

Channel Optimized Scalar Quantization over
Orthogonal Multiple Access Channels with Memory

by

Kiraseya Preusser

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

Queen's University
Kingston, Ontario, Canada

September 2017

Copyright ©Kiraseya Preusser 2017

Abstract

In this thesis, the joint source-channel coding method, channel optimized scalar quantization, is applied to real-valued, correlated data. The data is sent over the orthogonal multiple access channel, with non-binary noisy discrete channels with memory as the two sub-channels. Three different schemes are compared for this system: in the first scheme encoding and decoding are performed independently, in the second scheme encoding is done independently and joint decoding is carried out, and the third scheme is with jointly optimized encoders and joint decoding. The goal is to derive optimality conditions that will result in a lower end-to-end distortion. To this end, necessary optimality conditions for the two latter schemes are fully derived and implemented for the bivariate Gaussian and bivariate Laplacian distributions of varying correlation.

The first and second methods are then further compared, by implementing them for an image transmission system. Here the images are first processed with the 2 dimensional discrete cosine transform, and then encoded using channel optimized scalar quantization. At the decoder, two different methods are used, the independent and joint decoder.

In addition to comparing the different coding methods, various channels characteristics are exploited. For example, the non-binary noisy discrete channel can be used to model memory and the 2^q -ary output allows for performance improvement via soft-decision decoding. It is observed that by taking the source correlation into consideration, significant signal-to-distortion ratio gains can be achieved. For example, the highest gain incurred from the third scheme is when the bivariate Gaussian is compressed at rate 2, where the gain in signal-to-distortion ratio due to source correlation is 10.90 dB.

Acknowledgments

I would like to thank Professor Fady Alajaji and Professor Tamás Linder for all of their dedicated support and assistance throughout the past two years. I would also like to thank my sister, Calista, and my parents, for their love and wisdom. Thanks also to Emily for keeping me focused and motivated in the last months of my master's, and Taylor for the laughter and steak nights!

Running as many simulations as I did would have been impossible without the assistance from Jeff Stafford and the Centre of Advanced Computing. I'm grateful for them letting me use their servers, and Jeff for teaching me how to use them properly.

Thanks to Torsten from arctic.io for giving me permission to use the images from his website, and NASA for making those images available on the public domain.

Finally, I would like to thank my colleague Chasel for brainstorming solutions with me when things were not working the way I expected.

Contents

Chapter 1	Introduction	1
1.1	Motivation	1
1.2	Literature Review	2
1.3	Contributions	4
1.4	Thesis Overview	4
Chapter 2	Preliminaries	5
2.1	Channel Models	5
2.1.1	Discrete Memoryless Channels	5
2.1.2	Channels with Memory	7
2.1.3	Multiple Access Channel	11
2.2	Source and Channel Coding, and Quantization	12
2.2.1	Source Coding and Channel Coding	12
2.2.2	Scalar Quantization	13
2.3	Channel Optimized Quantization	16
2.3.1	Introduction	16
2.3.2	System Description	17
2.3.3	COSQ Design	17
2.3.4	Initial Codebook Design and Simulated Annealing	18
2.3.5	Results	21
Chapter 3	COSQ of Correlated Sources over Orthogonal MACs	26
3.1	Derivation of Optimality Conditions	26
3.1.1	COSQ II: Individually Optimized Encoders, Jointly Optimized Decoder	27

3.1.2	COSQ III: Jointly Optimized Encoders, Joint Optimized Decoder . . .	30
3.1.3	Encoding Complexity and Storage Requirements	37
3.2	Results	41
3.2.1	COSQ II	43
3.2.2	COSQ III	57
Chapter 4	Application to Correlated Images sent over Orthogonal MACs	62
4.1	Image Transmission System	62
4.2	Fixed Bit Allocation	65
4.3	Correlated Images	66
4.4	Results	69
Chapter 5	Conclusion and Future Work	84

List of Figures

1.1	Tandem Coding System [24]	2
2.1	The Binary Symmetric Channel [24]	6
2.2	Queue of Length M	8
2.3	Urn of Numbered Balls with 2^q Unique Numbers	8
2.4	The Orthogonal MAC	11
2.5	Capacity Region for Independent Binary Symmetric Channels [24]	12
2.6	Block Diagram of COSQ System	17
3.1	Block Diagram of Two User COSQ System	31
3.2	Two User Rate 1 COSQ II with Varying CSNR, Gaussian	45
3.3	Two User Rate 2 COSQ II with Varying CSNR, Gaussian	46
3.4	Two User Rate 3 COSQ II with Varying CSNR, Gaussian	47
3.5	Two User Rate 4 COSQ II with Varying CSNR, Gaussian	48
3.6	Two User Rate 1 COSQ II with Varying CSNR, Laplacian	49
3.7	Two User Rate 2 COSQ II with Varying CSNR, Laplacian	50
3.8	Two User Rate 3 COSQ II with Varying CSNR, Laplacian	51
3.9	Two User Rate 4 COSQ II with Varying CSNR, Laplacian	52
3.10	Two User Varying Rates COSQ II with Noiseless and Very Noisy Channels, Gaussian	55
3.11	Two User Rate 3 COSQ II with Varying CSNR, Channel Correlation $\text{cor} = 0$ and 0.9 , Gaussian	56
3.12	Two User Rate 1 COSQ III with Varying CSNR, Gaussian	58

3.13	Two User Rate 2 COSQ III with Varying CSNR, Gaussian	59
3.14	Two User Rate 1 COSQ III with Varying CSNR, $q = 2$, Gaussian	60
3.15	Two User Rate 1 COSQ III with Varying CSNR, Channel Correlation $\text{cor} =$ 0.9, Gaussian	61
4.1	Zig Zag Pattern [23]	64
4.2	Image Transmission System	64
4.3	Histogram of Correlated DC Coefficients	67
4.4	Histogram of First AC Coefficients	67
4.5	Histogram of Second AC Coefficients	68
4.6	Original, Correlated images	72
4.7	Images Compressed with 0.125 bpp for a Channel with CSNR = 2 dB	72
4.8	COSQ I Decoded Images, CSNR = 2 dB, 0.125 bpp	73
4.9	COSQ II Decoded Images, CSNR = 2 dB, 0.125 bpp	73
4.10	COSQ I Decoded Images, CSNR = 0 dB, 0.109 bpp	74
4.11	COSQ II Decoded Images, CSNR = 0 dB, 0.109 bpp	74
4.12	COSQ I Decoded Images, CSNR = 0 dB, 0.125 bpp, Channel $\text{cor} = 0.9$	75
4.13	COSQ II Decoded Images, CSNR = 0 dB, 0.125 bpp, Channel $\text{cor} = 0.9$	75
4.14	COSQ I Decoded Images, CSNR = 0 dB, 0.109 bpp, Channel $\text{cor} = 0.9$	76
4.15	COSQ II Decoded Images, CSNR = 0 dB, 0.109 bpp, Channel $\text{cor} = 0.9$	76
4.16	Original, Correlated Images	77
4.17	Images Compressed with 0.125 bpp for a Channel with CSNR = 15 dB	77
4.18	COSQ I Decoded Images, CSNR = 15 dB, 0.125 bpp	78
4.19	COSQ II Decoded Images, CSNR = 15 dB, 0.125 bpp	78
4.20	COSQ I Decoded Images, CSNR = 0 dB, 0.125 bpp	79
4.21	COSQ II Decoded Images, CSNR = 0 dB, 0.125 bpp	79
4.22	COSQ I Decoded Images, CSNR = 15 dB, 0.109 bpp	80
4.23	COSQ II Decoded Images, CSNR = 15 dB, 0.109 bpp	80

4.24 COSQ I Decoded Images, CSNR = 5 dB, 0.109 bpp	81
4.25 COSQ II Decoded Images, CSNR = 5 dB, 0.109 bpp	81

List of Tables

2.1	SDR Results for COSQ Training with Gaussian and Laplacian Data, Channel Correlation $\text{cor} = 0, \alpha = 1, M = 1$	24
2.2	SDR Results for COSQ Training, for Gaussian Data, $\alpha = 1$	25
2.3	The ρ Values for Corresponding CSNR for NBNDC-QB with Soft-Decision Resolution $q = 1, 2, 3$ [22]	25
3.1	Summary of Complexity and Storage Requirements for COSQ Schemes . . .	41
3.2	Gain due to Source Correlation, ρ , in dB for COSQ II, Channel Parameters: $M = 1, \text{cor} = 0, \alpha = 0$	43
4.1	Fixed Bit Allocation Tables for DCT	66
4.2	SDR Results for DC Coefficients	69
4.3	SDR Results for AC Coefficients, Rate 2	70
4.4	PSNR Results Consecutive May Images	82
4.5	PSNR Results Consecutive May Images Sent Over Channel with $\text{cor} = 0.9$.	82
4.6	PSNR Results for May Images sent with Noisy Version	83

List of Acronyms

BER	Bit Error Rate
BSC	Binary Symmetric Channel
CC	Centroid Condition
CSNR	Channel Signal-to-Noise Ratio
COSQ	Channel Optimized Scalar Quantization
DCT	Discrete Cosine Transform
DMC	Discrete Memoryless Channel
IID	Identically and Independently Distributed
JSCC	Joint Source-Channel Coding
MAC	Multiple Access Channel
MSE	Mean Squared Error
NBNDC	Non Binary Noisy Discrete Channel
NBNDC-QB	Non Binary Noisy Discrete Channel with Queue Based Noise
NNC	Nearest Neighbour Condition
PSNR	Peak Signal-to-Noise Ratio
SA	Simulated Annealing
SDR	Signal-to-Distortion Ratio
SNR	Signal-to-Noise Ratio
SQ	Scalar Quantization
WLOG	Without loss of generality

Chapter 1

Introduction

1.1 Motivation

The transmission of data signals from one location to another over noisy media has become commonplace in everyday life, and it is important that the transmission of data be efficient and reliable. The data being sent can take on various forms, such as text, sound (music or speech), images, or video. The source data is generally modelled as a random process, either discrete or continuous valued. This data is either stored, or transmitted over a wireless channel, or a cable. In lossy systems continuous data is discretized and represented as *bits* in order to store or transmit it. The process in which data is discretized and prepared for transmission is called *encoding* and the process which reconstructs the data is called *decoding*. There are multiple coding models and encoding functions designed for various conditions. One such model used to transmit data is the *tandem coding system*. In this system, shown in Figure 2.1, the *source coding* and *channel coding* are treated separately. In the source coding step the data is compressed and redundancy in the source is removed, in order to represent the data as succinctly as possible. This process makes the information more vulnerable to errors in the channel so the second step is channel coding, where the compressed data is protected from the channel noise by adding controlled redundancy. After this two-staged encoding process, a finite, discrete signal, that represents the data, is sent over the channel. On the other end, the process is reversed with first the channel decoder and then the source

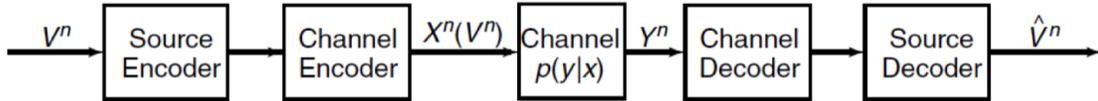


Figure 1.1: Tandem Coding System [24]

decoder. The source and channel coder should be designed in such a way that the resulting message is as close to the input data as possible, given the storage and transmission power restrictions. The source coder is designed assuming that the channel coder is optimal, and vice versa. The motivation behind the use of a tandem system is Shannon’s source-channel separation theorem. Shannon’s separation principles state that with a noisy channel of capacity C it is possible to obtain a reconstruction signal with distortion D provided that the capacity is greater than $R(D)$, where $R(D)$ is the rate-distortion function [1]. In practice, this system is not always ideal as the use of two-stage coding introduces delay and complexity, and the block lengths are not always sufficiently long [2]. Most importantly, this theorem cannot be generalized to multi-user systems. Because of this, *joint source-channel coding* (JSCC) schemes have been developed. These schemes incorporate elements of both source and channel coding into a single encoder/decoder pair.

The problem addressed in this thesis is designing a JSCC for two users sending correlated information to a common receiver. The two sources can either be spatially or temporally distinct. There are existing sensor systems where sensors from two locations send correlated data to the same receiver, or the same sensor sends information periodically. The goal with the JSCC in this thesis, is to use the correlation in the data to improve the end-to-end distortion.

1.2 Literature Review

Multiple studies into JSCC have been made and have shown it to be a viable scheme. There are a variety of different JSCC schemes and generally they can be categorized as implementing

either unequal error protection, zero-redundancy channel coding, or combined source-channel coding [3].

Unequal error protection sorts the bits by importance and prioritizes the bits that hold more information. These bits are given more protection and the less important bits are given less protection against channel noise. The bits of the least importance are sometimes given no protection at all. This method is used in [4] to protect the bits which contribute the most to the image reconstruction. With most image transform coding techniques the coefficients being sent across the channel hold varying amounts of energy, thus this method is commonly employed to obtain the highest quality reconstruction.

Zero-redundancy channel coding uses the redundancy from the source to protect against the channel noise. This method adds no redundancy for the channel, thus eliminating the use of channel coding. The complexity at the decoder increases; however the encoding is simpler.

The two previously mentioned methods do not take the channel characteristics into consideration in their design. The combination of source and channel coding, sometimes called channel-optimized or channel-matched coding, is a more common JSCC scheme, which combines elements of both source and channel code into a single code. Lloyd and Max [5], [6] worked on quantization techniques and Kurtenbach and Wintz [7] continued their work by deriving the necessary conditions for an optimal scalar quantizer for noisy channels. These conditions have since been applied to various channel models, different types of data, and Farvardin *et al.* studied the complexity and design issues related to the scheme [8], [9].

It is known that JSCC reduces the delay in the system as [10] showed that the error exponent for JSCC is twice that of separate source-channel coding. This implies that only half the encoding and decoding delay is present when using an optimal JSCC method. This in turn saves 2 dB in power to achieve the same overall probability of error. Other advantages of the JSCC scheme are the performance advantages in nonasymptotic regions over the separate code design [11]. In [12], Shahidi *et al.* numerically shows that soft-decision decoding results in gains in *signal-to-distortion ratio* (SDR) over hard-decision decoding, which supports the

work in [13], [14], [15], [16] asserting that soft-decision decoding can increase channel capacity and system performance.

Finally, JSCC has been implemented in the multi-user case in [17]. Here the orthogonal *multiple access channel* (MAC) is used to model multiple users sending information. In [18] two correlated Gaussian memoryless sources are sent across the Gaussian MAC. In [19] the achievable regions for jointly correlated source side information are considered and source and channel codes are derived. Finally [20] shows that correlated sources can be sent with arbitrarily small probability of error over a multiple access channel.

1.3 Contributions

In this thesis, the results from [21], [22], [23] are combined, in that the JSCC conditions for sending correlated data over the orthogonal MAC with non-binary noisy discrete sub-channels are derived and implemented. The performance of a jointly optimized decoder is also investigated for sending correlated images over the same orthogonal MAC.

1.4 Thesis Overview

The second chapter introduces the channel models used, the existing JSCC conditions and the resulting SDR for a single user scheme. Chapter 3 includes the derivation of two two-user JSCC methods. The first has independently optimized encoders and a jointly optimized decoder, while the second has jointly optimized encoders and a jointly optimized decoder. The schemes are compared in terms of encoding complexity and storage requirements and implemented in MATLAB to show their performance. Next, the first two-user JSCC scheme is used in the transmission of correlated images and the results are shown in Chapter 4. Finally, Chapter 5 is devoted to conclusions and further work.

Chapter 2

Preliminaries

2.1 Channel Models

2.1.1 Discrete Memoryless Channels

A communication channel with finite input alphabet \mathcal{X} and finite output alphabet \mathcal{Y} is called a discrete channel. The set of transition probabilities from the channel input to the channel output can be summarized by:

$$p(y|x) = P(Y = y|X = x), \quad \forall x \in \mathcal{X}, y \in \mathcal{Y},$$

which is the probability of observing the output signal $Y = y$ given the input signal $X = x$. In practice, the input takes the form of a sequence of n input variables $\mathbf{X} = (X_1, X_2, \dots, X_n)$ and the output is a corresponding sequence $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$. The channel is now governed by the transition probability:

$$p(\mathbf{y}|\mathbf{x}) = P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}), \quad \mathbf{x} \in \mathcal{X}^n, \mathbf{y} \in \mathcal{Y}^n.$$

Here \mathcal{X}^n and \mathcal{Y}^n are the n -fold cartesian products of \mathcal{X} and \mathcal{Y} , respectively. A discrete time channel is classified as *memoryless* if for a given input at time i , the output y_i is independent of all other channel inputs and outputs at times other than i . Specifically,

$p(y_i|x_1, \dots, x_i, y_1, \dots, y_{i-1}) = p(y_i|x_i)$. The conditional probability can be written as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n p(y_i|x_i)$$

when the channel is used *without feedback*. The *informational channel capacity* of a discrete memoryless channel is given by

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

where the maximum of the *mutual information*, $I(X;Y)$, is taken over all possible input distributions $p(x)$ [24].

Binary Symmetric Channel (BSC)

One example of a discrete memoryless channel is the *binary symmetric channel* (BSC), shown in Figure 2.1. This is a binary input, binary output channel with crossover probability p . It is a simple channel where when an error occurs, 0 is received as a 1 and vice versa. The

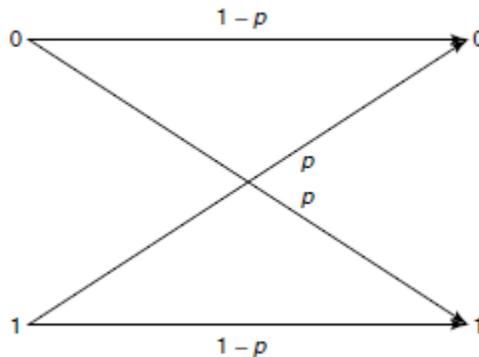


Figure 2.1: The Binary Symmetric Channel [24]

transition matrix, for which $p(j|i) = \mathbf{Q}_{i,j}$, of the BSC is given by

$$\mathbf{Q} = \begin{bmatrix} 1 - \epsilon & \epsilon \\ \epsilon & 1 - \epsilon \end{bmatrix}.$$

It can be shown that the mutual information is bounded by:

$$I(X; Y) \leq 1 - H_b(p)$$

where $H_b(\cdot)$ is the binary entropy function [24]. It can be shown the capacity of the BSC with crossover probability p is $1 - H_b(p)$. This channel is widely used for analysis, as the complexity of many problems can be reduced by using the BSC.

2.1.2 Channels with Memory

In practice, channels display characteristics of memory. In fact errors typically occur in bursts [25]. To capture this behaviour, the non-binary noisy discrete channel is introduced.

Non-Binary Noisy Discrete Channel

The *non-binary noisy discrete channel* (NBND) is a binary input, 2^q -ary output communication channel that incorporates the statistical memory and soft-decision information of a time-correlated discrete fading channel. The channel output at time j is given by

$$Y_j = (2^q - 1)X_j + (-1)^{X_j}Z_j \tag{2.1}$$

where Y_j is the channel output, Z_j is the channel noise, and X_j is the channel input. $Y_j, Z_j \in \{0, 1, \dots, 2^q - 1\}$ for $q \geq 1$, and the input $X_j \in \{0, 1\}$ [26]. For $j \geq 0$, the channel noise can be written in terms of the input and output symbols,

$$Z_j = \frac{Y_j - (2^q - 1)X_j}{(-1)^{X_j}}. \tag{2.2}$$

The advantage of using this channel is that the channel memory can easily be varied without adding complexity to the system. The non-binary output from the NBND makes it feasible to improve the system's performance by implementing soft-decision decoding. The complexity

increases exponentially with q , the soft-decision quantization resolution; however q is typically not larger than 3.

The distribution of the noise process, $\{Z_i\}_{i=1}^{\infty}$, can be any stochastic process. The distribution can, for example be a binary stationary memoryless process, with $q = 1$, and then the NBNDP reduces to the BSC. The noise distribution investigated in this thesis is that from [26], a non-binary generalization of the queue-based noise [27]. This channel is referred to as the queue-based NBNDP, or NBNDP-QB. In this model, the noise is a 2^q -ary M^{th} -order Markovian stationary ergodic process with $2^q + 2$ independent parameters. At time j , the noise symbol z_j is chosen from one of the following two packages:

- A queue with M cells, Figure 2.2, with balls each representing a noise symbol. Each cell contains one ball with values from 0 to $2^q - 1$.
- An urn with many numbered balls, Figure 2.3, also representing noise symbols.



Figure 2.2: Queue of Length M

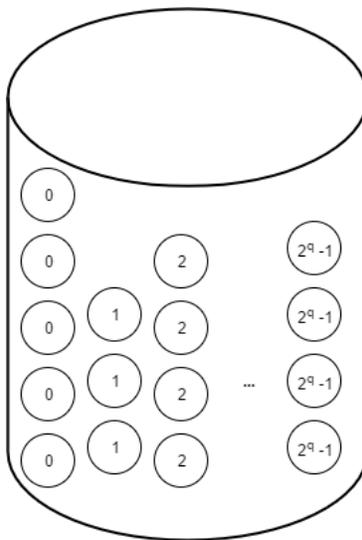


Figure 2.3: Urn of Numbered Balls with 2^q Unique Numbers

At each time iteration, the queue is selected with probability ϵ . When the queue is chosen, one of the M cells is chosen and the corresponding number of that cell becomes the noise symbol. The probability of selecting the k^{th} cell in the queue is given by

$$\begin{cases} \frac{1}{M-1+\alpha}, & \text{if } k = 1, 2, \dots, M-1; \\ \frac{\alpha}{M-1+\alpha}, & \text{if } k = M \end{cases} \quad (2.3)$$

with $\alpha \geq 0$. When the urn is selected, with probability $1 - \epsilon$, a ball is chosen from the urn and the noise symbol takes on the value of the selected ball. The balls in the urn take on the probability distribution $(\rho_0, \rho_1, \dots, \rho_{2^q-1})$ where each ρ_i is the probability that a ball of value i is chosen from the urn. Regardless of how the noise symbol was evaluated, a ball with that value now gets placed at the beginning of the queue, pushing the M^{th} ball out.

This resulting queue-based system is a stationary M^{th} order Markovian system with $2^q + 2$ independent parameters, the size of the queue M , the probability distribution of the balls in the urn, the correlation parameter, $0 \leq \epsilon < 1$ and $\alpha \geq 0$.

The queue-based noise $\{\mathbf{S}_n\}_{-\infty}^{\infty}$, defined by $\mathbf{S}_n = (Z_n, Z_{n-1}, \dots, Z_{n-M+1}) \in \{0, 1, \dots, 2^q-1\}^M$, is a homogeneous, first-order Markov state process. Let the noise state transition probability be defined as

$$Q(\mathbf{s}_n | \mathbf{s}_{n-1}) = P(\mathbf{S}_n = \mathbf{s}_n | \mathbf{S}_{n-1} = \mathbf{s}_{n-1}),$$

where $\mathbf{s}_n = (z_n, z_{n-1}, \dots, z_{n-M+1})$, $z_n \in \{0, 1, \dots, 2^q - 1\}$. [26] states that,

$$Q(\mathbf{s}_n | \mathbf{s}_{n-1}) = \left(\sum_{l=1}^{M-1} \delta_{z_n, z_{n-l}} + \alpha \delta_{z_n, z_{n-M}} \right) \frac{\epsilon}{M-1+\alpha} + (1-\epsilon)\rho_{z_n}, \quad (2.4)$$

where

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases}$$

Since the noise process is independent of the input, the transition probability can be written as

$$P(\mathbf{Y}^m = \mathbf{y}^m | \mathbf{X}^m = \mathbf{x}^m) = P(\mathbf{Z}^m = \mathbf{z}^m) \quad (2.5)$$

where $\mathbf{Y}^m = Y_1, \dots, Y_m$, $\mathbf{X}^m = X_1, \dots, X_m$, and $\mathbf{Z}^m = z_1, \dots, z_m$. Combining the previous distributions, the whole system can be summarized with the following m -fold channel transition probability $P(\mathbf{Z}^m = \mathbf{z}^m) = P_{(\text{NBND C-QB})}^{(m)}(\mathbf{z}^m) = P_{(\text{NBND C-QB})}(Z_1 = z_1, Z_2 = z_2, \dots, Z_m = z_m)$ [26].

- For $m > M$

$$P_{(\text{NBND C-QB})}^{(m)}(\mathbf{z}^m) = \prod_{i=M+1}^m \left[\left(\sum_{l=i-M+1}^{i-1} \delta_{z_i, z_l} + \alpha \delta_{z_i, z_{i-M}} \right) \frac{\epsilon}{M-1+\alpha} + (1-\epsilon)\rho_{z_i} \right] \pi_{(z_1, z_2, \dots, z_M)}, \quad (2.6)$$

where

$$\pi_{((z_1, z_2, \dots, z_M))} = \frac{\prod_{l=0}^{2^q-1} \prod_{j=0}^{\xi_l-1} \left((1-\epsilon)\rho_l + j \frac{\epsilon}{M-1+\alpha} \right)}{\prod_{k=0}^{M-1} \left((1-\epsilon) + k \frac{\epsilon}{M-1+\alpha} \right)}, \quad (2.7)$$

where $\xi_l = \sum_{k=1}^M \delta_{z_k, l}$

- For $m \leq M$

$$P_{(\text{NBND C-QB})}^{(m)}(\mathbf{z}^m) = \frac{\prod_{l=0}^{2^q-1} \prod_{j=0}^{\xi'_l-1} \left((1-\epsilon)\rho_l + j \frac{\epsilon}{M-1+\alpha} \right)}{\prod_{k=0}^{m-1} \left((1-\epsilon) + k \frac{\epsilon}{M-1+\alpha} \right)}, \quad (2.8)$$

where $\xi'_l = \sum_{k=1}^m \delta_{z_k, l}$.

When $m = 1$, the marginal noise distribution reduces down to $P_{(\text{NBND C-QB})}^{(1)}(z^1) = \rho_{z^1}$, where $z^1 \in \{0, 1, \dots, 2^q - 1\}$. The correlation coefficient of the system is given by:

$$\begin{aligned} \text{cor} &= \frac{E[Z_k Z_{k+1}] - E[Z_k]^2}{\text{Var}(Z_k)} \\ &= \frac{\frac{\epsilon}{M-1+\alpha}}{1 - (M-2+\alpha) \frac{\epsilon}{M-1+\alpha}} \end{aligned} \quad (2.9)$$

This channel model will be used extensively when designing the channel optimized scalar

quantization algorithm.

