FASTER DESIGN OF ROBUST BINARY JOINT WATERMARKING AND SCALAR QUANTIZATION UNDER ADDITIVE GAUSSIAN ATTACKS

FASTER DESIGN OF ROBUST BINARY JOINT WATERMARKING AND SCALAR QUANTIZATION UNDER ADDITIVE GAUSSIAN ATTACKS

By HAN ZHANG, B. ENG.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfilment of the Requirements for the Degree of Master of Applied Science

McMaster University © Copyright by Han Zhang, May 2022

McMaster University

MASTER OF APPLIED SCIENCE (2022)

Hamilton, Ontario (Electrical and Computer Engineering)

TITLE: Faster Design of Robust Binary Joint Watermarking and Scalar Quantization under Additive Gaussian Attacks

AUTHOR: Han Zhang B.Eng (Beijing University of Posts and Telecommunications)

SUPERVISOR: Dr. Sorina Dumitrescu

NUMBER OF PAGES: xiii, 60

Abstract

This thesis investigates the problem of optimal design of binary joint watermarking and scalar quantization (JWSQ) systems that are robust under additive Gaussian attacks. A binary JWSQ system consists of two quantizers with disjoint codebooks. The joint quantization and embedding are performed by choosing the quantizer corresponding to the embedded message. The optimal JWSQ design for both fixed-rate and variable-rate cases was considered in the past, but the solution approaches exhibited high computational complexity.

In this thesis, we propose faster binary JWSQ design algorithms for both the fixed-rate and variable-rate scenarios. We achieve the speed up by mapping the corresponding optimization problem to a minimum weight path problem in a certain weighted directed acyclic graph (with a constraint on the length of the path in the fixed-rate case). For this mapping to be possible we discretize the quantizer space and use an approximation for the probability of decoding error. The proposed solution algorithms have $O(LN^3)$ and $O(N^4)$ time complexity in the two cases respectively, where N is the size of discretized source alphabet, and in the fixed-rate scenario L is the number of cells in each quantizer.

The effectiveness of the proposed designs is assessed through extensive experiments on a Gaussian source. Our results show that our algorithms are able to achieve performance very close to the prior existing schemes, but only at a small fraction of their running time.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Dr. Sorina Dumitrescu who supervised and supported me throughout my graduate life. She is intelligent and has a great passion for research. Her guidance has helped me to become more disciplined and logical in my thesis. I always feel so lucky to become her student.

I also would like to thank committee members, Dr. Ratnasingham Tharmarasa and Dr. Jun Chen for reading my thesis and providing valuable suggestions for my work.

I am very grateful to my beloved family and friends. They have given me selfless support and love, making me more fearless in my exploration of the world. Each time I felt anxious, they always stand with me and give me power.

Everyone I have mentioned has helped me improve, grow, and become a better person.

Contents

List of Figures

List of Tables

List of Abbreviations

AWGN	Additive White Gaussian Noise
DNR	Distortion Noise Ratio
FJWSQ	Fixed-rate Joint Watermarking and Scalar Quantization
JWC	Joint Watermarking and Compression
JWSQ	Joint Watermarking and Scalar Quantization
MD	Minimum Distance (decoder)
MWP	Minimum-Weight Path
QIM	Quantization Index Modulation
SWC	Seperate Watermarking and Compression
VJWSQ	Variable-rate Joint Watermarking and Scalar Quantization
WDAG	Weighted Directed Acyclic Graph

List of Notation

m	The watermark symbol
s	The host signal sample
Q^m	A scalar quantizer for watermark symbol m
C_i^m	The i-th cell for quantizer Q^m
T^m	The set of thresholds for quantizer Q^m
Y^m	The code book for quantizer Q^m
L_m	The length of Y^m
D	The quantization distortion
P_e	The decoding probability of error
$P_{j,e}^m$	The conditional bit error probability given m and the fact that the host signal lies in the quantization cell C_j^m
R	The rate of a JWSQ system
$\varDelta_i$	The distance between two reconstruction points
$\varDelta_{min}$	The minimum accepted distance between two reconstruction points
S	The source alphabet

V	The vertex set $\{ab \mid 0 \leq a \leq b \leq N-1, s_b - s_a \geq \mathit{\Delta}_{min}\} \cup \{00, NN\}$
E	The edge set $\{(ab, bc) \mid ab, bc \in V\}$
G	The weighted directed acyclic graph with the vertex set V and the edge set E
$w(ab, bc)$	The weight function of G
V'	The vertex set $\{abc \mid 0 \leq a \leq b \leq c \leq N, a < c\}$
E'	The edge set $\{(abc, bcd) \mid 0 \leq a \leq b \leq c \leq d \leq N, a < d\}$
E^*	The general edge set $E' \setminus \{(000, 00a), (aNN, NNN)\}$
G'	The weighted directed acyclic graph with the vertex set V' and the edge set E'
$w(abc, bcd)$	The weight function of G'
C_{ac}	One cell of the quantizer $[s_a, s_c)$
$\gamma(a, c)$	The reconstruction value of cell C_{ac}

1 Introduction

In this thesis, we propose algorithms for the design of binary joint watermarking and compression schemes based on scalar quantization that are robust under additive Gaussian attacks. In this chapter we present the motivation of the problem, briefly review the prior work and outline the contribution of the thesis.

In Section 1.1, we introduce the general description and applications of digital watermarking. The related literature about joint watermarking and compression is briefly reviewed in Section 1.2. Finally, the contribution and organization of this thesis are summarized in Section 1.3.

1.1 Digital Watermarking and Applications

With the development of the Internet and of wireless networks, people can access, download and transfer multimedia data easily. This enriches our lives, while also introducing some potential risks. Malicious tampering and illegal reproduction of multimedia data affect the rights of authors and catch public's attention. To stop theft and tampering, it is necessary to use techniques that protect the copyright of content authors.

One common method to achieve this is cryptography. The sender encrypts the content before delivery, then the encrypted file can be sent over the network. Only legal receivers have access to the decryption key and can therefore decrypt the file. This method protects the content during transit, however, but risk still exists after the decryption. For instance, a pirate can purchase the legal decryption key and distribute the content illegally.

Digital watermarking can be used for copyright protection after decryption. Digital watermarking refers to embedding a signal, called watermark, in the original content, which is called the “host signal”. The result is called “watermarked signal”. The embedded information could be any meaningful mark, such as the author’s name, the company’s logo or the indicator of a patent.

The watermark must satisfy certain properties in order to fulfill its purpose. Namely, it should not cause degradation to the host signal. In other words, it has to be imperceptible, which means that it can not be detected or noticed by human perception system, and can only be extracted by a well-designed detector. On the other hand, the watermark should survive if the watermarked signal is subjected to some benign transformations or to malicious attacks. Examples of benign transformations are signal compression or other processing required by the application (for instance image cropping).

There are three major applications for digital watermarking, owner identification, transaction tracking and content authentication. Next we give some details about each of them.

- Owner Identification

  We used to use textual copyright to protect the content. However, textual copyright has a lot of limitations such as being easy to remove from a document when it is copied. Additionally, it can cover only a portion of the image. A famous example of the ineffectiveness of this technique is the image Lena. The image is cropped from Playboy magazine but without the textual copyright. This image was wildly used and nobody knew where it came from for a long time.

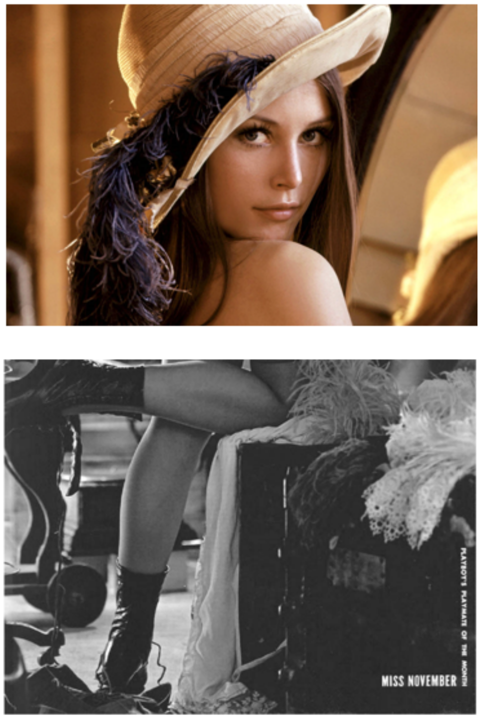


Figure 1.1: Lena image is cropped without the textual copyright

But watermarks can be made both imperceptible and inseparable from the original work that contains them, which rends them more useful for owner identification. There is an application called Digimarc's detector. This detector is widely distributed by bundling with a famous image processing software, Photoshop. Authors are encouraged to embed a digital watermark into their work and save it to the database. Once Digimarc's detector detects a watermark, the contact information of the original owner will appear. It provides an efficient way for creators to find out who they should contact to use an image.

