\cdots+r_k} \to \mathbb{R} \\ \mathcal{D}^{(k)}(y^{r_1}, \dots, y^{r_k}) &= c_{y^{r_1},\dots,y^{r_k}} \end{aligned} \tag{4.1}$$

where $\mathcal{X}$, $\mathcal{Y}$ are the channel input and the channel output alphabets respectively, and $c_{y^{r_1},\dots,y^{r_k}}$ is the codeword corresponding to the received sequence $(y^{r_1}, \dots, y^{r_k})$ such that $\forall k \in \{1, 2, \dots, L\}$, $y^{r_k} \in \mathcal{Y}^{r_k}$.

In what follows, the process of designing the adaptive COSQ is described. At the first step, an $r_1$-bit COSQ is designed using the method described in [15] for the source with pdf $f_U(\cdot)$. Let $\mathcal{P}^{r_1} = \{\mathcal{S}_{x^{r_1}} : x^{r_1} \in \mathcal{X}^{r_1}\}$ denote the corresponding partition set of the $2^{r_1}$-level quantizer. The uncertainty about the source can be updated based on the received sequence over the feedback link. The source conditional pdf[1] given the

[1] In this thesis, the source's conditional pdf given the received sequence at every encoding step is sometimes referred to as the source's *posterior* pdf.

received sequence $Y^{r_1} = y^{r_1} \in \mathcal{Y}^{r_1}$ is given by:

$$
\begin{aligned}
f_{U|Y^{r_1}}(u|y^{r_1}) &= \frac{f_U(u)P(Y^{r_1} = y^{r_1}|U = u)}{P(Y^{r_1} = y^{r_1})} \\
& \frac{f_U(u)P(Y^{r_1} = y^{r_1}|U = u)}{\sum\limits_{x^{r_1} \in \mathcal{X}^{r_1}} P(Y^{r_1} = y^{r_1}|X^{r_1} = x^{r_1}) \int\limits_{\mathcal{S}_{x^{r_1}}} f_U(u)\mathrm{d}u}
\end{aligned} \tag{4.2}
$$

where according to (4.1) at the first step i.e., $k = 1$

$$
P(Y^{r_1} = y^{r_1}|U = u) = \sum_{x^{r_1} \in \mathcal{X}^{r_1}} P(Y^{r_1} = y^{r_1}|X^{r_1} = x^{r_1}).\mathbf{1}\{u \in \mathcal{S}_{x^{r_1}}\},
$$

and $\mathbf{1}\{\cdot\}$ is the indicator function. In the second step, an $r_2$-bit COSQ is designed, for the source pdf given by (4.2), using the approach described by [15]. The $r_2$-tuple channel input, $X^{r_2} = x^{r_2} \in \mathcal{X}^{r_2}$, generated at the second step is then transmitted over the channel. Based on the received sequence $Y^{r_2} = y^{r_2} \in \mathcal{Y}^{r_2}$, the source pdf is updated for the design of the next step quantizer in the following way

$$
\begin{aligned}
f_{U|Y^{r_1},Y^{r_2}}(u|y^{r_1}, y^{r_2}) &= \frac{f_{U,Y^{r_1}}(u, y^{r_1})P(Y^{r_2} = y^{r_2}|U = u, Y^{r_1} = y^{r_1})}{P(Y^{r_1} = y^{r_1}, Y^{r_2} = y^{r_2})} \\
&= \frac{f_{U,Y^{r_1}}(u, y^{r_1})P(Y^{r_2} = y^{r_2}|U = u, Y^{r_1} = y^{r_1})}{\sum\limits_{x^{r_2} \in \mathcal{X}^{r_2}} \int\limits_{\mathcal{S}_{x^{r_2}}} P(Y^{r_2} = y^{r_2}|X^{r_2} = x^{r_2}, Y^{r_1} = y^{r_1}, U = u) f_{U,Y^{r_1}}(u, y^{r_1})\mathrm{d}u} \qquad (4.3) \\
&= \frac{f_{U|Y^{r_1}}(u|y^{r_1})P(Y^{r_2} = y^{r_2}|U = u, Y^{r_1} = y^{r_1})}{\sum\limits_{x^{r_2} \in \mathcal{X}^{r_2}} \int\limits_{\mathcal{S}_{x^{r_2}}} P(Y^{r_2} = y^{r_2}|X^{r_2} = x^{r_2}, Y^{r_1} = y^{r_1}, U = u) f_{U|Y^{r_1}}(u|y^{r_1})\mathrm{d}u}, \qquad (4.4)
\end{aligned}
$$

where (4.3) and (4.4) are due to the chain rule and the encoding structure in (4.1) provided that $P(Y^{r_1} = y^{r_1}) \neq 0$ holds for $\forall y^{r_1} \in \mathcal{Y}^{r_1}$. Also

$$\begin{aligned} & P(Y^{r_2} = y^{r_2} | U = u, Y^{r_1} = y^{r_1}) \\ & = \sum_{x^{r_2} \in \mathcal{X}^{r_2}} P(Y^{r_2} = y^{r_2} | X^{r_2} = x^{r_2}, Y^{r_1} = y^{r_1}, U = u) . \mathbf{1}\{u \in \mathcal{S}_{x^{r_2}}\} \\ & = \sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{x^{r_2} \in \mathcal{X}^{r_2}} P(Y^{r_2} = y^{r_2} | X^{r_2} = x^{r_2}, X^{r_1} = x^{r_1}, Y^{r_1} = y^{r_1}) . \mathbf{1}\{u \in \mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{x^{r_2}}\}, \end{aligned} \tag{4.5}$$

where (4.5) is due to encoding functions at the first two steps and $\mathcal{P}^{r_2} = \{\mathcal{S}_{x^{r_2}} : x^{r_2} \in \mathcal{X}^{r_2}\}$ denotes the quantization cells at the second step. For the special case of a DMC, as noted by (2.1), we have:

$$P(Y^{r_2} = y^{r_2} | U = u, Y^{r_1} = y^{r_1}) = \sum_{x^{r_2} \in \mathcal{X}^{r_2}} P(Y^{r_2} = y^{r_2} | X^{r_2} = x^{r_2}) . \mathbf{1}\{u \in \mathcal{S}_{x^{r_2}}\},$$

It can be shown that the optimal codebook at every step is the conditional expectation of the source given the received sequences, i.e. for the $k^{th}$ step we have

$$c_{y^{r_1},\ldots,y^{r_k}} = E[U | \mathbf{Y}^{r_1 + \cdots + r_k} = (y^{r_1}, \ldots, y^{r_k})], \tag{4.6}$$

and according to the received sequence $(y^{r_1}, y^{r_2}, \cdots, y^{r_{k-1}})$ the NNC condition at the $k^{th}$ step is given by

$$\begin{aligned} \mathcal{S}_{x^{r_k}} = \Big\{ & u : d(u, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, x^{r_k}) \\ & \leq d(u, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, \hat{x}^{r_k}), \hat{x}^{r_k} \neq x^{r_k}, \forall \hat{x}^{r_k} \in \mathcal{X}^{r_k} \Big\}, \forall x^{r_k} \in \mathcal{X}^{r_k} \end{aligned} \tag{4.7}$$

where

$$d(u, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, x^{r_k}) \triangleq \sum_{y^{r_k}} P(y^{r_k}|x^{r_k}, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, u)(u - c_{y^{r_1}, y^{r_2}, \ldots, y^{r_k}})^2,$$

$$\begin{aligned} P(y^{r_k}|x^{r_k}, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, u) = \\ \sum_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum_{x^{r_{k-1}} \in \mathcal{X}^{r_{k-1}}} P(y^{r_k}|x^{r_k}, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-1}}) \\ \times \mathbf{1}\{u \in \mathcal{S}_{x^{r_1}} \cap \cdots \cap u \in \mathcal{S}_{x^{r_{k-1}}}\}, \end{aligned}$$

the corresponding posterior pdf is

$$\begin{aligned} & f_{U|Y^{r_1}, \ldots, Y^{r_{k-1}}}(u|y^{r_1}, \ldots, y^{r_{k-1}}) \\ & = \frac{f_{U|Y^{r_1}, \ldots, Y^{r_{k-2}}}(u|y^{r_1}, \ldots, y^{r_{k-2}}) P(y^{r_{k-1}}|y^{r_{k-2}}, \ldots, y^{r_1}, u)}{\displaystyle\sum_{x^{r_{k-1}} \in \mathcal{X}^{r_{k-1}}} \int_{\mathcal{S}_{x^{r_{k-1}}}} P(y^{r_{k-1}}|x^{r_{k-1}}, y^{r_{k-2}}, \ldots, y^{r_1}, u) f_{U|Y^{r_1}, \ldots, Y^{r_{k-2}}}(u|y^{r_1}, \ldots, y^{r_{k-2}}) \mathrm{d}u} \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} P(y^{r_{k-1}}|y^{r_{k-2}}, \ldots, y^{r_1}, u) &= \sum_{x^{r_{k-1}} \in \mathcal{X}^{r_{k-1}}} P(y^{r_{k-1}}|x^{r_{k-1}}, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, u).\mathbf{1}\{u \in \mathcal{S}_{x^{r_{k-1}}}\} \\ &= \sum_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum_{x^{r_{k-1}} \in \mathcal{X}^{r_{k-1}}} P(y^{r_{k-1}}|x^{r_{k-1}}, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-2}}) \\ &\quad \times \mathbf{1}\{u \in \mathcal{S}_{x^{r_1}} \cap \cdots \cap u \in \mathcal{S}_{x^{r_{k-1}}}\}, \end{aligned}$$

and $\mathcal{P}^{r_k} = \{\mathcal{S}_{x^{r_k}} : x^{r_k} \in \mathcal{X}^{r_k}\}$ is a partition of $\mathbb{R}$ at the $k^{th}$ step corresponding to the received sequence over the feedback link. Note that the conditional distortion given the received sequences $(y^{r_1}, y^{r_2}, \cdots, y^{r_{k-1}})$ at the $k^{th}$ step is given as

$$\begin{aligned} D|_{Y^{r_1},\ldots,Y^{r_{k-1}}} &= E\Big[(U-\hat{U})^2 | \mathbf{Y}^{r_1+\cdots+r_{k-1}} = (y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}})\Big] \\ &= \sum_{x^{r_k}\in\mathcal{X}^{r_k}} \sum_{y^{r_k}\in\mathcal{Y}^{r_k}} \int_{\mathcal{S}_{x^{r_k}}} f_{U|Y^{r_1},Y^{r_2},\ldots,Y^{r_{k-1}}}(u|y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}) \\ &\times (u - c_{y^{r_1},\ldots,y^{r_k}})^2 P(y^{r_k}|x^{r_k}, y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}, u)\mathrm{d}u. \end{aligned}$$

This process is continued until the last step, where an $r_L$-bit COSQ is designed given all the previously received sequences i.e., $(y^{r_1}, \ldots, y^{r_{L-1}}) \in \mathcal{Y}^{r_1+\cdots+r_{L-1}}$. The overall system distortion is the weighted sum of all conditional distortions given the previously received sequences, i.e.,

$$D = \mathbf{E}_{Y^{r_1},\ldots,Y^{r_{L-1}}}[D|_{Y^{r_1},\ldots,Y^{r_{L-1}}}] \tag{4.9}$$

where

$$\begin{aligned} D|_{Y^{r_1},\ldots,Y^{r_{L-1}}} = \sum_{y^{r_L}\in\mathcal{Y}^{r_L}} \sum_{x^{r_L}\in\mathcal{X}^{r_L}} \int_{\mathcal{S}_{x^{r_L}}} & f(u|y^{r_1}, \ldots, y^{r_{L-1}}) \\ & \times (u - c_{\mathbf{y}})^2 P(y^{r_L}|x^{r_L}, y^{r_{L-1}}, \ldots, y^{r_1}, u)\mathrm{d}u, \end{aligned} \tag{4.10}$$

$\mathbf{y} \triangleq (y^{r_1}, \ldots, y^{r_L}) \in \mathcal{Y}^r$, $\mathcal{C} = \{c_{\mathbf{y}} \in \mathbb{R} : \mathbf{y} \in \mathcal{Y}^r\}$ is the codebook at the final step, and $\mathcal{P}^{r_L} = \{\mathcal{S}_{x^{r_L}} : x^{r_L} \in \mathcal{X}^{r_L}\}$ denotes the partition set in the last step. If the channel is memoryless, by (2.1) we have:

$$D|_{Y^{r_1},\dots,Y^{r_{L-1}}} = \sum_{y^{r_L}\in\mathcal{Y}^{r_L}} \sum_{x^{r_L}\in\mathcal{X}^{r_L}} P(y^{r_L}|x^{r_L}) \int\limits_{\mathcal{S}_{x^{r_L}}} f(u|y^{r_1},\dots,y^{r_{L-1}})(u-c_{\mathbf{y}})^2\mathrm{d}u. \tag{4.11}$$

The design procedure of the ACOSQ is presented in[2] Algorithm 4.