2.1.3 Multiple Access Channel

The multiple access channel is a multiple terminal communication channel over which two or more sources can transmit information to a single receiver. Generally the source symbols from the multiple users will interfere with one another, so the use of channel coding is necessary to protect the data. To avoid this interference, the orthogonal MAC is used in practice. Here the sources do not interfere; however the channel bandwidth must be shared among

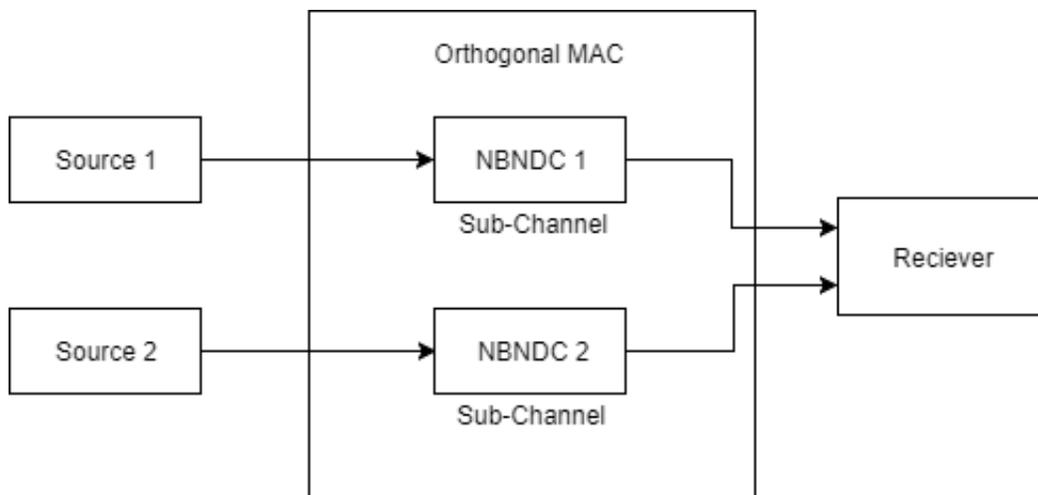


Figure 2.4: The Orthogonal MAC

several users. This can be achieved by implementing frequency division multiple access, time division multiple access or code division multiple access techniques. The orthogonal MAC is a suitable model for when multiple correlated sources are independently transmitted without interference, to a common receiver. This model will be used with two NBND channels acting as the orthogonal sub-channels, as shown in Figure 2.4. Figure 2.5 shows the capacity region for two orthogonal binary symmetric channels. Since there is no interference between the channels the capacity is simply the region between the capacities from the separate channels. Recall the capacity of a single BSC is $C = 1 - H_b(p)$.

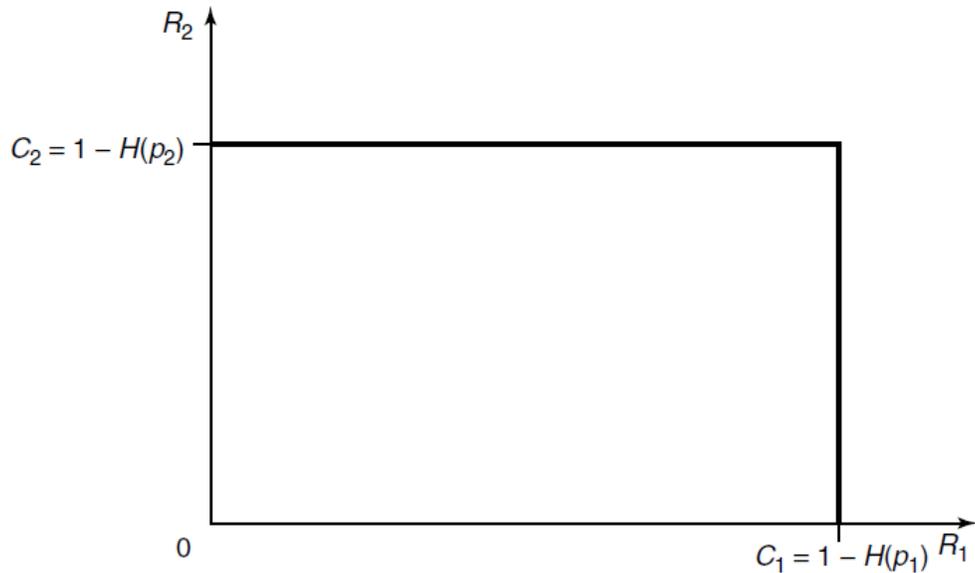


Figure 2.5: Capacity Region for Independent Binary Symmetric Channels [24]

2.2 Source and Channel Coding, and Quantization

2.2.1 Source Coding and Channel Coding

Source coding, also referred to as data compression, is the process of removing redundant information from the source with the purpose of reducing the requirement of bandwidth. Two types of data compression exist, *lossless* and *lossy*. In lossless data compression the reconstructed data is identical to the original source, i.e., no information is lost. The goal of lossy data compression is to reduce the amount of information stored or transmitted but keep the reconstructed data similar to the original. Lossy data compression is achieved by quantization, which will be discussed in a later section.

Channel coding is the process of mitigating the effect of channel noise on the information being transferred. This is done by adding controlled redundancies into the binary stream. The bandwidth must be allocated for error protection but the amount of bandwidth allocated is delicate; too many source bits in the binary stream will result in channel errors destroying the signal while too few makes the reconstruction quality poor.

The goal of source coding is to represent the source information as succinctly as possible. The redundancy of a source can be due to non-uniformity of its marginal probability distribution or its memory. The redundancy in information can be described using *entropy*. Entropy is an integral component of information theory. It is defined as the average number of bits required to store or communicate one signal in a message (for example a pixel or ASCII character). Entropy is defined as follows

$$H(X) = - \sum_{x \in \mathcal{X}} p_x(x) \log_2 p_x(x) = -E_X[\log_2 p_x(x)],$$

where $\mathcal{X} = \{0, 1, \dots, N - 1\}$ is the alphabet, of the source X and $p_x(x) = P(X = x)$ is the probability mass function associated.

2.2.2 Scalar Quantization

Quantization is used to compress data by mapping data points to a discrete set of points called *codewords*. These codewords make up the *codebook*. To generate these codewords, a set of training data is required. Training data can be data points from a probability distribution or, in the two user case, data generated from a bivariate distribution. To perform quantization the values to be quantized are mapped to a point in the codebook. This process is a form of lossy compression since the data points are no longer stored as their true value. Scalar quantization is used on single-valued data sets. The method assigns each element of a data set to a unique codeword. Specifically, for a rate n scalar quantizer, q , there are $N = 2^n$ codewords. The scalar quantizer maps the real-line to a set of finite values,

$$q : \mathbb{R} \rightarrow C$$

where

$$C = \{c_1, c_2, \dots, c_N\} \subset \mathbb{R},$$

where c_i 's are the codewords, or reproduction levels. In order to map the real-line to these codewords, the real-line is partitioned into N non-overlapping regions, $S_i, i = 1, 2, \dots, N$. These are defined as

$$S_i = \{x \in \mathbb{R} : q(x) = c_i\}, i = 1, 2, \dots, N$$

where the S_i 's satisfy

$$\cup_{i=1}^N S_i = \mathbb{R}$$

and

$$S_i \cap S_j = \emptyset,$$

for $i \neq j$ where $i, j = 1, 2, \dots, N$. For each input x to the quantizer q , the output is obtained as follows,

$$q(x) = c_i \text{ if } x \in S_i.$$

The quantizer is fully defined by the partition set $\{S_i\}$ and the codebook C . To optimize the quantizer, in other words, reduce the distortion of the quantization for a given number of codewords, there are two conditions that must be met. The goal is to minimize the statistical distortion, measured as the expected distortion,

$$D = E[d(X, q(X))] = \sum_i \int_{S_i} (x - c_i)^2 p(x) dx,$$

where $p(x)$ is the source probability density function and $d(x, y) = |x - y|^2$ is the squared error distortion measure, thus D is called the mean-squared error (MSE) distortion. The first condition is the *nearest neighbour condition*. This condition requires for a given codebook C that the data points be mapped to the codeword closest to them. The condition is given as follows,

$$S_i = \{x : |x - c_i|^2 < |x - c_j|^2, \forall j \neq i\} \quad \forall i = 1, \dots, N.$$

The *centroid condition* is the second condition satisfied by an optimal quantizer. Given a partition set $\{S_i\}$, the optimal codewords are given by

$$c_i = \arg \min_c E[d(X, c)|X \in S_i], \quad \forall i = 1, 2, \dots, N.$$

Owing to the two above necessary optimality conditions, the optimization of the quantizer can be done by an iterative algorithm which simplifies the design of the optimal quantizer. This algorithm is called the *Lloyd algorithm*. It is an iterative algorithm which starts with an initial, fixed, codebook. It proceeds to use the nearest neighbour condition to find the partition set, and then uses the centroid condition to find a new codebook. After each iteration the distortion will either decrease or stay the same. As a result the algorithm will always converge. The algorithm can be seen in more detail below. The system requires an initial codebook which will be introduced in the next section (2.3.2).

Input: pdf $f(x)$, initial codebook $C_1 = \{c_1^1, c_2^1, \dots, c_N^1\}$, threshold ϵ .

$m = 1$

$D_0 = \infty$

$D_1 = E[d(X, q^1(X))]$

while $\frac{D_{m-1}-D_m}{D_{m-1}} > \epsilon$ **do**

$m = m + 1$

$S_i^m = \{x : d(x, c_i^m) \leq d(x, c_j^m), j = 1, 2, \dots, N\} \quad i = 1, 2, \dots, N$

$c_i^m = \arg \min_c E[d(X, c)|X \in S_i^m] \quad i = 1, 2, \dots, N$

$C_m = \{c_1^m, c_2^m, \dots, c_N^m\}$

$D_m = E[d(X, q^m(X))]$

end

Output: C_m

Algorithm 1: Lloyd Algorithm for Scalar Quantizer (MSE distortion)

In practice the quantized values, or codewords c_i 's, are not sent across the channel, in-

stead they are assigned indices $i = 1, \dots, N$ which can be sent across the channel. This means that the codebook needs to be stored at both the encoder and decoder. By considering the transmission of the coded index, the input data $\{V_i\}$ can be mapped to a coded index via encoding function γ and reconstruct the data via a decoding function β , i.e.,

$$\gamma(v) = i \text{ if } v \in S_i,$$

$$\beta(j) = c_j, c_j \in C,$$

where c_j is the codeword corresponding with output index $j = 1, \dots, N$. In a noiseless channel, the received index is identical to the transmitted one; however, when there is noise present the transmitted indices may become corrupted. In a noiseless channel, the distortion of the quantizer can be directly calculated. In this scenario, the way in which the indices are assigned to the codewords has no effect on the final distortion. In the next section the case where the channel experiences noise will be investigated.

2.3 Channel Optimized Quantization

2.3.1 Introduction

As seen in the previous section, the indices from the quantization function are vulnerable to noise in the channel. The channel optimized quantization aims to protect the indices from this noise by considering the channel transition probabilities with the goal of reducing the expected distortion. By including these probabilities, the centroid and nearest neighbour conditions will be modified accordingly. The system is assumed to have perfect *a priori* information. In other words, the channel statistics are known to the encoder. There is a tradeoff between the quantization error and channel error, which avoids the addition of redundancy to protect against channel error. In addition one can exploit the non-binary output from the NBNDP to implement soft-decision decoding when $q > 1$.

2.3.2 System Description

Consider a real-valued and memoryless input source $\{V_i\}_{i=1}^{\infty}$ with zero mean and unit variance. The communication system is depicted in Figure 2.6.

The COSQ-Encoder γ is a mapping that takes the input source value $v \in \mathbb{R}$ and maps it to

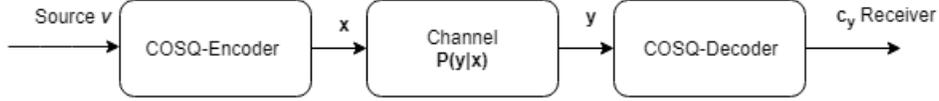


Figure 2.6: Block Diagram of COSQ System

an index $\mathbf{x} \in \{0, 1\}^n$ with relation

$$\gamma(v) = \mathbf{x} \text{ if } v \in S_{\mathbf{x}},$$

where $\{S_{\mathbf{x}} : \mathbf{x} \in \{0, 1\}^n\}$ is a partition of \mathbb{R} . The binary n -tuples \mathbf{x} are sent across the NBNDC-QB, resulting in a 2^q -ary output. Finally, the COSQ-decoder $\beta(\cdot)$, takes the 2^q -ary, n -tuple output \mathbf{y} and maps it to a reconstruction of v via,

$$\beta(\mathbf{y}) = c_{\mathbf{y}} \in \mathbb{R}, \mathbf{y} \in \{0, 1, \dots, 2^q - 1\}^n.$$

2.3.3 COSQ Design

The objective of the COSQ training algorithm is to find the codebook $C = \{c_{\mathbf{y}} : \mathbf{y} \in \{0, 1, \dots, 2^q - 1\}^n = \mathcal{Y}\}$ and the partition set $P = \{S_{\mathbf{x}} : \mathbf{x} \in \{0, 1\}^n = \mathcal{X}\}$ such that the average distortion-per-sample is minimized. This distortion measure is given by:

$$\begin{aligned} D(C, P) &= \sum_{\mathbf{x} \in \mathcal{X}} \int_{S_{\mathbf{x}}} p(v) \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) d(v, c_{\mathbf{y}}) dv \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \int_{S_{\mathbf{x}}} p(v) \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) (v - c_{\mathbf{y}})^2 dv \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) \int_{S_{\mathbf{x}}} p(v) (v - c_{\mathbf{y}})^2 dv \end{aligned}$$

where $p(v)$ is the source probability density function, $P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})$ is the channel transition probability, and \mathcal{X} and \mathcal{Y} are the alphabets at the input and output respectively. The encoding rate is $n = \log_2 N$. As before, with scalar quantization, there are necessary conditions for optimality. The nearest neighbour and centroid conditions are variations of the previous conditions. It has been proved that for a noisy system, the necessary optimality conditions are as follows [7], [8], [9]

- **Generalized Nearest Neighbour Condition**

$$S_i^* = \left\{ v : \sum_{\mathbf{y} \in J} P(\mathbf{y}|\mathbf{x}) |v - c_{\mathbf{y}}|^2 \leq \sum_{\mathbf{y} \in J} P(\mathbf{y}|\tilde{\mathbf{x}}) |v - c_{\mathbf{y}}|^2, \forall \tilde{\mathbf{x}} \in I \right\}$$

where $I = \{0, 1\}^n$, $J = \{0, 1, \dots, 2^q - 1\}^n$.

- **Generalized Centroid Condition**

$$\begin{aligned} c_{\mathbf{y}}^* &= \arg \min_{\hat{v} \in \mathbb{R}} E[d(V, \hat{v})|\mathbf{Y} = \mathbf{y}], \quad \mathbf{y} \in J \\ &= \frac{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \int_{S_{\mathbf{x}}} v f(v) dv}{\sum_{\mathbf{x}} P(\mathbf{y}|\mathbf{x}) \int_{S_{\mathbf{x}}} f(v) dv}. \end{aligned}$$

The channel error represents the probability of a binary value changing from its original value to a different value between 0 and $2^q - 1$. Note that the codebook and partition index sets are not the same size here. The partition set is indexed using binary n -tuples $I = \{0, 1\}^n$, while the codebook is indexed with 2^q -ary n -tuples $J = \{0, 1, \dots, 2^q - 1\}^n$, where q is the soft-decision granularity. As before, the Lloyd algorithm can be used to iterate between the conditions. The algorithm now also requires the channel statistics for the NBNDC-QB. Before the algorithm can iterate between these two conditions, an initial codebook is required.

2.3.4 Initial Codebook Design and Simulated Annealing

In order to implement the COSQ scheme the COSQ training algorithm needs to first optimize a codebook and partition set. For this algorithm, the input required is training data,

an initial codebook and an index assignment. The initial codebook is generated from a uniform quantizer and the index assignment is a result of simulated annealing. The uniform quantizer is a variation of scalar quantization which partitions the source space into equal, non-overlapping, partitions

$$\begin{aligned}\cup_{i=1}^N S_i &= \mathbb{R}, \\ S_i \cap S_j &= \emptyset,\end{aligned}$$

for $i \neq j$ where $i, j = 1, \dots, N$. As the quantization is uniform, an interval Δ is calculated $\Delta = \frac{v_{\max} - v_{\min}}{n}$, where v_{\max} and v_{\min} are the maximum and minimum values of the training data, respectively. The quantizer is defined as

$$q_{\text{uniform}}(v) = a_i \in \{a_1, \dots, a_n\}, \text{ if } v \in \left[a_i - \frac{\Delta}{2}, a_i + \frac{\Delta}{2} \right]$$

where the a_i 's are the partition midpoints. Using this quantizer, one can obtain an initial codebook that is fed into the noiseless Lloyd algorithm to obtain an optimized initial codebook.

The index assignment needs to be optimized for the given channel conditions. Farvardin in [9] proposed to use a method known as simulated annealing. This is a form of a randomized stochastic relaxation algorithm. It is adapted from the process of studying crystal growth, where the material is heated to the melting point and then the temperature is gradually decreased, allowing for the material to form a crystal. In mathematics, it is a non-linear technique used for optimization problems. Specifically for channel coding, simulated annealing is used to optimize the codeword indices.

The simulated annealing method starts by defining the initial state of the system. To determine the next state, different configurations are generated in a probabilistic way to allow for local "hill climbing". This is done to evade local minima. If the cooling is done sufficiently slowly, the algorithm can converge to the global minimum in probability; however in practice

faster cooling is used to find a local minimum. A common cooling schedule is

$$T_k = \alpha T_{k-1} - 1,$$

and $0 < \alpha < 1$ where T is the effective temperature of the system [23]. To determine the state at which the system should change, the energy of different states are calculated and compared. When the indices are sent across the noisy channel, an error in the binary representation of the index $i = \{0, \dots, 2^n - 1\}$, where n is the rate of the encoder, causes a discrepancy of 2^i in the codebook index. For example, when $n = 4$, if 0100, the binary string for index 4 is received as 1100, the index at the receiver is now 12. To avoid this kind of impact from a single error, the binary representations of the index values need to be reordered such that a change in a binary value changes the codebook value as little as possible. To do this simulated annealing finds a locally optimal state by comparing the energy of different states. This algorithm is done in the following four steps:

Simulated Annealing Algorithm

Step 1: Set the temperature to the initial high temperature T_0 and randomly chose an initial state b , corresponding to a permutation of the indexing for the codebook.

Step 2: Choose the next state b' randomly and calculate the change in distortion:

$$\Delta \varepsilon^2 = \varepsilon^2(b') - \varepsilon^2(b)$$

where ε^2 is the expected distortion $D(C, P)$. If $\Delta \varepsilon^2 < 0$, replace b with b' , and go to **step 3**, otherwise replace b by b' with probability $\exp(-\Delta \varepsilon^2/T)$ and go to **step 3**.

Step 3: If after a predefined number of perturbations, N , the energy is no longer decreasing, go to **step 4**. Otherwise go to **step 2**.

Step 4: Decrease the temperature using the cooling schedule given. If the temperature falls below the prescribed freezing temperature T_f or the system appears stable, the algorithm is

completed and the final state is state b . Otherwise, go to **step 2**.

The cooling schedule used for the implementation of this algorithm is:

$$T_k = \alpha T_{k-1} - 1$$

for $0 < \alpha < 1$. The other constants used for the algorithm are shown in the table below:

T_0	10
T_f	0.00025
α	0.97
N	200

These parameters were taken from [23]. Note that T_0 is a high temperature relative to $\Delta\varepsilon^2$.

2.3.5 Results

To obtain the following results, Algorithm 2 (see p.23) was used. To further optimize the results, the noise in the channel is gradually increased so that each progressively noisy channel simulation is initialized using a codebook from an already somewhat noisy channel. The results in [22] demonstrate that this method can result in up to 2 dB of gain in the signal-to-distortion ratio (SDR). This method is called increase-decrease and was repeated 5 times to obtain the results shown below. In the results, the value of comparison is the SDR. This quantity, which, under the assumption that the source has zero mean and unit variance, is given by

$$\text{SDR}_{dB} = -10 \log_{10}((V - \hat{V})^2).$$

The larger the SDR the lower the mean-squared error between the reproduction and the source, and hence the better the performance.

For the results shown in Tables 2.1 and 2.2, the threshold for the relative decrease in distortion, $\epsilon = 0.0001$, and 500,000 training vectors were used. The probability distribution $(\rho_0, \rho_1, \dots, \rho_{2^q-1})$ for the NBNDC-QB for soft-decision granularity $q = 1, 2, 3$ is given in Table 2.3 for a range of channel signal-to-noise ratios (CSNR). The CSNRs used are 15 dB, 10 dB, 5 dB, 2 dB, and 0 dB. To calculate the probability distributions in the channel, based on the CSNR in dB, refer to equations (1)-(3) in [26]. The results correspond with those from [22], sometimes performing better as the increase-decrease method introduced in [22] was used for all results. As expected, the performance increases with rate and soft-decision parameter q , and the memory in the channel.

Run uniform quantization and noiseless Lloyd algorithm to obtain initial codebook

$$C_1 = \{c_1^1, c_2^1, \dots, c_N^1\}.$$

for $CSNR = [15, 10, 5, 2, 0]$ dB repeated in alternating order 5 times **do**

Run simulated annealing to obtain index assignment.

Input: pdf $f(x)$, initial codebook $C_1 = \{c_1^1, c_2^1, \dots, c_N^1\}$, index assignment, threshold ϵ .

$$m = 1$$

$$D_0 = \infty$$

$$D_1 = E[d(V, \hat{V})]$$

while $\frac{D_{m-1} - D_m}{D_m} > \epsilon$ **do**

$$m = m + 1$$

$$S_i^m = \{v : \sum_{y \in J} P(y|x)d(v, c_y^m) \leq \sum_{y \in J} P(y|\tilde{x})d(v, c_y^m), \forall \tilde{x} \in I\} \quad i = 1, 2, \dots, N$$

$$c_y^m = \arg \min_{\hat{v} \in \mathbb{R}} E[d(V, \hat{V})|Y = y] \quad y \in J$$

$$C_m = \{c_1^m, c_2^m, \dots, c_N^m\}$$

$$D_m = E[d(V, \hat{V})]$$

end

Output: C_m

end

Algorithm 2: COSQ Algorithm for optimal training codebook

Table 2.1: SDR Results for COSQ Training with Gaussian and Laplacian Data, Channel Correlation $\text{cor} = 0$, $\alpha = 1$, $M = 1$

		CSNR				
q	R	15 dB	10 dB	5 dB	2 dB	0 dB
Gaussian Distribution						
1	1	4.16	3.76	2.87	2.20	1.67
	2	8.12	6.69	4.66	3.44	2.70
	3	11.02	8.50	5.80	4.45	3.57
	4	13.09	10.39	7.51	5.57	4.32
	5	14.79	12.13	8.23	6.22	4.96
	6	16.80	14.00	9.83	7.35	5.84
	7	19.74	15.86	11.10	8.35	6.51
	8	21.25	17.32	12.55	9.166	7.14
2	1	4.23	3.85	3.10	2.42	1.95
	2	8.47	7.00	5.10	4.00	3.26
	3	11.54	9.21	6.84	5.26	4.32
3	1	4.24	3.90	3.14	2.48	2.05
	2	6.93	5.80	4.07	3.20	2.63
	3	10.30	7.93	5.66	4.28	3.53
	4	12.09	10.47	7.67	5.77	4.55
Laplacian Distribution						
1	1	2.88	2.64	2.08	1.60	1.26
	2	6.49	5.29	3.62	2.66	2.08
	3	9.13	6.97	4.62	3.46	2.76
	4	11.24	8.70	5.76	4.22	3.37
2	1	2.93	2.70	2.23	1.78	1.46
	2	6.90	5.65	3.99	3.08	2.51
	3	10.16	7.46	5.53	4.12	3.35

Table 2.2: SDR Results for COSQ Training, for Gaussian Data, $\alpha = 1$

		CSNR				
q	R	15 dB	10 dB	5 dB	2 dB	0 dB
Gaussian Distribution, $M = 1$, cor = 0.9						
1	1	4.17	3.77	2.88	2.16	1.67
	2	8.85	8.01	6.70	5.69	5.08
	3	13.41	11.85	9.47	8.06	7.24
Gaussian Distribution, $M = 1$, cor = 0.5						
1	1	4.17	3.77	2.88	2.16	1.67
	2	8.27	6.83	4.63	3.38	2.68
	3	11.28	8.66	5.78	4.18	3.22
Gaussian Distribution, $M = 5$, cor = 0.9						
1	1	4.17	3.77	2.88	2.16	1.67
	2	8.88	8.19	6.76	5.96	5.15
	3	13.49	11.94	9.68	8.31	7.52

Table 2.3: The ρ Values for Corresponding CSNR for NBNDC-QB with Soft-Decision Resolution $q = 1, 2, 3$ [22]

CSNR	q	ρ_0	ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7
15 dB	1	0.992	0.008	-	-	-	-	-	-
	2	0.972	0.020	0.006	0.001	-	-	-	-
	3	0.955	0.018	0.012	0.008	0.004	0.002	0.001	0.000
10 dB	1	0.977	0.023	-	-	-	-	-	-
	2	0.924	0.053	0.019	0.005	-	-	-	-
	3	0.865	0.051	0.037	0.024	0.013	0.006	0.003	0.001
5 dB	1	0.936	0.064	-	-	-	-	-	-
	2	0.782	0.154	0.054	0.010	-	-	-	-
	3	0.703	0.100	0.078	0.055	0.034	0.018	0.008	0.005
2 dB	1	0.892	0.108	-	-	-	-	-	-
	2	0.695	0.196	0.085	0.024	-	-	-	-
	3	0.563	0.132	0.112	0.084	0.054	0.031	0.015	0.009
0 dB	1	0.854	0.146	-	-	-	-	-	-
	2	0.627	0.227	0.110	0.036	-	-	-	-
	3	0.427	0.145	0.132	0.104	0.071	0.041	0.021	0.013

Chapter 3

COSQ of Correlated Sources over Orthogonal MACs

3.1 Derivation of Optimality Conditions

Instead of a single user accessing the channel, consider now two users, with real-valued messages V_1, V_2 that have joint probability density $f(v_1, v_2)$, where $(v_1, v_2) \in \mathbb{R}^2$. These messages are sent from different locations over the orthogonal MAC to a common receiver. The objective is to send both messages across separate, noisy channels and decode them simultaneously. This means the two encoders cannot communicate with one another; however the decoding can be done jointly. The motivation behind developing a new method for this scenario is that there are existing systems in which correlated data is sent either concurrently, or periodically. Instead of the decoder working with only the information sent from one transmission, it can use the second transmission to improve the end-to-end distortion. In the following sections, two schemes will be introduced. The first scheme uses the same encoding as the previously described COSQ algorithm, but decodes two sources jointly. The second scheme uses an optimized encoder that predicts the other user's values based on the correlation of the sources and a jointly optimized decoder. Both methods are described more explicitly below.