- Transaction Tracking

  In this application, the owner can place a different watermark in each copy so that they can track which copy is leased. An example is the following. In 2004, Technicolor, a division of Thompson, used video watermarking technology licensed from Philips to individually watermark each of the 5,803 voting members' Oscar screeners. After distributing these copies, pirated videos of the films appeared on the Internet. Analysis of these pirated copies revealed that the original source material had been Oscar screeners provided to the actor Carmine Caridi. [11]

- Content Authentication

  In the previous applications, a watermark is designed to be robust. In this application, the algorithms used are called fragile watermarking. Any watermarks embedded into the original content are sensitive to intentional modification. We call this type of watermark an authentication mark. Another field is called semi-fragile watermark, which survives small transformations, such as lossy compression.

1.2 Joint Watermarking and Compression

In most applications, the watermarked signal has to be compressed for a more efficient use of resources when storing or transmitting it. The most common way to do this is by performing the watermarking and compression separately. An alternative approach is to perform them jointly. The latter variant calls for methods that jointly design watermarking and compression systems. We will call such schemes joint watermarking and compression (JWC) schemes. As shown in [4], JWC can achieve better performance than separate watermarking and compression (SWC). Although there are plenty of research works in the digital watermarking field, not too much research has been done in the JWC area. Some ad hoc JWC algorithms were proposed in [12], [13] and [14] for images, audio and video seperately. [15] analyze the minimum achievable composite rate for both the public and the private versions of the JWC problem.

Optimized practical designs for JWC were proposed in [4, 5]. In both works scalar quantization is used for watermarking and compression and one bit per sample is embedded, thus they are binary JWC schemes. Additionally, the robustness against additive Gaussian attacks is maximized subject to constraints on the quantization distortion and rate. In the sequel we will use the phrase "joint watermarking and scalar quantization" (JWSQ) for JWC based on scalar quantization. The main difference between the JWSQ schemes of [4] and [5] is that the fixed rate is considered in the former and the variable rate case is addressed in the latter. In both cases, the constrained optimization problem is converted to the unconstrained problem of minimizing the Lagrangian. The solution algorithm is an iterative algorithm which alternates between optimizing the encoder while keeping the decoder fixed and optimizing the decoder while keeping the encoder fixed.

For the information-theoretic perspective of JWC, [7] provides the capacity definition of a fixed discrete memoryless channel with the public decoder. Information hiding capacity is obtained in [8] whether or not the decoder knows the host data. When the covertext and the attack channel are Gaussian, the upper bound of the achievable rate region is discussed in [10].

1.3 Thesis Contribution and Organization

One drawback of the JWSQ design algorithms of [4, 5] is their high computational complexity. In this work we address this shortcoming by proposing faster design algorithms for binary JWSQ ensuring robustness under addittive Gaussian attacks. We consider both the fixed rate and variable rate cases. In each case, we model the related optimization problems to a minimum weight path problem (possibly with constraints on the number of edges) in a certain weighted directed acyclic graph. In order to achieve this, we resort to some approximations (for instance by discretizing the quantization space) which might lead to suboptimal designs. We show empirically that the sacrifice in performance in comparison to [4] and [5] is very small, while the speed up is considerable.

This thesis is structured as follows. The next chapter includes the background knowledge related to our problem including digital watermarking techniques, evaluation, and information-theoretic perspective of JWC. In Chapter 3, previous works on binary JWSQ are reviewed. In Chapter 4, we present the proposed JWSQ systems in both fixed-rate and variable-rate scenarios. Chapter 5 presents the simulation results and their discussion. Chapter 6 concludes this thesis and lists some potential directions for future work.

2 Background

This chapter reviews some basic knowledge regrading digital watermarking. We first review the common models for digital watermarking in Section 2.1, followed by the watermarking evaluation criteria in Section 2.2. Some results from the information-theoretic perspective of joint compression and digital watermarking are presented in Section 2.3. Finally, Section 2.4 concludes the chapter.

2.1 Models of Digital Watermarking

Many digital watermarking techniques have been developed based on the traditional communications model, which is illustrated in Figure 2.1. In this model, the input message consists of the source data we want to transmit. The encoder maps the source data to a codeword that can be transmitted over the channel. There are several types of channel noise, onre of them being the additive noise, which is commonly chosen when considering watermarking. The goal of the decoder is to reconstruct the source input reliably. Generally, an encoder contains two parts: the source encoder and the channel encoder. The source encoder maps the sequence of source samples to a sequence of bits. The channel encoder converts the sequence of bits into a physical signal.

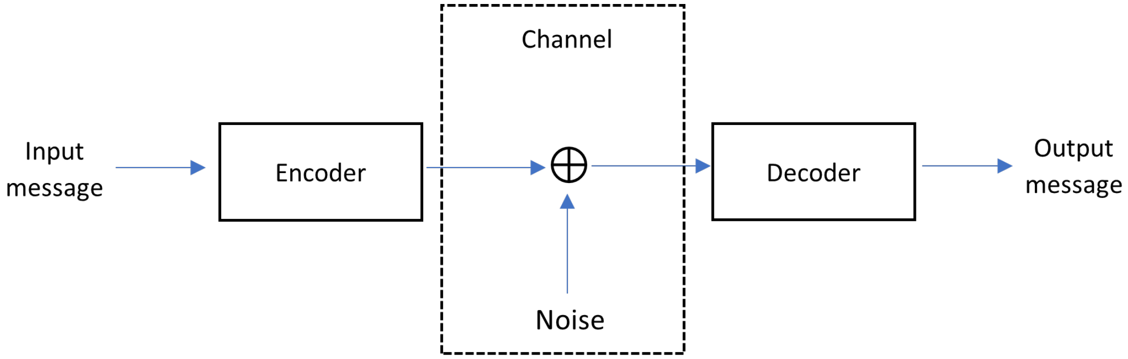


Figure 2.1: Traditional communications model

One basic model for digital watermarking is depicted in Figure 2.2. In that model, the watermark is the input message that we want to transmit, which is added to the host signal. In that way, we can treat the host signal as a noise of the channel. However, this model cannot be used in all algorithms since the watermark and host signal are not independent in many scenarios. To solve this problem, another model is developed which uses the host signal in the encoder. This model is a communication system with side information at the watermark transmitter, as illustrated in Figure 2.3. The watermarking techniques considered in this thesis are based on this model.