**Algorithm 4:** The ACOSQ design procedure at the $k^{th}$ step

```
Input: Conditional source pdf $f_{U|Y^{r_1},Y^{r_2},\cdots,Y^{r_{k-1}}}$, initial codebook $\mathcal{C}^{(0)}$, and
  the stopping threshold $T$
Output: Voronoi regions, $\mathcal{P}^{(m)}$, reconstruction codebook, $\mathcal{C}^{(m+1)}$, and the
  ultimate distortion value $D^{(m+1)}$
$D^{(0)} \leftarrow \infty$
$D^{(1)} \leftarrow 0$
m $\leftarrow$ 0
while $\frac{D^{(m)}-D^{(m+1)}}{D^{(m)}} > T$ do
    $\mathcal{S}^{(m)}_{x^{r_k}} \leftarrow \Big\{u : d(u, y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}}, x^{r_k}) \leq$
      $d(u, y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}}, \hat{x}^{r_k}),\ \forall \hat{x}^{r_k} \neq x^{r_k}, \hat{x}^{r_k} \in \mathcal{X}^{r_k}\Big\},\ \forall x^{r_k} \in \mathcal{X}^{r_k}$
    $c^{(m+1)}_{y^{r_1},y^{r_2},\dots,y^{r_k}} \leftarrow E\Big[U|\mathbf{Y}^{r_1+\cdots+r_k} = (y^{r_1}, y^{r_2}, \dots, y^{r_k})\Big]$
    $D^{(m+1)} \leftarrow E\Big[(U-\hat{U})^2|\mathbf{Y}^{r_1+\cdots+r_{k-1}} = (y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}})\Big]$
    $m \leftarrow m+1$
end
```

The MATLAB scripts corresponding to this algorithm with $r = (1, 1, 1, 1)$ are available at `https://github.com/Saeed-Rezazadeh/ACOSQ-1-1-1-1.git`.

As mentioned earlier, once the quantizers are designed, the source is also encoded in $L$ steps. More specifically, at the first step, the source output is compared to the partition set $\mathcal{P}^{r_1}$, mapped to the nearest quantization cell index and transmitted over the channel. Due to the presence of feedback, the encoder knows about the received sequence at the receiver; therefore, for the transmission of the next $r_2$ bits

[2] The MATLAB scripts corresponding to the ACOSQ design of Section. 4.1 with different bit allocations are available at `https://github.com/Saeed-Rezazadeh/`.

in the second step, the source output is compared to the partition set $\mathcal{P}^{r_2}$ for the posterior distribution corresponding to the received sequence. This means that the source output is compared to a *different* partition set depending on which sequence had been received in the previous step. The third step involves the transmission of the next $r_3$ bits based on the previously received sequences $y^{r_1} \in \mathcal{Y}^{r_1}$ and $y^{r_2} \in \mathcal{Y}^{r_2}$. This process is continued until the entire $r$ bits are transmitted.

It is necessary to mention that based on (4.10), only the reconstruction codebook from the last step is of significance for the computation of the overall system distortion; however, the intermediate quantization cells are needed during the transmission phase. Moreover, the quantization cells at every encoding step is optimized subject to minimizing the conditional distortion given all the previous received sequences up to that particular step without considering the effects of future channel inputs that will be generated in the succeeding encoding steps. Finally, note that, for a discrete channel with memory, exploiting the intra-dependency between the channel inputs by considering future transmissions to design the intermediate quantizers could potentially lead to improving the overall system distortion. In the next section, we propose an adaptive COSQ scheme which takes into account future transmissions to *design* the quantizers at every encoding step.

## 4.2 Improvements to the Single-User Adaptive COSQ With Feedback

Similar to the scheme described in Section 4.1, we propose an $N$-level adaptive COSQ which is designed within $L$ steps. Also, the transmission of the source is a multistep process such that $r = r_1 + \cdots + r_L$ where $r = \log_2 N$ is the total number of bits to be transmitted and $r_k$, for $k = 1, 2, \ldots, L$, is the number of bits used to encode and transmit the source at the $k^{th}$ step. At the first step, $r_1$ bits, $x^{r_1} \in \mathcal{X}^{r_1}$, are used to encode the source with pdf $f_U(\cdot)$. Using the feedback channel, the encoder is informed about the received $r_1$-tuple channel output $y^{r_1} \in \mathcal{Y}^{r_1}$. In the second step, $r_2$ bits are generated based on the received channel outputs in the previous step (i.e., $y^{r_1} \in \mathcal{Y}^{r_1}$). This process is continued until the entire $r$ bits are transmitted. In the following, we delineate the differences in the design of the proposed ACOSQ and the adaptive scheme of Section 4.1.

To design the adaptive quantizer, in the first step, an optimal $r$-bit COSQ is designed where the encoder (i.e., quantization cells) and the decoder (i.e., the codebook) are optimized subject to minimizing the following distortion function

$$\begin{aligned} D &= \mathbf{E}[(U-\hat{U})^2] \\ &= \sum_{\mathbf{x}^{(1)}\in\mathcal{X}^r} \sum_{\mathbf{y}^{(1)}\in\mathcal{Y}^r} P(\mathbf{Y}^{(1)} = \mathbf{y}^{(1)} | \mathbf{X}^{(1)} = \mathbf{x}^{(1)}) \int_{\mathcal{S}_{\mathbf{x}^{(1)}}} f_U(u)(u - c_{\mathbf{y}^{(1)}})^2 \mathrm{d}u \qquad (4.12) \end{aligned}$$

where $\mathcal{P}^{(1)} = \{\mathcal{S}_{\mathbf{x}^{(1)}} \subset \mathbb{R} : \mathbf{x}^{(1)} \in \mathcal{X}^r\}$ and[3] $\mathcal{C}^{(1)} = \{c_{\mathbf{y}^{(1)}} \in \mathbb{R} : \mathbf{y}^{(1)} \in \mathcal{Y}^r\}$ are the partition set and the codebook for the first step, respectively such that

$$c_{\mathbf{y}^{(1)}} = E[U|\mathbf{Y} = \mathbf{y}^{(1)}].$$

Although, an optimal $r$-bit COSQ is designed, only $r_1$ bits, $x^{r_1} \in \mathcal{X}^{r_1}$, out of the possible $r$ bits, $\mathbf{x}^{(1)} \in \mathcal{X}^r$, are chosen for transmission. Note that the transmitted $r_1$-tuple, $x^{r_1} \in \mathcal{X}^{r_1}$, is a selection of $r_1$ bits from the $r$-tuple $\mathbf{x}^{(1)} \in \mathcal{X}^r$, hence, there are a total of $\binom{r}{r_1} \times r_1!$ possibilities to choose the $r_1$-tuple channel input. In the second

[3] Note that the digit embraced with parentheses used as super-script denotes the encoding step; e.g., $\mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}$ and $\mathbf{y}^{(k)} \in \mathcal{Y}^{r-(r_1+\cdots+r_{k-1})}$ represent the $r-(r_1+\cdots+r_{k-1})$-tuple channel input and channel output at the $k^{th}$ step, respectively.

step, another optimal $r$-bit COSQ is designed to optimize the *remaining* $r - r_1$ bits based on the previously received sequence, i.e., $y^{r_1} \in \mathcal{Y}^{r_1}$ which is the channel output corresponding to the transmitted sequence $x^{r_1} \in \mathcal{X}^{r_1}$ in the first step. In other words, in the second step, the encoding and the decoding functions are subject to minimizing the following distortion function

$$\begin{aligned} D = &\sum_{y^{r_1} \in \mathcal{Y}^{r_1}} \sum_{\mathbf{y}^{(2)} \in \mathcal{Y}^{r-r_1}} \sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}} P(Y^{r_1} = y^{r_1} | X^{r_1} = x^{r_1}) \\ &\times P(\mathbf{Y}^{(2)} = \mathbf{y}^{(2)} | \mathbf{X}^{(2)} = \mathbf{x}^{(2)}, Y^{r_1} = y^{r_1}, X^{r_1} = x^{r_1}) \int_{\mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{\mathbf{x}^{(2)}}} f_U(u)(u - c_{y^{r_1}, \mathbf{y}^{(2)}})^2 \mathrm{d}u \end{aligned} \tag{4.13}$$

where $\mathcal{S}_{x^{r_1}} = \bigcup_{\mathbf{x}' \in \mathcal{X}^{r-r_1}} \mathcal{S}_{\mathbf{x}^{(1)}}$, $\mathbf{x}' \in \mathcal{X}^{r-r_1}$ is the group of bits *not* chosen for transmission in the first step, and $\mathcal{P}^{(2)} = \{\mathcal{S}_{\mathbf{x}^{(2)}} : \mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}\}$ is the partition set for the second step such that for every $y^{r_1} \in \mathcal{Y}^{r_1}$

$$\begin{aligned} \mathcal{S}_{\mathbf{x}^{(2)}} = \Big\{ &u \in \mathbb{R} : d^{(2)}(u, y^{r_1}, x^{r_1}, \mathbf{x}^{(2)}) \\ &\leq d^{(2)}(u, y^{r_1}, x^{r_1}, \hat{\mathbf{x}}^{(2)}),\ \hat{\mathbf{x}}^{(2)} \neq \mathbf{x}^{(2)},\ \forall \hat{\mathbf{x}}^{(2)} \in \mathcal{X}^{r-r_1} \Big\},\ \forall \mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}, \end{aligned} \tag{4.14}$$

where

$$\begin{aligned} d^{(2)}(u, y^{r_1}, x^{r_1}, \mathbf{x}^{(2)}) &= \mathbf{E}[(U - \hat{U})^2 | U = u, Y^{r_1} = y^{r_1}, X^{r_1} = x^{r_1}, \mathbf{X}^{(2)} = \mathbf{x}^{(2)}] \\ &= \sum_{\mathbf{y}^{(2)} \in \mathcal{Y}^{r-r_1}} P(\mathbf{Y}^{(2)} = \mathbf{y}^{(2)} | \mathbf{X}^{(2)} = \mathbf{x}^{(2)}, Y^{r_1} = y^{r_1}, X^{r_1} = x^{r_1})(u - c_{y^{r_1}, \mathbf{y}^{(2)}})^2. \end{aligned} \tag{4.15}$$

For the special case of a memoryless channel we have:

$$d^{(2)}(u, y^{r_1}, x^{r_1}, \mathbf{x}^{(2)}) = \sum_{\mathbf{y}^{(2)} \in \mathcal{Y}^{r-r_1}} P(\mathbf{Y}^{(2)} = \mathbf{y}^{(2)} | \mathbf{X}^{(2)} = \mathbf{x}^{(2)})(u - c_{y^{r_1}, \mathbf{y}^{(2)}})^2. \tag{4.16}$$

Here, $\mathcal{C}^{(2)} = \{c_{y^{r_1}, \mathbf{y}^{(2)}} : (y^{r_1}, \mathbf{y}^{(2)}) \in \mathcal{Y}^r\}$ denotes the codebook for the second step such that

$$\begin{aligned} c_{y^{r_1}, \mathbf{y}^{(2)}} &= \mathbf{E}[U | \mathbf{Y} = (y^{r_1}, \mathbf{y}^{(2)})] \\ &= \frac{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}} P(y^{r_1} | x^{r_1}) P(\mathbf{y}^{(2)} | \mathbf{x}^{(2)}, y^{r_1}, x^{r_1}) \int_{\mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{\mathbf{x}^{(2)}}} u f_U(u) \mathrm{d}u}{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}} P(y^{r_1} | x^{r_1}) P(\mathbf{y}^{(2)} | \mathbf{x}^{(2)}, y^{r_1}, x^{r_1}) \int_{\mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{\mathbf{x}^{(2)}}} f_U(u) \mathrm{d}u}. \end{aligned} \tag{4.17}$$

If the channel is memoryless we have:

$$\begin{aligned} c_{y^{r_1}, \mathbf{y}^{(2)}} &= \mathbf{E}[U | \mathbf{Y} = (y^{r_1}, \mathbf{y}^{(2)})] \\ &= \frac{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}} P(y^{r_1} | x^{r_1}) P(\mathbf{y}^{(2)} | \mathbf{x}^{(2)}) \int_{\mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{\mathbf{x}^{(2)}}} u f_U(u) \mathrm{d}u}{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \sum_{\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}} P(y^{r_1} | x^{r_1}) P(\mathbf{y}^{(2)} | \mathbf{x}^{(2)}) \int_{\mathcal{S}_{x^{r_1}} \cap \mathcal{S}_{\mathbf{x}^{(2)}}} u f_U(u) \mathrm{d}u}. \end{aligned} \tag{4.18}$$

Similar to the previous step, only $r_2$ bits out of the possible $r - r_1$ bits are used for encoding the source at the second step. The $r_2$-tuple channel input, $x^{r_2} \in \mathcal{X}^{r_2}$ which is a selection of $r_2$ bits from the $(r - r_1)$-tuple $\mathbf{x}^{(2)} \in \mathcal{X}^{r-r_1}$, is then transmitted over the noisy channel. Note that for every received sequence $y^{r_1}$, there are a total of $\binom{r-r_1}{r_2} \times r_2!$ possibilities to choose the channel input $X^{r_2} = x^{r_2}$.