3.1.1 COSQ II: Individually Optimized Encoders, Jointly Optimized Decoder

When considering two users sending messages across independent channels and received at the same location, a natural solution is to optimize the decoder to take both received signals in consideration. This method does not require any changes to the encoder and is therefore simple to implement in existing systems; however there is an increase in complexity at the receiver. For this scheme the nearest neighbour condition previously derived is used as the encoder remains the same. At the output however, the two outputs have a new, joint codebook, with size $|C_u| = 2^{q(n_1+n_2)}$ per user $u \in \{1, 2\}$, where n_1 and n_2 are the rates corresponding to users 1 and 2 and q is the soft-decision granularity of the NBNDC. The new codebooks have codewords $c_{\mathbf{y}_1, \mathbf{y}_2} \in C_1$ and $\tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} \in C_2$

$$(c_{\mathbf{y}_1, \mathbf{y}_2}, \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}) = \arg \min_{(\hat{v}_1, \hat{v}_2) \in \mathbb{R}^2} E[d(V_1, \hat{v}_1) + d(V_2, \hat{v}_2) | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2] \quad \mathbf{y}_1 \in J_1, \mathbf{y}_2 \in J_2 \quad (3.1)$$

where $\mathbf{y}_1 \in J_1 = \{0, 1, \dots, 2^q - 1\}^{n_1}$, $\mathbf{y}_2 \in J_2 = \{0, 1, \dots, 2^q - 1\}^{n_2}$. Here, V_1, V_2 are the two correlated sources, \hat{v}_1, \hat{v}_2 are the outputs of the corresponding source samples, Y_1, Y_2 are the 2^q -ary output samples from the channel, where $n_u = \log_2 N_u$, $u = 1, 2$. This equation will be explicitly derived in the next section.

To implement this method a new codebook needs to be trained for the decoder. The final, optimized, codebook from the previously described, one user scheme is used as the initial codebook at each encoder. Correlated data is encoded using these codebooks and then at the decoder, the new codebook is calculated using the centroid condition in equation 3.1. The codewords reflect values which minimize the expected distortion between the reconstruction values and the data points in the partitions. As the encoding function and the codebook at the encoder remains the same, the algorithm is only iterated through once, as repeating it would result in no change.

Some interesting results were discovered when implementing this method. For one, when the correlation between the two Gaussian users $\rho = 0$, i.e. the sources are independent, the scheme is equivalent to the one user scheme. This is because the joint codebook condition can be reduced to the single user codebook. It was observed that as source correlation ρ increases, the distortion decreases; however when the two users are identical, and there is no noise in the system, the scheme is also equivalent to the one user scheme. This result is explicitly proven for rate one below; however the behaviour was also noticed at higher rates.

Lemma 1: When the bivariate Gaussian source is encoded with the rate one individually optimized COSQ encoder for a single user and the decoder is jointly optimized, the expected distortion $D = E[(X_1 - \hat{X}_1)^2 + (X_2 - \hat{X}_2)^2] = 2E[(X - \hat{X})^2]$ when the sources X_1 and X_2 are independent, where $E[(X - \hat{X})^2]$ is the expected distortion for a single user system. Furthermore, for a noiseless channel, the expected distortion is equal to the case when the sources are independent.

Proof: The case where the sources are independent is easily shown to be equivalent to the single user case as there is no gain from the source correlation and the optimality conditions simply reduce down to the single user case.

When the sources are identical, the proof that the performance is equivalent to the independent method is a little more interesting. $(X_1, X_2) \sim N([0, 0], 1, 1, \rho)$, are two jointly Gaussian sources, both mean 0, variance 1, and correlation ρ . The marginal distributions of this jointly Gaussian source is also Gaussian, $X_1, X_2 \sim N(0, 1)$.

The encoding codebook is constant in this scenario and can be shown to be, $(\hat{X}_1, \hat{X}_2) \in \left[\left(\sqrt{\frac{2}{\pi}}, \sqrt{\frac{2}{\pi}} \right), \left(-\sqrt{\frac{2}{\pi}}, \sqrt{\frac{2}{\pi}} \right), \left(\sqrt{\frac{2}{\pi}}, -\sqrt{\frac{2}{\pi}} \right), \left(-\sqrt{\frac{2}{\pi}}, -\sqrt{\frac{2}{\pi}} \right) \right]$. For simplicity, let $\sqrt{\frac{2}{\pi}} = x_+$

and $-\sqrt{\frac{2}{\pi}} = x_-$ then the expected distortion can be written as:

$$\begin{aligned}
D &= E[(X_1 - \hat{X}_1)^2 + (X_2 - \hat{X}_2)^2] \\
&= E[(X_1 - x_+)^2 + (X_2 - x_+)^2 | X_1 > 0, X_2 > 0]P(X_1 > 0, X_2 > 0) \\
&\quad + E[(X_1 - x_-)^2 + (X_2 - x_+)^2 | X_1 < 0, X_2 > 0]P(X_1 < 0, X_2 > 0) \\
&\quad + E[(X_1 - x_+)^2 + (X_2 - x_-)^2 | X_1 > 0, X_2 < 0]P(X_1 > 0, X_2 < 0) \\
&\quad + E[(X_1 - x_-)^2 + (X_2 - x_-)^2 | X_1 < 0, X_2 < 0]P(X_1 < 0, X_2 < 0).
\end{aligned}$$

When the two sources are not correlated so that $\rho = 0$, they are independent, and the distortion becomes:

$$\begin{aligned}
D_{\rho=0} &= (E[(X_1 - x_+)^2 | X_1 > 0] + E[(X_2 - x_+)^2 | X_2 > 0])P(X_1 > 0)P(X_2 > 0) \\
&\quad + (E[(X_1 - x_-)^2 | X_1 < 0] + E[(X_2 - x_+)^2 | X_2 > 0])P(X_1 < 0)P(X_2 > 0) \\
&\quad + (E[(X_1 - x_+)^2 | X_1 > 0] + E[(X_2 - x_-)^2 | X_2 < 0])P(X_1 > 0)P(X_2 < 0) \\
&\quad + (E[(X_1 - x_-)^2 | X_1 < 0] + E[(X_2 - x_-)^2 | X_2 < 0])P(X_1 < 0)P(X_2 < 0).
\end{aligned}$$

Since the marginals of both sources are normally distributed with $X_1, X_2 \sim N(0, 1)$, the generic notation can be used, $X \sim N(0, 1)$.

$$\begin{aligned}
D_{\rho=0} &= 2E[(X - x_+)^2 | X > 0]P(X > 0)^2 \\
&\quad + 2(E[(X - x_+)^2 | X > 0] + E[(X - x_-)^2 | X < 0])P(X > 0)P(X < 0) \\
&\quad + 2E[(X - x_-)^2 | X < 0]P(X < 0)^2.
\end{aligned}$$

The distribution is symmetric about 0 and hence the distortion can be written as:

$$\begin{aligned}
D_{\rho=0} &= 8E[(X - x_+)^2 | X > 0]P(X > 0)^2 \\
&= 8\left(\frac{1}{2}\right)^2 E[(X - x_+)^2 | X > 0] \\
&= 2E[(X - x_+)^2 | X > 0].
\end{aligned}$$

Similarly, for a single source the expected distortion is,

$$\begin{aligned}
E[(X - \hat{X})^2] &= E[(X - x_+)^2|X > 0]P(X > 0) + E[(X - x_-)^2|X < 0]P(X < 0) \\
&= 2E[(X - x_+)^2|X > 0]P(X > 0) \\
&= E[(X - x_+)^2|X > 0]
\end{aligned}$$

per user.

When the sources are identical implying $X_1 = X_2$, and $\rho = 1$ then following similar steps, and writing $X_1 = X_2 = X \sim N(0, 1)$, the distortion becomes:

$$\begin{aligned}
D_{\rho=1} &= E[(X_1 - \hat{X}_1)^2 + (X_2 - \hat{X}_2)^2] \\
&= 2E[(X - \hat{X})^2] \\
&= 2E[(X - x_+)^2|X > 0]P(X > 0) + 2E[(X - x_-)^2|X < 0]P(X < 0) \\
&= \frac{4}{2}E[(X - x_+)^2|X > 0] \\
&= 2E[(X - x_+)^2|X > 0].
\end{aligned}$$

As shown, these values for the expected distortion are all equal. The expected distortion can be explicitly calculated as,

$$D = 2E[(X - x_+)^2|X > 0] = 2 \left(E[X^2|X > 0] - 2E[X|X > 0]\sqrt{\frac{2}{\pi}} + \frac{2}{\pi} \right) = 2 \left(1 - \frac{2}{\pi} \right).$$

This corresponds to a distortion of $1 - \frac{2}{\pi}$ per user, or $-10 \log_{10}(1 - \frac{2}{\pi}) = 4.396$ dB. \square

This result can be extended to non-Gaussian bivariate distributions assuming the distribution has identical marginal distributions and the encoders are the same for both sources.

3.1.2 COSQ III: Jointly Optimized Encoders, Joint Optimized Decoder

The previous method did not make any changes to the encoder; however it is possible to alter the encoder to take into consideration the other user when choosing the index to send across

the channel. This is done by deriving the nearest neighbour condition using the end-to-end distortion of the two-user system.

Consider two real-valued, stationary, and ergodic sources $\{(V_1, V_2)^t\}_{t=1}^{\infty}$ with zero mean, unit variance, and correlation, ρ . The communication system is depicted in Figure 3.1.

They are independently encoded with a mapping $\gamma_u(\cdot), u = 1, 2$. γ_u maps the input source

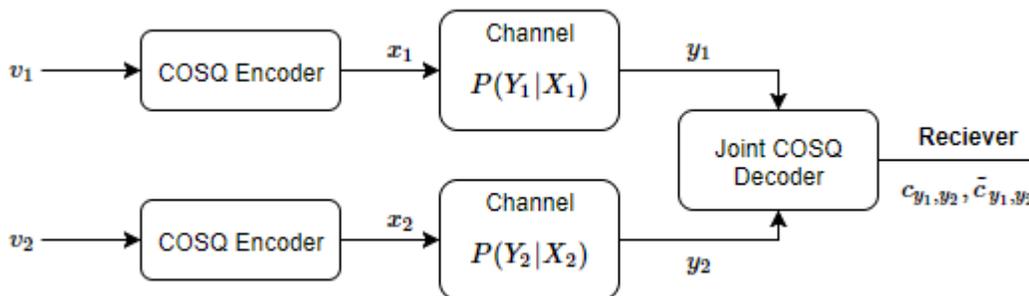


Figure 3.1: Block Diagram of Two User COSQ System

$v_u \in \mathbb{R}$, to binary string $\mathbf{x}_u = \{0, 1\}^n$, where n is the rate used to encode the source samples. Note that for simplicity n is used as the rate for both users; however the two sources can be encoded with non-equal rates. For $u = 1, 2$, the source v_u is encoded by

$$\gamma_u(v_u) = \mathbf{x}_u \text{ if } v_u \in S_{\mathbf{x}_u},$$

where $S_{\mathbf{x}_u}$ is the partition of \mathbb{R} corresponding with the \mathbf{x}_u^{th} binary string, $\{S_{\mathbf{x}_u} : \mathbf{x}_u \in \{0, 1\}^n\}$. These binary messages are sent across the channel. The channel outputs make a 2^q -ary message pair $\{(\mathbf{y}_1, \mathbf{y}_2) : \mathbf{y}_u \in \{0, 1, \dots, 2^q - 1\}^n = \mathcal{Y}\}$. The two output streams are jointly decoded with the following function:

$$\beta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^2.$$

The decoding function maps each pair of 2^q -ary outputs to an unique codebook pair,

$$\{(\mathbf{y}_1, \mathbf{y}_2)\} \rightarrow \{(c_{\mathbf{y}_1, \mathbf{y}_2}, \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})\} \in \mathbb{R}^2.$$

As before, the channel optimized scalar quantization algorithm aims to find the optimal codebook and partition set. Here $C = \{(c_{\mathbf{y}_1, \mathbf{y}_2} \in C_1, \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} \in C_2)\}$ is the codebook and $P = \{(S_{\mathbf{x}_1^j} \in P_1, S_{\mathbf{x}_2^k} \in P_2) : j, k \in \{1, \dots, N\}\}$ is the partition set for the quantization function. Since the encoder function is one-to-one, the partition is defined in terms of \mathbf{x}_u^j instead of only the index set, $\{1, \dots, N\}$. The end-to-end distortion of the system can be written as

$$D(C, P) = E[(V_1 - \hat{V}_1)^2 + (V_2 - \hat{V}_2)^2].$$

This can be expanded as

$$\begin{aligned} D(C, P) &= \sum_{\mathbf{x}_1 \in \mathcal{X}} \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} E[(V_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 + (V_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})^2 | V_1 \in S_{\mathbf{x}_1}, V_2 \in S_{\mathbf{x}_2}] \\ &\quad P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2) P(V_1 \in S_{\mathbf{x}_1}, V_2 \in S_{\mathbf{x}_2}) \\ &= \sum_{\mathbf{x}_1 \in \mathcal{X}} \sum_{\mathbf{x}_2 \in \mathcal{X}} \int_{S_{\mathbf{x}_1}} \int_{S_{\mathbf{x}_2}} f(v_1, v_2) \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2) \\ &\quad (|v_1 - c_{\mathbf{y}_1, \mathbf{y}_2}|^2 + |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2) dv_1 dv_2 \end{aligned} \tag{3.2}$$

where $f(v_1, v_2)$ is the joint density function of the two sources.

Nearest Neighbour Condition

First the optimal partition set is derived. For the derivation, the first user's conditions will be considered, and similarly the second user will have equivalent conditions. To find these partitions, the distortion is minimized for a given codebook. Considering that $V_1 = v_1 \in S_{\mathbf{x}_1^j}$

where $j = \{1, \dots, N\}$, the nearest neighbour can be derived as follows

$$\begin{aligned}
D(C, P) &= \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} ((v_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 + E[(V_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})^2 | V_1 = v_1, V_2 \in S_{\mathbf{x}_2}]) \\
&\quad P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) P(V_2 \in S_{\mathbf{x}_2} | V_1 = v_1) \\
&\geq \min_{\mathbf{x}_1^j} \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} ((v_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 + E[(V_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})^2 | V_1 = v_1, V_2 \in S_{\mathbf{x}_2}]) \\
&\quad P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) P(V_2 \in S_{\mathbf{x}_2} | V_1 = v_1)
\end{aligned} \tag{3.3}$$

To realize the minimization in (3.3) the partition sets can be defined to include all source values for which the $D(C, P)$ is minimized. This means the optimal partition set can be defined as follows:

$$S_{\mathbf{x}_u^j} = \{v_u : d_{v_u}(v_u, \mathbf{x}_u^j) \leq d_{v_u}(v_u, \mathbf{x}_u^k), \forall k \neq j\}, \forall j \in I, u \in \{1, 2\}$$

where $I = \{1, \dots, N\}$. Here, $d_{v_u}(v_u, \mathbf{x}_u^j)$, measures the distortion between the source v_u and the j^{th} binary string. For user 1, this distortion is given by

$$\begin{aligned}
d_{v_1}(v_1, \mathbf{x}_1^j) &= \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} ((v_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 + E[(V_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})^2 | V_1 = v_1, V_2 \in S_{\mathbf{x}_2}]) \\
&\quad P(V_2 \in S_{\mathbf{x}_2} | V_1 = v_1) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2)
\end{aligned} \tag{3.4}$$

and for V_2 :

$$\begin{aligned}
d_{v_2}(v_2, \mathbf{x}_2^j) &= \sum_{\mathbf{x}_1 \in \mathcal{X}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} ((v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2})^2 + E[(V_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 | V_2 = v_2, V_1 \in S_{\mathbf{x}_1}]) \\
&\quad P(V_1 \in S_{\mathbf{x}_1} | V_2 = v_2) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2^j)
\end{aligned} \tag{3.5}$$

Centroid Condition

To find the optimal codebook value, for a fixed partition region, the derivative of the distortion is taken with respect to the codebook value and set to zero. Again this is only shown for

user 1. The same derivation is applied to the second user.

$$\frac{\partial D(C,P)}{\partial c_{\mathbf{y}_1^j, \mathbf{y}_2^k}} = \sum_{\mathbf{x}_1 \in \mathcal{X}} \sum_{\mathbf{x}_2 \in \mathcal{X}} P(\mathbf{Y}_1 = \mathbf{y}_1^j, \mathbf{Y}_2 = \mathbf{y}_2^k | \mathbf{X}_1 = \mathbf{x}_1, \mathbf{X}_2 = \mathbf{x}_2) \int_{S_{\mathbf{x}_1}} \int_{S_{\mathbf{x}_2}} 2f(v_1, v_2)(v_1 - c_{\mathbf{y}_1^j, \mathbf{y}_2^k}) dv_1 dv_2. \quad (3.6)$$

Setting this to zero and isolating the codebook term results in

$$c_{\mathbf{y}_1^j, \mathbf{y}_2^k} = \frac{\sum_{\mathbf{x}_1 \in \mathcal{X}} P(\mathbf{Y}_1 = \mathbf{y}_1^j | \mathbf{X}_1 = \mathbf{x}_1) \sum_{\mathbf{x}_2 \in \mathcal{X}} P(\mathbf{Y}_2 = \mathbf{y}_2^k | \mathbf{X}_2 = \mathbf{x}_2) \int_{S_{\mathbf{x}_1}} \int_{S_{\mathbf{x}_2}} v_1 f(v_1, v_2) dv_1 dv_2}{\sum_{\mathbf{x}_1 \in \mathcal{X}} P(\mathbf{Y}_1 = \mathbf{y}_1^j | \mathbf{X}_1 = \mathbf{x}_1) \sum_{\mathbf{x}_2 \in \mathcal{X}} P(\mathbf{Y}_2 = \mathbf{y}_2^k | \mathbf{X}_2 = \mathbf{x}_2) \int_{S_{\mathbf{x}_1}} \int_{S_{\mathbf{x}_2}} f(v_1, v_2) dv_1 dv_2} \quad (3.7)$$

for user 1. Also,

$$c_{\mathbf{y}_1^j, \mathbf{y}_2^k} = \frac{\sum_{\mathbf{x}_2 \in \mathcal{X}} P(\mathbf{Y}_2 = \mathbf{y}_2^k | \mathbf{X}_2 = \mathbf{x}_2) \sum_{\mathbf{x}_1 \in \mathcal{X}} P(\mathbf{Y}_1 = \mathbf{y}_1^k | \mathbf{X}_1 = \mathbf{x}_1) \int_{S_{\mathbf{x}_2}} \int_{S_{\mathbf{x}_1}} v_2 f(v_1, v_2) dv_2 dv_1}{\sum_{\mathbf{x}_2 \in \mathcal{X}} P(\mathbf{Y}_2 = \mathbf{y}_2^k | \mathbf{X}_2 = \mathbf{x}_2) \sum_{\mathbf{x}_1 \in \mathcal{X}} P(\mathbf{Y}_1 = \mathbf{y}_1^k | \mathbf{X}_1 = \mathbf{x}_1) \int_{S_{\mathbf{x}_2}} \int_{S_{\mathbf{x}_1}} f(v_1, v_2) dv_2 dv_1} \quad (3.8)$$

for user 2.

Implementation of Encoding Condition

The encoding function derived above cannot be directly implemented as in the one user case. To implement the nearest neighbour condition, equations 3.4 and 3.5 need to be expanded and evaluated. Since the probability distributions of the sources used are known, the integrals can be explicitly solved. The source distributions used are Gaussian and Laplacian, so the bivariate probability distributions are required. The data used from these distributions will always be normalized to have unit variance and 0 mean. The bivariate Gaussian density, for correlation ρ and unit variance, 0 mean, is given by:

$$f_G(v_1, v_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(\frac{-1}{1-\rho^2}(v_1^2 + v_2^2 - 2\rho v_1 v_2)\right)$$

and the bivariate Laplacian:

$$f_L(v_1, v_2) = \frac{1}{\pi\sqrt{(1-\rho^2)}} K_0 \left(\sqrt{\frac{2(v_1^2 - 2\rho v_1 v_2 + v_2^2)}{1-\rho^2}} \right),$$

where $K_0(u)$ is the third Bessel function given by $K_0(u) = \frac{1}{2} \int_0^\infty s^{-1} \exp(-s - \frac{u^2}{4s}) ds, u > 0$ [28].

When implementing the NNC, the partition sets and codebooks change between each iteration, denoted by t , and the NNC is calculated using the partition sets and codebooks from the previous iteration, $t - 1$. Expanding the NNC (showing only for user 1, but process for user 2 is equivalent):

$$\begin{aligned} & d_{v_1}^t(v_1, \mathbf{x}_1^j) \\ &= \sum_{\mathbf{x}_2 \in \mathcal{X}} \int_{S_{\mathbf{x}_2}^{t-1}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} (|v_1 - c_{\mathbf{y}_1, \mathbf{y}_2}|^2 + |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2) f(v_1, v_2) \\ & \quad P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) dv_2 \\ &= \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} (v_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 P(V_1 = v_1, V_2 \in S_{\mathbf{x}_2}^{t-1}) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) \\ & \quad + \sum_{\mathbf{x}_2 \in \mathcal{X}} \int_{S_{\mathbf{x}_2}^{t-1}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2 f(v_1, v_2) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) dv_2. \end{aligned}$$

To calculate this value, the second term needs to be examined. It can be broken down into the following integrals:

$$\begin{aligned} & \int_{S_{\mathbf{x}_2}^{t-1}} |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2 f(v_1, v_2) dv_2 \\ &= \int_{S_{\mathbf{x}_2}^{t-1}} (v_2^2 - 2v_2 \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} + \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}^2) f(v_1, v_2) dv_2 \\ &= \int_{S_{\mathbf{x}_2}^{t-1}} v_2^2 f(v_1, v_2) dv_2 - 2\tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} \int_{S_{\mathbf{x}_2}^{t-1}} v_2 f(v_1, v_2) dv_2 + \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}^2 P(V_1 = v_1, V_2 \in S_{\mathbf{x}_2}^{t-1}). \end{aligned}$$

The first integral can be calculated for the bivariate Gaussian as follows; WLOG $S_{\mathbf{x}_2}^{t-1} = (a, b) \in \mathbb{R}$,

$$\begin{aligned}
& \int_{S_{\mathbf{x}_2}^{t-1}} v_2^2 f(v_1, v_2) dv_2 \\
&= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \int_{S_{\mathbf{x}_2}^{t-1}} v_2^2 \exp \left\{ \frac{-v_1^2 + 2\rho v_1 v_2 - v_2^2}{2(1-\rho^2)} \right\} dv_2 \\
&= \frac{1}{2} \left[\exp \left\{ \frac{v_1^2 - v_2^2}{2(1-\rho^2)} \right\} (\rho^2 v_1^2 + 1 - \rho^2) \operatorname{erf} \left\{ \frac{v_2 - \rho v_1}{\sqrt{2(1-\rho^2)}} \right\} \exp \left\{ \frac{\rho^2 v_1^2 + v_2^2}{2(1-\rho^2)} \right\} \right. \\
&\quad \left. - \frac{2(1-\rho^2)(\rho v_1 + v_2)}{\sqrt{2\pi(1-\rho^2)}} \exp \left\{ \frac{\rho v_1 v_2}{1-\rho^2} \right\} \right]_{v_2=b}^a
\end{aligned}$$

where $\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^x e^{-t^2} dt$.

The second term is similarly calculated; WLOG $S_{\mathbf{x}_2}^{t-1} = (a, b) \in \mathbb{R}$,

$$\begin{aligned}
& \int_{S_{\mathbf{x}_2}^{t-1}} v_2 f(v_1, v_2) dv_2 \\
&= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \int_{S_{\mathbf{x}_2}^{t-1}} v_2 \exp \left\{ \frac{-v_1^2 + 2\rho v_1 v_2 - v_2^2}{2(1-\rho^2)} \right\} dv_2 \\
&= \frac{1}{2} \left[\exp \left\{ \frac{v_1^2 - v_2^2}{2(1-\rho^2)} \right\} \rho v_1 \operatorname{erf} \left\{ \frac{v_2 - \rho v_1}{\sqrt{2(1-\rho^2)}} \right\} \exp \left\{ \frac{\rho^2 v_1^2 + v_2^2}{2(1-\rho^2)} \right\} - \frac{2(1-\rho^2)}{\sqrt{2\pi(1-\rho^2)}} \exp \left\{ \frac{\rho v_1 v_2}{1-\rho^2} \right\} \right]_{v_2=b}^a
\end{aligned}$$

Finally, the last term; WLOG $S_{\mathbf{x}_2}^{t-1} = (a, b) \in \mathbb{R}$,

$$\begin{aligned}
& P(V_1 = v_1, v_2 \in S_{\mathbf{x}_2}^{t-1}) \\
&= \int_{v_2 \in S_{\mathbf{x}_2}^{t-1}} f_{v_1, v_2}(v_1, v_2) dv_2 \\
&= \int_{v_2 \in S_{\mathbf{x}_2}^{t-1}} \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp \left\{ \frac{-v_1^2 + 2\rho v_1 v_2 - v_2^2}{2(1-\rho^2)} \right\} dv_2 \\
&= \frac{1}{\sqrt{2\pi(1-\rho^2)}} \left[\frac{1}{2} \sqrt{2\pi(1-\rho^2)} \operatorname{erf} \left\{ \frac{v_2 - \rho v_1}{\sqrt{2(1-\rho^2)}} \right\} \exp \left\{ \frac{-v_1^2}{2} \right\} \right]_{v_2=b}^a \\
&= \frac{1}{2} \left[\operatorname{erf} \left\{ \frac{v_2 - \rho v_1}{\sqrt{2(1-\rho^2)}} \right\} \exp \left\{ \frac{-v_1^2}{2} \right\} \right]_{v_2=b}^a
\end{aligned}$$

The NNC terms for the Laplacian distribution can also be calculated. These integrals have no definite solutions; however MATLAB can obtain numerical solutions. The indefinite

integrals required are

$$\begin{aligned}
& \int_{S_{\mathbf{x}_2}^{t-1}} |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2 f(v_1, v_2) dv_2 \\
&= \int_{S_{\mathbf{x}_2}^{t-1}} (v_2^2 - 2v_2 \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} + \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}^2) f(v_1, v_2) dv_2 \\
&= \int_{S_{\mathbf{x}_2}^{t-1}} v_2^2 f(v_1, v_2) dv_2 - 2\tilde{c}_{\mathbf{y}_1, \mathbf{y}_2} \int_{S_{\mathbf{x}_2}^{t-1}} v_2 f(v_1, v_2) dv_2 + \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}^2 P(V_1 = v_1, V_2 \in S_{\mathbf{x}_2}^{t-1})
\end{aligned}$$

The integrals are calculated on MATLAB and to generate the Laplacian data required to train the COSQ system, the following generator was implemented in MATLAB [28]

- Generate a bivariate normal variable (X_1, X_2) with mean zero, and covariance matrix Σ .
- Generate a standard exponential variable W .
- Set $(Y_1, Y_2) \leftarrow \sqrt{W}(X_1, X_2)$.
- Return (Y_1, Y_2) .