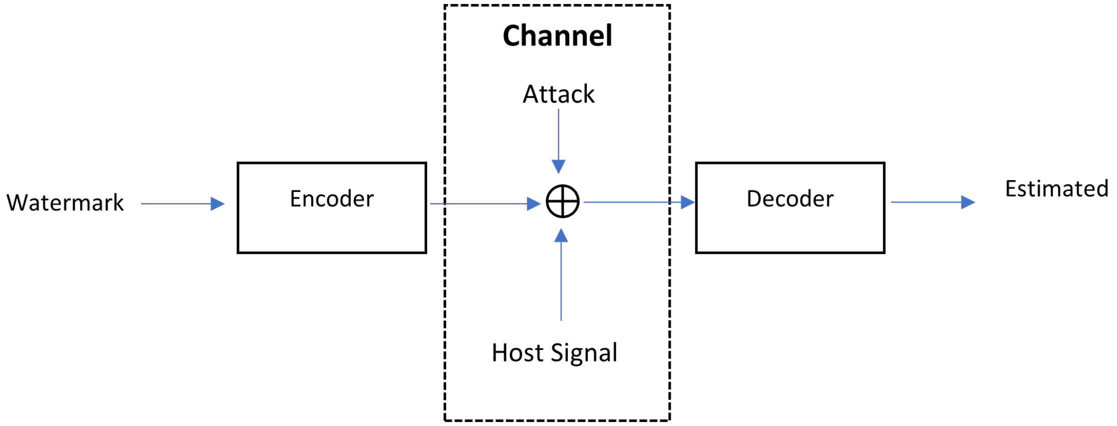


Figure 2.2: Basic digital watermarking model

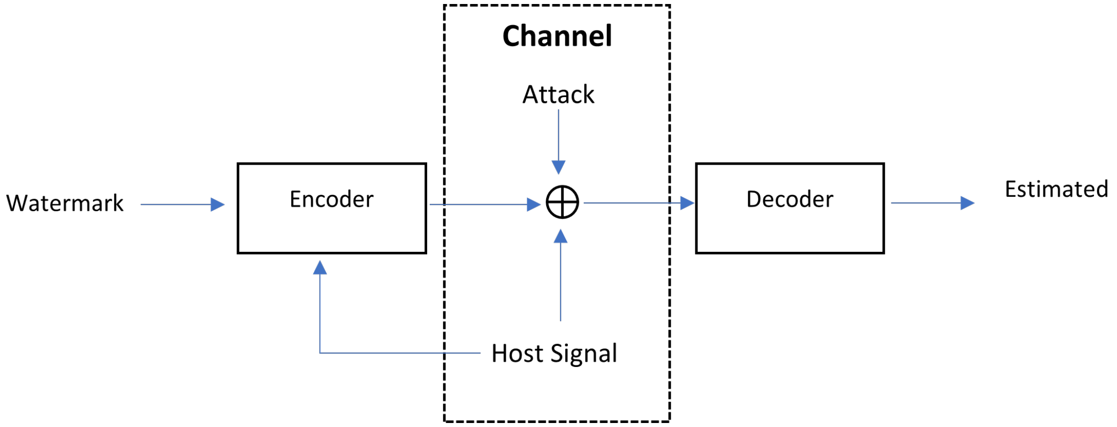


Figure 2.3: Digital watermarking model with side information

2.2 Digital Watermarking System Evaluation

To evaluate a watermarking system, we need to consider different aspects. Generally, improvement in one aspect everything else being equal means a better digital watermarking system. However, different applications require different properties. In this section, several commonly desired properties are introduced.

2.2.1 Fidelity

Fidelity refers to the perceptual similarity between the original signal and the watermarked signal. Different applications have different tolerance for fidelity. For signals that are transmitted in a low-quality channel, fidelity is not important compared to the channel degradations. On the contrary, the system requires high fidelity watermarks when signals are very high in quality, such as HDTV audio and video. In this thesis, the fidelity is evaluated as the squared error distortion D. The higher D is, the lower the fidelity is.

2.2.2 Data Payload

Data payload refers to the number of watermark bits embedded into the host signal. Generally, we consider the normalized version, i.e., the number of embedded bits divided by the number of samples of the host signal. This is termed “embedding bitrate” (and is measured in bits per sample). A watermark is called N-bit watermark if it encodes N bits. The required data payload may vary a lot for different applications. For example, copy control applications only require 4-8 bits while television broadcast monitoring applications require at least 24 bits.

2.2.3 Effectiveness

The effectiveness is the probability that a watermark is embedded successfully at the encoder. Although 100% effectiveness is desired, ensuring this might lead to other properties being sacrificed. Correspondingly, we can improve other properties by reducing effectiveness if other properties are more important in some applications.

2.2.4 Robustness

Robustness means the ability to detect the watermark at the decoder in the presence of attacks . A watermarked signal can be attacked by attackers or be processed by common signal processing operations (e.g. spatial filtering, lossy compression, digital-to-analog conversion, etc.) or be added with a channel noise. Any of these activities will cause a distortion to the watermarked signal. It is desired to guarantee that the hidden message can still be detected.

No watermarked signal can be robust to all types of signal processing operations. For different types of applications, the watermarked signal should be robust against the possible operations it may encounter. For example, in video broadcast monitoring, the watermark needs to survive some small amount of vertical and horizontal translation. When it comes to broadcast radio, we obviously have no need to consider it.

In this paper, robustness is evaluated by the error probability at the decoder, which is denoted by P_e. The lower the P_e is, the more robust the system is.

2.3 Information-theoretic Perspective of Joint Compression and Digital Watermarking

As we illustrate in Fig 2.3, a digital watermarking system can be modeled as a communication system with side information at the watermark transmitter. We first represent each part mathematically. Let M denote the watermark to be embedded, and let $Z^N = (z_1, z_2, ..., z_N)$ denote the host data sequence that consists of independent and identically distributed (i.i.d.) samples drawn according to $p(z)$. The watermarked signal is denoted by X^N and the input of the decoder is Y^N. After the reconstruction that is processed by the decoder, we can get the estimated watermark $\hat{M}$. The Fig 2.3 can be represented as Fig 2.4.

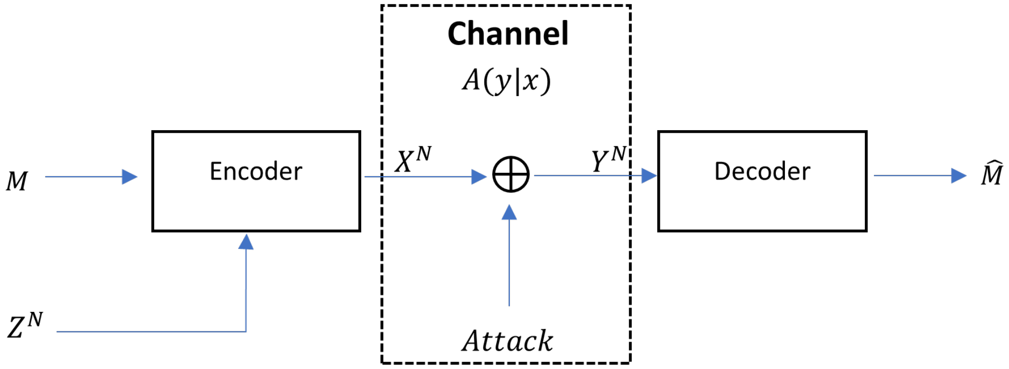


Figure 2.4: Digital watermarking model with symbol notations

Joint compression and digital watermarking means embedding watermarks into the host signal while compressing the host signal subject to distortion constraints. The watermark needs to be reconstructed without the information of the host signal. This kind of detection is called blind detection. Expressions of encoder and decoder are:

- The watermarking encoder is expressed as a function $f_N : Z^N \times M \to X^N$. The watermarked signal $x^N = f_N(z^N, m)$ is generated.

- The watermarking decoder maps the received sequence y^N to a reconstructed watermark $\hat{M}$ using $\phi_N : Y^N \to M, \hat{m} = \phi_N(y^N)$.
- The error probability of watermarking is $P_e = Pr\{\hat{M} \neq M\}$ when encoder and decoder pair (f_N, ϕ_N) is provided.
- $R = \frac{1}{N} \log |M|$ is the watermark embedding rate where $|M|$ is the cardinality of message set M.

The distortion function for the encoder is function $d_1 : Z \times X \to \mathbb{R}_+$.

$$d^N(z^N, x^N) = \frac{1}{N} \sum_{k=1}^{N} d_1(z_k, x_k)$$

The distortion function for the attacker is function $d_2 : X \times Y \to \mathbb{R}_+$.

$$d^N(x^N, y^N) = \frac{1}{N} \sum_{k=1}^{N} d_2(x_k, y_k)$$

To make it clear, we use d_i to represent d_i^N in this chapter.

The memoryless attack channel can be modeled as a conditional pmf $A(y|x)$ from X to Y such that $\mathbb{E}d_2(x^N, y^N) \leq D_2$. The class $\mathcal{A}(f_N, D_2)$ is defined as the set of all memoryless attack channels subject to D_2 under the channel with encoding function f_N. Also, define $\mathcal{A}(f_N)$ as $\mathcal{A}(f_N, D_2) \cap \mathcal{B}$, where $\mathcal{B}$ is some compact set of channels [8].[1]

The capacity of a fixed discrete memoryless channel with public decoder is given by [7]

$$C = \max_{p(x,u|z)} [I(U;Y) - I(U;Z)] \tag{2.1}$$

[1] $\mathcal{B}$ could be fixed attack channel, a finite-dimensional parametric family, or a class of channels that introduce no signal bias: $E(Y) = X$

where U is an auxiliary random variable defined over a finite set $\mathcal{U}$ of cardinality $|\mathcal{U}| \leq |\mathcal{X}||\mathcal{Z}|+1$. $I(;)$ means the mutual information. This capacity also requires the distortion constraint D_1, i.e. $\mathbb{E}d_1(z^N, x^N) \leq D_1$.