In general, the NNC and CC conditions along with the corresponding distortion function at the $k^{th}$ step for the given received sequence $(y^{r_1}, y^{r_2}, \ldots, y^{r_{k-1}}) \in \mathcal{Y}^{r_1+r_2+\cdots+r_{k-1}}$ are as follows

$$\begin{aligned}\mathcal{S}_{\mathbf{x}^{(k)}} = \Big\{u : d^{(k)}&(u, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-1}}, \mathbf{x}^{(k)}) \\ &\leq d^{(k)}(u, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-1}}, \hat{\mathbf{x}}^{(k)}), \\ &\forall \hat{\mathbf{x}}^{(k)} \neq \mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}\Big\},\ \forall \mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})} \qquad (4.19)\end{aligned}$$

where

$$\begin{aligned}&d^{(k)}(u, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-1}}, \mathbf{x}^{(k)}) \triangleq \\ &\qquad \sum_{\mathbf{y}^{(k)} \in \mathcal{Y}^{r-(r_1+\cdots+r_{k-1})}} P(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}, y^{r_1}, \ldots, y^{r_{k-1}}, x^{r_1}, \ldots, x^{r_{k-1}})(u - c_{\mathbf{y}})^2,\end{aligned}$$

$$\begin{aligned}&c_{\mathbf{y}} = E[U|\mathbf{Y} = (y^{r_1}, \ldots, y^{r_{k-1}}, \mathbf{y}^{(k)})] \\ &= \frac{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum_{\mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}} P(y^{r_1}|x^{r_1}) \cdots P(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}, \ldots, y^{r_1}, x^{r_1}) \int_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{\mathbf{x}^{(k)}}} u f_U(u) \mathrm{d}u}{\displaystyle\sum_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum_{\mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}} P(y^{r_1}|x^{r_1}) \cdots P(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}, \ldots, y^{r_1}, x^{r_1}) \int_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{\mathbf{x}^{(k)}}} f_U(u) \mathrm{d}u},\end{aligned} \qquad (4.20)$$

and $\mathcal{P}^{r_k} = \{\mathcal{S}_{x^{r_k}} : x^{r_k} \in \mathcal{X}^{r_k}\}$, such that $\mathcal{S}_{x^{r_k}} = \bigcup_{\mathbf{x}' \in \mathcal{X}^{r-(r_1+r_2+\cdots+r_k)}} \mathcal{S}_{\mathbf{x}^{(k)}}$, for $\mathbf{x}^{(k)} \in$

$\mathcal{X}^{r-(r_1+\cdots+r_{k-1})}$ and $\mathbf{x}' \in \mathcal{X}^{r-(r_1+r_2+\cdots+r_k)}$ is the group of bits not chosen for transmission in the $k^{th}$ step, and

$$\begin{aligned}
D &= \mathbf{E}[(U-\hat{U})^2] \\
&= \sum_{x^{r_1}\in\mathcal{X}^{r_1}} \cdots \sum_{\mathbf{x}^{(k)}\in\mathcal{X}^{r-(r_1+\cdots+r_{k-1})}} \sum_{y^{r_1}\in\mathcal{Y}^{r_1}} \cdots \sum_{\mathbf{y}^{(k)}\in\mathcal{Y}^{r-(r_1+\cdots+r_{k-1})}} P(y^{r_1}|x^{r_1}) \times \cdots \\
&\times P(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}, y^{r_{k-1}}, x^{r_{k-1}}, \ldots, y^{r_1}, x^{r_1}) \times \int_{\mathcal{S}_{x^{r_1}}\cap\cdots\cap\mathcal{S}_{\mathbf{x}^{(k)}}} f_U(u)(u-c_{\mathbf{y}})^2 \mathrm{d}u.
\end{aligned} \tag{4.21}$$

This process is continued until the last step, where the *last* $r_L$ bits are optimized based on all the previously received sequences over the feedback channel. More specifically, at the last step, an $r$-bit COSQ is designed such that the last $r_L$ bits are optimized subject to minimizing the following distortion function based on all the previously received sequences $(y^{r_1}, y^{r_2}, \ldots, y^{r_{L-1}}) \in \mathcal{Y}^{r_1+\cdots+r_{L-1}}$:

$$\begin{aligned}
D &= \mathbf{E}[(U-\hat{U})^2] \\
&= \sum_{x^{r_1}\in\mathcal{X}^{r_1}} \cdots \sum_{x^{r_L}\in\mathcal{X}^{r_L}} \sum_{y^{r_1}\in\mathcal{Y}^{r_1}} \cdots \sum_{y^{r_L}\in\mathcal{Y}^{r_L}} P(y^{r_1}|x^{r_1}) \times \cdots \\
&\times P(y^{r_L}|x^{r_L}, y^{r_{L-1}}, x^{r_{L-1}}, \ldots, y^{r_1}, x^{r_1}) \times \int_{\mathcal{S}_{x^{r_1}}\cap\cdots\cap\mathcal{S}_{x^{r_L}}} f_U(u)(u-c_{\mathbf{y}})^2 \mathrm{d}u
\end{aligned} \tag{4.22}$$

where $\mathbf{y} \triangleq (y^{r_1}, \ldots, y^{r_L}) \in \mathcal{Y}^r$,

$$\begin{aligned}
\mathcal{S}_{x^{r_L}} &= \{u \in \mathbb{R} : d^{(L)}(u, y^{r_1}, \cdots, y^{r_{L-1}}, x^{r_1}, \ldots, x^{r_{L-1}}, x^{r_L}) \\
&\leq d^{(L)}(u, y^{r_1}, \cdots, y^{r_{L-1}}, x^{r_1}, \ldots, x^{r_{L-1}}, \hat{x}^{r_L}),\ \forall \hat{x}^{r_L} \neq x^{r_L},\ \hat{x}^{r_L} \in \mathcal{X}^{r_L}\},\ \forall x^{r_L} \in \mathcal{X}^{r_L},
\end{aligned} \tag{4.23}$$

and

$$\begin{aligned}
&d^{(L)}(u, y^{r_1}, \cdots, y^{r_{L-1}}, x^{r_1}, \ldots, x^{r_{L-1}}, x^{r_L}) \\
&= E[(U - \hat{U})^2 | U = u, Y^{r_1} = y^{r_1}, \ldots, Y^{r_{L-1}} = y^{r_{L-1}}, X^{r_1} = x^{r_1}, \ldots, X^{r_{L-1}} = x^{r_L}] \\
&= \sum_{y^{r_L} \in \mathcal{Y}^{r_L}} P(y^{r_L} | x^{r_L}, \ldots, y^{r_1}, x^{r_1})(u - c_{\mathbf{y}})^2. \qquad (4.24)
\end{aligned}$$

If the channel is memoryless then

$$d^{(L)}(u, y^{r_1}, \cdots, y^{r_{L-1}}, x^{r_1}, \ldots, x^{r_{L-1}}, x^{r_L}) = \sum_{y^{r_L} \in \mathcal{Y}^{r_L}} P(y^{r_L} | x^{r_L})(u - c_{\mathbf{y}})^2.$$

The codeword value in the last step is given by

$$\begin{aligned}
&c_{\mathbf{y}} = E[U | \mathbf{Y} = (y^{r_1}, \ldots, y^{r_L})] \\
&= \frac{\sum\limits_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum\limits_{x^{r_L} \in \mathcal{X}^{r_L}} P(y^{r_1} | x^{r_1}) \cdots P(y^{r_L} | x^{r_L}, \ldots, y^{r_1}, x^{r_1}) \int\limits_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{x^{r_L}}} u f_U(u) \mathrm{d}u}{\sum\limits_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum\limits_{x^{r_L} \in \mathcal{X}^{r_L}} P(y^{r_1} | x^{r_1}) \cdots P(y^{r_L} | x^{r_L}, \ldots, y^{r_1}, x^{r_1}) \int\limits_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{x^{r_L}}} f_U(u) \mathrm{d}u}, \qquad (4.25)
\end{aligned}$$

and for a memoryless channel we have

$$\begin{aligned}
&c_{\mathbf{y}} = E[U | \mathbf{Y} = (y^{r_1}, \ldots, y^{r_L})] \\
&= \frac{\sum\limits_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum\limits_{\mathbf{x}^{r_L} \in \mathcal{X}^{r_L}} P(y^{r_1} | x^{r_1}) \cdots P(y^{r_L} | x^{r_L}) \int\limits_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{x^{r_L}}} u f_U(u) \mathrm{d}u}{\sum\limits_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum\limits_{x^{r_L} \in \mathcal{X}^{r_L}} P(y^{r_1} | x^{r_1}) \cdots P(y^{r_L} | x^{r_L}) \int\limits_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{x^{r_L}}} f_U(u) \mathrm{d}u}.
\end{aligned}$$

Likewise, the distortion function in the last step for a memoryless channel is given by

$$D = \sum_{x^{r_1} \in \mathcal{X}^{r_1}} \cdots \sum_{x^{r_L} \in \mathcal{X}^{r_L}} \sum_{y^{r_1} \in \mathcal{Y}^{r_1}} \cdots \sum_{y^{r_L} \in \mathcal{Y}^{r_L}} P(y^{r_1}|x^{r_1}) \times \cdots \times P(y^{r_L}|x^{r_L}) \\ \times \int_{\mathcal{S}_{x^{r_1}} \cap \cdots \cap \mathcal{S}_{x^{r_L}}} f_U(u)(u - c_{\mathbf{y}})^2 \mathrm{d}u$$

The proposed ACOSQ at the $k^{th}$ step is presented[4] in Algorithm 5. Note that the sequence for transmission at every step is chosen exhaustively. In other words, at every step, $r$ bits are used to annotate the quantization cells; however, we exhaustively search for the best channel input sequence $x^{r_k} \in \mathcal{X}^{r_k}$ that minimizes the ultimate overall system distortion given by (4.22); making the design phase of the proposed ACOSQ complex as the total number bits (i.e., $r$) and the number of encoding steps (i.e., $L$) grow large. However, once the quantizers are designed, the ultimate ACOSQ is a robust JSCC with low complexity and low delay. We note that a simple approach to reduce the system design complexity is to use the first $r_k$ bits of the generated sequence at every step for transmission instead of searching for the best $r_k$ tuple exhaustively. The experimental results for the proposed ACOSQ with and without exhaustive search are provided in Tables A.1, A.2, and A.3 in Appendix A.