3.1.3 Encoding Complexity and Storage Requirements

As a method of comparison, the encoding complexity and storage requirements of the schemes introduced are evaluated. The encoding complexity gives an indication of the work required for the sensor to encode the source information. The storage requirements on the other hand combines the memory required for the entire system, in the form of pre-evaluated codebooks, or channel statistics. More formally, the values are defined as:

- **Encoding complexity:** Defined as the total number of multiplications required to encode a source sample.
- **Storage Requirements:** Total number of scalar values stored at the encoder and decoder in order to implement the quantizer.

The methods that will be compared, are the single user scheme, the two user scheme with a independently optimized encoder, jointly optimized decoder, and the two user scheme with a jointly optimized encoder and jointly optimized decoder. For the remainder of the thesis these schemes will be referred to as COSQ I, II, and III, respectively.

COSQ I: Single User COSQ

Encoding Complexity In order to implement the COSQ, the nearest neighbour condition with channel statistics used. For this method, the codebook is stored at both the encoder and decoder. The calculation at the encoder, the NNC, can be reduced to the following equation:

$$\begin{aligned}
& \operatorname{argmin}_{i \in I} \sum_{j=1}^N P_{Y|X}(j|i) |x - c_j|^2 \\
&= \operatorname{argmin}_{i \in I} \sum_{j=1}^N P_{Y|X}(j|i) \{x^2 - 2xc_j + c_j^2\} \\
&= \operatorname{argmin}_{i \in I} \{ \sum_{j=1}^N P_{Y|X}(j|i) c_j^2 - 2 \sum_{j=1}^N P_{Y|X}(j|i) x c_j \} \\
&= \operatorname{argmin}_{i \in I} \{ \sum_{j=1}^N P_{Y|X}(j|i) c_j^2 - 2x \sum_{j=1}^N P_{Y|X}(j|i) c_j \}.
\end{aligned} \tag{3.9}$$

The two sums are constant, therefore can be computed beforehand and stored at the encoder. Thus the multiplications that need to be done at the encoder, for each source, is equal to the number of codewords being used.

$$\text{COSQ I Complexity} = 2^n,$$

where n is the rate of the encoder.

Storage Requirements For the NNC from above to be implemented, the two sums need to be stored at the encoder. The size of both of these is that of the codebook, 2^n . For the decoder the codebook is required to map the output message back into real values. Thus the overall storage requirements for both encoder and decoder are 3 times the size of the codebook.

$$\text{COSQ I Storage} = 3 \cdot 2^n.$$

COSQ II: Individually Optimized Encoder, Jointly Optimized Decoder

Encoder Complexity Despite the fact that there are now two sources, the encoder functions the same way as for the COSQ with a single user; however the rates of the two users may not be the same. Thus the complexities of the two encoders are not necessarily equal and the complexity instead is written as the maximum number of multiplications required to encode a source sample.

$$\text{COSQ II Complexity} = \max(2^{n_1}, 2^{n_2})$$

where n_1 and n_2 are the rates of user 1 and 2 respectively.

Storage Requirements In order to implement the encoder function, once again the summation terms need to be stored by the encoder. This requires at most the codebook size of the larger codebook. The decoder codebook is now defined differently as both sources are taken into consideration. The size of the codebook at the output is $2^{n_1+n_2}$.

$$\text{COSQ II Storage} = 2 \max(2^{n_1}, 2^{n_2}) + 2^{n_1+n_2}.$$

COSQ III: Jointly Optimized Encoders, Jointly Optimized Decoder

Encoder Complexity The implementation of the NNC for the two user optimally, independently, encoded scheme, is more complex than the previous two. The calculation done for each source sample requires calculating the expected value of the other second source.

$$\begin{aligned} & d_{v_1}^t(v_1, \mathbf{x}_1^j) \\ = & \sum_{\mathbf{x}_2 \in \mathcal{X}} \int_{S_{\mathbf{x}_2}^{t-1}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} (|v_1 - c_{\mathbf{y}_1, \mathbf{y}_2}|^2 + |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2) f(v_1, v_2) \\ & P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) dv_2 \\ = & \sum_{\mathbf{x}_2 \in \mathcal{X}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} (v_1 - c_{\mathbf{y}_1, \mathbf{y}_2})^2 P(V_1 = v_1, V_2 \in S_{\mathbf{x}_2}^{t-1}) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) \\ & + \sum_{\mathbf{x}_2 \in \mathcal{X}} \int_{S_{\mathbf{x}_2}^{t-1}} \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} |v_2 - \tilde{c}_{\mathbf{y}_1, \mathbf{y}_2}|^2 f(v_1, v_2) P(\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2 | \mathbf{X}_1 = \mathbf{x}_1^j, \mathbf{X}_2 = \mathbf{x}_2) dv_2 \end{aligned}$$

For the bivariate Gaussian these integrals have been derived in the previous section. When combining them the final equation that needs to be solved at the encoder for each source has a total of 13 unique multiplication terms. These terms are multiplied together in 23 particular ways. This means that for each source sample, 36 multiplications for each codebook value is required to assess the codebook values which results in the lowest distortion. The codebook size is now larger too, as there is a codeword corresponding to each permutation at the output. The size of the new codebook is $2^{n_1+n_2}$.

$$\text{COSQ III Complexity} = 36 \cdot 2^{n_1+n_2}.$$

Storage Requirements Due to the complexity of the encoder, this scheme requires much more memory than the previous two. This includes the boundary values for the integrals over the partition sets, which is the size of the codebook plus one and the channel transition distribution which has the same size as the codebook. Some terms in the NNC are constants which need to be stored ahead of time, such as $\frac{2(1-\rho^2)}{\sqrt{2\pi(1-\rho^2)}}$ and similar terms. In total there are 7 vectors that require storage, which are the size of the output codebook. In addition there are 6 scalars which can be computed and stored at the encoder. The codebook also needs to be stored at the decoder, this has size $2^{n_1+n_2}$.

$$\text{COSQ III Storage} = 7 \cdot 2^{n_1+n_2} + 6.$$

These complexity and storage requirements of the three methods are summarized in table 3.1. The joint schemes add exponential complexity and storage at the receiver regardless of whether or not the encoder is changed.

COSQ III is the most complex scheme at the encoder, and at low rates, for example rate 2 at both users, $2^{n_1+n_2} = 8$ and the 36 factor makes a large impact. At higher rates the 36 factor becomes less important; however the exponential increases twice as fast than the single scheme, when both users have the same rate. In order to implement the COSQ III method, the encoder also has to be able to compute exponentials and the error function.

Table 3.1: Summary of Complexity and Storage Requirements for COSQ Schemes

Method	Complexity	Storage
COSQ I	2^n	$3 \cdot 2^n$
COSQ II	$\max(2^{n_1}, 2^{n_2})$	$2 \max(2^{n_1}, 2^{n_2}) + 2^{n_1+n_2}$
COSQ III	$36 \cdot 2^{n_1+n_2}$	$7 \cdot 2^{n_1+n_2} + 6$

3.2 Results

The numerical results for the COSQ II and COSQ III are presented here. First the algorithm used for the COSQ III scheme is shown in Algorithm 3. As with the COSQ I scheme, the noise is gradually increased and the final codebook from the previous channel noise is used to initialize the next iteration of the algorithm. As before, 500,000 data points are used per user and the threshold $\epsilon = 0.0001$. The channel parameters used are $M = 1$, $\alpha = 1$, $\text{cor} = 0$ and 0.9 , and $\text{CSNR} = 15, 10, 5, 2$, and 0 dB.

Run uniform quantization and noiseless Lloyd algorithm to obtain initial codebooks, $C^1 = \{C_1^1, C_2^1\}$;

for *Source Correlation*,

$\rho = [0, 0.01, 0.02, \dots, 0.98, 0.99, 1, 0.99, 0.98, \dots, 0.02, 0.01, 0]$ **do**

for $CSNR = [15, 10, 5, 2, 0]$ *repeated 5 times, alternating the order* **do**

Run simulated annealing to obtain optimized index assignment;

Input: Joint pdf $f(v_1, v_2)$, initial codebook $C^1 = \{C_1^1, C_2^1\}$, index assignment, threshold ϵ ;

$m = 1$;

$D_0 = \infty$;

$D_1 = E[d(V_1, \hat{V}_1) + d(V_2, \hat{V}_2)]$;

while $\frac{D_{m-1} - D_m}{D_m} > \epsilon$ **do**

$m = m + 1$;

$S_{x_u}^m = \{v_u : d_{v_u}^m(v_u, x_u^j) \leq d_{v_u}^m(v_u, x_u^k), \forall k \neq j, j \in I, u \in \{1, 2\}\}$;

$C_{y_1^j, y_2^k} = \frac{\sum_{x_1 \in \mathcal{X}} P(Y_1=y_1^j|X_1=x_1) \sum_{x_2 \in \mathcal{X}} P(Y_2=y_2^k|X_2=x_2) \int_{S_{x_1}^m} \int_{S_{x_2}^m} v_1 f(v_1, v_2) dv_1 dv_2}{\sum_{x_1 \in \mathcal{X}} P(Y_1=y_1^j|X_1=x_1) \sum_{x_2 \in \mathcal{X}} P(Y_2=y_2^k|X_2=x_2) \int_{S_{x_1}^m} \int_{S_{x_2}^m} f(v_1, v_2) dv_1 dv_2}$;

$\tilde{C}_{y_1^j, y_2^k} = \frac{\sum_{x_1 \in \mathcal{X}} P(Y_1=y_1^j|X_1=x_1) \sum_{x_2 \in \mathcal{X}} P(Y_2=y_2^k|X_2=x_2) \int_{S_{x_2}^m} \int_{S_{x_1}^m} v_2 f(v_1, v_2) dv_2 dv_1}{\sum_{x_1 \in \mathcal{X}} P(Y_1=y_1^j|X_1=x_1) \sum_{x_2 \in \mathcal{X}} P(Y_2=y_2^k|X_2=x_2) \int_{S_{x_2}^m} \int_{S_{x_1}^m} f(v_1, v_2) dv_2 dv_1}$;

$C^m = \{C_1^m, C_2^m\}$;

$D_m = E[d(V_1, \hat{V}_1) + d(V_2, \hat{V}_2)]$;

end

Output: C^m ;

end

end

Algorithm 3: Two User COSQ Algorithm

Table 3.2: Gain due to Source Correlation, ρ , in dB for COSQ II, Channel Parameters: $M = 1$, $\text{cor} = 0$, $\alpha = 0$

	CSNR (dB)					
Rate	Noiseless	15	10	5	2	0
Gaussian						
1	0.6	0.6	0.7	1.1	0.9	0.8
2	1.0	1.2	1.6	1.4	1.3	1.2
3	0.8	1.7	1.8	1.7	1.7	1.8
4	0.3	2.5	2.0	2.5	3.0	2.5
Laplacian						
1	0.3	0.3	0.3	0.5	0.6	0.6
2	0.7	1.0	1.0	1.1	1.1	1.1
3	0.8	1.3	1.3	1.5	1.5	1.4
4	0.5	1.9	1.8	2.7	1.7	1.5

3.2.1 COSQ II

Table 3.2 shows the increase in SDR due to the source correlation ρ of the input source for Gaussian and Laplacian sources over channels of varying CSNR. It shows the maximum increase in SDR, i.e. the distance between the lowest SDR, which occurs when the sources are independent and the highest SDR which generally occurs close to when the sources are dependent. The general trend is that as the noise increases, the gain also increases, to a point. As the rate increases, the performance gain also increases.

The graphs below show the performance of the COSQ II algorithm at varying rates and channel noise, with the bivariate Gaussian and Laplacian used as the training data. Figures 3.2, 3.3, 3.4, and 3.5 show the COSQ for rates from 1 to 4 trained on the bivariate Gaussian, at varying channel SNR (CSNR). The general trend is an increase in SDR, with the increase of the source correlation. The numerical results support the Lemma 1, when the channel is noiseless, the performance is the same for the independent and dependent sources with rate one encoding. This behaviour is also observed at higher rates. Figures 3.6, 3.7, 3.8, and 3.9 show the SDR results for rates from 1 to 4 with varying CSNR, for the bivariate Laplacian. With this distribution Lemma 1 does not hold for noiseless channels, as the

SDR for dependent sources is lower than the SDR for independent sources, at high rates. This is likely due to numerical error as the SDR difference between the independent and identical sources is only 0.1 to 0.2 dB. Figure 3.10 shows the COSQ II algorithm when the two Gaussian users are encoded with different rates. As expected, the higher the rate of the second user, the higher the SDR the for other user. If the second user is encoded with a lower rate, the gain is not as substantial. As with the other scheme, the higher performance gains occur at higher channel noise. Finally Figure 3.11 shows the gain due to channel memory. In these graphs, the channel correlation parameter $\text{cor} = 0.9$ and is represented by the dotted line. Generally the channel correlation contributes a gain of 2 to 4 dB, with the highest gain occurring when the channels have high noise.

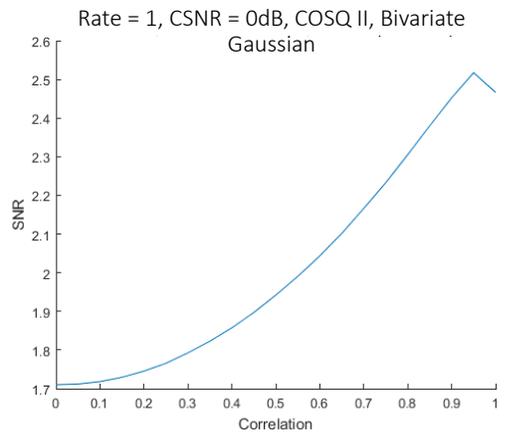
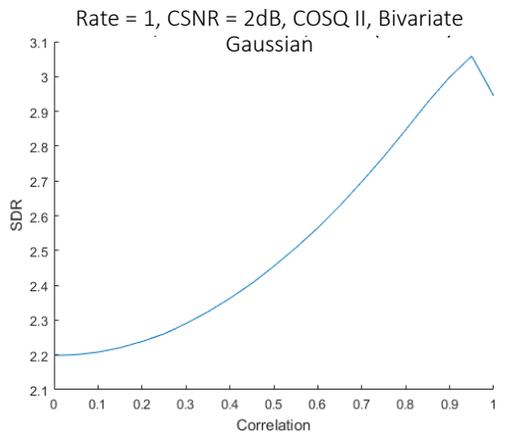
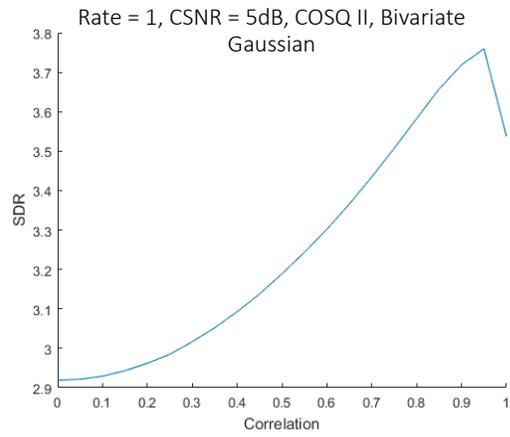
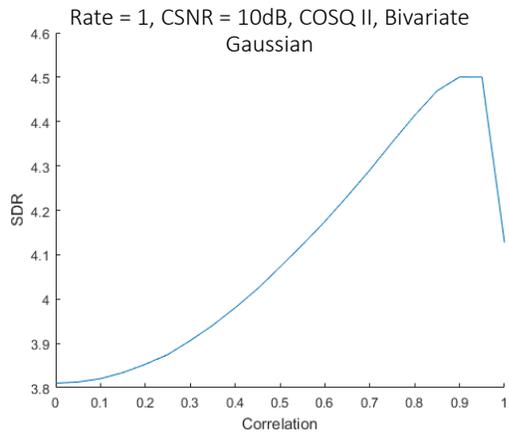
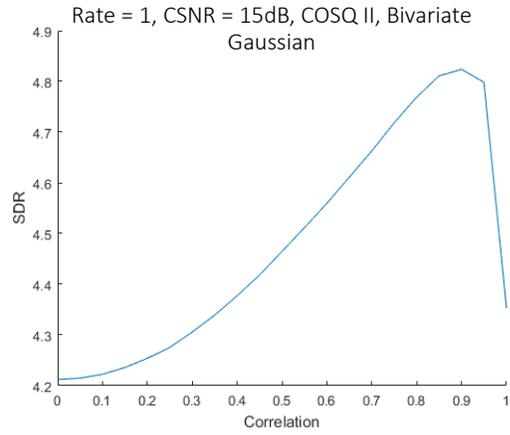
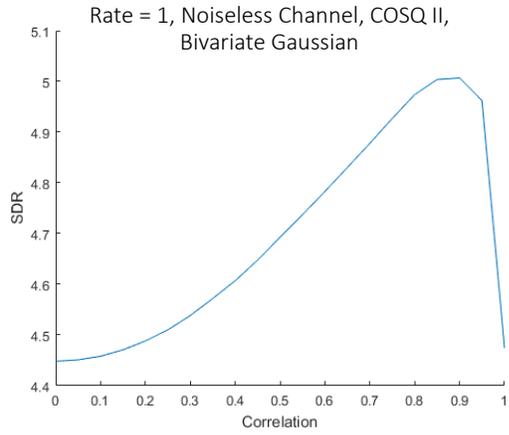


Figure 3.2: Two User Rate 1 COSQ II with Varying CSNR, Gaussian

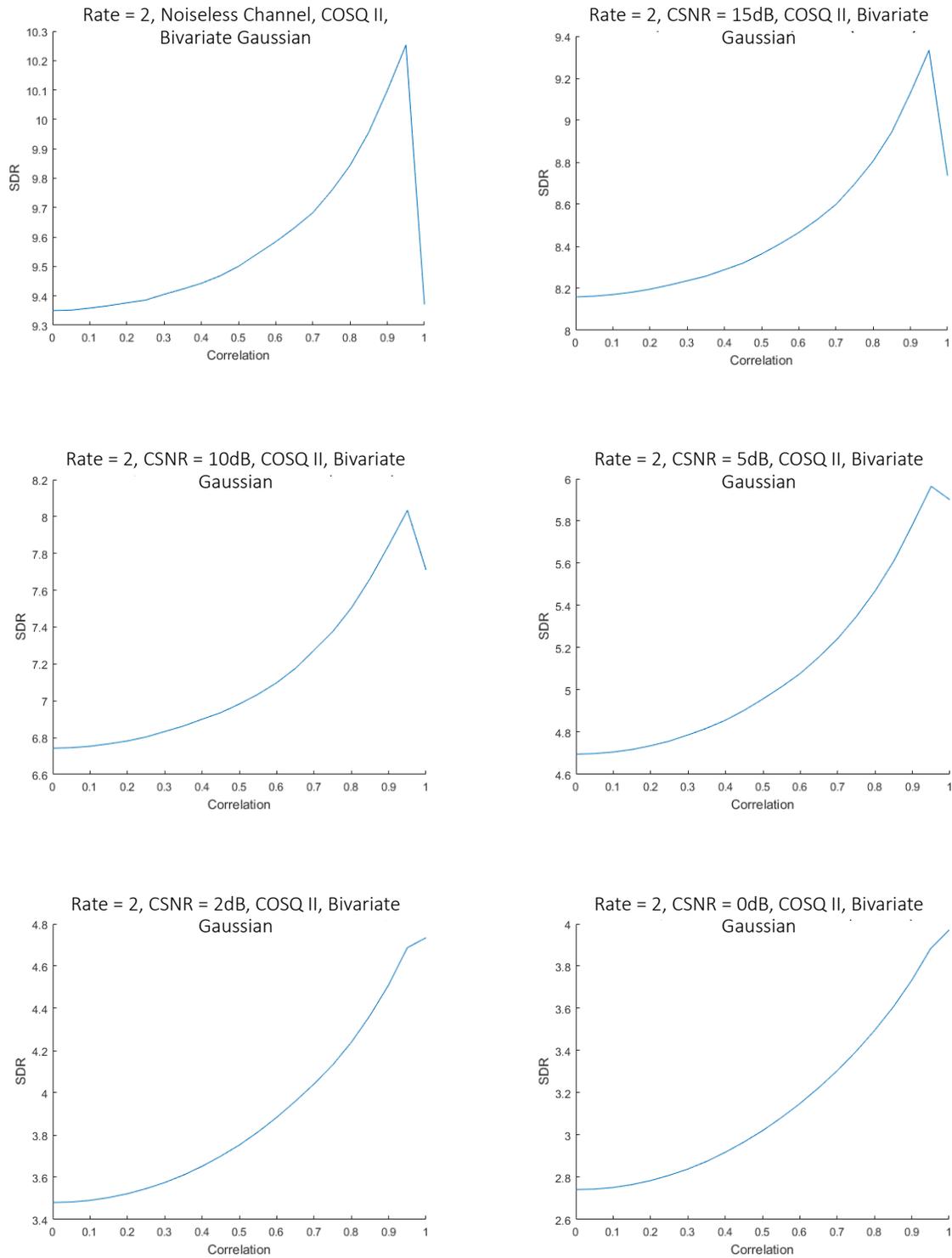


Figure 3.3: Two User Rate 2 COSQ II with Varying CSNR, Gaussian

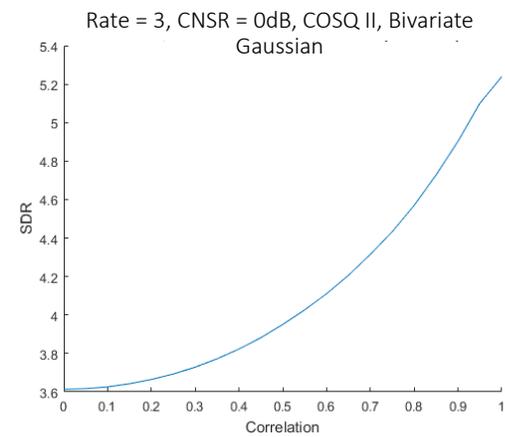
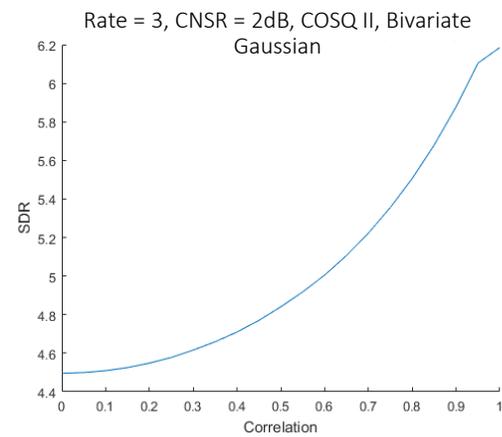
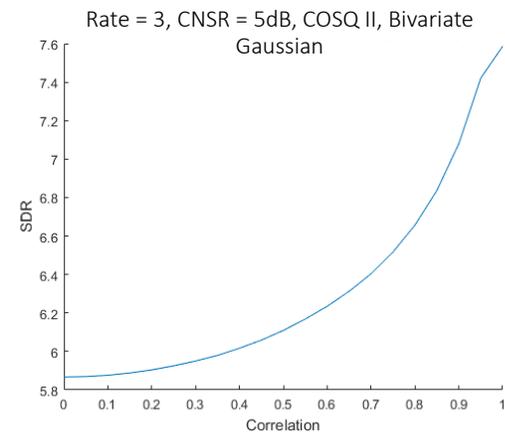
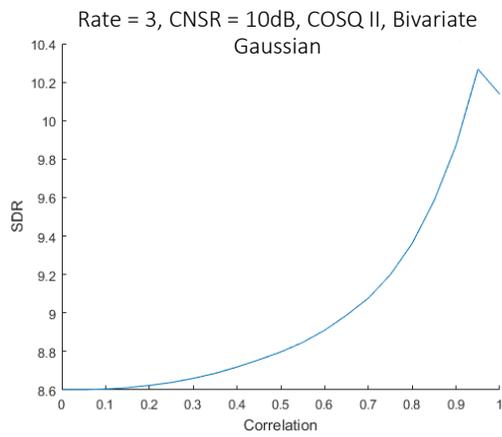
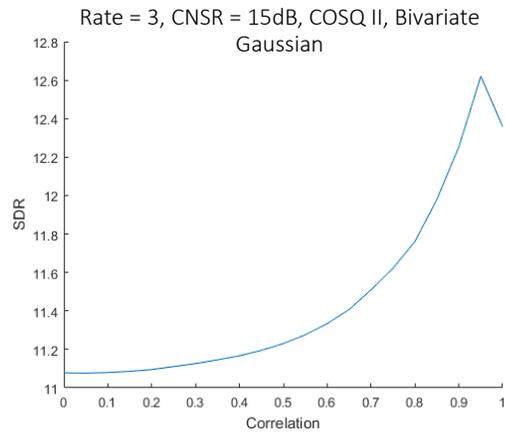
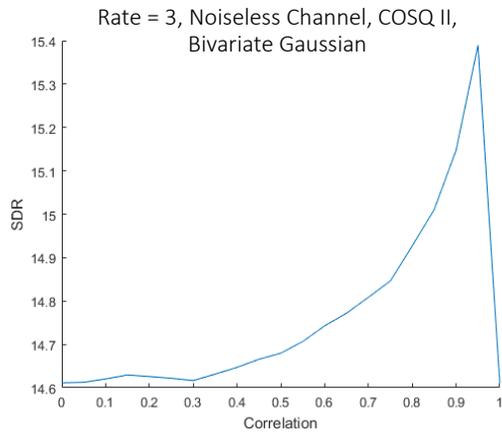