In [8], the information hiding capacity is generated upon the above definitions. One theorem is that assume for any $N \geq 1$, the attacker knows f_N, and the decoder knows both f_N and the attack channel. Let $\mathcal{Q}$ be the set of all memoryless covert channels subject to distortion D_1. A rate R is achievable for distortion D_1 and attacks in the class $\mathcal{A}(f_N)$ if and only if $R < C$, where

$$C = \max_{Q_{X,U|Z}(x,u|z)\in\mathcal{Q}} \min_{A(y|x)\in\mathcal{A}(f_N)} \{I(U;Y) - I(U;Z)\} \tag{2.2}$$

and the joint distribution for U, Z, X, Y forms a Markov chain $(U, Z) \rightarrow X \rightarrow Y$.

An interesting result has been developed in [8]: the achievable rate of reliable transmission is the same whether or not the host data are known at the decoder. The corresponding result is shown below.

Let $\mathcal{Z} = \mathcal{X} = \mathcal{Y} = \mathbb{R}$ and $d_1(x, y) = d_2(x, y) = (x - y)^2$ be the squared-error distortion measure. Assume that $Z \sim \mathcal{N}(0, \sigma^2)$ and the class of attack channels is $\mathcal{A}(f_N, D_2)$. The condition $D_2 < (\sigma + \sqrt{D_1})^2$ need to be satisfied. Let a be the maximizer of expression

$$f(a) = \frac{[(2a-1)\sigma^2 - D_2 + D_1][D_1 - (a-1)^2\sigma^2]}{[D_1 + (2a-1)\sigma^2]D_2} \tag{2.3}$$

in the interval $(a_{inf}, 1 + \sqrt{D_1}/\sigma)$, where $a_{inf} = \max(1, \frac{\sigma^2+D_2-D_1}{2\sigma^2})$.

Then, the hiding capacity is upperbounded by

$$C = \frac{1}{2}\log(1 + \frac{[(2a-1)\sigma^2 - D_2 + D_1][D_1 - (a-1)^2\sigma^2]}{[D_1 + (2a-1)\sigma^2]D_2}) \tag{2.4}$$

The optimal covert channel [2] is given by $X = aZ + B$ and $U = \alpha Z + B$ where $B \sim \mathcal{N}(0, D_1 - (a-1)^2\sigma^2)$ is independent of Z. The optimal attack $A(y|x)$is the Gaussian test channel from rate-distortion theory [9]

$$A^*(y|x) = \mathcal{N}(\beta^{-1}x, \beta^{-1}D_2) \tag{2.5}$$

where $\beta = \frac{(2a-1)\sigma^2+D_1}{(2a-1)\sigma^2-(D_2-D_1)}\alpha = \frac{D1-(a-1)^2\sigma^2}{D_1-(a-1)^2\sigma^2+\beta D_2}$

For the case where the covertext and the attack channel are Gaussian, closed-form expressions for the rate region of watermarking embedding rate R_w versus composite rate R_c have not been found yet. However, an achievable rate region using relations between the composite rate, the embedding rate, and the prescribed distortion constraint for the private decoder is established in [10]. It can be treated as an outer bound of the Gaussian case for the public decoder. Details are shown below.

Assume the covertext Z^N is i.i.d Gaussian with zero mean and variance σ_z^2 and the attack is additive i.i.d Gaussian noise with zero mean and variance D_2. A private, continuous alphabet joint watermarking and compression code $(2^{nR_c}, 2^{nR_w}, n)$ satisfies the requirements

$$\frac{1}{N}\mathbb{E}\|Z^N - X^N\|^2 \leq D_1 \tag{2.6}$$

and

$$Pr\{\hat{M} \neq M\} \to 0 \; as \; N \to \infty \tag{2.7}$$

respectively, if and only if $(R_c, R_w) \in R_{D_1,D_2}$, where R_{D_1,D_2} is defined as

[2] Covert channel is optimally designed by information hider

$$R_{D_1,D_2} = \begin{cases} (R_c, R_w) : R_c \geq [\frac{1}{2}\log(\frac{\sigma_z^2}{D_1})]^+ \\ R_w \leq \max_{\gamma\in[\sigma_z^2, 2^{2R_c}]} \min\{R_c - \frac{1}{2}\log(\gamma), \frac{1}{2}\log(1 + \frac{P_w(\gamma)}{D_2})\} \end{cases} \tag{2.8}$$

where

$$P_w(\gamma) = \frac{\gamma(\sigma_z^2 + D_1) - 2\sigma_z^2 + 2\sqrt{\sigma_z^2(\gamma D_1 - \sigma_z^2)(\gamma - 1)}}{\gamma} \quad \sigma_z^2 \geq D_1 \tag{2.9}$$

2.4 Conclusion

In this chapter, we have first summarized some commonly used models for digital watermarking and system evaluation criteria. Further, we have reviewed some information-theoretic results related to joint compression and digital watermarking. More specifically, an upper bound on the information hiding capacity was defined and an upper bound on the achievable rate region in terms of embedding rate, composite rate and distortion constraint was also introduced.

3 Review of Previous Work on Binary JWC Based on Scalar Quantization

In this chapter, we review the previous work related to binary joint watermarking and compression (JWC) based on scalar quantization. Recall that we use the term binary JWSQ for such systems. First of all, a widely used coding strategy, namley quantization index modulation, is presented in Section 3.1. The fixed-rate JWSQ (FJWSQ) scheme proposed in [4] is presented in Section 3.2. In Section 3.3, the variable-rate JWSQ (VJWSQ) system introduced by [5] is reviewed. Finally, this chapter is summarized in Section 3.4.

3.1 Quantization Index Modulation

Since the need for copyright protection of contents has arisen, many digital watermarking schemes have been developed. Quantization index modulation (QIM) proposed by Chen and Wornell [2] is considered as one of the most efficient embedding methods since it is capacity achiving in many scenarios when the statistics of the attack channel are known at the watermark encoder.

In QIM, each watermak symbol m is associated a quantizer Q^m. In order to embed a watermark symbol m into a host signal sample s, s is quantized using the quantizer Q^m. In this work we focus on the case of binary watermarking and scalar quantization. This means that we aim to embed a one bit message ($m \in \{0,1\}$) into one sample $s \in \mathbb{R}$ of the host signal and use scalar quantizers Q^m.

Let $Q(s)$ denote a uniform scalar quantizer with step size Δ, which is defined as

$$Q(s) = \Delta \cdot round(s/\Delta), \tag{3.1}$$

where $round(x)$ maps x to the closest integer. To generate two different quantizers we can use the technique of dither modulation. More specifically, the quantizers can be defined as follows:

$$Q^m(s) = Q(s - d^m) + d^m,\ m = 0, 1 \tag{3.2}$$

where $d^0 = -\Delta/4$ and $d^1 = \Delta/4$. The reconstruction values of quantizer Q^0 and Q^1 are shown in Fig 3.1 as circles and crosses, respectively. The circles and crosses are treated as both quantizer reconstruction points and signal constellations points

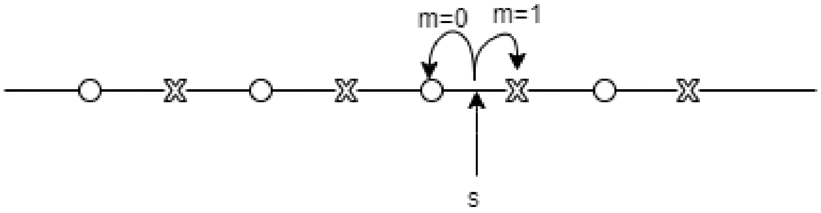


Figure 3.1: Embedding one bit $m \in \{0, 1\}$ into one sample s using QIM

The above dither modulation approach can be generalized by using general quantizers $Q^m, m \in \{0, 1\}$. Each quantizer is a mapping from $\mathbb{R}$ to a codebook $Y^m = \{y_1^m, y_2^m, ..., y_{L_m}^m\}$. The output values $y_j^m, 1 \leqslant j \leqslant L_m$ are also called reconstruction points. The nonintersection property needs to be met to lead to host-signal interference rejection. In other words, the two codebooks have to be disjoint[3].