Similar to the ACOSQ design of Section 4.1, the intermediate partitions sets are required to transmit the source; however, the codebook from only the last step of the design process is significant in the computation of overall system distortion. The distortion function in the last step given by (4.22) determines the quality of the quantizer.

[4] The MATLAB scripts corresponding to the ACOSQ design of Section. 4.2 with different bit allocations are available at `https://github.com/Saeed-Rezazadeh/`.

**Algorithm 5:** The proposed ACOSQ design procedure at the $k^{th}$ step

```
Input: The source pdf $f$, initial codebook $\mathcal{C}^{(0)}$, and the stopping threshold $T$
Output: Voronoi regions, $\mathcal{P}^{(m)}$, reconstruction codebook, $\mathcal{C}^{(m+1)}$, and the
  ultimate distortion value $D^{(m+1)}$
$D^{(0)} \leftarrow \infty$
$D^{(1)} \leftarrow 0$
m ← 0
while $\frac{D^{(m)}-D^{(m+1)}}{D^{(m)}} > T$ do
    $\mathcal{S}^{(m)}_{\mathbf{x}^{(k)}} \leftarrow \Big\{u : d^{(k)}(u, y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}}, \mathbf{x}^{(k)}) \leq d^{(k)}(u, y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}}, \hat{\mathbf{x}}^{(k)}),\ \forall \hat{\mathbf{x}}^{(k)} \neq \mathbf{x}^{(k)}, \hat{\mathbf{x}}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}\Big\},\ \forall \mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}$
    $c^{(m+1)}_{\mathbf{y}} \leftarrow E\Big[U|\mathbf{Y} = (y^{r_1}, y^{r_2}, \dots, y^{r_{k-1}}, \mathbf{y}^{(k)})\Big]$
    $D^{(m+1)} \leftarrow E(U - \hat{U})^2$, as given by (4.21)
    $m \leftarrow m + 1$
end
```

The MATLAB scripts corresponding to this algorithm with $r = (1, 1, 1, 1)$ are available at `https://github.com/Saeed-Rezazadeh/Proposed-ACOSQ-1-1-1-1.git`.

Similar to the scheme of [34], also described in Section 4.1, in the proposed ACOSQ, the source outputs are encoded in $L$ steps. Let $\mathcal{P}^{r_1} = \{\mathcal{S}_{x^{r_1}} : x^{r_1} \in \mathcal{X}^{r_1}\}$ denote the partition set in the first step. The source output is compared to the partition set $\mathcal{P}^{r_1}$, mapped to the nearest quantization index, and transmitted. Likewise in the second step, the source output is compared with the partition set $\mathcal{P}^{r_2} = \{\mathcal{S}_{x^{r_2}} : x^{r_2} \in \mathcal{X}^{r_2}\}$, where $\mathcal{S}_{x^{r_2}} = \bigcup_{\mathbf{x}' \in \mathcal{X}^{r-r_1-r_2}} \mathcal{S}_{\mathbf{x}^{(2)}}$ and $\mathbf{x}' \in \mathcal{X}^{r-r_1-r_2}$ is the group of bits not chosen for transmission in the second step, corresponding to the received sequence over the feedback link. Note that due to the presence of feedback at every step, the source output is compared to a *different* partition set according to the previously received sequence over the feedback link. This process is continued until the entire $r$ bits are transmitted.

## 4.3 Numerical Results

In this section, we provide the experimental results for designing a 4-bit adaptive quantizer. For such a quantization rate, different possible coding strategies are also used to further illustrate the effect of feedback throughout the transmission process. For instance to design a 4-bit quantizer, we consider $r_k = 1$ for $k = 1, 2, 3, 4$. In other words, one bit is used to encode and transmit the source at every step such that every single bit is adapted to the previously received channel outputs; utilizing the most amount of feedback information. Other possible coding strategies are also provided in this section.

The experimental results for the scheme of [34] is referred to as ACOSQ whereas the adaptive scheme presented in Section 4.2 is referred to as the proposed ACOSQ. For the sake of comparison, we provide the experimental results for the non-adaptive COSQ design [15], referred to as NA-COSQ, where the encoder outputs are not adapted to the previous channel outputs so that the source is encoded in a single step. The source to be encoded is a memoryless Gaussian source with zero mean and unit variance. The encoder outputs are transmitted over a discrete channel with additive Markov noise with memory order $M = 1$ as described in Section 2.1.1. The parameter $\delta$ determines the amount of noise correlation. The performance of the two aforementioned adaptive COSQs and the non-adaptive COSQ are presented in terms of SDR values such that $\text{SDR} = \log_{10} \frac{1}{D}$ as the source has zero-mean and unit-variance. Numerical results for the 4-bit ACOSQ design of [34], also described in Section 4.1, and the proposed ACOSQ over the binary Markov channel are presented in Tables 4.1, 4.2, and 4.3 for different values of $\delta$. Also provided in these tables is the system's OPTA obtained by evaluating $D(rC)$, where $D(\cdot)$ is the distortion-rate function of the source for the squared-error distortion measure and $C$ is the channel capacity in bits per channel use. For a given bit allocation, the best performance obtained by comparing the two adaptive schemes described in Sections 4.1 and 4.2 is shown with boldface values whereas the best performance achieved among all different bit allocations are underlined.

### Initial codebook selection for adaptive quantizers

As noted earlier, for the scheme of [34] (also referred to as ACOSQ) at every step $k$, an $r_k$-bit COSQ is designed for the source with the conditional pdf given a received sequence over the feedback link. To initialize the quantizer at each step, we use the same approach as in [15]. In other words, at each step an SQ, considering a noiseless channel, is designed for the source with the posterior pdf corresponding to the received sequence using the splitting algorithm as the initial codebook selection. The resulting codebook is used to initialize the $r_k$-bit COSQ for the channel with the lowest cross-over probability $\epsilon$. Once the quantizers for all encoding steps are designed, we slightly increase the channel cross-over probability and design all of the quantizers again, setting the previously obtained codebook (for small $\epsilon$) at every step as the initial state of the quantizer at the corresponding encoding step for the new $\epsilon$.

Moreover, the design of the proposed ACOSQ is also a multi-step process where at every step an $r$-bit COSQ is designed based on the information obtained from the feedback channel. The ultimate codebook for the last step in the ACOSQ is used as the initial codebook for the $r$-bit COSQ at the first step of the proposed ACOSQ. The codebook obtained at every step is used to initialize the $r$-bit quantizer in the next step. Similar to ACOSQ, once the quantizers in all steps are designed, we slightly increase the channel cross-over probability, $\epsilon$, and design the quantizers again, using the previously obtained codebook in the first step (for small $\epsilon$) as the initial state of the $r$-bit COSQ in the first step with new $\epsilon$. The resulting codebook obtained in the first step is then used to initialize the quantizer in the next step.

Table 4.1: SDR (in dB) performance results of the single-user ACOSQs compared to the NA-COSQ over a discrete memoryless BSC ($\delta = 0$).

| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | ACOSQ | 19.30 | 16.96 | **15.20** | 10.39 | **7.38** |
| | Proposed ACOSQ | **19.37** | **17.07** | 15.12 | **10.41** | 7.37 |
| $r = (1, 1, 2)$ | ACOSQ | 19.22 | **16.75** | **14.99** | 9.77 | 6.77 |
| | Proposed ACOSQ | **19.27** | 16.70 | 14.82 | **9.89** | **6.80** |
| $r = (1, 2, 1)$ | ACOSQ | **19.14** | **16.45** | 14.58 | 9.89 | 6.89 |
| | Proposed ACOSQ | 19.02 | 16.26 | **14.72** | **9.90** | **6.98** |
| $r = (2, 1, 1)$ | ACOSQ | 18.80 | 15.71 | 14.44 | 9.77 | 6.72 |
| | Proposed ACOSQ | **18.90** | **16.10** | **14.64** | **9.97** | **7.16** |
| $r = (1, 3)$ | ACOSQ | **18.81** | **15.82** | **13.99** | 9.20 | 6.30 |
| | Proposed ACOSQ | 18.74 | 15.73 | 13.84 | **9.20** | **6.30** |
| $r = (3, 1)$ | ACOSQ | 18.14 | 14.78 | 13.27 | 9.36 | 6.47 |
| | Proposed ACOSQ | **18.56** | **15.29** | **13.92** | **9.47** | **6.55** |
| $r = (2, 2)$ | ACOSQ | **18.74** | **15.60** | **14.21** | 8.68 | 6.07 |
| | Proposed ACOSQ | 18.63 | 15.42 | 13.78 | **9.31** | **6.52** |
| $r = 4$ | NA-COSQ | 17.51 | 14.24 | 12.50 | 8.28 | 5.88 |
| | OPTA | 23.80 | 22.98 | 22.13 | 17.18 | 12.78 |

To examine how the adaptive designs combat channel errors, consider a 4-bit ACOSQ that is to encode the source within four steps such that at every encoding step, one bit is used to encode the source samples. In the beginning, the first bit is

Table 4.2: SDR (in dB) performance results of the single-user ACOSQs compared to the NA-COSQ over the additive Markov noise channel with memory $\delta = 5$.

| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | ACOSQ | 19.30 | 17.00 | 15.27 | 11.86 | **9.99** |
| | Proposed ACOSQ | **19.68** | **17.79** | **16.22** | **12.03** | 9.96 |
| $r = (1, 1, 2)$ | ACOSQ | 19.26 | 16.89 | 15.39 | **11.69** | 9.48 |
| | Proposed ACOSQ | **19.53** | **17.40** | **15.82** | 11.52 | **9.71** |
| $r = (1, 2, 1)$ | ACOSQ | 19.13 | 16.43 | 15.02 | 10.92 | 8.31 |
| | Proposed ACOSQ | **19.57** | **17.64** | **16.15** | **11.73** | **9.51** |
| $r = (2, 1, 1)$ | ACOSQ | 19.42 | 17.36 | 15.68 | 11.73 | **9.91** |
| | Proposed ACOSQ | **19.74** | **18.05** | **16.60** | **11.73** | 9.64 |
| $r = (1, 3)$ | ACOSQ | 19.33 | 16.56 | 15.21 | **11.23** | **8.93** |
| | Proposed ACOSQ | **19.43** | **17.09** | **15.40** | 11.04 | 8.76 |
| $r = (3, 1)$ | ACOSQ | 19.35 | 17.10 | 15.50 | 11.07 | 9.11 |
| | Proposed ACOSQ | **19.50** | **17.53** | **16.10** | **11.76** | **9.67** |
| $r = (2, 2)$ | ACOSQ | 19.33 | 17.30 | 15.55 | 11.16 | 9.21 |
| | Proposed ACOSQ | **19.70** | **17.94** | **16.45** | **11.54** | **9.34** |
| $r = 4$ | NA-COSQ | 19.18 | 16.77 | 15.25 | 11.10 | 8.95 |
| | OPTA | 24.01 | 23.77 | 23.50 | 21.73 | 19.96 |

generated and transmitted over the channel. Through the feedback link, the encoder knows about the outcome of the previous transmission. As a result, the encoder either refines the source in case of the perfect reception (i.e., no error) or compensates for the erroneous reception that occurred previously using the next channel input. Particularly, for the ACOSQ presented in Section 4.1 and in the low noise level regime where channel errors rarely occur, the bit-by-bit quantization (successive refinement) is sub-optimal compared to the NA-COSQ that generates the channel input sequence

Table 4.3: SDR (in dB) performance results of the single-user ACOSQs compared to the NA-COSQ over the additive Markov noise channel with memory $\delta = 10$.