Figure 3.4: Two User Rate 3 COSQ II with Varying CSNR, Gaussian

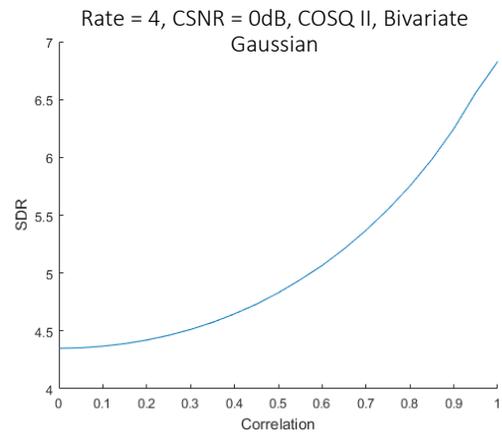
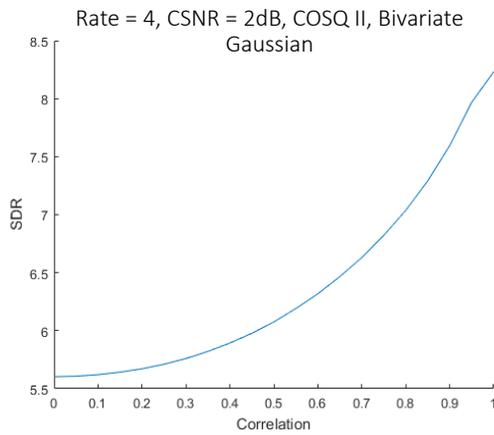
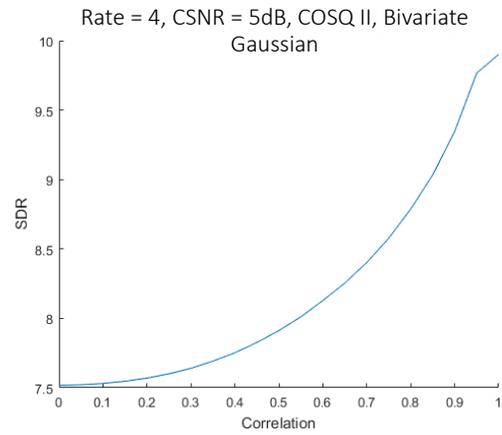
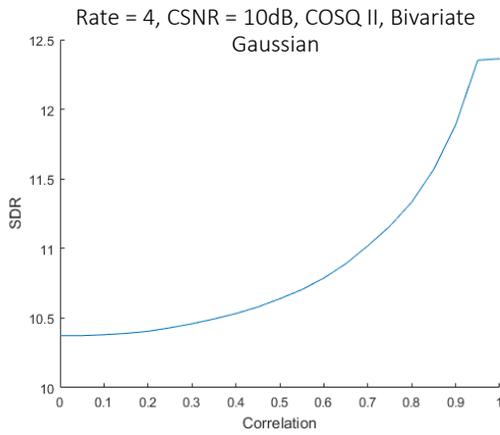
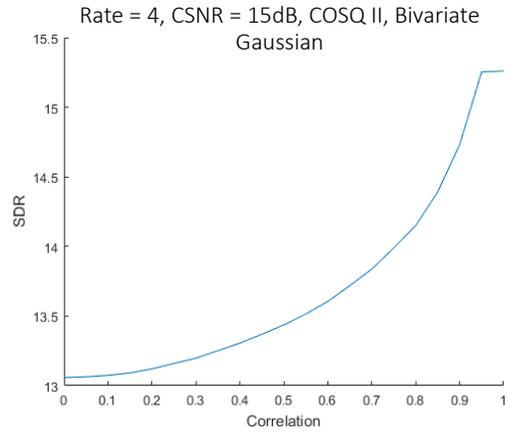
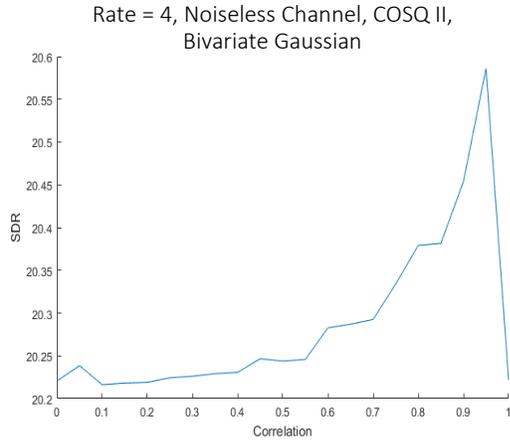


Figure 3.5: Two User Rate 4 COSQ II with Varying CSNR, Gaussian

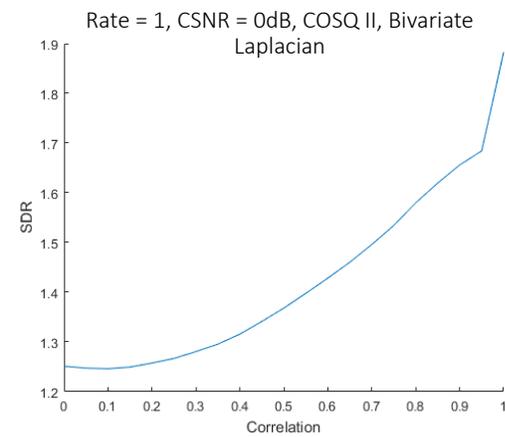
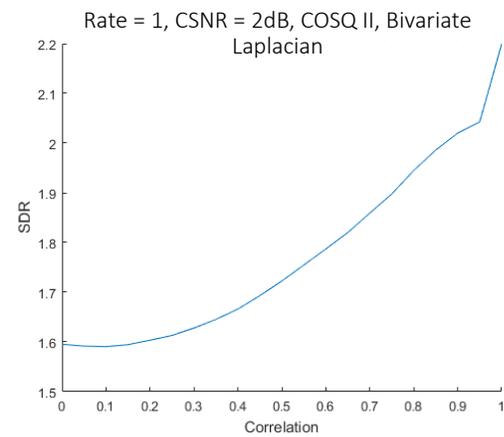
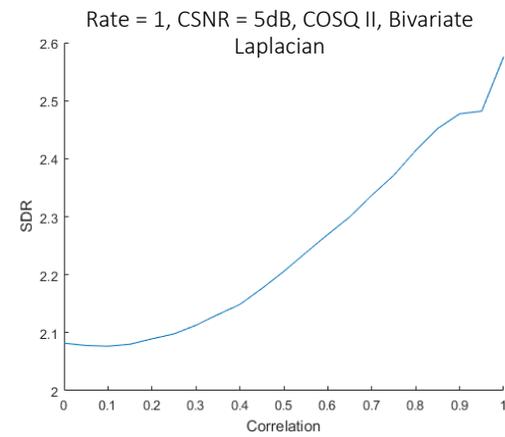
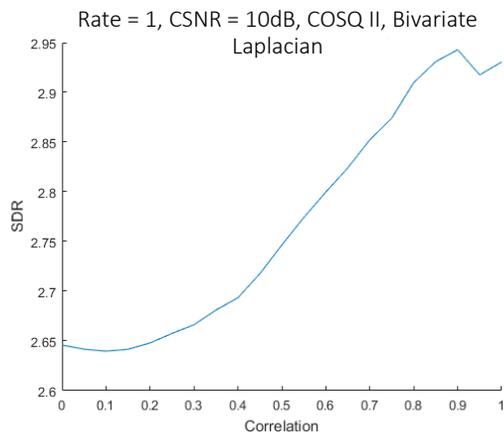
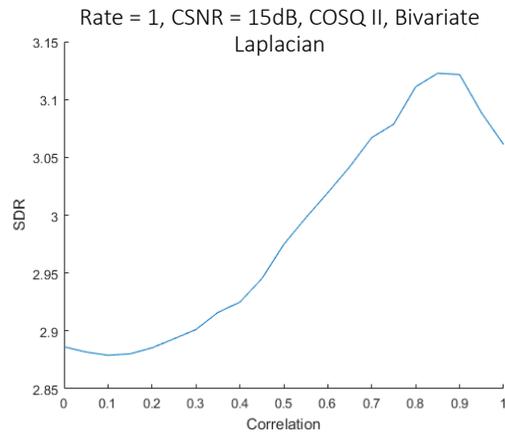
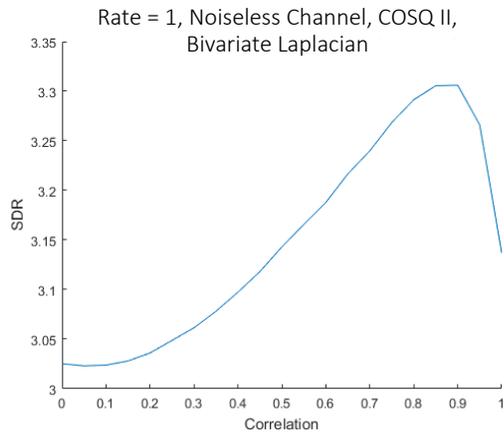


Figure 3.6: Two User Rate 1 COSQ II with Varying CSNR, Laplacian

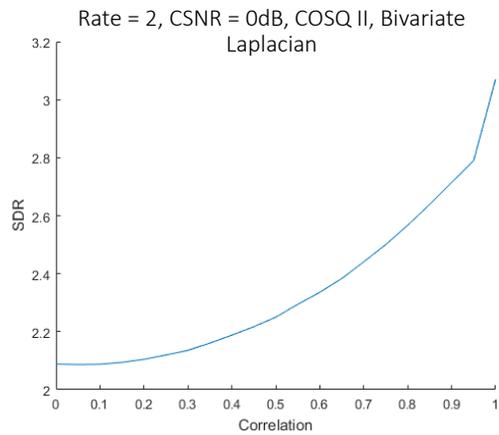
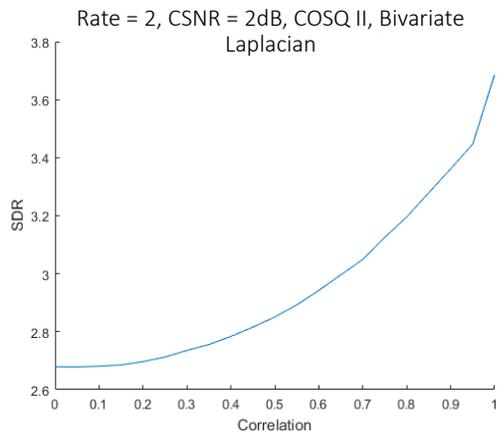
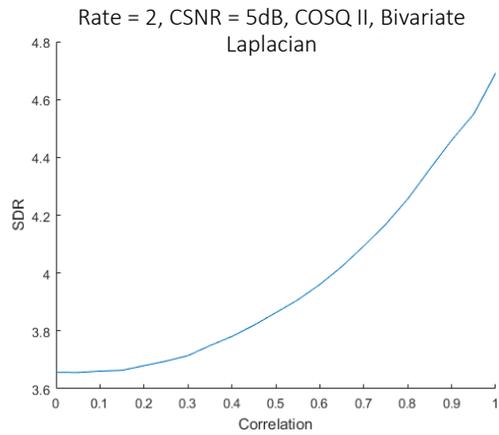
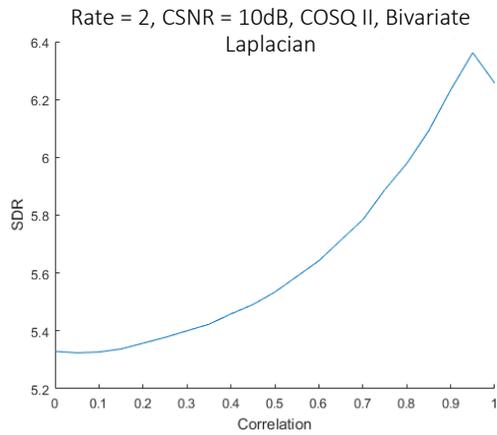
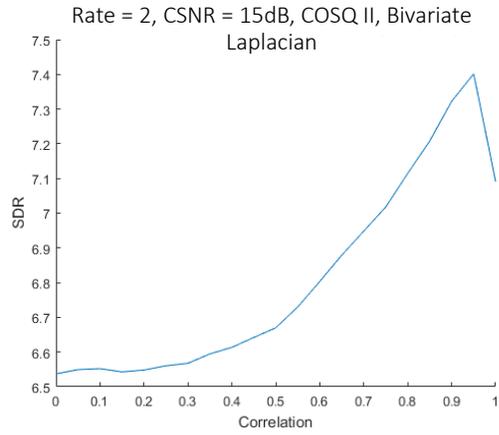
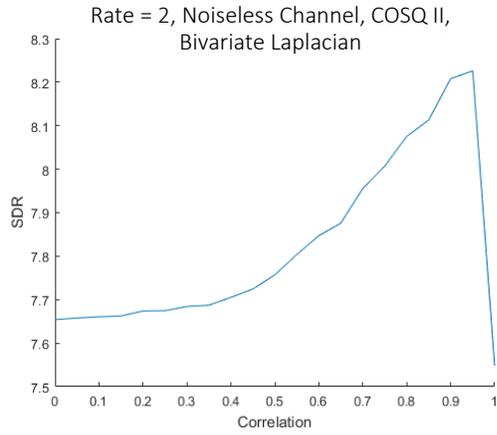


Figure 3.7: Two User Rate 2 COSQ II with Varying CSNR, Laplacian

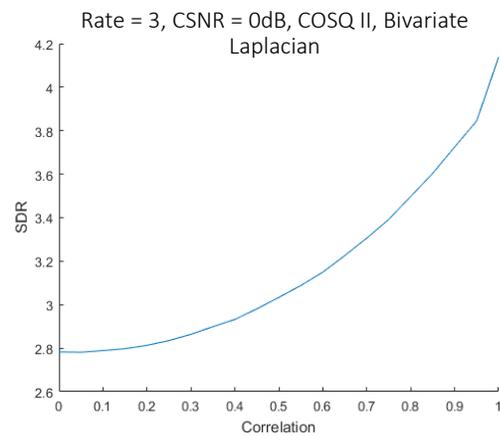
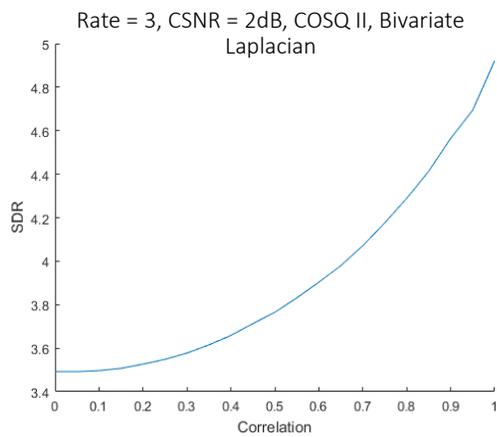
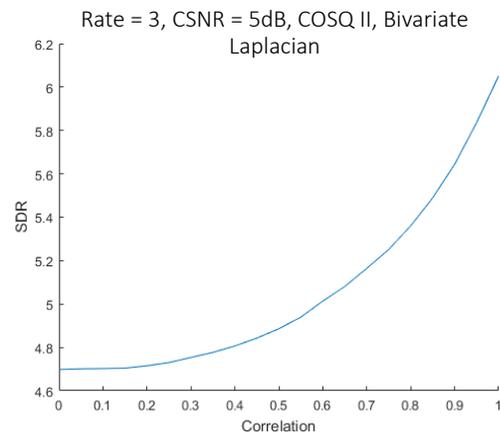
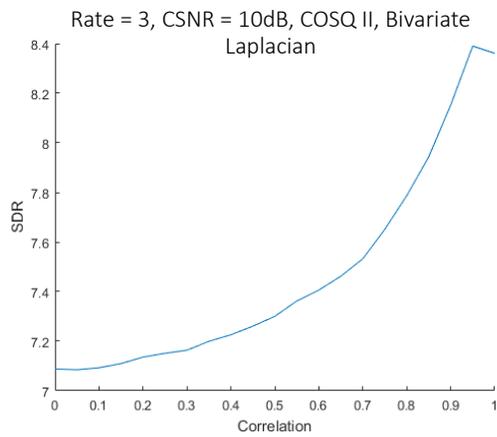
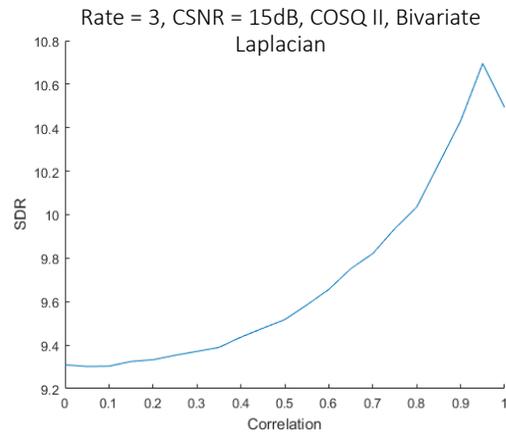
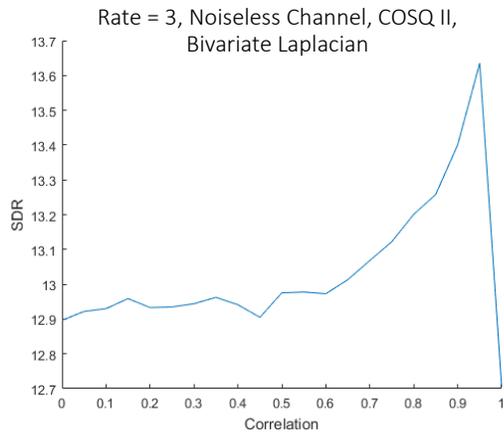


Figure 3.8: Two User Rate 3 COSQ II with Varying CSNR, Laplacian

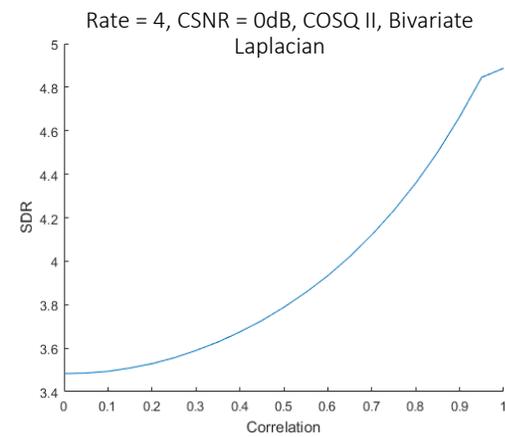
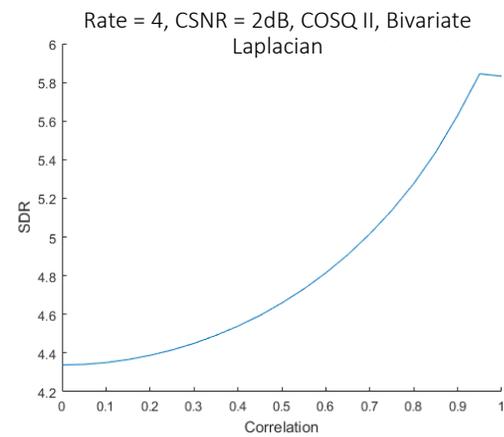
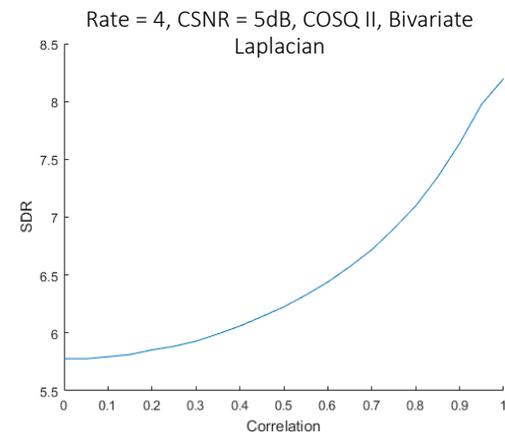
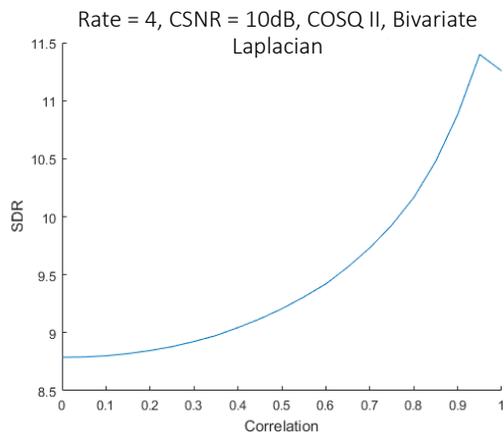
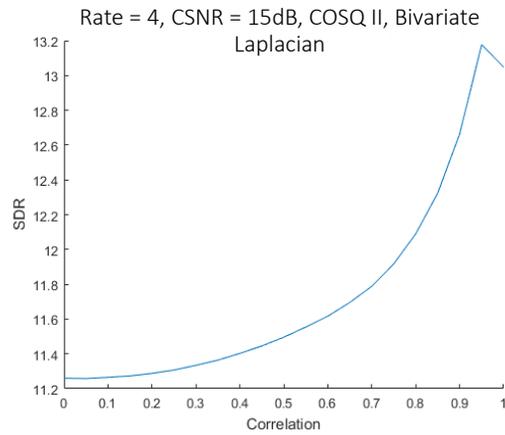
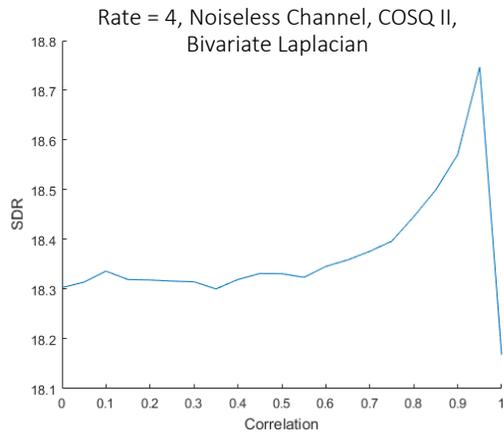
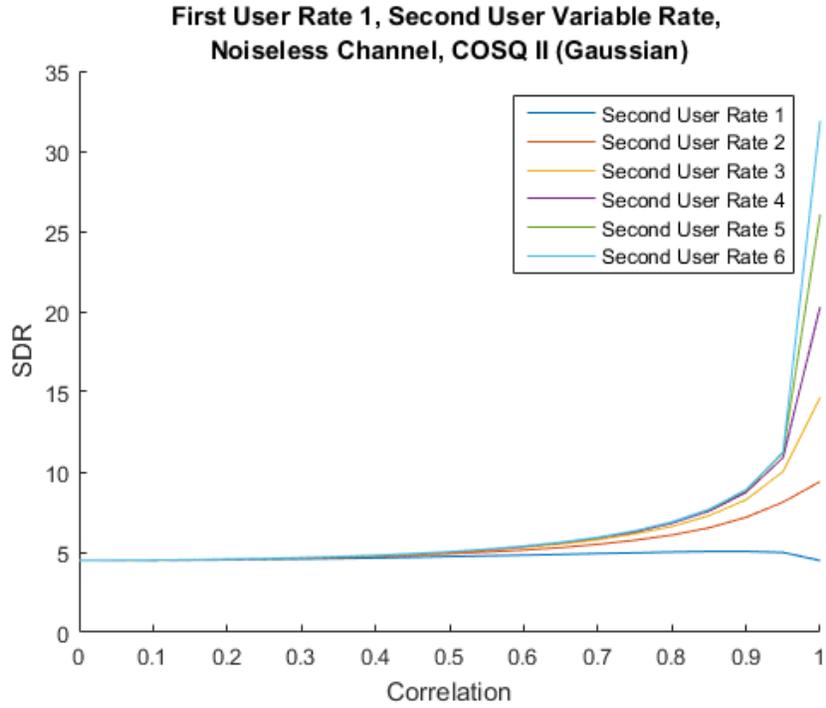
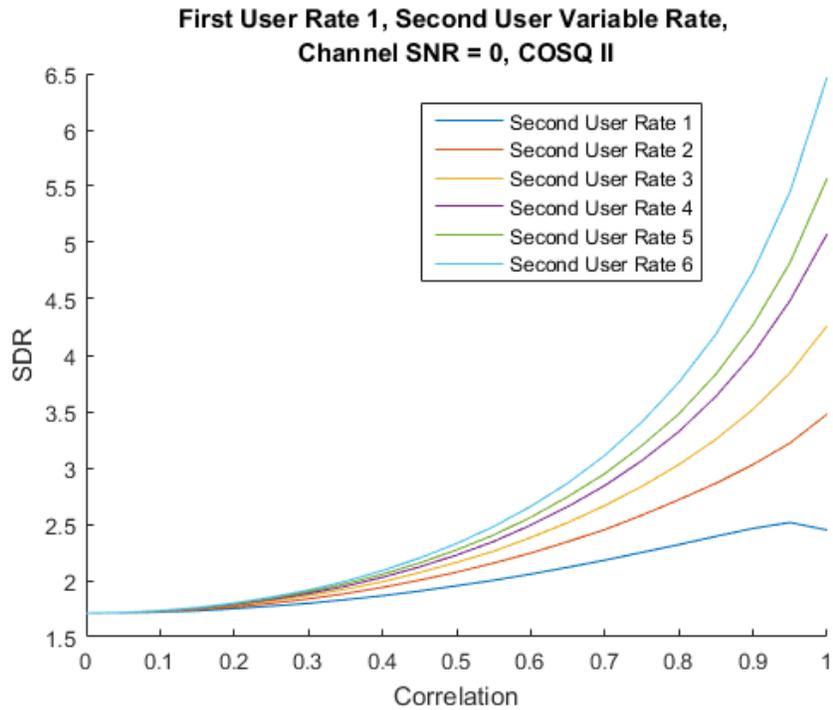


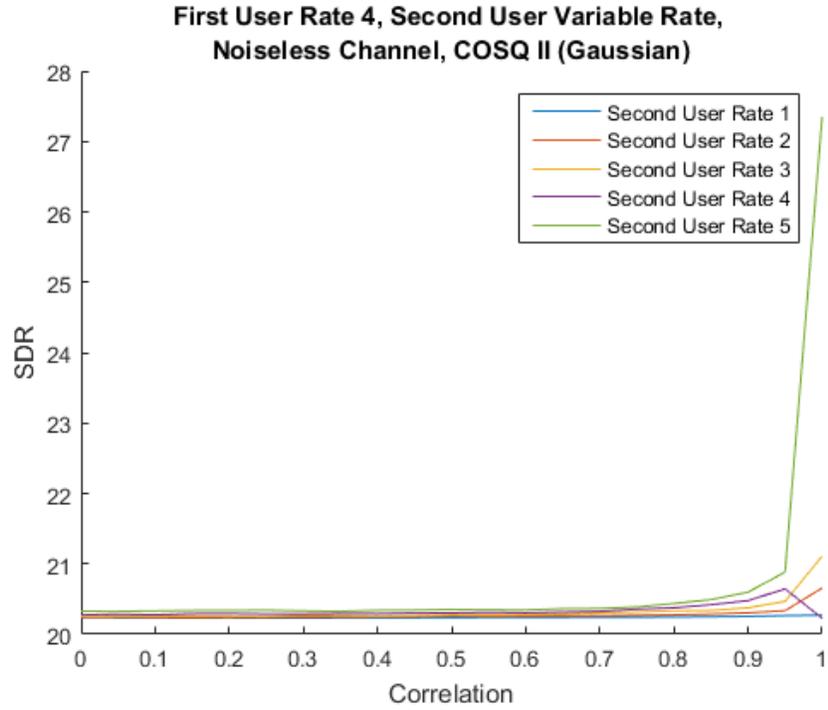
Figure 3.9: Two User Rate 4 COSQ II with Varying CSNR, Laplacian



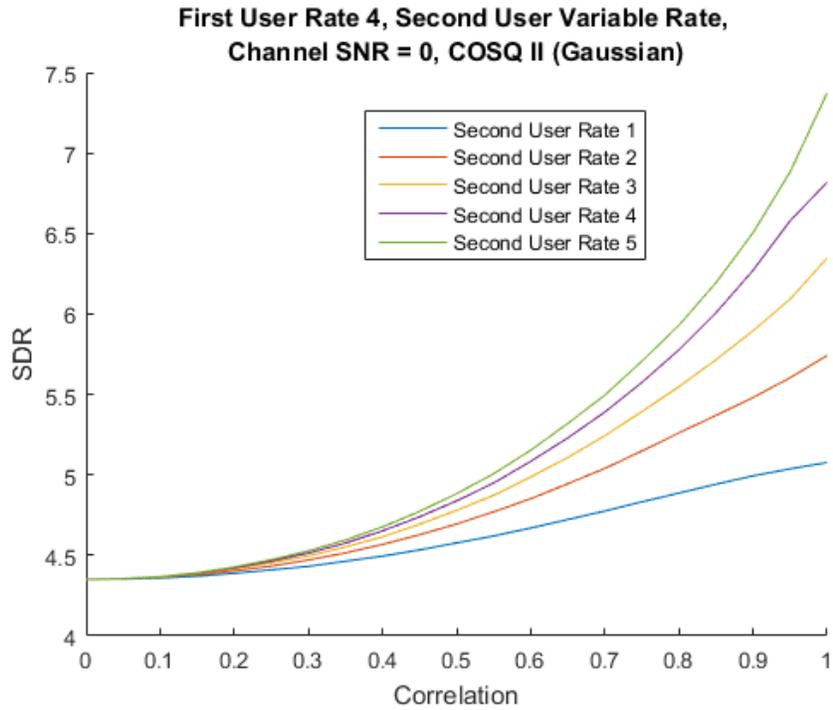
First User Rate 1, Second User Varying Rate, Noiseless Channel