The decoder receives the watermarked signal $\hat{y}$, which is possibly interrupted or distorted. Based on that, it has to estimate the transmitted message. Denote by $\hat{m}$ the estimated message. The objective of the decoder is to make the error probability $P_e = P\{\hat{m} \neq m\}$ as small as possible. One can use the minimum distance (MD) decoder, which chooses the closest reconstruction point, checks to which quantizer

[3] According to [2], host-interference rejection can be achieved when knowledge of the host signal at the encoder is adequately exploited in system design.

this point belongs, and then extracts the watermark, in other words, computes

$$\hat{m}(\hat{y}) = arg \min_{m \in \{0,1\}} (\hat{y} - Q^m(\hat{y}))^2. \tag{3.3}$$

3.2 Joint Watermarking and Compression Using Fixed-rate Scalar Quantization

In [4], Wu and Yang designed a binary FJWSQ to minimize the probability of error in case of additive white Gaussian noise (AWGN) attacks. The codebooks Y^0 and Y^1 of the two quantizers have the same size L, i.e., $L_0 = L_1 = L$. The set $T^m = \{t_0^m, t_1^m, t_2^m, ..., t_L^m\}$ is the set of thresholds for the quantizer Q^m, which divides the real line into L cells. More specifically, $t_0^m = -\infty$ and $t_L^m = \infty$, and $t_i^m < t_{i+1}^m$, for $0 \leq i \leq L-1$, $m \in \{0,1\}$. The i-th cell of Q^m is $C_i^m = \{s \in \mathbb{R} : Q^m(s) = y_i^m\} = \left[t_{i-1}^m, t_i^m\right)$. All the cells are pairwise disjoint and their union equals the whole quantized space $\mathbb{R}$, i.e.,

$$\begin{cases} C_i^m \cap C_j^m = \emptyset, \text{ for all } i \neq j \\ \bigcup_{i=1}^{L} C_i^m = \mathbb{R} \end{cases} \tag{3.4}$$

For the quantization distortion, which is measured by using squared error, to be minimized, the threshold between two adjacent cells has to be the midpoint of the two corresponding reconstruction points, i.e.,

$$t_j^m = \frac{1}{2}(y_j^m + y_{j+1}^m), \ 1 \leq j \leq L-1. \tag{3.5}$$

Additionally, the codebooks Y^0 and Y^1 have to satisfy the following condition

$$y_1^0 \le y_1^1 \le y_2^0 \le y_2^1 \le \cdots \le y_j^0 \le y_j^1 \le \cdots \le y_L^0 \le y_L^1. \tag{3.6}$$

Let Y denote the codebook set $Y = Y^0 \cup Y^1$, and let T denote the end points set $T = T^0 \cup T^1$.

Fig.3.2 illustrates the thresholds and the reconstruction values of a FJWSQ. The circles and the crosses are reconstruction points of Q^0 and Q^1, respectively. The positions of thresholds are marked by dashed lines.

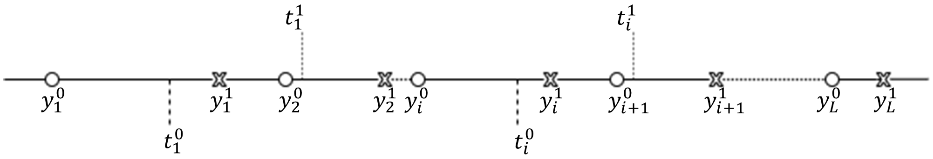


Figure 3.2: Representation of a binary FJWSQ. The circles and the crosses are reconstruction points of Q^0 and Q^1, respectively. The positions of the thresholds in T^m are marked by the dashed vertical lines.

By choosing the quantizer that corresponds to the embedded message m, the output signal is jointly watermarked and compressed. In other words, the watermark message m is embedded into the source signal s to generate the watermarked signal $y = Q^m(s)$. The compression rate of the FJWSQ is

$$R_Q = \log_2(2 * L). \tag{3.7}$$

To design a JWC system, an appropriate decoding rule needs to be chosen first. By doing simulations, Wu and Yang observed that in small distortion scenario (distortion $\ll 1$) and when the distortion to noise ratio (DNR) is larger than 4.77 dB[4], the

[4] 4.77 dB is the minimum DNR required to support the reliable embedding rate of one bit per sample[6].

performance of the MD decoder was similar to that of the maximum likelihood (ML) decoder. In addition, the MD decoder is preferable since it does not need to know the source statistics and has lower implementation complexity than the ML decoder. Therefore, in the DNR region of practical interest, the MD decoder defined in (3.3) is used. The decoding bit error probability $P_e(Y,T)$ can be defined as

$$P_e(Y,T) = \frac{1}{2} \sum_{m \in \{0,1\}} \sum_{j=1}^{L_m} P(s \in C_j^m) P_{j,e}^m \tag{3.8}$$

where $P(s \in C_j^m)$ is the probability that the host signal s lies in C_j^m. $P_{j,e}^m$ is the conditional bit error probability given m and the fact that the host signal lies in the quantization cell C_j^m. Assuming that the watermarked symbol is attacked by an AWGN attack channel with noise variance σ_n^2, $P_{j,e}^m$ can be expressed as [4]

$$\begin{cases} P_{j,e}^0 = Q(|\frac{(y_L^0+y_L^1)-2y_j^0}{2\sigma_n}|) + \sum_{i=1}^{L-1} |Q(|\frac{(y_i^0+y_i^1)-2y_j^0}{2\sigma_n}|) - Q(|\frac{(y_i^1+y_{i+1}^0)-2y_j^0}{2\sigma_n}|)| \\ P_{j,e}^1 = Q(|\frac{(y_1^0+y_1^1)-2y_j^1}{2\sigma_n}|) + \sum_{i=2}^{L} |Q(|\frac{(y_{i-1}^1+y_i^0)-2y_j^1}{2\sigma_n}|) - Q(|\frac{(y_i^0+y_i^1)-2y_j^1}{2\sigma_n}|)| \end{cases} \tag{3.9}$$

where $Q(x) = (1/\sqrt{2\pi}) \int_x^\infty e^{-\frac{t^2}{2}} dt$.

The squared error is used to measure the distortion since it is differential everywhere on the real line. Assuming that the watermark messages $m \in \{0,1\}$ are equally likely, the expected quantization distortion can be written as

$$D(Y,T) = \frac{1}{2} \sum_{m \in \{0,1\}} \sum_{j=1}^{L_m} \int_{t_{j-1}^m}^{t_j^m} (s - y_j^m)^2 p(s) ds, \tag{3.10}$$

where $p(s)$ is the probability density function of the host signal. The authors of [4] formulate the optimization problem as the problem of minimizing $P_e(Y,T)$ with a constraint on the distortion, i.e.,

$$\begin{cases} minimize_{Y,T}\ P_e(Y,T)\ , \\ \ subject\ to\ D(Y,T) \leq D_1, \end{cases} \tag{3.11}$$

where D_1 is the target distortion. They solve the problem by using Lagrangian relaxation, in other words, by solving

$$minimize_{Y,T}\ P_e(Y,T) + \mu D(Y,T), \tag{3.12}$$

where $\mu \geq 0$ is the Lagrangian multiplier. Note that the solution (Y^*, T^*) to the problem (3.11) can be found by solving (3.12) for some $\mu \geq 0$ if and only if the point $(D(Y^*,T^*), P_e(Y^*,T^*))$ is situated on the lower boundary of the convex hull of the set of all pairs $(D(Y,T), P_e(Y,T))$.

Note that in view of (3.5), the probability of error P_e only depends on Y. Then P_e can be rewritten as

$$P_e(Y) = \frac{1}{2} \sum_{m \in \{0,1\}} \sum_{j=1}^{L_m} \int_{\frac{y_{j-1}^m + y_j^m}{2}}^{\frac{y_j^m + y_{j+1}^m}{2}} p(s) ds P_{j,e}^m \tag{3.13}$$

To solve the problem (3.12), Wu and Yang use a locally optimal algorithm that alternates between two steps. One step fixes the codebook set Y and finds T that minimizes $D(Y,T)$, i.e., it sets the thresholds according to (3.8). The other step fixes the end points T and updates Y using the feasible direction method to minimize the weighted sum in (3.12). According to their reported experimental results, about 1000 to 3000 iterations of these two steps are needed to generate the final quantizer, which may result in a long running time.