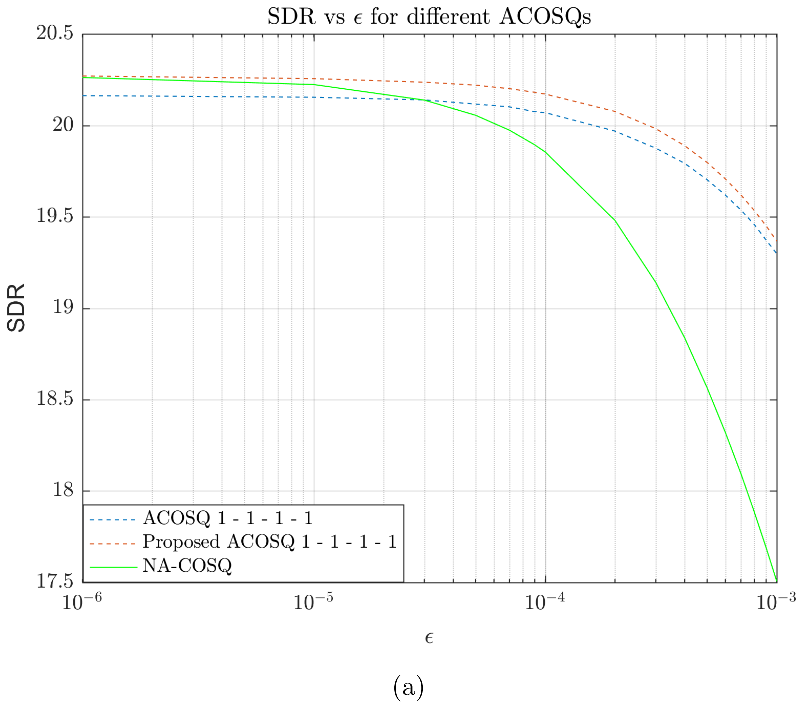
| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | ACOSQ | 19.45 | 17.47 | 16.32 | 12.93 | 11.53 |
| | Proposed ACOSQ | **19.93** | **18.77** | **17.62** | **13.82** | **11.87** |
| $r = (1, 1, 2)$ | ACOSQ | 19.46 | 17.51 | 16.48 | 12.86 | 11.08 |
| | Proposed ACOSQ | **19.84** | **18.29** | **17.23** | **13.14** | **11.18** |
| $r = (1, 2, 1)$ | ACOSQ | 19.51 | 17.22 | 15.56 | 12.28 | 10.14 |
| | Proposed ACOSQ | **19.86** | **18.54** | **17.42** | **13.51** | **11.50** |
| $r = (2, 1, 1)$ | ACOSQ | 19.64 | 18.06 | 16.73 | 13.32 | 11.68 |
| | Proposed ACOSQ | **19.95** | **18.81** | **17.71** | **13.82** | **12.01** |
| $r = (1, 3)$ | ACOSQ | 19.40 | 17.21 | 16.06 | 12.15 | 9.87 |
| | Proposed ACOSQ | **19.74** | **18.11** | **16.81** | **12.99** | **11.03** |
| $r = (3, 1)$ | ACOSQ | 19.69 | 18.05 | 16.73 | 12.50 | 10.58 |
| | Proposed ACOSQ | **19.84** | **18.41** | **17.17** | **12.97** | **10.86** |
| $r = (2, 2)$ | ACOSQ | 19.60 | 17.98 | 16.54 | 13.16 | 11.12 |
| | Proposed ACOSQ | **19.81** | **18.38** | **17.15** | **13.41** | **11.30** |
| $r = 4$ | NA-COSQ | 19.52 | 17.59 | 16.17 | 12.08 | 10.08 |
| | OPTA | 24.03 | 23.89 | 23.72 | 22.61 | 21.47 |

in a single step with the same channel parameters (see Figures 4.2(a) and 4.3(a)). This means that, for the ACOSQ design, due to the imposed tree-structure on the ultimate codebook obtained in the last step, the resulting system is a sub-optimal quantizer [19]. Thus, other allocations of quantization rate perform better for this range of noise level (see Figures 4.2(b) and 4.3(b)). In other words, the intermediate quantizers do not satisfy the optimality conditions for the overall system design. On the other hand, for the proposed ACOSQ presented in Section 4.2, the intermediate quantizers are designed subject to minimizing the overall system distortion and hence even in low noise level regime, the proposed ACOSQ performs no worse than the corresponding NA-COSQ.

Moreover, in the high noise level regime where channel errors occur frequently the error correction capability of the two adaptive schemes help compensate for channel distortion and hence achieve a higher SDR value compared to the corresponding NA-COSQ with the same channel parameters.

Similar to the two-user COSQ described in Chapter 3, for the adaptive single-user schemes of this chapter we consider two scenarios in dealing with channels with memory. If there is no prior knowledge about the channel memory, the best approach is to interleave the channel so that the channel is rendered memoryless. Then an adaptive quantizer is designed for a memoryless channel. The designed adaptive scheme is then used over an interleaved single-user channel which is the combination of the interleaver, discrete one-way channel with additive Markov noise, and the de-interleaver. Similar to the proposed two-user COSQ, we herein assume the length of the interleaving is sufficiently large so that the Markov single-user channel along with the interleaver and de-interleaver is perfectly equivalent to a single-user DMC. If the channel's memory characteristic is known a priori, then we may be better off designing our adaptive systems optimally for the channel with memory.

As noted earlier, the NA-COSQ encodes the source in a single step without adapting the channel input sequence to the previously received symbols. On the other hand if the channel inputs are adapted to the previously received symbols, then the largest gain the information on the feedback link for an interleaved channel (i.e., $\delta = 0$) can



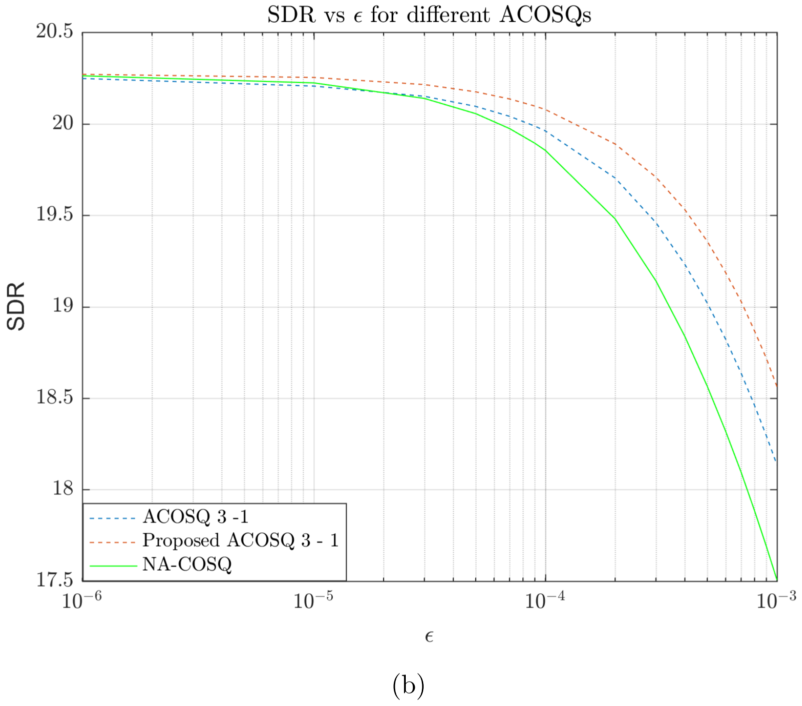
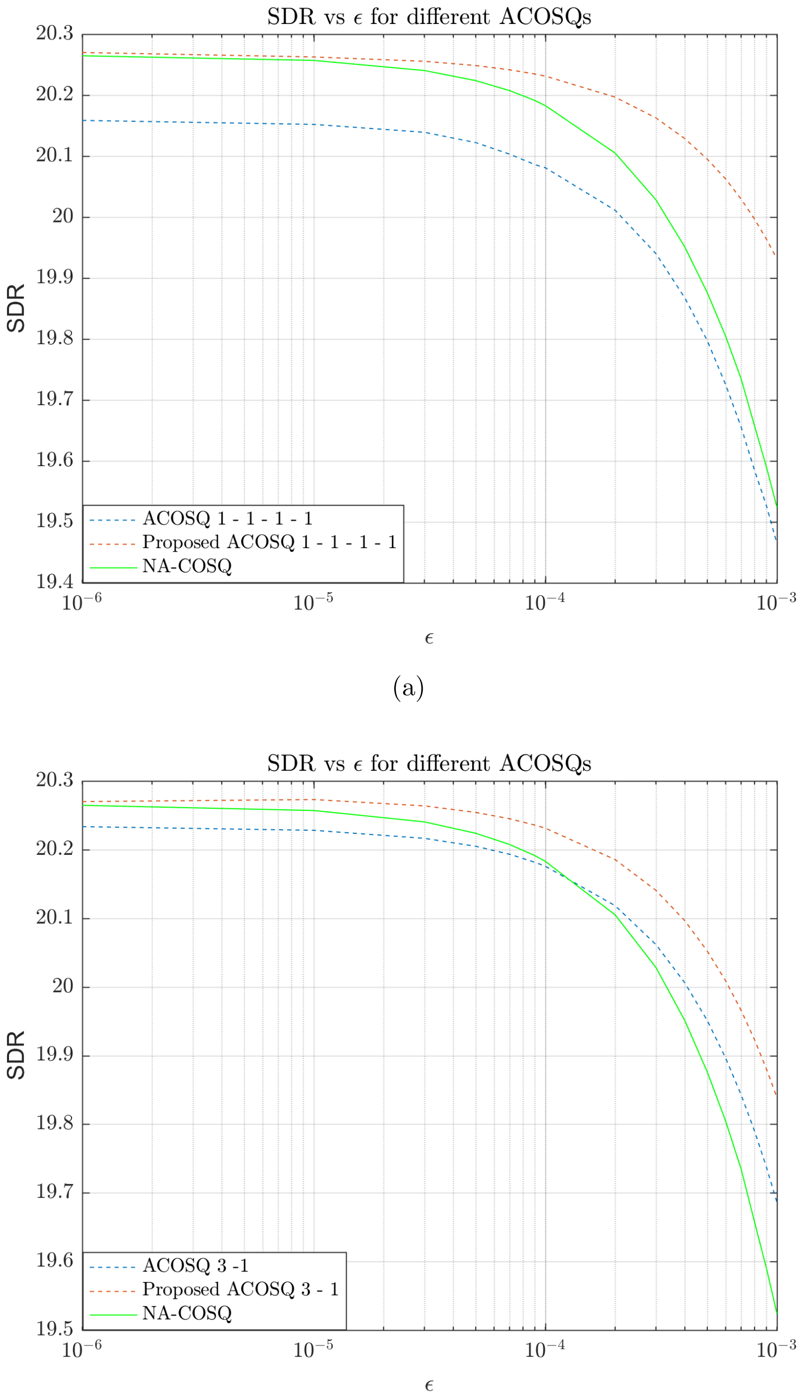


Figure 4.2: ACOSQ schemes for a discrete one-way channel with $\delta = 0$.



(a)

(b)

Figure 4.3: ACOSQ schemes for a discrete one-way channel with $\delta = 10$.

provide is 2.83 dB, compared to the corresponding NA-COSQ with the same channel parameters, which occurs for the proposed ACOSQ with $r = (1, 1, 1, 1)$ and the cross-over probability $\epsilon = 0.005$ (see Table. 4.1). Moreover, if the adaptive schemes are optimally designed for a channel with memory, the largest gain provided by the adaptive design compared to the corresponding NA-COSQ with the same channel's cross-over probability $\epsilon$ is 1.35 dB for $\delta = 5$ and 1.93 dB for $\delta = 10$ which occur for $\epsilon = 0.01$ and $\epsilon = 0.10$, respectively, using the proposed ACOSQ with $r = (2, 1, 1)$ for both values of $\delta$ (see Tables 4.2 and 4.3).

Comparing the best performance obtained among all adaptive schemes with different bit allocations[5] for a channel with highly correlated noise (i.e., $\delta = 10$) against the performance of the NA-COSQ over an interleaved channel (i.e., $\delta = 0$) given a fixed $\epsilon$ states that the largest gain is 6.13 dB which occurs at $\epsilon = 0.1$ and is provided by the proposed ACOSQ with $r = (2, 1, 1)$. This means that one can achieve up to 6.13 dB gain by incorporating both the feedback information and channel memory into the system's design and transmission processes.

It is noteworthy to mention that the two adaptive schemes with different bit allocations can very well be extended to a two-user system setup. In Appendix A, we extend the adaptive scheme of Section 4.1 to a two-user system where we consider two orthogonal one-way channels for each direction of transmission.

[5] The boldface underlined SDR values in Tables 4.1, 4.2, and 4.3.