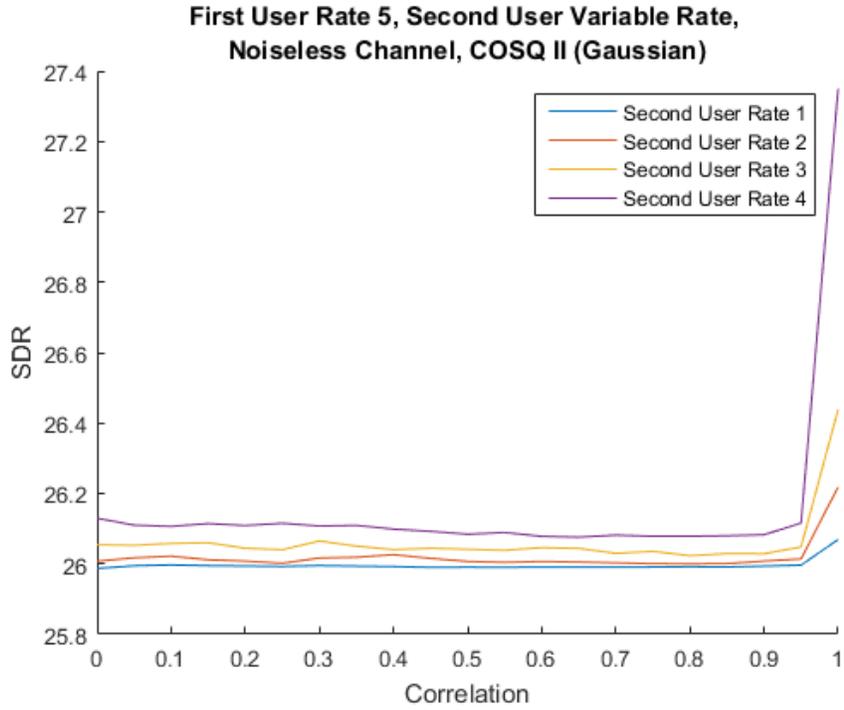
First User Rate 1, Second User Varying Rate, CSNR = 0 dB



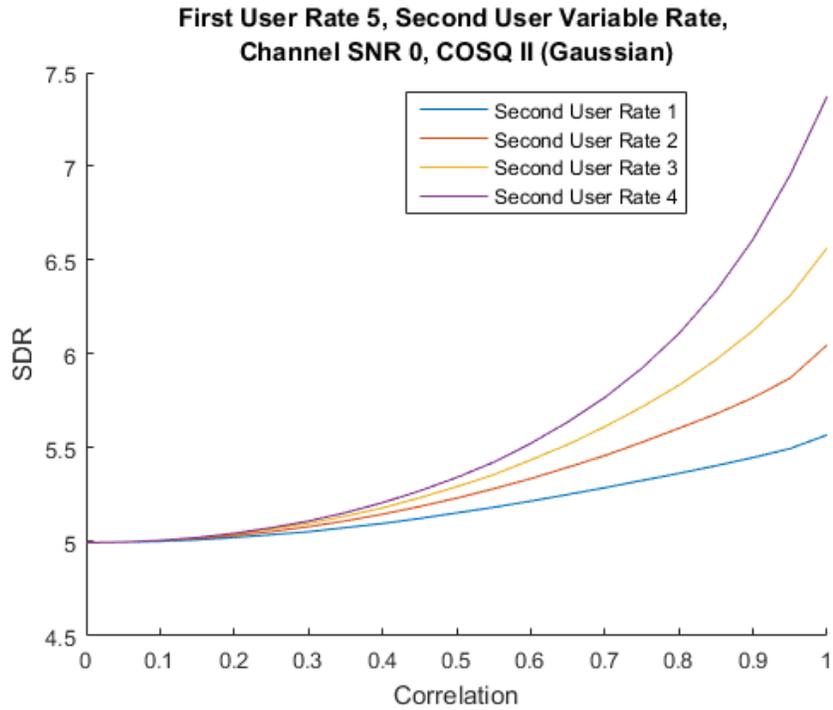
First User Rate 4, Second User Varying Rate, Noiseless Channel



First User Rate 4, Second User Varying Rate, CSNR = 0 dB



First User Rate 5, Second User Varying Rate, Noiseless Channel



First User rate 5, second user varying, CSNR = 0 dB

Figure 3.10: Two User Varying Rates COSQ II with Noiseless and Very Noisy Channels, Gaussian

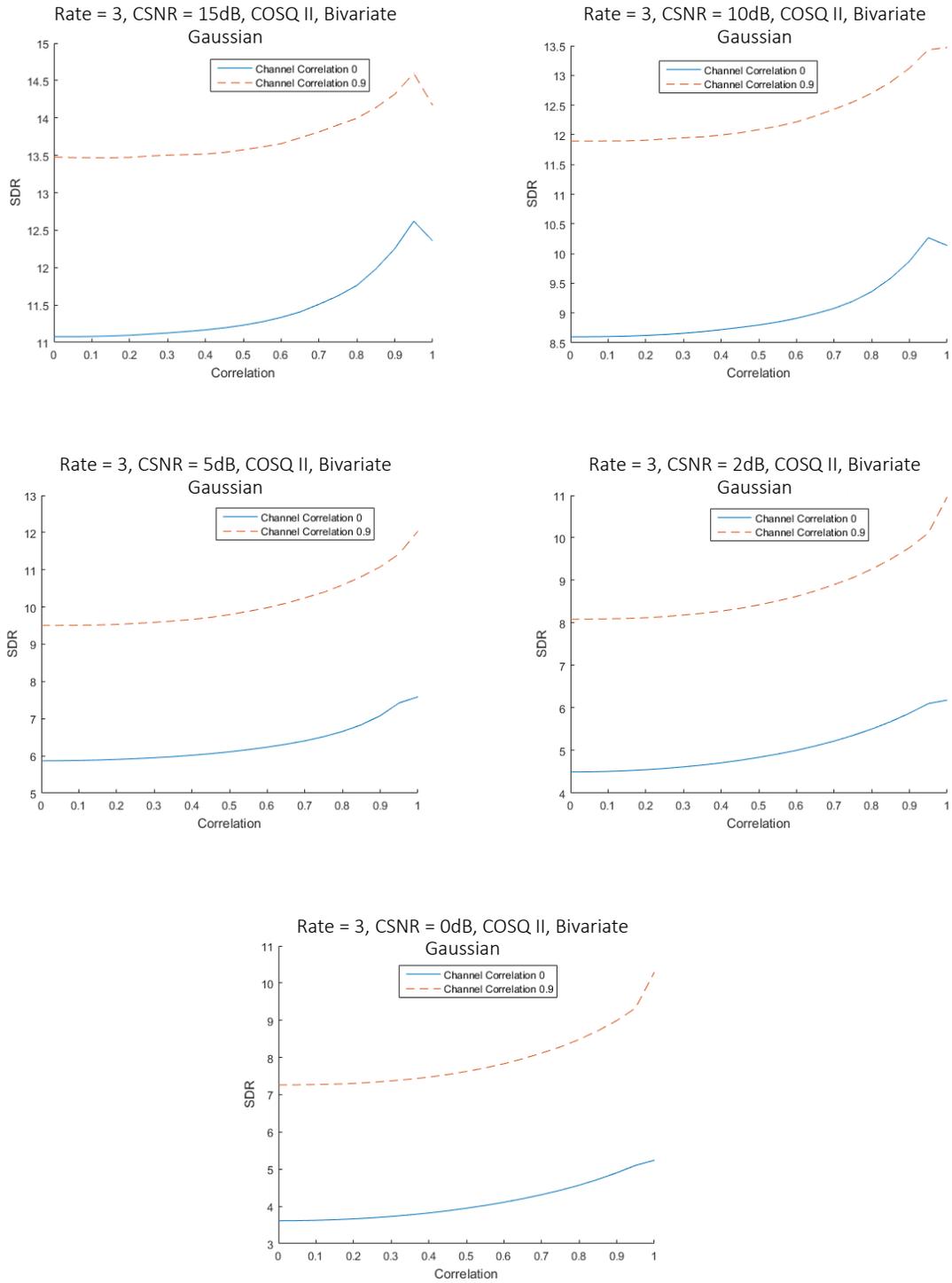


Figure 3.11: Two User Rate 3 COSQ II with Varying CSNR, Channel Correlation $\text{cor} = 0$ and 0.9, Gaussian

3.2.2 COSQ III

The next set of graphs show the SDR results for the COSQ III scheme. Figure 3.12 shows the results for rate 1. Once again the horizontal axis shows the source correlation, from 0 to 1, and the vertical axis shows the SDR. The increase due to source correlation from the COSQ III scheme is significantly more than in the COSQ II scheme; in the noiseless channel there is a 3.6 dB gain over the COSQ II method when the sources are highly correlated. The gain is not as high as the noise increases; however the COSQ III always performs better. The next set of graphs, in Figure 3.13 shows the COSQ III scheme for rate 2. The gain in the noiseless channel due to source correlation is 11 dB, which is substantial. The largest increase in performance occurs around $\rho = 0.75$. The gain is consistent throughout the varying channel noise. For the CSNR = 15 dB and CSNR = 10 dB channels the increase is not strictly monotonical, with dips in the gain around source correlation with value 0.6. Next, Figure 3.14 examines the gain due to the soft-decoding granularity, where $q = 2$. There is an increase in performance compared to the $q = 1$ system in Figure 3.12, especially in the speed at which the SDR increases. Finally, Figure 3.15 has graphs demonstrating the increase in gain due to channel memory. Once again the channel correlation parameter is $\text{cor} = 0.9$. As with the system with $q = 2$, the gain starts increasing at lower source correlation, ρ around 0.6, instead of 0.75.

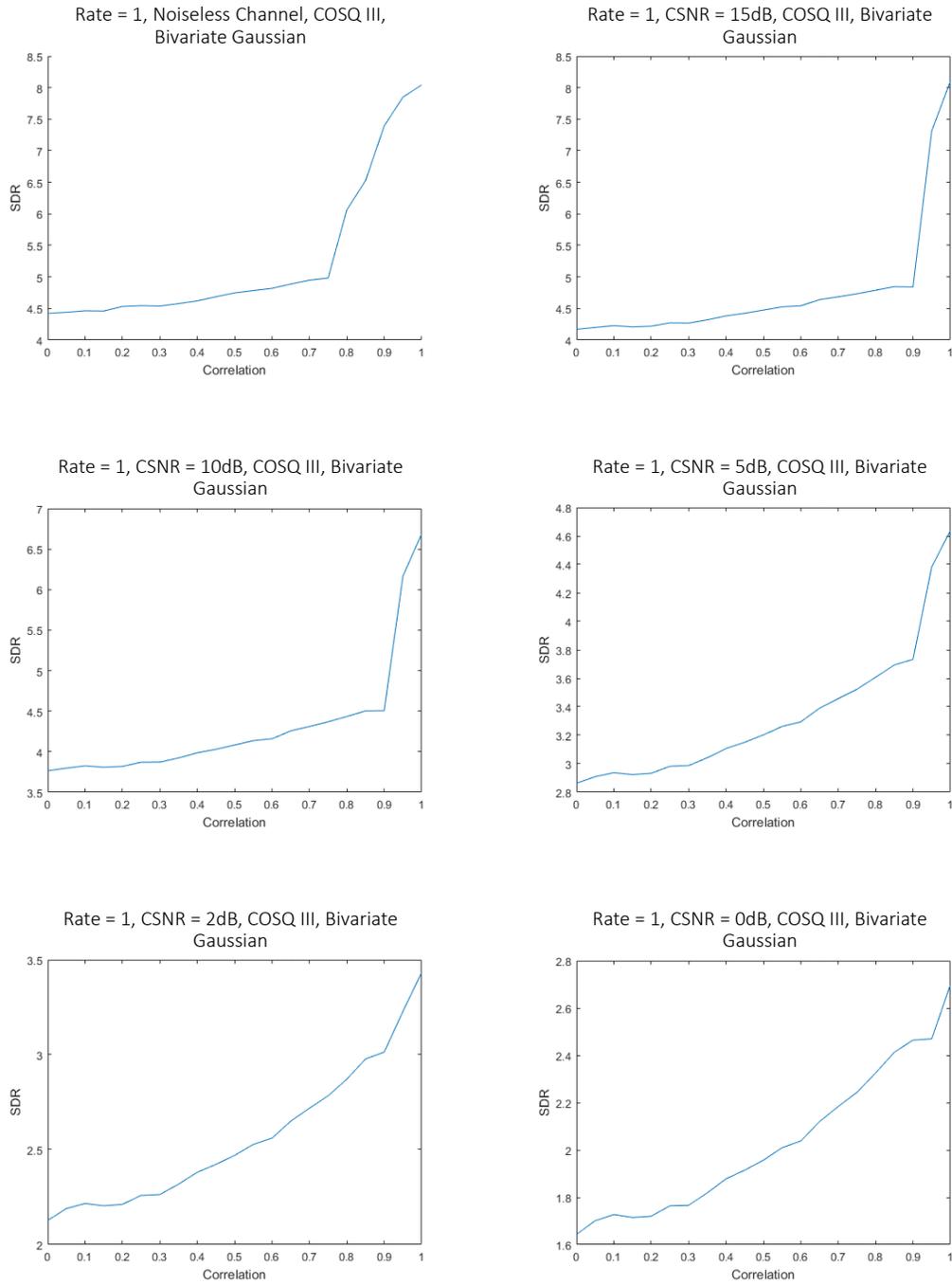


Figure 3.12: Two User Rate 1 COSQ III with Varying CSNR, Gaussian

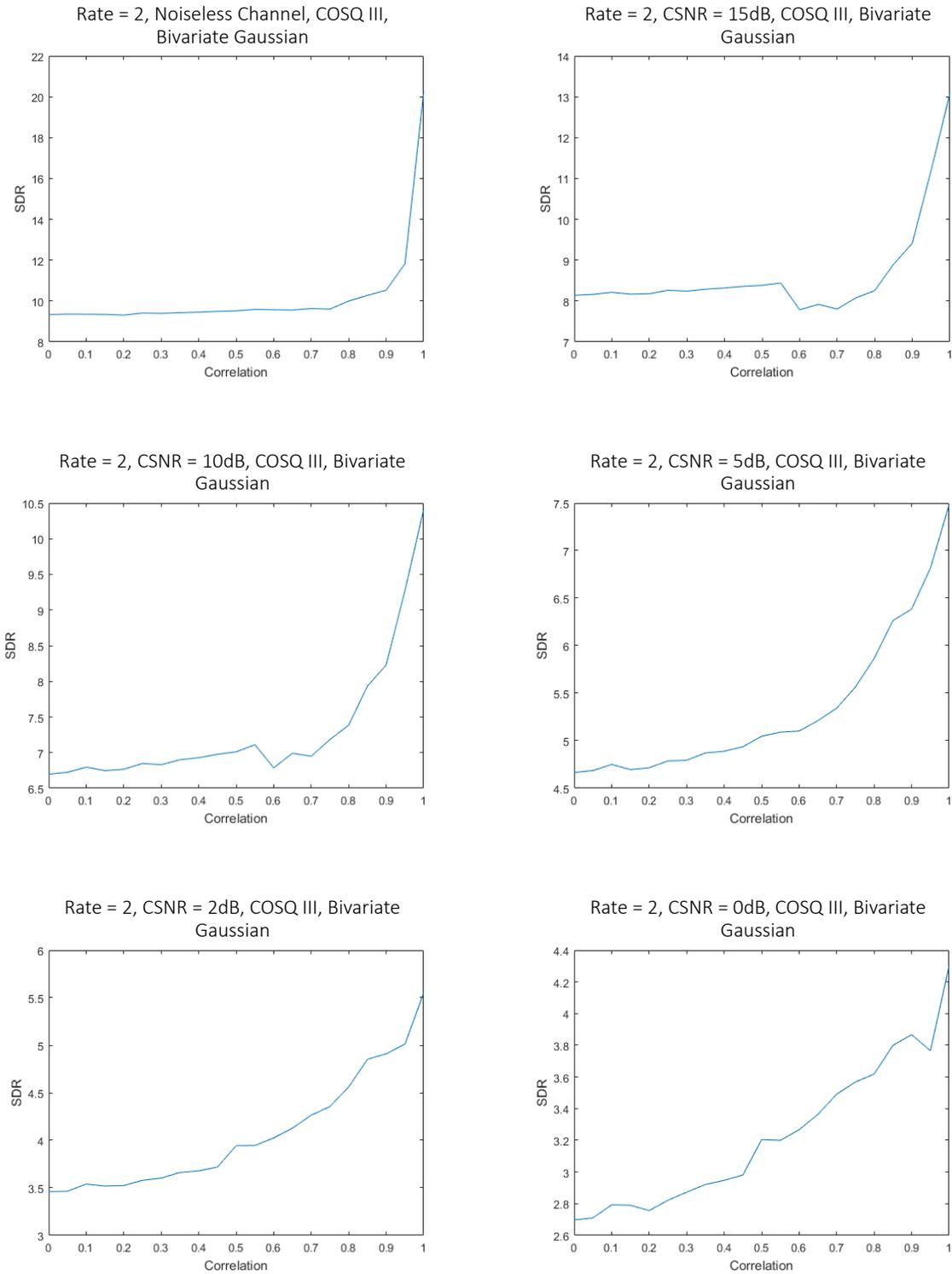


Figure 3.13: Two User Rate 2 COSQ III with Varying CSNR, Gaussian

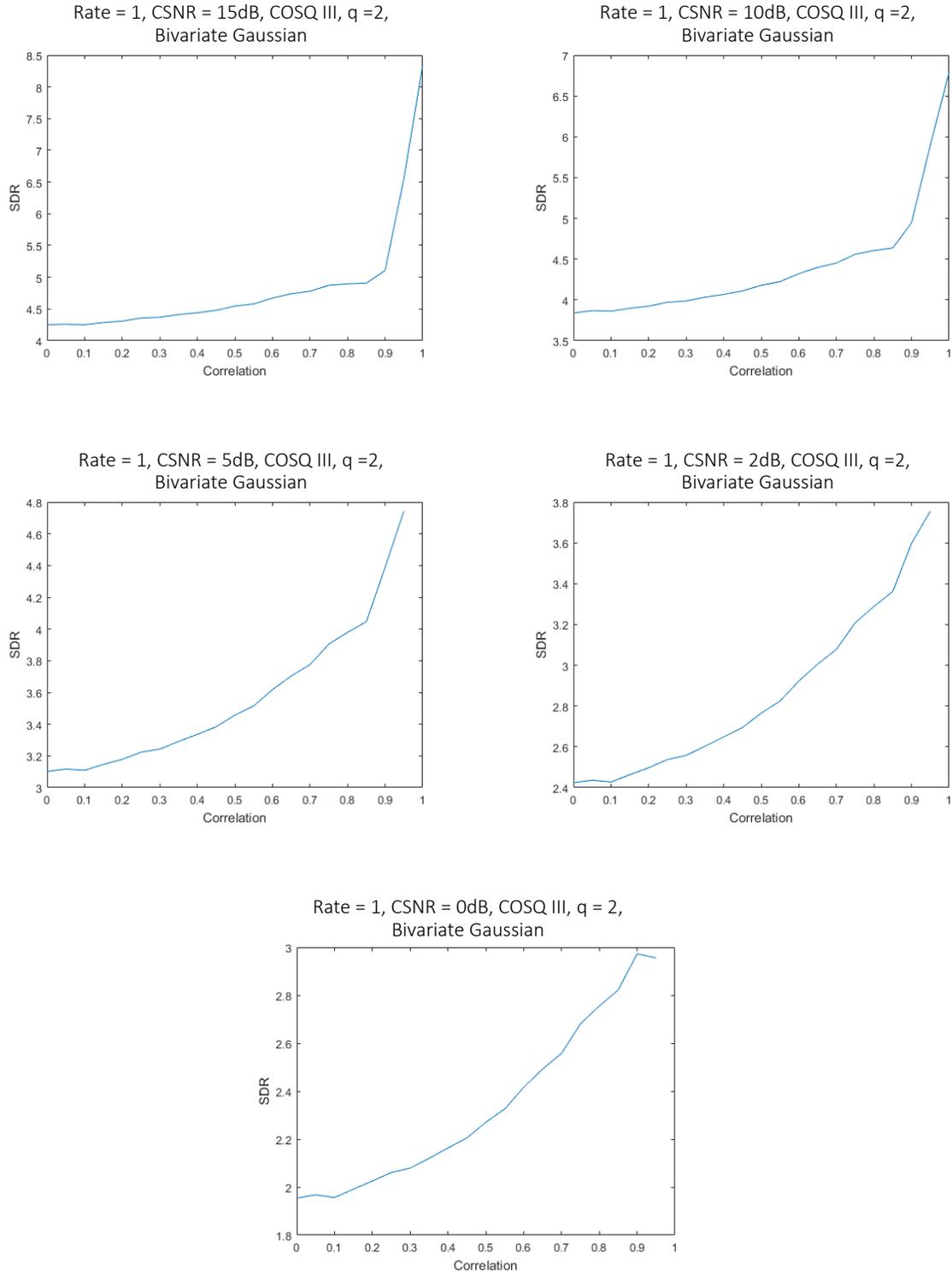


Figure 3.14: Two User Rate 1 COSQ III with Varying CSNR, $q = 2$, Gaussian

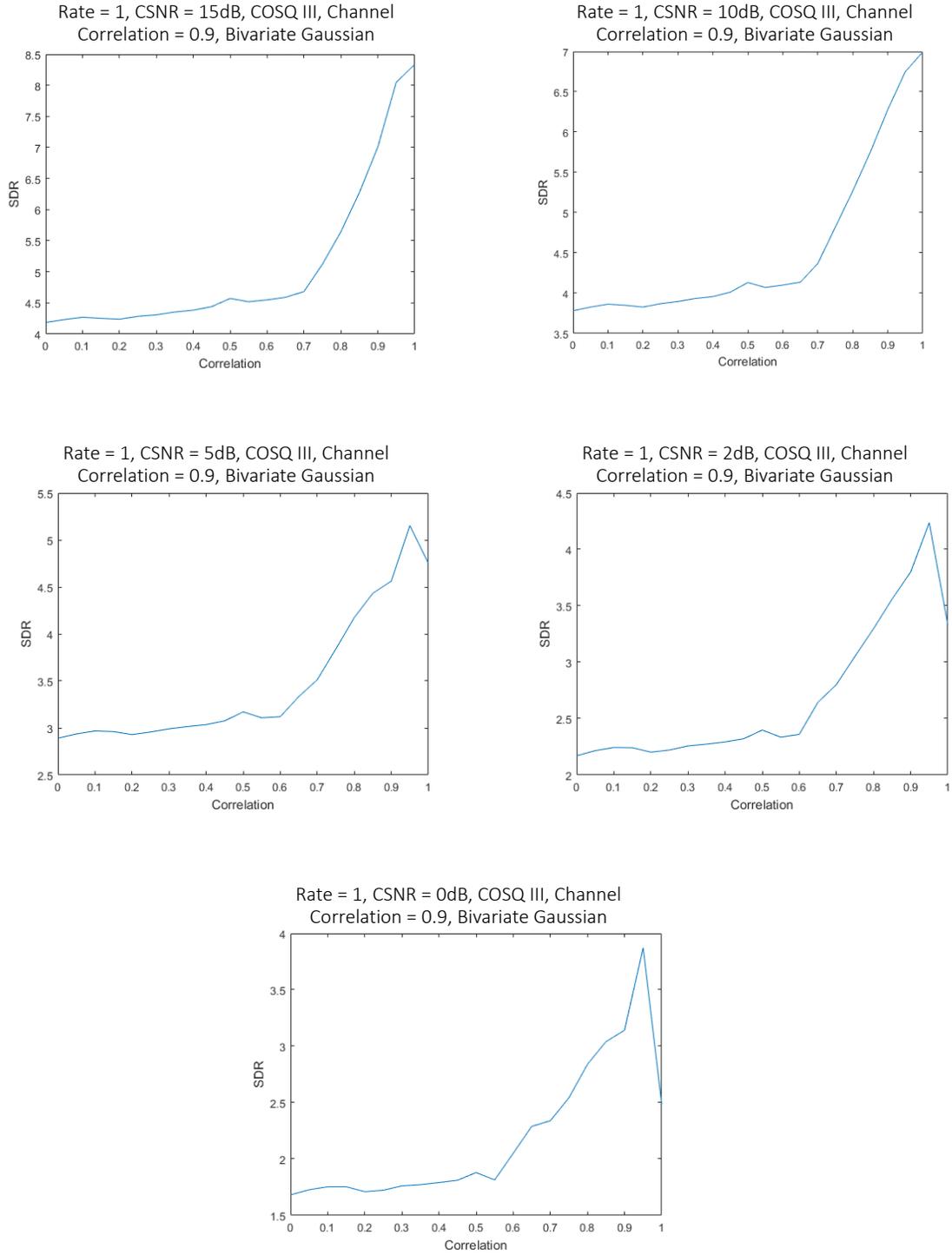


Figure 3.15: Two User Rate 1 COSQ III with Varying CSNR, Channel Correlation $\text{cor} = 0.9$, Gaussian

Chapter 4

Application to Correlated Images sent over Orthogonal MACs

In this chapter the single user COSQ I and the two user COSQ II are used to send correlated images over the orthogonal MAC. Given the results in the previous section, an increase in performance is expected when the images are highly correlated and when there is high noise in the channel. First the method used for the image transmission is introduced and then results will be presented.