Wu and Yang also compare the performance of their JWSQ system with a JWSQ using uniform quantizers. Based on their experiments, they find that the above algorithm to generate optimal binary JWSQ systems using nonuniform quantizers achieves better performance than using the uniform one.

3.3 Joint Watermarking and Compression Using Variable-rate Scalar Quantization

In [5], Zhou and Yang improved the performance of the FJWSQ of [4] by using variable-rate scalar quantization. The two quantizers Q^0 and Q^1 of their VJWSQ have the same number of cells $L_0 = L_1 = L$ and the condition (3.6) still has to be satisfied. It is assumed that the output of each quantizer is encoded losslessly using an entropy coder, in other words, an encoder which is able to achieve a rate close to the entropy of the quantizer output. Thus the rate of the VJWSQ is defined as the entropy of the watermarked signal, i.e.,

$$R = \frac{1}{2}[H(Y^0) + H(Y^1)] + 1, \tag{3.14}$$

where $H(Y^m)$ denotes the entropy of Y^m. It is clear that the rate R can be expressed as a function of threshold set T as follows:

$$R(T) = 1 - \frac{1}{2} \sum_{m\in\{0,1\}} \sum_{j=1}^{L_m} \int_{t_{j-1}^m}^{t_j^m} p(s)ds \log \int_{t_{j-1}^m}^{t_j^m} p(s)ds. \tag{3.15}$$

The optimization problem formulated in [5] is

$$\begin{cases} minimize_{Y,T}\ P_e(Y,T), \\ subject\ to\ D(Y,T) \leq D_2 \\ R(T) \leq R_0 \end{cases} \tag{3.16}$$

where R_0 and D_2 are given values, while $D(Y,T)$ and $P_e(Y,T)$ are defined as in the fixed-rate case. Zhou and Yang convert the problem to the unconstrained problem of minimizing the Lagrangian, i.e.,

$$minimize\ P_e(Y,T) + \mu_1 D(Y,T) + \mu_2 R(T), \tag{3.17}$$

with $\mu_1 \geq 0$ and $\mu_2 \geq 0$ are the Lagrangian multipliers. Note that the solution (Y^*, T^*) to the problem (3.16) can be found by solving (3.17) for some $\mu_1, \mu_2 \geq 0$ if and only if the point $(D(Y^*, T^*), R(T^*), P_e(Y^*, T^*))$ is situated on the lower boundary of the convex hull of the set of all triples $(D(Y,T), R(T), P_e(Y,T))$.

Since the relation $t_j^m = \frac{1}{2}\left(y_j^m + y_{j+1}^m\right)$ does no longer hold for the variable-rate scenario, the bit error probability will depend on both the set T of end points T and the codebook Y. More specifically, it can be written as

$$P_e(Y,T) = \frac{1}{2} \sum_{m \in \{0,1\}} \sum_{j=1}^{L_m} \int_{t_{j-1}^m}^{t_j^m} p(s) ds P_{j,e}^m \tag{3.18}$$

The following two constraints are mandatory to make equation (3.18) hold all the time:

$$\begin{cases} y_1^0 \leq y_1^1 \leq ... \leq y_j^0 \leq y_j^1 \leq ... \leq y_L^0 \leq y_L^1 \end{cases} \tag{3.19}$$

An alternating algorithm that is similar in spirit to the algorithm of [4] is developed to solve this optimization problem. The running time is even longer than the running time of the algorithm of [4] since the optimization problem is more complex now.

According to the simulation result in [5], Zhou and Yang observed that the optimum binary VJWSQ has better performance than FJWSQ in [4]. There is about 0.3-dB DNR gain when i.i.d. Gaussian source is used.

3.4 Chapter Summary

In this chapter, we have reviewed the previous work on binary JWSQ. QIM that developed in [2] embeds information by first modulating a sequence of indices with the embedded information and then quantizing the host signal with the sequence of quantizers. It is easy to implement and flexible to compute. Based on QIM, a JWSQ strategy using fixed-rate scalar quantization is formulated in [4]. The optimal quantizers are found by solving the constrained optimization problem. The problem is converted to the unconstrained problem of minimizing the Lagrangian. The end points are updated by getting the midpoint of two consecutive thresholds of one quantizer and feasible direction method is used to update codebooks. An improved JWSQ strategy using variable-rate scalar quantization is performed in [5]. A similar Lagragian function that contains a new parameter, rate R, is used to relax the constraint and get the optimal quantizers. In the next chapter, we will present our faster solution for FJWSQ and VJWSQ coding problem.

4 Proposed Binary Joint Watermarking and Scalar Quantization Systems

In this chapter, we propose faster design algorithms for the FJWSQ and VJWSQ scenarios. The main idea of our designs is to model the related optimization problem to a minimum weight path problem (possibly with constraints on the number of edges) in a certain weighted directed acyclic graph. For this to be possible we discretize the quantization space and use an approximation for the probability of watermark decoding error.

In this chapter, we have two sections. Section 4.1 treats the Fast JWSQ design in the fixed-rate. We introduce the problem formulation and propose a dynamic programming solution. Section 4.2 presents the problem configuration and solution algorithm for the variable-rate case.

4.1 Proposed FJWSQ Design

In this section we present the proposed FJWSQ design. We first introduce the problem configuration in subsection 4.1.1. Similarities and differences with Wu and Yang's work are listed here. We then present the proposed solution algorithm using the graph model in subsection 4.1.2.

4.1.1 Problem Formulation

The configuration of the proposed binary FJWSQ scheme is the same as the scheme of Wu and Yang's [4]. In other words, the JWSQ consists of a pair of quantizers Q^0 and Q^1, where for each $m \in \{0, 1\}$, quantizer Q^m is determined by the set of thresholds $T^m = \{t_0^m, t_1^m, t_2^m, ..., t_{L_m}^m\}$ and the codebook $Y^m = \{y_1^m, y_2^m, ..., y_{L_m}^m\}$, which satisfy the relations (3.6) and (3.5). As in [4], we assume that the watermarked signal is subjected to an AWGN attack and utilize the MD decoder given in (3.3) for watermark detection.

We formulate the optimization problem slightly differently than in [4] by switching the roles of the distortion and the probability of error. More specifically, we assume that a value P_{e0} for the maximum acceptable probability of error is given and try to minimize the distortion while keeping the probability of error no larger than P_{e0}. This formulation is more realistic in applications where losses in the recovery of the watermark are less tolerated in comparison with distortions of the source signal. For example, if the watermark is the name of the author of a photo, then we would like to recover it exactly without any loss, while the reconstruction of the photo can still have very high quality even with some loss (in other words, even if it is not exactly as the original).

To conclude, we tentatively formulate the optimization problem as follows

$$\begin{cases} \textit{minimize } D(T,Y), \\ \textit{subject to } P_e(T,Y) \leq P_{e0} \end{cases} \tag{4.1}$$

where $D(T,Y)$ and $P_e(T,Y)$ are defined as in Section 3.2. Further, in order to handle the problem more easily, we consider a modification based on the intuition that for P_e to be sufficiently low, the distance between consecutive reconstruction values corresponding to different quantizers has to be sufficiently large. This intuition is illustrated in the Fig. 4.1. Crosses and circles are the reconstruction points of two side quantizers respectively. Assume that the watermarked signal x is one of the reconstruction value of one side quantizer. The probability of error is calculated by adding up the probability that the received point lies in all shaded area. Since the channel noise is AWGN, the further away from the center, the lower the possibility.

As $\mathit{\Delta}_i$ increasing, the addition of possibility of all shaded area decreases.