# Chapter 5

# Conclusions

We extended the optimality conditions in [15] for the COSQ design to the TWC setup such that the statistical dependency between the two users can be exploited as side information at the decoders. Our numerical results indicated that the statistical correlation between the two users can boost the decoders reliability. Also for correlated sources, the resulting encoders use identical binary indices for multiple separated quantization cells, leading to significant reduction in distortion values caused by the quantizers. Moreover, we compared the performance of our designs with half-duplex schemes that avoid interference. We showed that our proposed full-duplex system can considerably outperform the corresponding half-duplex scheme (with identical overall transmission rate) for the BA-TWC with additive noise. Also, for the BM-TWC with additive noise where the interference cannot be perfectly eliminated, the full-duplex design provides superior performance compared to the half-duplex system. Additionally, we investigated the performance of the proposed two-user COSQ over a discrete TWC with memory. Our experimental results showed that if the system is designed optimally for the TWC with memory, one can achieve significant improvement compared to the case where the TWC is interleaved and a two-user COSQ is designed for a memoryless TWC.

Amanullah et al. proposed in [34] an adaptive quantizer for a single-user one-way DMC where the design of the quantizer as well as the source encoding are multi-step processes. In this scheme, the channel input sequence at every step is interactively adapted to the received information over the feedback link. In this work we additionally investigated the performance of the ACOSQ of [34] over discrete single-user one-way channels with additive Markovian noise. Note that the proposed scheme of [34] is totally ignorant toward the effects of channel inputs in the remaining steps when generating the channel inputs in a given encoding step. This means that the quantizers at every step are designed subject to minimizing the distortion in that particular step for the given received sequence and not subject to minimizing the overall system distortion. We proposed an ACOSQ such that not only is every channel input sequence generated adaptively vis-a-vis the received symbols over the feedback link, but also the effects of the channel inputs in the succeeding encoding steps are taken into consideration when designing the quantizer. Our experimental results showed that our proposed ACOSQ compares favorably with the ACOSQ of [34] particularly over channels with memory. Unlike the proposed ACOSQ described in Section 4.2, for the ACOSQ of Section 4.1, the effect of future channel inputs is not considered in the design of the intermediate quantizers. To have a better comparison between the proposed ACOSQ of Section 4.2 and ACOSQ of Section 4.1, a new mechanism can be proposed to consider the effect of future channel inputs when encoding the source samples using the ACOSQ of Section 4.1; therefore, both the adaptive schemes, once designed, will benefit from considering the effects of future channel inputs for transmitting their source information.

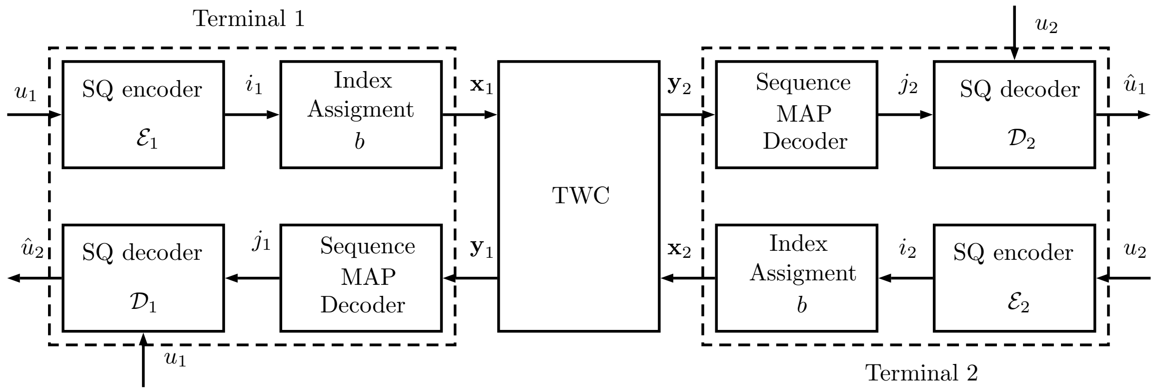


Figure 5.1: Block diagram of a JSCC system using scalar quantization and joint MAP decoder over a discrete TWC with memory.

There are several extensions of the current work that can be investigated in the future studies. Another JSCC scheme can be discussed that benefits from the channel's and sources' memory. In other words, the problem of joint sequence MAP decoding for two quantized sources transmitted over a discrete TWC with an arbitrary user-interference can be an interesting future research direction. As depicted in Figure 5.1, each real-valued source samples is first encoded by an SQ designed for a noiseless channel. Using a proper index mapping function, the quantizer outputs are transmitted over a discrete TWC which in general can have memory as described in Section 2.1.2. The received sequence is then fed into a joint sequence MAP decoder at each terminal that exploits the source and channel memory. Future work could include developing necessary and sufficient conditions to replace the delay prone joint sequence MAP decoder with an instantaneous (symbol-by-symbol) decoding rule to significantly reduce the decoding delay and complexity.

# Appendix A

# Additional Analysis

## A.1 Proposed ACOSQ With and Without Exhaustive Search

In this section we show the performance gain that can be realized via an exhaustive search determining the best channel input against the case where the first $r_k$ bits of the generated sequence are chosen as the channel input at every encoding step. Tables A.1, A.2, and A.3 show[1] numerical results , in terms of SDR values, with different channel parameters (i.e., $\epsilon$ and $\delta$) for the proposed ACOSQ scheme where in these tables for every bit allocations the SDR values in the upper row corresponds to the case where at the $k^{th}$ step, the first $r_k$ bits of the quantization index $\mathbf{x}^{(k)} \in \mathcal{X}^{r-(r_1+\cdots+r_{k-1})}$ is chosen for transmission whereas the second row shows the results of exhaustive search for choosing the channel input sequence at every step.

## A.2 ACOSQ for a Two-User System

All of the adaptive schemes discussed in Chapter 4 can be extended to a two-user setup. Consider the two-user system depicted in Figure A.1 where at each terminal a

[1] In Tables A.1, A.2, and A.3 the best approach for choosing the channel input sequence is shown in boldface letters for a given bit allocation.

Table A.1: SDR (in dB) performance results of the proposed single-user ACOSQ with and without exhaustive search compared to the NA-COSQ over a discrete memoryless BSC ($\delta = 0$).

| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | Proposed | 19.37 | 17.07 | 15.09 | 9.44 | 6.79 |
| | ACOSQ | 19.37 | 17.07 | **15.12** | **10.41** | **7.37** |
| $r = (1, 1, 2)$ | Proposed | 19.27 | 16.70 | 14.82 | 9.08 | 6.46 |
| | ACOSQ | 19.27 | 16.70 | 14.82 | **9.89** | **6.80** |
| $r = (1, 2, 1)$ | Proposed | 18.92 | 16.14 | 14.72 | 9.54 | 6.83 |
| | ACOSQ | **19.02** | **16.26** | 14.72 | **9.90** | **6.98** |
| $r = (2, 1, 1)$ | Proposed | 18.44 | 15.13 | 13.27 | 9.59 | 6.60 |
| | ACOSQ | **18.90** | **16.10** | **14.64** | **9.97** | **7.16** |
| $r = (1, 3)$ | Proposed | 18.74 | 15.72 | 13.84 | 8.85 | 6.15 |
| | ACOSQ | 18.74 | 15.73 | 13.84 | **9.20** | **6.30** |
| $r = (3, 1)$ | Proposed | 17.92 | 15.29 | 13.92 | 9.22 | 6.49 |
| | ACOSQ | **18.56** | 15.29 | 13.92 | **9.47** | **6.55** |
| $r = (2, 2)$ | Proposed | 17.96 | 14.92 | 13.15 | 8.88 | 6.19 |
| | ACOSQ | **18.63** | **15.42** | **13.78** | **9.31** | **6.52** |
| $r = 4$ | NA-COSQ | 17.51 | 14.24 | 12.50 | 8.28 | 5.88 |
| | OPTA | 23.80 | 22.98 | 22.13 | 17.18 | 12.78 |

zero-mean unit-variance Gaussian source is to be encoded and transmitted over the channel. Similar to the proposed two-user NA-COSQ design described in Chapter 3, the correlated source at each terminal can be treated as side information for the source at the other terminal. However, in this system setup, the TWC consists of two *orthogonal* one-way sub-channels each accompanied with a noiseless feedback link. In this system setup since the two sub-channels are orthogonal, the two users' transmission do not interfere with each other in both directions of transmission; therefore, the outputs of the binary orthogonal one-way channels with additive Markov noise

Table A.2: SDR (in dB) performance results of the proposed single-user ACOSQ with and without exhaustive search compared to the NA-COSQ over the additive Markov noise channel with memory $\delta = 5$.

| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | Proposed | 19.68 | 17.73 | 16.18 | 11.83 | 9.64 |
| | ACOSQ | 19.68 | **17.79** | **16.22** | **12.03** | **9.96** |
| $r = (1, 1, 2)$ | Proposed | 19.31 | 17.40 | 15.82 | 11.29 | 9.17 |
| | ACOSQ | **19.53** | 17.40 | 15.82 | **11.52** | **9.71** |
| $r = (1, 2, 1)$ | Proposed | 19.57 | 17.64 | 16.15 | 11.73 | 9.51 |
| | ACOSQ | 19.57 | 17.64 | 16.15 | 11.73 | 9.51 |
| $r = (2, 1, 1)$ | Proposed | 19.73 | 18.03 | 16.60 | 11.72 | 9.64 |
| | ACOSQ | **19.74** | **18.05** | 16.60 | **11.73** | 9.64 |
| $r = (1, 3)$ | Proposed | 19.35 | 17.04 | 15.40 | 11.04 | 8.76 |
| | ACOSQ | **19.43** | **17.09** | 15.40 | 11.04 | 8.76 |
| $r = (3, 1)$ | Proposed | 19.50 | 17.53 | 16.10 | 11.75 | 9.67 |
| | ACOSQ | 19.50 | 17.53 | 16.10 | **11.76** | 9.67 |
| $r = (2, 2)$ | Proposed | 19.69 | 17.92 | 16.45 | 11.54 | 9.34 |
| | ACOSQ | **19.70** | **17.94** | 16.45 | 11.54 | 9.34 |
| $r = 4$ | NA-COSQ | 19.18 | 16.77 | 15.25 | 11.10 | 8.95 |
| | OPTA | 24.01 | 23.77 | 23.50 | 21.73 | 19.96 |

at time $i$ for $i = 1, 2, \ldots$ can be described as

$$\begin{cases} Y_{1i} = X_{2i} \oplus Z_{1i}, \\ Y_{2i} = X_{1i} \oplus Z_{2i}, \end{cases} \tag{A.1}$$

where $Y_{ji}$, $X_{ji}$, and $Z_{ji}$ are the channel outputs, inputs and noise variables at terminal $j$ for $j = 1, 2$, respectively. The alphabets $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1\}$ are all binary. The noise variables $Z_1$ and $Z_2$, which are independent of each other and of the channel inputs, are identical to those in (3.3) and (3.4). The ACOSQ design of Section 4.1

Table A.3: SDR (in dB) performance results of the single-user ACOSQ with and without exhaustive search compared to the NA-COSQ over the additive Markov noise channel with memory $\delta = 10$.

| bit allocation | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|
| $r = (1, 1, 1, 1)$ | Proposed | 19.87 | 18.58 | 17.41 | 13.51 | 11.61 |
| | ACOSQ | **19.93** | **18.77** | **17.62** | **13.82** | **11.87** |
| $r = (1, 1, 2)$ | Proposed | 19.53 | 18.12 | 16.84 | 13.14 | 11.18 |
| | ACOSQ | **19.84** | **18.29** | **17.23** | 13.14 | 11.18 |
| $r = (1, 2, 1)$ | Proposed | 19.84 | 18.53 | 17.42 | 13.51 | 11.50 |
| | ACOSQ | **19.86** | **18.54** | 17.42 | 13.51 | 11.50 |
| $r = (2, 1, 1)$ | Proposed | 19.94 | 18.77 | 17.66 | 13.68 | 11.65 |
| | ACOSQ | **19.95** | **18.81** | **17.71** | **13.82** | **12.01** |
| $r = (1, 3)$ | Proposed | 19.70 | 18.08 | 16.81 | 12.99 | 11.03 |
| | ACOSQ | **19.74** | **18.11** | 16.81 | 12.99 | 11.03 |
| $r = (3, 1)$ | Proposed | 19.83 | 18.41 | 17.17 | 12.97 | 10.86 |
| | ACOSQ | **19.84** | 18.41 | 17.17 | 12.97 | 10.86 |
| $r = (2, 2)$ | Proposed | 19.81 | 18.38 | 17.15 | 13.04 | 11.13 |
| | ACOSQ | 19.81 | 18.38 | 17.15 | **13.41** | **11.30** |
| $r = 4$ | NA-COSQ | 19.52 | 17.59 | 16.17 | 12.08 | 10.08 |
| | OPTA | 24.03 | 23.89 | 23.72 | 22.61 | 21.47 |

with $r = (1, 1, 1, 1)$ is extended to the aforementioned two-user system setup[2]. This means that at each terminal four bits are used to encode and transmit the source. Similar to the scheme of Section 4.1, the quantizers at both terminals are designed within four steps such that at every step, one bit is generated adaptive to the received symbols over the feedback links. The memory of each sub-channel is modeled via the Polya urn contagion process described in Section 2.1.2 for a general non-orthogonal TWC where $P(Z_{ij} = 1) = \epsilon_j$ and $\delta_j$ determines the amount of noise correlation at terminal $j$ for $j = 1, 2$. Particularly in this section we considered $\epsilon_1 = \epsilon_2 = \epsilon$ and $\delta_1 = \delta_2 = \delta$.