4.1 Image Transmission System

The two user joint-source channel coding scheme is now applied to correlated, grayscale images. A grayscale image is represented by a two dimensional array of pixel values. Each pixel value ranges from 0 to 255 in base two. Colour images are based on the same principle; however instead of each pixel value containing one value, it contains three values. The first value represents the intensity of red in the pixel, the second value the intensity of green, and the third value the intensity of blue. As before the three values can vary from 0 to 255 as binary representations. An algorithm that works for grayscale images can be generalized for use with color images. For this reason, the rest of this thesis assumes the use of grayscale images. In a single image, the binary pixel values, which take on values between 0 and 255 in base two, are highly correlated. The *discrete cosine transform* (DCT) is used to decorrelate

the pixel values by mapping the pixels into the frequency domain. The discrete cosine transform is a linear transform that takes values from the spatial domain to the frequency domain. The one dimensional transform is given by:

$$C_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad \text{for } k = 0, \dots, N - 1$$

where x_n is the value undergoing the transform and N is the number of values being transformed. The two dimensional transform (2D DCT) is given by,

$$C_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right],$$

where x_{n_1, n_2} is the pixel value, and N_1 and N_2 are the sizes of the block in their respective dimension. As the name suggests, the DCT is a finite sum of cosine functions at different frequencies. When using the 2D DCT, the output from the transform is a two dimensional array of coefficients. The coefficients in the upper left hand corner contains the highest amount of energy at the lowest frequency. The top leftmost coefficient is called the DC coefficient and can be modeled using the Gaussian distribution. The other coefficients are all called the AC coefficients and are modeled by Laplacian distribution [23]. The human eye can not easily discern high frequencies and as such the AC coefficients become less important as the bottom right hand corner of the array of coefficients is approached. The zig-zag pattern shown in Figure 4.1 below demonstrates the order of importance of the coefficients for reconstructing pixel values in a reliable fashion.

The DCT is performed on 8×8 pixel blocks of the image. The coefficients in the top left corner are required to reconstruct the image in the spatial domain. The rates used to encode the coefficients are given by a bit allocation table, which will be further discussed in the next section. Once the DC and AC coefficients that are being kept have been generated, they are

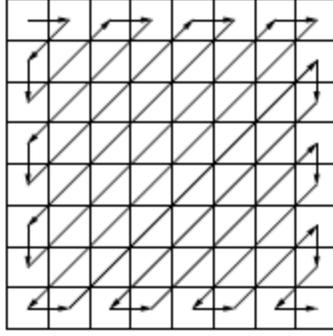


Figure 4.1: Zig Zag Pattern [23]

normalized. In this case the normalization method used is:

$$X_{\text{norm}} = \frac{X - \mu_X}{\sqrt{\sigma_X^2}},$$

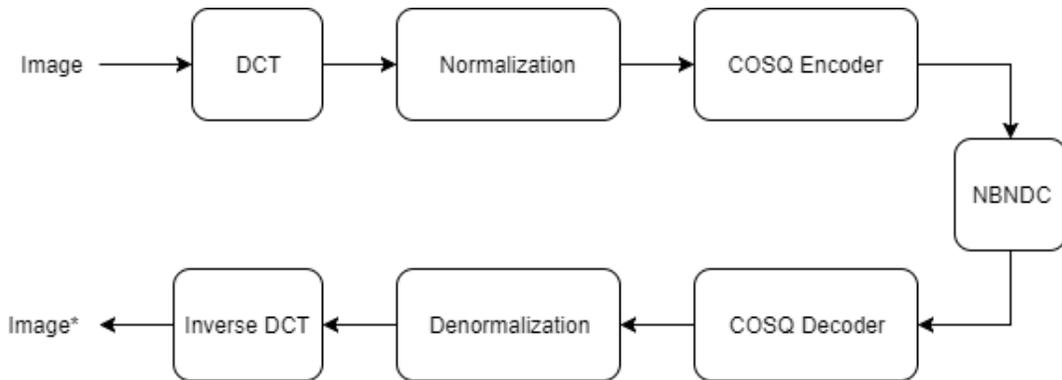


Figure 4.2: Image Transmission System

where μ_X is the mean of the data, and σ_X^2 the variance. The normalized data is now encoded, the DC and AC coefficients are encoded with different rates, determined by the bit allocation table. The binary indices are sent across the NBNDC and the images are decoded. The decoding methods used for this section are the benchmark COSQ I method, and the COSQ II method. Both decoded values are denormalized by reversing the above described normalization methods, and the inverse DCT is performed on the 8x8 blocks that have been reconstructed. Any AC coefficient that was not sent across the channel is filled in with a

0. Once the image has been recreated, the peak signal-to-noise ratio can be calculated by comparing the reconstructed image with the one that was sent across the channel. This value, in dB, gives an idea of the quality of the reconstruction, the higher the value, the closer to the encoded image. It is calculated as:

$$\text{PSNR}_{\text{dB}} = 10 \log_{10} \frac{255^2}{\sum_{i=1}^{512} \sum_{j=1}^{512} (X_{ij} - \hat{X}_{ij})^2},$$

where X_{ij} is the compressed image sent over the channel, and \hat{X}_{ij} is the received, reconstructed image. The images are 512x512 pixels in size, and the maximum value of the pixels is 255.

When applying the algorithm to two images, it needs to be ensured that the densities of the two sets of corresponding DC, and AC coefficients are jointly distributed. Images were used with a source correlation $\rho \in (0.7, 1)$ for the DC coefficient.

4.2 Fixed Bit Allocation

As mentioned previously, another method of joint-source channel coding, is unequal error protection, where the values sent across the channel are sorted by importance and the more important values are encoded with a higher rate. Bit allocation is one such method. The DCT coefficients generated from the 2D DCT hold varying degrees of information and thus they can be encoded differently. The DC coefficient is generally encoded with a higher rate and the AC coefficients with a lower rate, or not at all. For this experiment rates 3 and 4 are used for the DC coefficient and only the first two AC coefficients are encoded, using rate 2.

Table 4.1: Fixed Bit Allocation Tables for DCT

BPP = 0.125								BPP = 0.109							
4	2	0	0	0	0	0	0	3	2	0	0	0	0	0	0
2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3 Correlated Images

The images used for the image transmission system were satellite images of the Arctic taken every 24 hours. The average source correlation between the DC coefficients of consecutive images was $\rho_{DC} = 0.75$ and, $\rho_{AC_1} = 0.58$ and $\rho_{AC_2} = 0.51$ for the first and second AC coefficients respectively. For a large class of images the DC coefficients are well modelled by the Gaussian distribution. The histogram shown in Figure 4.5, shows that the DC coefficients from 22 Arctic images do not however behave like a bivariate Gaussian distribution. For this reason, instead of using codebooks trained on the bivariate Gaussian distribution, the codebooks were trained directly on the DC and AC coefficients from a set of Arctic images.

To train the codebook, 22 images from April 2nd to April 24th, 2017 were used. The DCT was performed on 8x8 pixel blocks, so that each 512x512 image yielded 4,096 DC coefficients, and AC coefficients (of which the first two were kept). These codebooks, from COSQ I and COSQ II were used to encode and decode images from May 2017.

Histogram of Correlated DC Coefficients from 22 Arctic Images

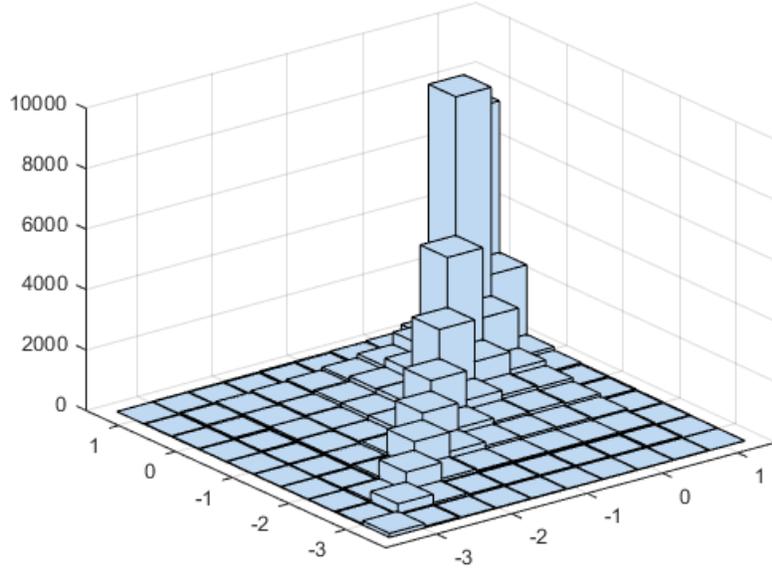


Figure 4.3: Histogram of Correlated DC Coefficients

Histogram of First Correlated AC Coefficients from 22 Arctic Images

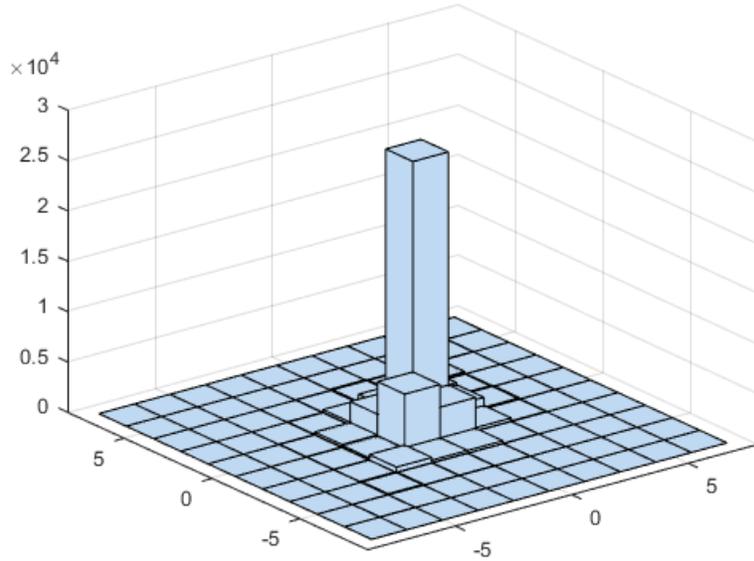


Figure 4.4: Histogram of First AC Coefficients

Histogram of Second Correlated AC Coefficients from 22 Arctic Images

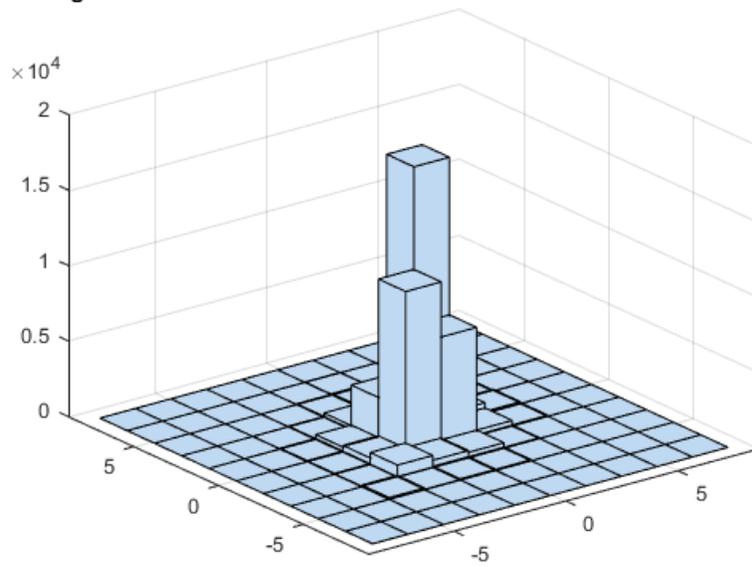


Figure 4.5: Histogram of Second AC Coefficients

4.4 Results

Table 4.2 shows the SDR in dB of the COSQ I and COSQ II methods trained on the DC coefficients from 22 Arctic images, for the memoryless channel and the channel with $\text{cor} = 0.9$. All channels used have $M = 1, \alpha = 1$. For both rates, and all channel conditions, the COSQ II method performs better than the COSQ I method. The gain is larger on average for the channel with memory, with a few exceptions. The average gain due to the method for the memoryless channel and rate 3 is 0.70 dB, and for rate 4 is 0.81 dB. For the channel with memory, $\text{cor} = 0.9$, the average gain due to the method is 0.97dB for rate 3 and 0.85 dB for rate 4. Table 4.3 shows similar SDR results for the first and second AC coefficient at rate 2. In this case the gain due to the scheme is higher for the memoryless channel, with the average gain for the memoryless channel being 0.23 dB, and 0.09 dB for the channel with memory. The gain due to the scheme for the AC coefficients is lower, as expected, due to the fact that the correlation ρ_{AC_1} and ρ_{AC_2} is lower than for the DC coefficients.

Table 4.2: SDR Results for DC Coefficients

Channel Correlation, cor	Method	CSNR in dB				
		15	10	5	2	0
Rate 3						
0	COSQ I	12.68	9.96	6.92	5.19	4.09
	COSQ II	13.05	10.52	7.66	6.09	4.99
0.9	COSQ I	13.54	13.48	10.87	9.38	8.52
	COSQ II	15.53	13.73	11.30	9.94	9.16
Rate 4						
0	COSQ I	14.66	11.68	8.27	6.30	5.03
	COSQ II	15.25	12.33	9.21	7.24	5.97
0.9	COSQ I	18.60	16.09	13.05	11.35	10.39
	COSQ II	18.84	16.74	14.04	12.49	11.60

For the image results, images from outside the training set were sent across channels of

Table 4.3: SDR Results for AC Coefficients, Rate 2

Channel Correlation, cor	Method	CSNR in dB				
		15	10	5	2	0
First AC Coefficient						
0	COSQ I	5.50	4.30	2.84	2.04	1.57
	COSQ II	5.71	4.63	3.14	2.29	1.80
0.9	COSQ I	6.14	5.25	3.66	2.60	1.95
	COSQ II	6.22	5.34	3.77	2.70	2.05
Second AC Coefficient						
0	COSQ I	5.58	4.39	2.88	2.07	1.60
	COSQ II	5.73	4.62	3.15	2.29	1.78
0.9	COSQ I	6.21	5.32	3.72	2.63	1.97
	COSQ II	6.29	5.40	3.79	2.71	2.04

varying noise, compressed with either 0.125 *bits per pixel* (bpp) or 0.109 bpp using the fixed bit allocation from above. For consistency, the results shown are from the May 1st and May 2nd images; however the averages quoted are from multiple dates.

Figure 4.6 shows the original consecutive images used, and below, in Figure 4.7 are the images compressed at 0.125 bpp with the encoder designed for the 2 dB channel. The next set of Figures, 4.8 and 4.9 show the images received from a 2 dB channel. The top two are decoded independently, while the bottom were decoded jointly. The average performance gain from the jointly decoded scheme was 0.44 dB. Figures 4.10 and 4.11 show independently and jointly decoded images received from a CSNR = 0 dB channel, and encoded at rate 0.109. These conditions resulted in a 0.52 dB gain on average from using the jointly decoded scheme. The next two sets of images use a channel with memory; the correlation parameter for the channel $\text{cor} = 0.9$. The first set of Figures 4.12 and 4.13, use a 0 dB channels, at 0.125 bpp compression and 0.109 bpp respectively. The average gain from using the joint decoding was 1.1 dB and 1 dB respectively. Finally, to increase the correlation between the images being sent, a noisy image was sent alongside the original May 1st image. Figures 4.14 and 4.15 show the original images and the 0.125 bpp compressed images that would be sent across the 15 dB channel. The following two pages show images compressed at 0.125 bpp, the first set

is sent across a 15 dB channel, while the second is sent across a 0 dB channel. The average gain from the jointly decoded scheme was 1.21 dB and 1.94 dB respectively. The final two pages of results show images compressed at 0.109 dB and sent over 15 dB and 5 dB channels. The average gain from the jointly decoded scheme was 1 dB and 2.11 dB respectively. These results support the findings from above, where bivariate Gaussian and Laplacian data was used. The largest performance gains are obtained at higher source correlation and higher channel noise. It is interesting to note that the independently decoded scheme has more detail in the edges of the images; however there are more artifacts. When the channel had memory, the performance gain due to the joint decoding was also higher than when the memoryless channel was used. After the image results, Tables 4.4, 4.5 and 4.6 show all the PSNR averages for the two methods, the memory and memoryless channel, and the differently correlated images.



Image 1 (Arctic on May 1st, 2017)



Image 2 (Arctic on May 2nd, 2017)

Figure 4.6: Original, Correlated images

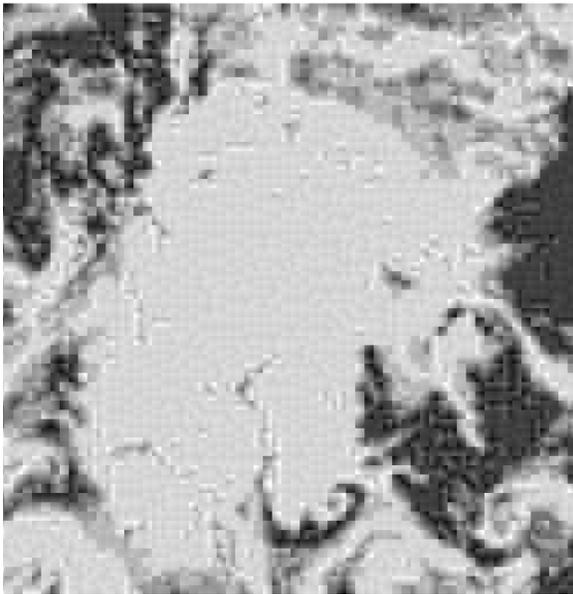


Image 1 Compressed

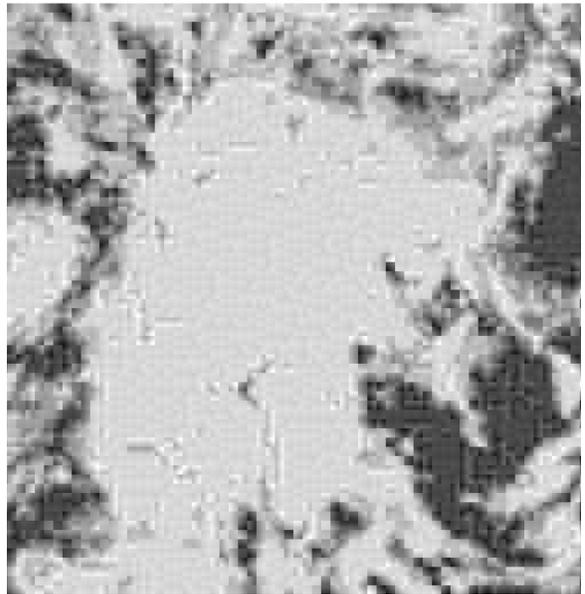


Image 2 Compressed

Figure 4.7: Images Compressed with 0.125 bpp for a Channel with CSNR = 2 dB



Image 1, PSNR = 19.35 dB



Image 2, PSNR = 19.33 dB

Figure 4.8: COSQ I Decoded Images, CSNR = 2 dB, 0.125 bpp

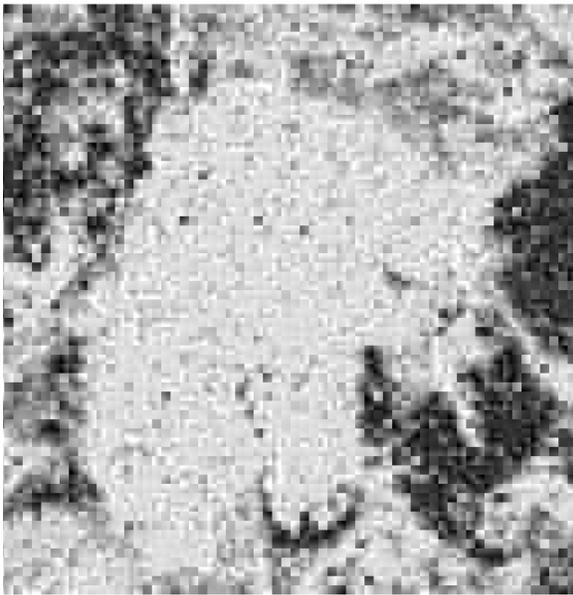


Image 1, PSNR = 19.95 dB



Image 2, PSNR = 19.85 dB

Figure 4.9: COSQ II Decoded Images, CSNR = 2 dB, 0.125 bpp



Image 1, PSNR = 14.85 dB



Image 2, PSNR = 14.74 dB

Figure 4.10: COSQ I Decoded Images, CSNR = 0 dB, 0.109 bpp



Image 1, PSNR = 15.17 dB



Image 2, PSNR = 15.21 dB

Figure 4.11: COSQ II Decoded Images, CSNR = 0 dB, 0.109 bpp



Image 1, PSNR = 18.55 dB

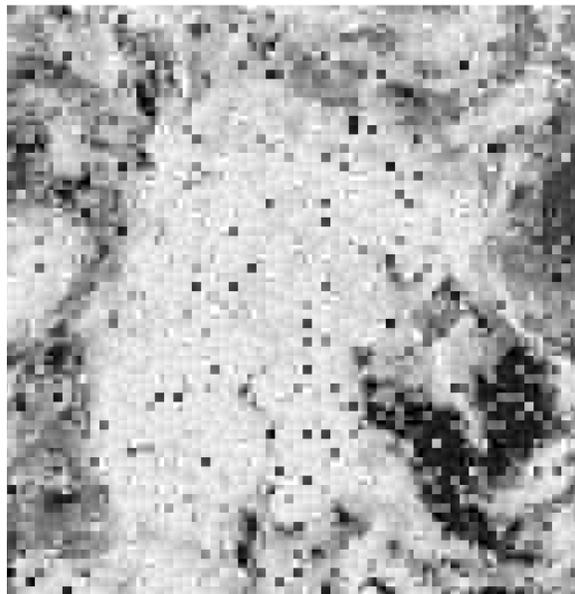


Image 2, PSNR = 19.14 dB

Figure 4.12: COSQ I Decoded Images, CSNR = 0 dB, 0.125 bpp, Channel cor = 0.9



Image 1, PSNR = 19.71 dB

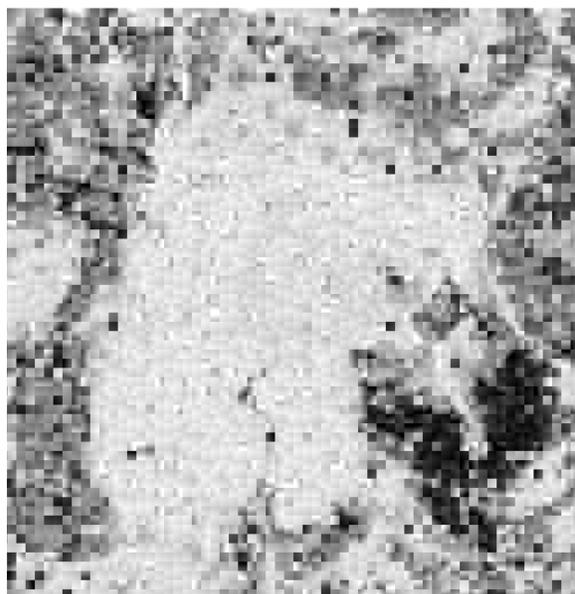


Image 2, PSNR = 21.01 dB

Figure 4.13: COSQ II Decoded Images, CSNR = 0 dB, 0.125 bpp, Channel cor = 0.9



Image 1, PSNR = 17.21 dB



Image 2, PSNR = 17.44 dB

Figure 4.14: COSQ I Decoded Images, CSNR = 0 dB, 0.109 bpp, Channel cor = 0.9



Image 1, PSNR = 18.02 dB

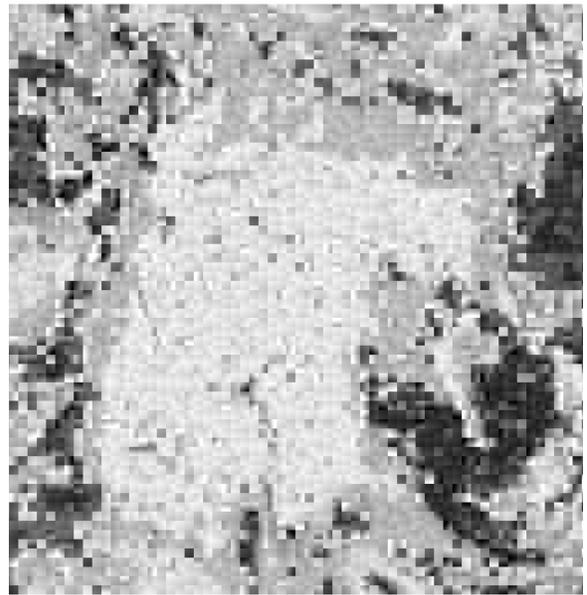


Image 2, PSNR = 18.43 dB

Figure 4.15: COSQ II Decoded Images, CSNR = 0 dB, 0.109 bpp, Channel cor = 0.9



Image 1 (Arctic on May 1st, 2017)

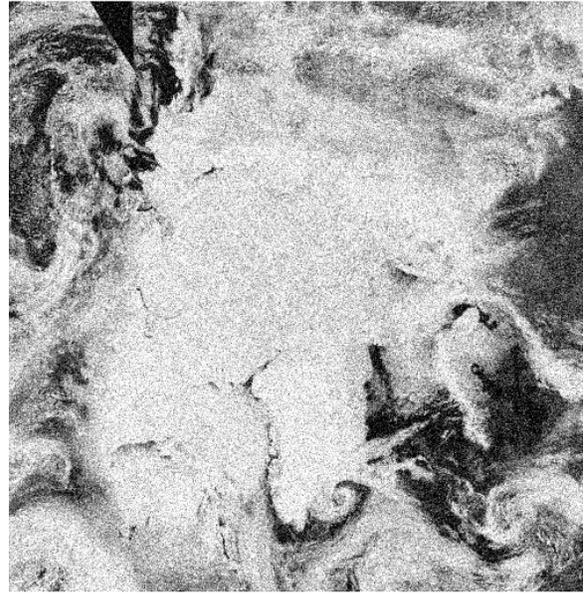


Image 2 (Arctic on May 1st, 2017 with Noise)