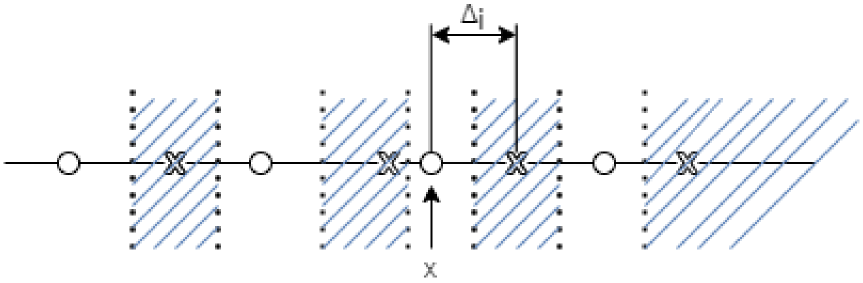


Figure 4.1: An example of how P_e is inflected by $\mathit{\Delta}_i$. x is one of the reconstruction value of one quantizer. Crosses and circles are the reconstruction points of two quantizers respectively. The probability of error is calculated by adding up the probability of received point lies in all shaded area.

Consider now the following notation:

$$\mathit{\Delta}_i = \begin{cases} y_j^1 - y_j^0, \ i = 2j-1, \ 1 \le j \le L-1 \\ y_j^0 - y_{j-1}^1, \ i = 2j, \ 1 \le j \le L-1 \\ y_j^1 - y_j^0, \ i = 2L-1 \end{cases} . \tag{4.2}$$

In the light of above insight, we formulate another optimization problem as follows

$$\begin{cases} \mathit{minimize}\ D, \\ \mathit{subject\ to}\ \mathit{\Delta}_i \ge \mathit{\Delta}_{min}, \ 1 \le i \le 2L-1 \end{cases} \tag{4.3}$$

It can be easily seen that if the condition in (4.3) is satisfied then

$$P_{j,e}^m \le 2Q(\mathit{\Delta}_{min}/(2\sigma_n)) \tag{4.4}$$

for all j, which leads to

$$P_e \le 2Q(\mathit{\Delta}_{min}/(2\sigma_n)). \tag{4.5}$$

Thus, if the target value $\mathit{\Delta}_{min}$ is sufficiently large, then P_e is sufficiently small.

Finally, we assume that the reconstruction points can take values only in some finite set $S = \{s_1, \cdots, s_{N-1}\}$. One way to obtain S is to take N points equally spaced in a bounded interval.

We formulate the final optimization problem as

$$\begin{cases} minimize_Y\ D(Y), & \\ subject\ to & \mathit{\Delta}_i \geq \mathit{\Delta}_{min},\ 1 \leq i \leq 2L-1 \\ & y_j^m \in S,\ 1 \leq j \leq L. \end{cases} \tag{4.6}$$

We consider only Y as a variable in the optimization problem, since, in virtue of (3.8), the FJWSQ is completely specified by the set Y of reconstruction points.

4.1.2 Algorithm for Optimal FJWSQ Design

In this section, we develop a globally optimal algorithm for the problem (4.6). For this, we show that the problem is equivalent to the minimum-weight path (MWP) problem with a constraint on the number of edges in a certain weighted directed acyclic graph(WDAG). Let G denote the WDAG. G consists of the triple (V, E, w), where V denotes the set of vertices, E denotes the set of edges, and w denotes the weight function that assigns a real number to each edge.

The vertex set is $V = \{ab | 0 \leq a \leq b \leq N-1, s_b - s_a \geq \mathit{\Delta}_{min}\} \cup \{00, NN\}$ and the edge set is $E = \{(ab, bc) | ab, bc \in V\}$. A vertex ab represents a pair of possible consecutive reconstruction points from the two quantizers. In other words, s_a is a reconstruction point of one quantizer and s_b is a reconstruction point of the other quantizer. Note that the condition $s_b - s_a \geq \mathit{\Delta}_{min}$ ensures that the constraint

$\mathit{\Delta}_i \geq \mathit{\Delta}_{min}$ is satisfied. An edge (ab, bc) represents a triple of possible consecutive reconstruction points, namely s_a and s_c from one quantizer and s_b from the other quantizer. The graph is directed, which means that the order in the pair (ab, bc) specifying an edge matters (in other words, the edge connects the two vertices in a specified direction). A path in a graph is a sequence of connected edges. Alternatively, a path can be regarded as a sequence of vertices, where any two consecutive vertices are connected by an edge. The length of the path is the number of edges. The graph is acyclic since there is no path with at least one edge ends in the node where it started.

The source vertex of the graph is 00 and the final vertex is NN. A path of $2L + 2$ edges from the source to the final node corresponds to a pair of quantizers (Q^0, Q^1) of a FJWSQ, where each quantizer has L cells. Specifically, if the vertices in the path are $00, 0a_1, a_1b_1, b_1a_2, a_2b_2, \cdots, a_Lb_L, b_LN, NN$, then the path corresponds to the FJWSQ with reconstruction values $y_i^0 = s_{a_i}$ and $y_i^1 = s_{b_i}$, for all $1 \leq i \leq L$. This correspondence is one-to-one. Figure 4.2 illustrates the correspondence between a path and an FJWSQ, for $L = 3$.

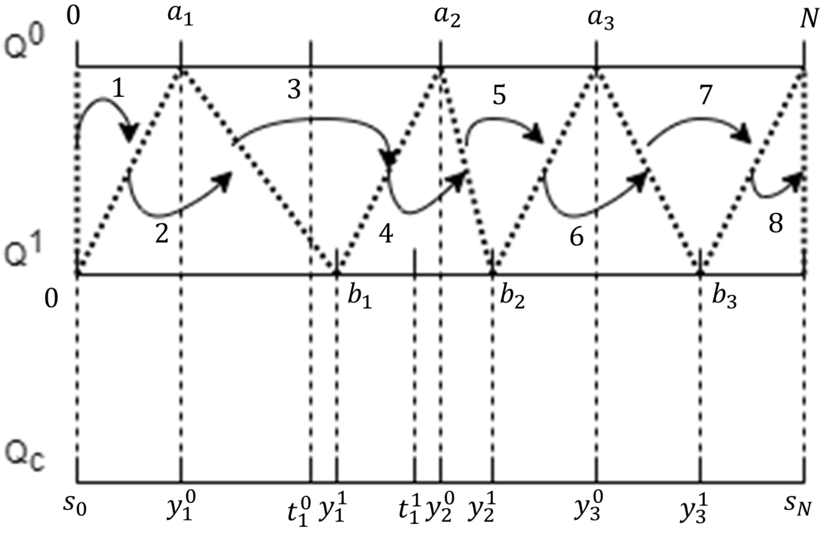


Figure 4.2: An example of a path with 8 edges in the WDAG G, the reconstruction points $\{y_1^0, y_1^1, y_2^0, y_2^1, y_3^0, y_3^1\}$ of the corresponding FJWSQ and two end points $\{t_1^0, t_1^1\}$. A node ab is represented by a dotted segment line connnecting a and b. An edge (ab, bc) is represented by an arc with corresponding sequence number on it.

According to the correspondence described above, if the edge (ab, bc) with $0 < a < c < N$ is in such a path, then s_a and s_c are consecutive reconstruction values in one of the quantizers. Then the threshold separating the corresponding quantization cells is $t = (s_a + s_c)/2$. This means that any sample s in $[s_a, t)$ is reconstructed as s_a, while any sample s in $[t, s_c)$ is reconstructed as s_c. Then we define the weight of the edge as the contribution of the samples s in $[s_a, s_c)$ to the distortion of this quantizer, i.e.,

$$w(ab, bc) = \frac{1}{2}\left(\int_{s_a}^{\frac{s_a+s_c}{2}} (s - s_a)^2 p(s) ds + \int_{\frac{s_a+s_c}{2}}^{s_c} (s - s_c)^2 p(s) ds\right) \tag{4.7}$$

when $0 < a < c < N$. When $a = 0$ or $c = N$, the weight of the edge is defined as follows

$$w(ab, bc) = \frac{1}{2}\int_{-\infty}^{s_c}(s - s_c)^2 p(s)ds \;\; when \; a = 0, \tag{4.8}$$

$$w(ab, bc) = \frac{1}{2}\int_{s_a}^{\infty}(s - s_a)^2 p(s)ds \;\; when \; c = N. \tag{4.9}$$

It can be easily seen that the weight of any path with $2L+2$ edges from the source to the final node is equal to the distortion of the corresponding FJWSQ. Since the correspondence between such paths and the FJWSQs with L cells in each quantizer is one-to-one, it follows that the problem (4.6) is equivalent to the problem of finding the maximum weight path in G among all paths from the source to the final node that contain exactly $2L + 2$ edges.