[2] The MATLAB scripts corresponding to the ACOSQ design of Section. 4.1 with $r = (1, 1, 1, 1)$ extended to a two-user system setup are available at `https://github.com/Saeed-Rezazadeh/two-user-ACOSQ-1-1-1-1.git`.

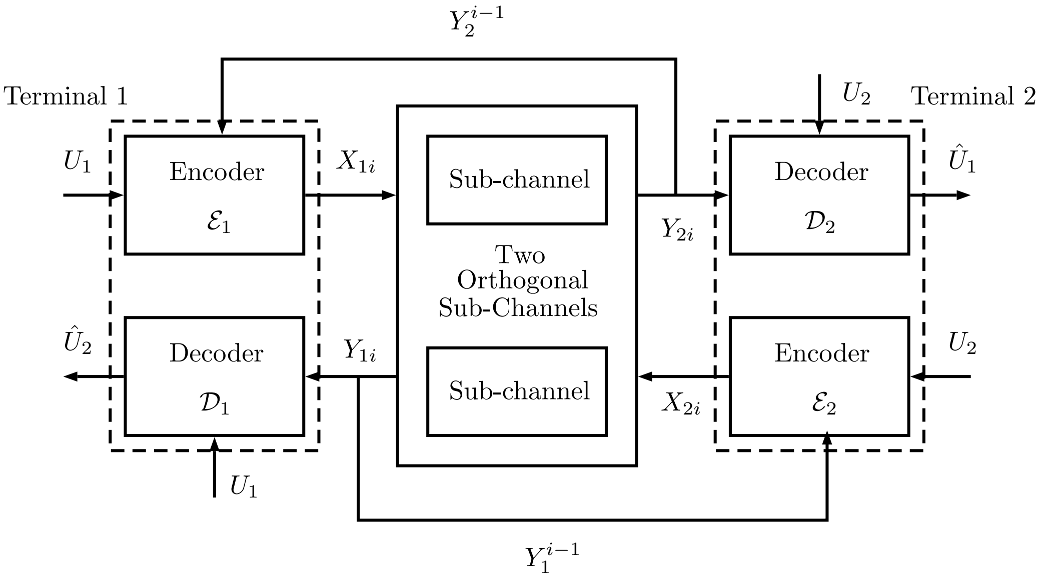


Figure A.1: The ACOSQ design with $r = (1, 1, 1, 1)$ for two orthogonal one-way channels.

Comparing the SDR values in Table A.4 shows that the two-user ACOSQ outperforms the corresponding two-user NA-COSQ design with the same channel and source parameters. More specifically, comparing the performance of the two-user ACOSQ with correlated Gaussian sources (i.e., $\rho = 0.9$) for the two-orthogonal one-way channels with the highly correlated noise (i.e., $\delta = 10$) against the performance of the two-user NA-COSQ over an interleaved channel with no side information at the decoders (i.e., $\delta = 0$ and $\rho = 0$) reveals that the largest gain is 9.06 dB which occurs for $\epsilon = 0.1$. In other words, the largest gain one can achieve by incorporating the feedback information, channel's memory, and statistical correlation between the two sources is 9.06 dB. Also, for an interleaved channel (i.e., $\delta = 0$), comparing the SDR values for the two-user ACOSQ with correlated sources (i.e., $\rho = 0.9$) and SDR values for the two-user NA-COSQ without any side information at the decoders for different values of $\epsilon$ shows that the largest gain provided by the information received

Table A.4: SDR (in dB) performance results of the two-user ACOSQ with $r = (1, 1, 1, 1)$ over two-orthogonal one-way channels with additive Markov noise compared to the two-user NA-COSQ. OPTA values are also included.

| $\delta$ | $\rho$ | | $\epsilon = 0.001$ | $\epsilon = 0.005$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | ACOSQ | 19.30 | 16.96 | 15.20 | 10.39 | 7.38 |
| | | NA-COSQ | 17.51 | 14.24 | 12.50 | 8.28 | 5.88 |
| | | OPTA | 23.80 | 22.98 | 22.13 | 17.18 | 12.78 |
| | 0.5 | ACOSQ | 19.32 | 17.02 | 15.29 | 10.65 | 7.74 |
| | | NA-COSQ | 17.76 | 14.39 | 13.01 | 8.25 | 6.04 |
| | | OPTA | 25.05 | 24.23 | 23.38 | 18.43 | 14.03 |
| | 0.9 | ACOSQ | 19.70 | 18.14 | 16.93 | 13.14 | 10.98 |
| | | NA-COSQ | 19.59 | 17.50 | 16.08 | 12.73 | 10.85 |
| | | OPTA | 31.02 | 30.20 | 29.34 | 24.39 | 20.00 |
| 10 | 0 | ACOSQ | 19.45 | 17.47 | 16.32 | 12.93 | 11.53 |
| | | NA-COSQ | 19.52 | 17.59 | 16.17 | 12.08 | 10.08 |
| | | OPTA | 24.03 | 23.89 | 23.72 | 22.61 | 21.47 |
| | 0.5 | ACOSQ | 19.48 | 17.51 | 16.25 | 12.86 | 11.30 |
| | | NA-COSQ | 18.70 | 16.17 | 15.44 | 12.21 | 10.33 |
| | | OPTA | 25.28 | 25.14 | 24.97 | 23.86 | 22.72 |
| | 0.9 | ACOSQ | 20.07 | 19.49 | 18.65 | 16.74 | 14.94 |
| | | NA-COSQ | 20.22 | 19.14 | 18.24 | 15.29 | 13.74 |
| | | OPTA | 31.25 | 31.10 | 30.93 | 29.82 | 28.69 |

over the feedback links plus the decoders' side information is 5.1 dB which occurs for $\epsilon = 0.1$. Likewise, for orthogonal one-way channels with highly correlated noise in both directions of transmission (i.e., $\delta = 10$) the largest gain one can achieve by exploiting sources' statistical correlation along with feedback information is 4.86 dB which occurs for $\epsilon = 0.1$.

## A.3 Computation of OPTA values

As noted in Section. 3.2.2, we obtain the OPTA values using the complete JSCC theorem in [26, Theorem 3] for transmission of two correlated Gaussian sources over a memoryless BA-TWC with additive noise. Using this theorem we have:

$$R(D_1) \leq rI(X_1; Y_2|X_2) \tag{A.2}$$

$$R(D_2) \leq rI(X_2; Y_1|X_1), \tag{A.3}$$

and

$$\begin{aligned} I(X_1; Y_2|X_2) &= H(Y_2|X_2) - H(Y_2|X_1, X_2) && \text{(A.4)} \\ &\leq H(Y_2) - H(Y_2|X_1, X_2) && \text{(A.5)} \\ &\leq 1 - H(Z_2), && \text{(A.6)} \end{aligned}$$

where (A.4) is due to the chain rule for mutual information, (A.5) and (A.6) hold since $H(Y_2|X_2) \leq H(Y_2) \leq 1$. Note that equality in (A.5) and (A.6) is achieved under a uniform i.i.d. input $\{X_{1i}\}$ at Terminal 1. A similar result holds for $I(X_2; Y_1|X_2)$. Also,

$$H(Z_2) = (1 - \epsilon_2)h_b(\epsilon_2) + \epsilon_2 h_b(1 - \epsilon_2),$$

where $h_b(x) = -x \log_2 x - (1 - x) \log_2(1 - x)$ is the binary entropy function. Moreover, in case of a BA-TWC with additive Markov noise, the right-hand side mutual information quantities in (A.2) and (A.3) are replaced with the corresponding mutual information rates for each channel direction. In this case, we have that

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} I(X_1^n; Y_2^n|X_2^n) &\leq 1 - H(\mathcal{Z}_2) \\ &= 1 - H(Z_{22}|Z_{21}) && \text{(A.7)} \\ &= 1 - (1 - \epsilon_2)h_b(\frac{\epsilon_2}{1 + \delta_2}) - \epsilon_2 h_b(\frac{1 - \epsilon_2}{1 + \delta_2}) \end{aligned}$$

where (A.7) is the result of the Markov process with order $M = 1$, where equality is achieved under a uniform i.i.d. input at Terminal 1. A similar result holds for the mutual information rate from Terminal 2 to Terminal 1. Therefore using the rate-distortion function for a bi-variate Gaussian source sent over a BA-TWC with additive Markov noise, lower bounds for the system distortions $D_1$ and $D_2$ can be derived as follows

$$R(D_1) = \frac{1}{2}\log_2 \frac{1-\rho^2}{D_1} \leq r\Big\{1 - (1-\epsilon_2)h_b(\frac{\epsilon_2}{1+\delta_2}) - \epsilon_2 h_b(\frac{1-\epsilon_2}{1+\delta_2})\Big\} \tag{A.8}$$

$$R(D_2) = \frac{1}{2}\log_2 \frac{1-\rho^2}{D_2} \leq r\Big\{1 - (1-\epsilon_1)h_b(\frac{\epsilon_1}{1+\delta_1}) - \epsilon_1 h_b(\frac{1-\epsilon_1}{1+\delta_1})\Big\}, \tag{A.9}$$

hence the system's overall OPTA value is given by

$$\text{OPTA} = 10 \times \log_{10} \frac{\sigma_1^2 + \sigma_2^2}{D_1 + D_2}. \tag{A.10}$$

Note that $\delta_1 = \delta_2 = 0$ for a memoryless TWC and hence (A.8) and (A.9) also apply to a discrete memoryless BA-TWC with additive noise.

Since there is no complete JSCC theorem for the discrete memoryless BM-TWC even in the absence of additive noise, we include an upper bound on the system's OPTA using the converse result in [25, Lemma 2]. To that end, $I(X_1; Y_2|X_2)$ and $I(X_2; Y_1|X_1)$ are maximized over all possible joint probability distribution $P(X_1, X_2)$. Assume that the channel input probability $P(X_1, X_2)$, for $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$, is uniformly drawn from a standard 3-simplex, we generate a large number of joint probability distributions to cover the joint distortion region of $(D_1, D_2)$ as shown in Figure. A.2. For every probability distribution $P(X_1, X_2)$, the mutual information $I(X_1; Y_2|X_2)$ is computed using (A.4) and the lower bounds for the system distortions

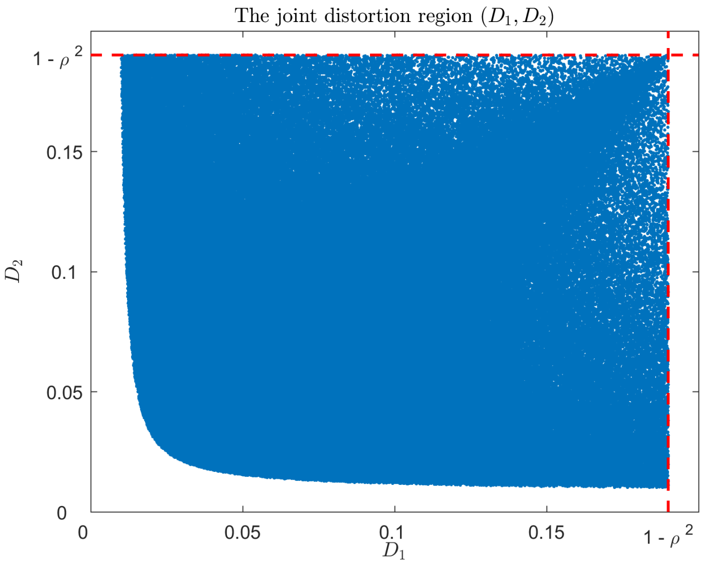


Figure A.2: The joint distortion region for a bi-variate Gaussian source with $\rho = 0.9$, a discrete memoryless BM-TWC with $\epsilon_1 = \epsilon_2 = 0.1$, and $r = 4$.