Figure 4.16: Original, Correlated Images

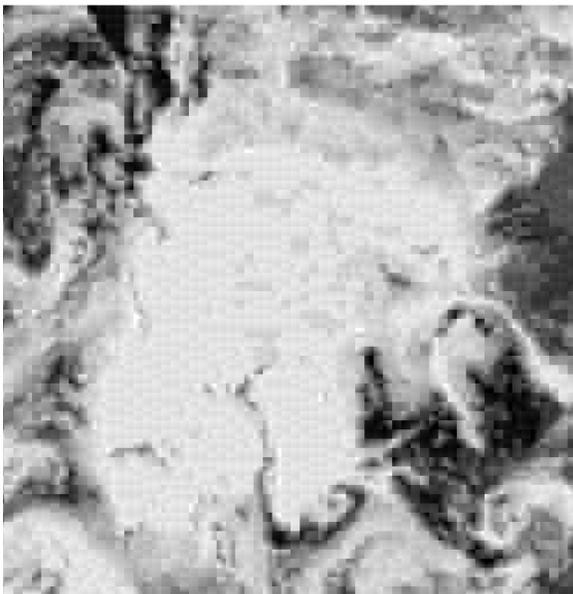


Image 1 Compressed

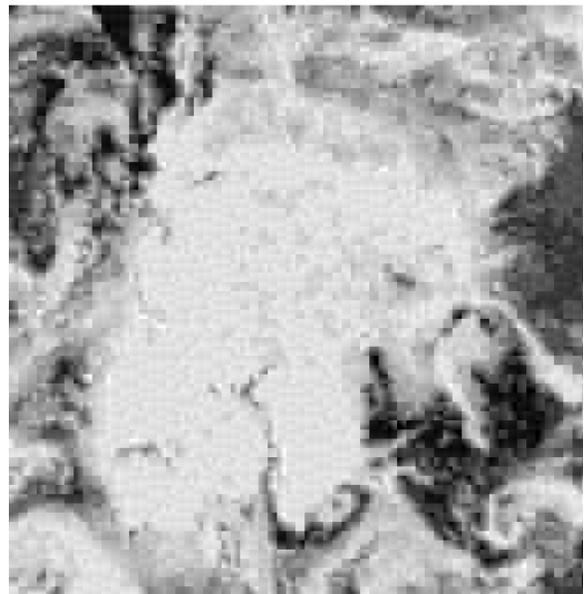


Image 2 Compressed

Figure 4.17: Images Compressed with 0.125 bpp for a Channel with CSNR = 15 dB

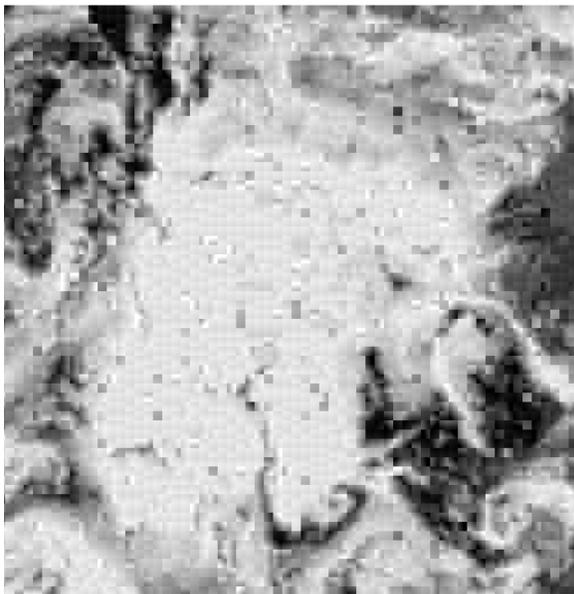


Image 1, PSNR = 27.67 dB



Image 2, PSNR = 28.79 dB

Figure 4.18: COSQ I Decoded Images, CSNR = 15 dB, 0.125 bpp



Image 1, PSNR = 29.22 dB



Image 2, PSNR = 30.04 dB

Figure 4.19: COSQ II Decoded Images, CSNR = 15 dB, 0.125 bpp



Image 1, PSNR = 16.31 dB



Image 2, PSNR = 16.17 dB

Figure 4.20: COSQ I Decoded Images, CSNR = 0 dB, 0.125 bpp

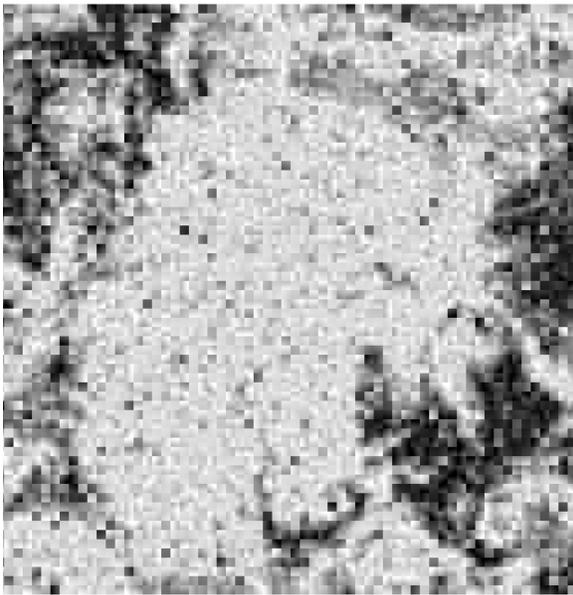


Image 1, PSNR = 18.16 dB

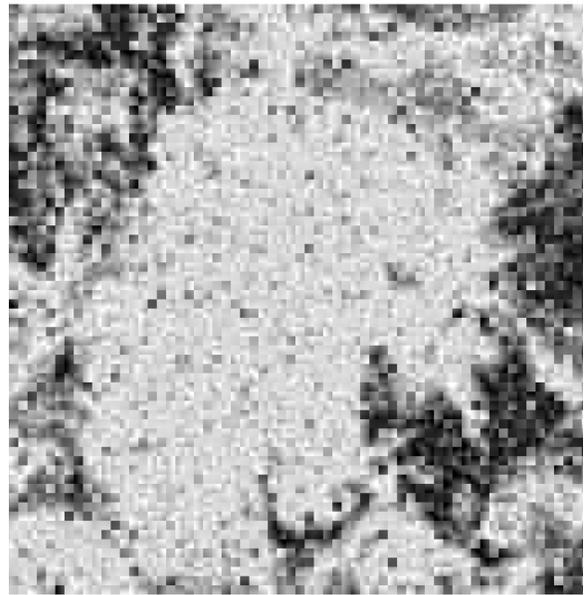


Image 2, PSNR = 18.15 dB

Figure 4.21: COSQ II Decoded Images, CSNR = 0 dB, 0.125 bpp

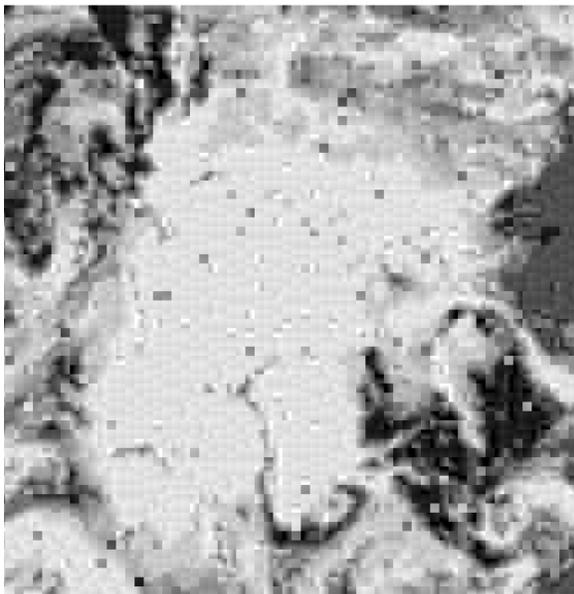


Image 1, PSNR = 26.66 dB

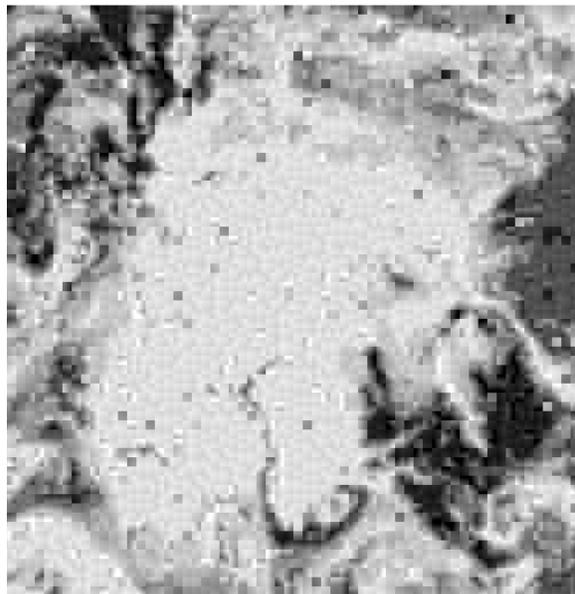


Image 2, PSNR = 27.50 dB

Figure 4.22: COSQ I Decoded Images, CSNR = 15 dB, 0.109 bpp

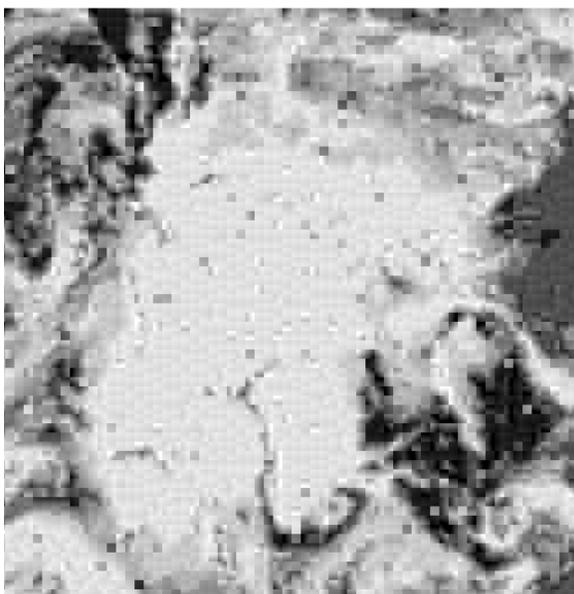


Image 1, PSNR = 27.60 dB

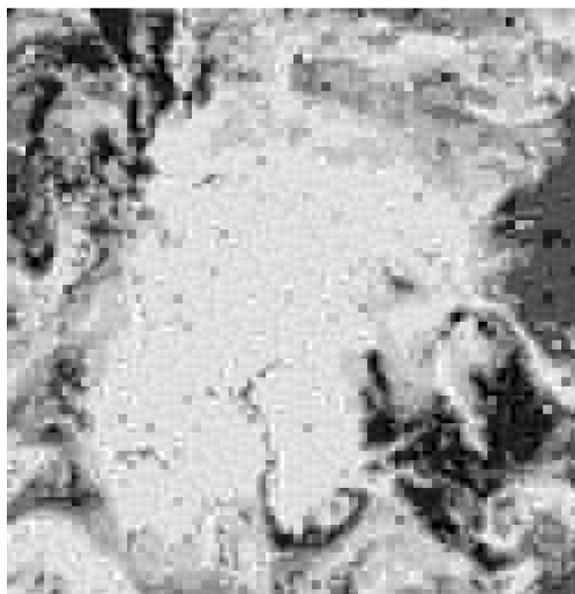


Image 2, PSNR = 28.26 dB

Figure 4.23: COSQ II Decoded Images, CSNR = 15 dB, 0.109 bpp

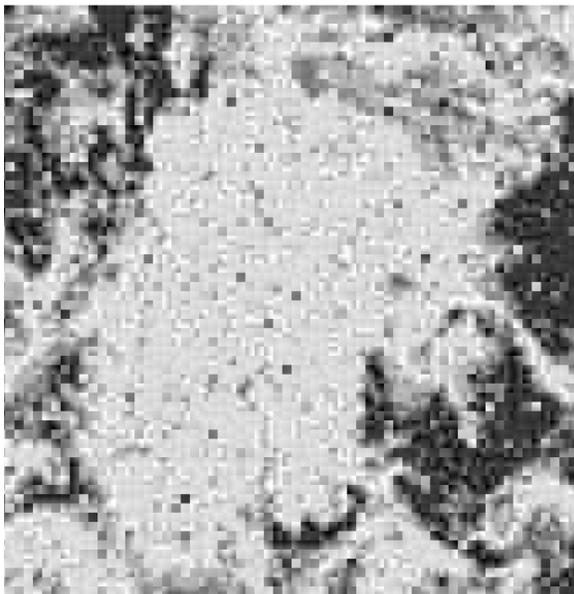


Image 1, PSNR = 20.62 dB

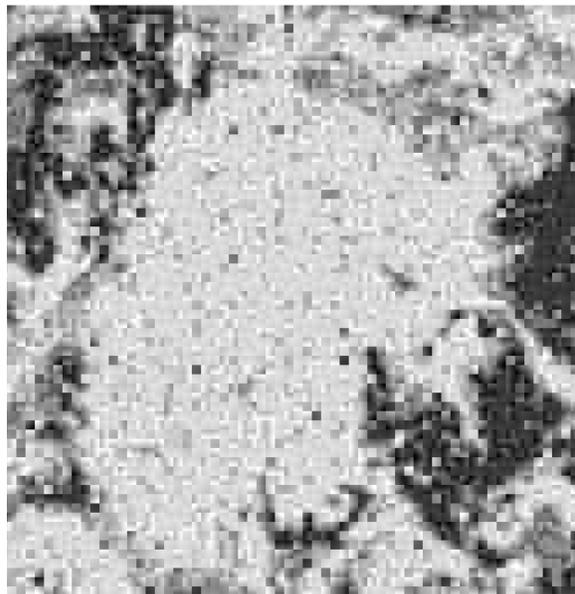


Image 2, PSNR = 20.43 dB

Figure 4.24: COSQ I Decoded Images, CSNR = 5 dB, 0.109 bpp

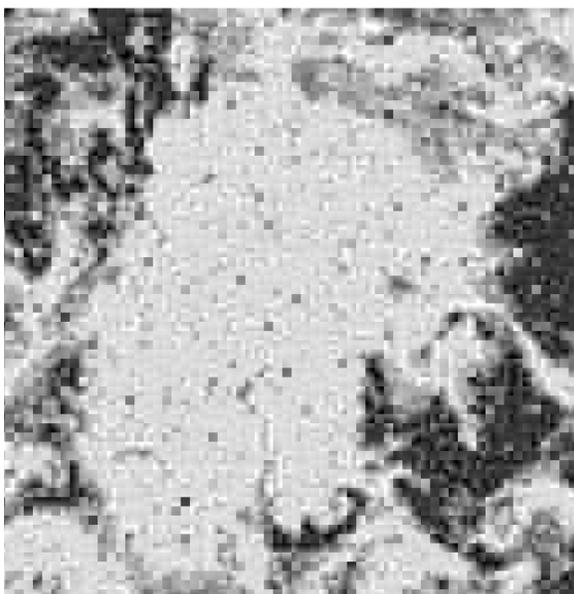


Image 1, PSNR = 22.54 dB

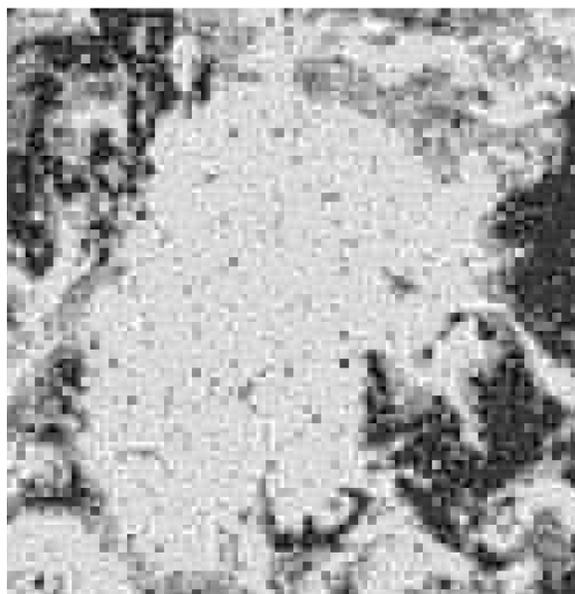


Image 2, PSNR = 22.50 dB

Figure 4.25: COSQ II Decoded Images, CSNR = 5 dB, 0.109 bpp

Table 4.4: PSNR Results Consecutive May Images

Method	User	CSNR in dB				
		15	10	5	2	0
Compressed at 0.125 bpp						
COSQ I	Image 1	28.20	24.84	21.09	19.35	16.13
	Image 2	27.42	24.83	20.81	19.33	16.65
COSQ II	Image 1	28.34	25.18	21.41	19.95	16.80
	Image 2	27.61	25.10	21.15	19.85	16.95
Compressed at 0.109 bpp						
COSQ I	Image 1	26.65	22.96	20.36	17.11	14.85
	Image 2	26.69	23.22	20.54	17.54	14.74
COSQ II	Image 1	26.87	23.22	20.68	17.68	15.17
	Image 2	27.14	23.47	20.81	17.63	15.21

Table 4.5: PSNR Results Consecutive May Images Sent Over Channel with $\text{cor} = 0.9$

Method	User	CSNR in dB				
		15	10	5	2	0
Compressed at 0.125 bpp						
COSQ I	Image 1	32.72	29.6	25.25	23.26	18.55
	Image 2	32.57	30.53	25.35	22.52	19.14
COSQ II	Image 1	32.91	29.76	25.61	23.92	19.71
	Image 2	32.69	30.65	25.87	23.23	21.01
Compressed at 0.109 bpp						
COSQ I	Image 1	31.53	28.31	19.30	18.46	17.21
	Image 2	31.65	28.04	19.12	18.15	17.44
COSQ II	Image 1	31.66	28.53	21.31	20.61	18.02
	Image 2	31.72	28.19	20.94	19.65	18.43

Table 4.6: PSNR Results for May Images sent with Noisy Version

Method	User	CSNR in dB				
		15	10	5	2	0
Compressed at 0.125 bpp						
COSQ I	Image 1	27.67	22.93	21.15	19.68	16.31
	Image 2	28.79	22.92	21.38	19.60	16.17
COSQ II	Image 1	29.22	24.81	21.55	20.91	18.16
	Image 2	30.04	24.78	22.02	21.18	18.15
Compressed at 0.109 bpp						
COSQ I	Image 1	26.66	23.02	20.62	17.16	15.14
	Image 2	27.50	23.39	20.43	17.27	15.08
COSQ II	Image 1	27.60	23.82	22.54	18.68	16.41
	Image 2	28.26	24.21	22.50	18.83	16.55

Chapter 5

Conclusion and Future Work

Necessary conditions for the optimality for COSQ for a two-user MAC system were derived in this thesis. Three coding schemes were introduced, the independently optimized encoder and decoder scheme, COSQ I, the independently optimized encoder, jointly optimized decoder scheme, COSQ II, and the jointly optimized encoder and decoder scheme, COSQ III. The methods were designed to take the advantage of the source correlation and the channel characteristics of the NBNDC-QB. After the algorithms were fully derived, the results showed that for source data from both the bivariate Gaussian and Laplacian distribution, the COSQ II performed better, or equal to the single user scheme at all rates and levels of channel noise. It was proved that for rate 1, the COSQ II performs equally when decoding the bivariate Gaussian with correlation, $\rho = 0$ and $\rho = 1$. The performance for independent sources is also equivalent to the single user scheme. All three COSQ methods successfully exploited the channel memory, with a maximum gain of 4 dB in the COSQ II scheme. The COSQ III had the highest gain due to source correlation, with the rate 2 encoder, designed for the noiseless channel increasing from 9.33 dB to 20.23 dB. When comparing the complexity and storage of the three methods, the COSQ III is shown to be the most complex of the three schemes. To further investigate the usefulness of the schemes, the independent and dependent decoders were compared on images. The images were consecutive images of the Arctic throughout the months of April and May, 2017. First the images from April were processed using the DCT, and then the COSQ I and II codebook trained on the DC coefficient and the first two AC

coefficients. The May 2017 images were independently encoded and decoded using the two different codebooks. The COSQ II scheme consistently resulted in better performance, both in terms of PSNR and also visually, the method resulted in fewer artifacts in the final images. There are a variety of ways this research can be continued. The obvious continuation is to add more users; however the optimality conditions of the schemes imply that each user would exponentially increase the complexity and storage requirements. As a result, it is worthy investigating how to simplify the encoder term of the COSQ III such that the encoding complexity can be reduced. One way would be to calculate the expected values for the second user for each partition of the first users codebook, and store these values in a lookup table. This would only reduce the linear complexity of the system; however the exponential complexity, due to the size of the codebook, would still be present. To mitigate this, the second user could be designed with a low rate, to boost the performance of the first user. Finally, an interesting result would be using the COSQ III scheme to encode and decode the images from Chapter 4. This would require either an estimation of the distribution of the DC and AC coefficients from these images, or an empirical estimation of the expected values for the encoder.

Bibliography

- [1] C. E. Shannon “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. Jul., pp. 379-423, 623-656, 1948.
- [2] C. E. Shannon “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec.*, 4(142-163), 1959.
- [3] N. Phamdo “Quantization over Discrete Noisy Channels under Complexity Constraints,” *PhD thesis, Department of Electrical Engineering, University of Maryland*, 1993.
- [4] J. W. Modestino, D. G. Daut and A. L. Vickers, “Combined source-channel coding of images using the block cosine transform,” *IEEE Trans. Comm.*, vol. COM-27, pp. 1644-1659, Nov. 1979.
- [5] S. P. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, pp. 129-137, Mar. 1982.
- [6] J. Max, “Quantizing for minimum distortion,” *IRE Trans. Inf. Theory*, vol. IT-6, pp. 7-12, Mar. 1960.
- [7] A. Kurtenback and P. Wintz, “Quantizing for noisy channels,” *IEEE Trans. Commun. Technology*, vol. 17, pp. 291-302, Apr. 1969.
- [8] N. Farvardin and V. Vaishampayan, “On the performance and complexity of channel-optimized vector quantizers”, *IEEE Trans. Inf. Theory*, vol. 37, pp. 155-160, Jan. 1991.
- [9] N. Farvardin, “A Study of vector quantization for noisy channels,” *IEEE Trans. Inf. Theory*, vol. 36, pp. 799-809, Jul. 1990.

- [10] Y. Zhong, F. Alajaji, and 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, Apr. 2006.
- [11] V. Kostina and S. Verdú, "Lossy joint source-channel coding in the finite block-length regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2545-2575, May 2013.
- [12] S. Shahidi, F. Alajaji, and T. Linder, "MAP detection and robust lossy coding over soft-decision correlated fading channels," *IEEE Trans. Veh. Technology*, vol. 62, no. 7, pp. 3175-3187, Sep. 2013.
- [13] F. Behnamfar, F. Alajaji, and T. Linder, "Channel-optimized quantization with soft-decision demodulation for space-time orthogonal block-coded channels," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3935-3946, Oct. 2006.
- [14] N. Phamdo and F. Alajaji, "Soft-decision demodulation design for COVQ over white, colored, and ISI Gaussian channels," *IEEE Trans. Comm.*, vol. 48, no. 9, pp. 1499-1506, Sep. 2000.
- [15] J. Singh, O. Dabeer, and U. Madhow, "On the limits of communication with low-precision analog-to-digital conversion at the receiver," *IEEE Trans. Comm.*, vol. 57, no. 12, pp. 3629-3639, Dec. 2009.
- [16] N. Wernersson and M. Skoglund, "On source decoding based on finite-bandwidth soft information," in *Proc. IEEE Int. Symp. Inf. Theory, Adelaide, SA, Australia*, Oct. 2005, pp. 87-91.
- [17] J. Kron, F. Alajaji, and M. Skoglund, "Low-delay joint source-channel mappings for the Gaussian MAC," *IEEE Trans. Inf. Theory*, vol. 18, no. 2, pp. 249-252, Feb. 2014.

- [18] P. Floor, A. Kim, N. Wernersson, T. Ramstad, M. Skoglund, and I. Balasingham, “Zero-delay joint source-channel coding for a bivariate Gaussian on a Gaussian MAC,” *IEEE Trans. Comm.*, vol. 6, no. 10, pp. 3091-3102, Oct. 2012.
- [19] D. Gündüz, E. Erkip, A. Goldsmith, and H. V. Poor, “Source and channel coding for correlated sources over multiuser channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 3927 - 3944 Sep. 2009.
- [20] T. M. Cover, A. E. Gamal, and M. Salehi, “Multiple access channels with arbitrarily correlated sources,” *IEEE Trans. Inf. Theory*, vol. 26, no. 6, Nov. 1980.
- [21] S. P. Beheshti, “MAP Decoding of Correlated Sources over Soft-Decision Orthogonal Multiple Access Fading Channels with Memory,” *MSc Thesis, Department of Mathematics and Statistics, Queen’s University*, Sep. 2014.
- [22] S. Shahidi, “Robust Lossy Source Coding for Correlated Fading Channels,” *MSc Thesis, Department of Mathematics and Statistics, Queen’s University*, Sep. 2011.
- [23] J. Cheng, “Channel optimized quantization of images over binary channels with memory,” *Department of Mathematics and Statistics, Queen’s University*, Sep. 1997.
- [24] T. M. Cover and J. A. Thomas “Elements of Information Theory 2nd ed.” Hoboken, New Jersey: *Prentice-Hall Inc*, 2002.
- [25] M. Mushkin and I. Bar-David, “Capacity and coding for the Gilbert-Elliott channels,” *IEEE Trans. Inf. Theory*, vol. 35, no. 6, Nov. 1989.
- [26] C. Pimentel, F. Alajaji, P. Melo, “A discrete queue-based model for capturing memory and soft-decision information in correlated fading channels,” *IEEE Trans. on Comm.*, vol. 60, no.6 Jan. 2012.
- [27] L. Zhong, F. Alajaji, and G. Takahara, “A model for correlated Rician fading channels based on a finite queue,” *IEEE Trans. Veh. Tech.*, vol. 57, no. 57, Jan. 2008.

- [28] S. Kotz, T. J. Kozubowski, K. Podgórski, “The Laplace Distribution and Generalizations - A Revisit with New Applications,” *Birkhäuser*, Jan. 2001.
- [29] V. P. Lineswala and J. N. Patel “JPEG image compression and transmission over wireless channel” *Advances in Computing, Control, and Telecommunication Technologies, International Conference on*, pp. 643-645, 2009.
- [30] H. E. Saffar “Channel Optimized Vector Quantization: Iterative Design Algorithms,” *MSc Thesis, Department of Mathematics and Statistics, Queen’s University*, Aug. 2008.
- [31] T. Eltoft, T. Kim, and T. Lee, “On the multivariate Laplace distribution,” *IEEE Sig. Proc. Letters*, vol. 13, no. 5, May 2006.
- [32] R. Chandramouli and N. Ranganathan, “Computing the bivariate Gaussian probability integral,” *IEEE Sig. Proc. Letters*, vol. 6, no. 6, Jun. 1999.
- [33] D. P. Bertsekas and J. N. Tsitsiklis, “The bivariate normal distribution,” *Introduction to Probability, 1st Edition*, Section 4.7, 2002.
- [34] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Tras. Comm.*, vol. 28, no. 1, Jan. 1980.
- [35] G. C. Zhu and F. Alajaji, “Soft-decision COVQ for turbo-coded AWGN and Rayleigh fading channels,” *IEEE Comm. Letters* vol. 5, no. 6, Jun. 2001.
- [36] F. Behnamfar, F. Alajaji, and T. Linder, “Image transmission over the Polya channel via channel-optimized quantization,” *IEEE Trans. Sig. Proc.*, Feb. 2005.
- [37] J. Gai, “A Computational Study of the Bivariate Normal Probability Function,” *MASc Thesis, Department of Mathematics and Statistics, Queen’s University*, Dec. 2001.
- [38] V. A. Vaishampayan and N. Farvardin, “Optimal block cosine transform image coding for noisy channels,” *IEEE Trans. on Comm.*, vol. 38, no. 3, Mar. 1990.

- [39] N. Farvardin and V. A. Vaishampayan, "Optimal quantizer design for noisy channels: an approach to combined source-channel coding," *IEEE Trans. Inf. Theory*, vol. 33, no. 6, Nov. 1987.
- [40] F. Alajaji and T. Fuja, "A communication channel modeled on contagion," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, Nov. 1994.