$D_1$ and $D_2$ are given by (A.2) and (A.3) using the maximum values obtained for $I(X_1; Y_2|X_2)$ and $I(X_2; Y_1|X_1)$, respectively, and the rate-distortion function for a bi-variate Gaussian source. The system's overall OPTA upper bound for transmission of two correlated Gaussian sources over a discrete memoryless BM-TWC with additive noise is then obtained using (A.10)[3].

## A.4 Two-user COSQ without adaptation using training set

As mentioned in Section 3.1, the proposed two-user COSQ considers that the joint pdf of the two sources is known a priori; however, in real-world communication systems, the source pdf is not known but rather only samples from the two sources

[3] The MATLAB scripts corresponding to exact/upper bound OPTA values are available at `https://github.com/Saeed-Rezazadeh/OPTA.git`

are available and we use the training sequences to replace the integrals with summations and the density function with empirical weights. Thus, for a training set $T = \big\{(u_{1i}, u_{2i})\big\}_{i=1}^{W}$, (3.11b) and (3.12a) are modified as

$$d_1(u_1, \mathbf{x}_1) = \sum_{\mathbf{x}_2 \in \mathcal{X}^r} \sum_{\mathbf{y}_1 \in \mathcal{Y}^r} \sum_{\mathbf{y}_2 \in \mathcal{Y}^r} P_{\mathbf{Y}_1,\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2) \sum_{u_2 \;\in\; T_{u_1} \;\cap\; \mathcal{S}_{\mathbf{x}_2}} \{(u_1 - c_{\mathbf{y}_2,u_2})^2 + (u_2 - c_{\mathbf{y}_1,u_1})^2\} \quad \text{(A.11)}$$

$$c_{\mathbf{y}_2,u_2} = \frac{\displaystyle\sum_{\mathbf{x}_1 \in \mathcal{X}^r} P_{\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_2|\mathbf{x}_1, \mathcal{E}_2(u_2)) \sum_{u_1 \in \;T_{u_2} \;\cap\; \mathcal{S}_{\mathbf{x}_1}} u_1}{\displaystyle\sum_{\mathbf{x}_1 \in \mathcal{X}^r} P_{\mathbf{Y}_2|\mathbf{X}_1,\mathbf{X}_2}(\mathbf{y}_2|\mathbf{x}_1, \mathcal{E}_2(u_2)) \times |T_{u_2} \;\cap\; \mathcal{S}_{\mathbf{x}_1}|} \quad \text{(A.12)}$$

where $T_{u_j}$ is a narrow strip standing on $u_j$ such that

$$T_{u_j} = \big\{(u_1', u_2') : u_j' \in (u_j - \frac{\alpha}{2}, u_j + \frac{\alpha}{2})\big\} \quad j = 1, 2,$$

and $\alpha$ determines the width of the strip. Similarly, (3.11d) and (3.12b) can also be modified to only use the training set. Finally, the distortion function in (3.8) is modified as

$$D = \sum_{\mathbf{x}_1 \in \mathcal{X}^r} \sum_{\mathbf{x}_2 \in \mathcal{X}^r} \sum_{\mathbf{y}_1 \in \mathcal{Y}^r} \sum_{\mathbf{y}_2 \in \mathcal{Y}^r} P(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2) \sum_{(u_1,u_2) \;\in\; \mathcal{S}_{\mathbf{x}_1} \;\cap\; \mathcal{S}_{\mathbf{x}_2}} \Big\{(u_1 - c_{\mathbf{y}_2,u_2})^2 + (u_2 - c_{\mathbf{y}_1,u_1})^2\Big\}. \quad \text{(A.13)}$$

It is noteworthy to mention that implementation of the proposed two-user COSQ for TWCs using the training sequence is feasible only when $T_{u_j} \cap \mathcal{S}_{\mathbf{x}_k} \neq \emptyset$ for $j, k = 1, 2$ and $j \neq k$. It is observed that when the two sources are highly correlated, depending on the value of $\rho$, we may have $T_{u_j} \cap \mathcal{S}_{\mathbf{x}_k} = \emptyset$. Hence, the proposed two-user COSQ with training set is only applicable to uncorrelated sources.

# Bibliography

[1] C. E. Shannon, “A mathematical theory of communication,” *The Bell Syst. Tech. J*, vol. 27, no. 3, pp. 379–423, July 1948.

[2] ——, “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, p. 1, 1959.

[3] T. S. Han, “Multicasting multiple correlated sources to multiple sinks over a noisy channel network,” *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 4–13, Jan 2011.

[4] C. Tian, J. Chen, S. N. Diggavi, and S. Shamai, “Optimality and approximate optimality of source-channel separation in networks,” *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 904–918, Feb 2014.

[5] A. El Gamal and Y.-H. Kim, *Network Information Theory.* Cambridge University press, 2011.

[6] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley & Sons, 2012.

[7] Y. Zhong, F. Alajaji, and L. L. Campbell, “On the joint source-channel coding error exponent for discrete memoryless systems,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1450–1468, April 2006.

[8] ——, “Joint source-channel coding error exponent for discrete communication systems with Markovian memory,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4457–4472, Dec 2007.

[9] ——, “Joint source-channel coding excess distortion exponent for some memoryless continuous-alphabet systems,” *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1296–1319, March 2009.

[10] R. G. Gallager, *Information Theory and Reliable Communication.* Springer, 1968, vol. 2.

[11] F. Alajaji and P.-N. Chen, *An Introduction to Single-User Information Theory.* Springer, 2018.

[12] V. Kostina and S. Verd, “Lossy joint source-channel coding in the finite blocklength regime,” *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2545–2575, May 2013.

[13] J. Lim and D. L. Neuhoff, “Joint and tandem source-channel coding with complexity and delay constraints,” *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 757–766, May 2003.

[14] V. B. Balakirsky, “Joint source-channel coding with variable length codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, June 1997, p. 419.

[15] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source - channel coding," *IEEE Trans Inf. Theory*, vol. 33, no. 6, pp. 827–838, November 1987.

[16] ——, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 155–160, Jan 1991.

[17] N. Phamdo, N. Farvardin, and T. Moriya, "A unified approach to tree-structured and multistage vector quantization for noisy channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 835–850, May 1993.

[18] R. Gray and Y. Linde, "Vector quantizers and predictive quantizers for Gauss-Markov sources," *IEEE Trans. Commun.*, vol. 30, no. 2, pp. 381–389, February 1982.

[19] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Springer Science & Business Media, 2012, vol. 159.

[20] J. Bakus and A. K. Khandani, "Quantizer design for channel codes with soft-output decoding," *IEEE Trans. Veh. Tech.*, vol. 54, no. 2, pp. 495–507, March 2005.

[21] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Math. Stat. and Prob.* The Regents of the University of California, 1961.

[22] A. Kaspi, "Two-way source coding with a fidelity criterion," *IEEE Trans. Inf. Theory*, vol. 31, no. 6, pp. 735–740, November 1985.

[23] A. Maor and N. Merhav, "Two-way successively refined joint source-channel coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1483–1494, April 2006.

[24] M. Heindlmaier, O. Ican, and C. Rosanka, “Scalar quantize-and-forward for symmetric half-duplex two-way relay channels,” in *2013 IEEE Inter. Symp. Inf. Theory*, July 2013, pp. 1322–1326.

[25] J.-J. Weng, F. Alajaji, and T. Linder, “Lossy transmission of correlated sources over two-way channels,” in *2017 IEEE Inf. Theory Workshop (ITW)*, Nov 2017, pp. 354–358.

[26] ——, “Joint source-channel coding for the transmission of correlated sources over two-way channels,” *2019 IEEE Inter. Symp. Inf. Theory*, 2019.

[27] P. Minero, S. H. Lim, and Y. Kim, “A unified approach to hybrid coding,” *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1509–1523, April 2015.

[28] O. Y. Bursalioglu, G. Caire, M. Fresia, and H. V. Poor, “Joint source-channel coding at the application layer for parallel Gaussian sources,” in *2009 IEEE Inter. Symp. Inf. Theory*, June 2009, pp. 2126–2130.

[29] S. P. Beheshti, F. Alajaji, and T. Linder, “Optimal joint decoding of correlated data over orthogonal multiple-access channels with memory,” *IEEE Trans. Veh. Tech.*, vol. 66, no. 1, pp. 79–94, Jan 2017.

[30] S. Shahidi, F. Alajaji, and T. Linder, “MAP detection and robust lossy coding over soft-decision correlated fading channels,” *IEEE Trans. Veh. Tech.*, vol. 62, no. 7, pp. 3175–3187, Sep. 2013.

[31] J. Weng, F. Alajaji, and T. Linder, “Optimized signaling of binary correlated sources over Gaussian multiple access channels,” in *2018 IEEE 88th Veh. Tech. Conf. (VTC-Fall)*, Aug 2018, pp. 1–5.

[32] T. P. Mitchell, F. Alajaji, and T. Linder, “Binary signaling of correlated sources over orthogonal multiple-access channels,” *IEEE Wireless Commun. Lett.*, vol. 4, no. 5, pp. 501–504, Oct 2015.

[33] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, Fourthquarter 2014.

[34] A. S. Amanullah and M. Salehi, “Joint source-channel coding in the presence of feedback,” in *Proc of 27th Asilomar Conf. Signals, Syst. Comput.*, Nov 1993, pp. 930–934 vol.2.

[35] M. Mushkin and I. Bar-David, “Capacity and coding for the gilbert-elliott channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 6, pp. 1277–1290, Nov 1989.

[36] F. Alajaji and T. Fuja, “A communication channel modeled on contagion,” *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 2035–2041, Nov 1994.

[37] S. Lloyd, “Least squares quantization in pcm,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, March 1982.

[38] J. Kieffer, “Uniqueness of locally optimal quantizer for log-concave density and convex error weighting function,” *IEEE Trans. Inf. Theory*, vol. 29, no. 1, pp. 42–47, January 1983.

[39] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, January 1980.

[40] N. Farvardin, “A study of vector quantization for noisy channels,” *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 799–809, July 1990.

[41] B. Hajek, “A tutorial survey of theory and applications of simulated annealing,” in *1985 24th IEEE Conf. on Decision and Control*, Dec 1985, pp. 755–760.

[42] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, “Convergence and finite-time behavior of simulated annealing,” *Advances in applied probability*, vol. 18, no. 3, pp. 747–771, 1986.

[43] S. Rezazadeh, F. Alajaji, and W.-Y. G. Chan, “Scalar quantizer design for Two-Way channels,” in *2019 16th Canadian Workshop Inf. Theory (CWIT 2019)*, Hamilton, Canada, Jun. 2019.

[44] H. S. Wang, *Finite-state modeling, capacity, and joint source/channel coding for time-varying channels.* Rutgers, The State University of New Jersey, 1992.

[45] A. C. Hung and H. . Meng, “Adaptive channel optimization of vector quantized data,” in *Proc DCC ‘93: Data Comp. Conf.*, March 1993, pp. 282–291.

[46] C. Shannon, “The zero error capacity of a noisy channel,” *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.

[47] F. Alajaji, “Feedback does not increase the capacity of discrete channels with additive noise,” *IEEE Trans. Inf. Theory*, vol. 41, no. 2, pp. 546–549, March 1995.