The algorithm to solve the $(2L + 2)$-MWP problem in G is shown in Algorithm 1. If the weights of edges are precomputed, the time complexity of the algorithm amounts to $O(LN^3)$ since the number of nodes is $O(N^2)$, the number of edges is $O(N^3)$ and the number of different i values is $O(L)$.

Algorithm 1 Solution for $(2L + 2)$-edge MWP problem in G

```
Let $s_i$ be an array of size $(N + 1)^2$, which holds the total weight of the MWP with
$i$ edges from node 00 to any node. $s_i[00] = 0, s_i[NN] = \infty$ for all $i$.
Let $t_i$ be an array of size $(N + 1)^2$, which holds the last visited node of the MWP
with $i$ edges from node 00 to any node. All elements are initialized to 00.
for i=3 to 2L+2 do
    for a=2 to N do
        for b=a+1 to N do
            for c=0 to a-1 do
                Let $w_1$ be the weight of the edge from $ca$ to $ab$;
                Find MWP to $ab$ with $i$ edges:
                if $i = 3$ then
                    Let $w_2$ be the weight of the sum of first two edges
                    $w_2 = w(00, 0c) + w(0c, ca)$
                    if $s_i[ab] > w_2 + w_1$ then
                        $s_i[ab] \leftarrow w_2 + w_1$;
                        $t_i[ab] \leftarrow ca$;
                    end if
                else
                    if $s_i[ab] > s_{i-1}[ca] + w_1$ then
                        $s_i[ab] \leftarrow s_{i-1}[ca] + w_1$;
                        $t_i[ab] \leftarrow ca$;
                    end if
                end if
            end for
        end for
    end for
end for
Recover the optimal path using backtracking based on the arrays $t_i$
```

Note that the aforementioned time complexity can still be achieved if the weight of each edge is computed in $O(1)$ time instead of being precomputed. For this, we precompute and store the cumulative kth order moments $\alpha_k(t) = \int_{-\infty}^{t} s^k p(s)ds$ for all possible thresholds t and $k = 0, 1, 2$. Note that the number of all possible thresholds t is $O(N^2)$ in general, thus this precomputation requires $O(N^2)$ amount of time. In the case when the values in S are equally spaced, i.e., $s_a = \alpha + a\beta$, for all $1 \leq a, \leq N-1$, with fixed α and β, the possible thresholds take values in the set $\{\alpha + j\beta/2 | 1 \leq j \leq 2N-2\}$, thus the precomputation can be done in $O(N)$ time.

When the weight of an edge is needed, its computation can be performed using the cumulative moments by exploiting the following relations

$$\begin{aligned}\int_{s_{start}}^{s_{end}} (s - s_{re})^2 p(s)ds &= \int_{s_{start}}^{s_{end}} (s^2 - 2s_{re}s + s_{re}^2)p(s)ds \\ &= \int_{s_{start}}^{s_{end}} s^2 p(s)ds - 2s_{re} \int_{s_{start}}^{s_{end}} sp(s)ds + s_{re}^2 \int_{s_{start}}^{s_{end}} p(s)ds,\end{aligned} \tag{4.10}$$

$$\int_{s_{start}}^{s_{end}} s^k p(s)ds = \int_{-\infty}^{s_{end}} s^k p(s)ds - \int_{-\infty}^{s_{start}} s^k p(s)ds, \ k = 0, 1, 2. \tag{4.11}$$

Finally, if our goal is to solve the problem (4.1), then we solve the problem (4.3) for various values of $\mathit{\Delta}_{min}$ until P_e becomes close enough to P_{e0}, while satisfying $P_e \leq P_{e0}$. The search over $\mathit{\Delta}_{min}$ can be implemented by using bisection search.

4.2 Proposed VJWSQ Design

This section is devoted to the proposed VJWSQ design. In subsection 4.2.1, we introduce the problem formulation. The following section presents thr graph model of the problem amd the solution algorithm.

4.2.1 Problem Formulation

We consider a VJWSQ as defined in Section 3.3 with the difference that the number of cells of two quantizers can be either equal ($L_0 = L_1 = L$) or different by 1, namely $L_0 + 1 = L_1$. For each j and m, the reconstruction value of cell C_j^m is defined as the center of mass C_j^m,

$$y_j^m = \frac{\int_{t_{j-1}^m}^{t_j^m} sp(s)ds}{\int_{t_{j-1}^m}^{t_j^m} p(s)ds}, \ 1 \leq j \leq L_m. \tag{4.12}$$

The above condition is necessary for the distortion to be minimized. In virtue of (4.12), it follows that the VJWSQ is completely specified by the set T of thresholds. Further, in order to guarantee that condition (3.6) is also satisfied, we impose the following constraint on the thresholds

$$t_1^0 \leq t_1^1 \leq ... \leq t_j^0 \leq t_j^1 \leq ... \leq t_{L-1}^0 \leq t_{L-1}^1 \ , \text{ if } L_0 = L_1 = L,$$

$$t_1^1 \leq t_1^0 \leq ... \leq t_j^1 \leq t_j^0 \leq ... \leq {t^1}_{L_1-2} \leq {t^0}_{L_0-1} \leq {t^1}_{L_1-1} \ , \text{ if } L_0 + 1 = L_1. \tag{4.13}$$

According to [16], relations (4.12) and (4.13) imply that (3.6) hold as well. An example of the case when $L_0 + 1 = L_1$ is shown in Fig. 4.3. Here, $L_0 = 3$ and $L_1 = 4$.

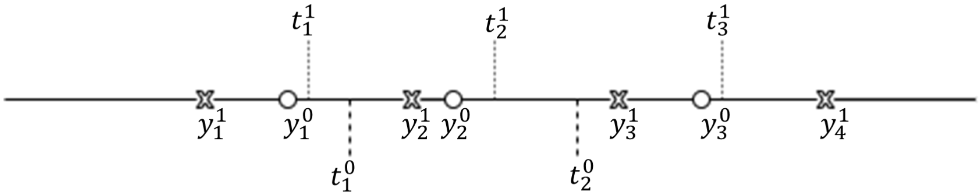


Figure 4.3: Representation of a VJWSQ system with different number of cells in the two quantizers. The circles and the crosses are reconstruction points of Q^0 and Q^1 respectively. The positions of the thresholds in T^m are marked by dashed vertical lines.

The quantization distortion and the rate are defined as in Section 3.3. The probability of watermark decoding error is the same (3.18), but the relations (3.9) for $P_{j,e}^m$ become

$$\begin{cases} P_{j,e}^0 = Q(|\frac{(y_{L_0+1}^0+y_{L_1+1}^1)-2y_j^0}{2\sigma_n}|) + Q(|\frac{(y_1^0+y_1^1)-2y_j^1}{2\sigma_n}|) \\ \qquad + \sum_{i=2}^{L_0} |Q(|\frac{(y_{i-1}^0+y_i^1)-2y_j^0}{2\sigma_n}|) - Q(|\frac{(y_i^1+y_i^1)-2y_j^1}{2\sigma_n}|)| \\ P_{j,e}^1 = \sum_{i=1}^{L_0+1} |Q(|\frac{(y_i^1+y_i^0)-2y_j^1}{2\sigma_n}|) - Q(|\frac{(y_i^0+y_{i+1}^1)-2y_j^1}{2\sigma_n}|)| \end{cases} \tag{4.14}$$

Using (4.12), the distortion, the rate and the probability of decoding error can be written as functions of T only. We formulate the optimization problem as

$$\begin{cases} minimize_T \ D, & \\ subject\ to & P_e(T) \leq P_{e0} \cdot \\ & R(T) \leq R_0 \end{cases} \tag{4.15}$$

We relax the constraints and transform the problem to the problem of minimizing the Lagrangian

$$minimize_T D(T) + \lambda_1 P_e(T) + \lambda_2 R(T) \tag{4.16}$$

with $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are the Lagrangian multipliers. Note that the solution (T^*) to the problem (4.15) can be found by solving (4.16) for some $\lambda_1, \lambda_2 \geq 0$ if and only if the point $(P_e(T^*), R(T^*), D(T^*))$ is situated on the lower boundary of the convex hull of the set of all triples $(P_e(T), R(T), D(T))$.

Additionally, we impose the condition that the thresholds can only take values in some finite set $S = \{s_1, \cdots, s_{N-1}\}$ and formulate the final optimization problem as

$$\begin{gathered} minimize_T D(T) + \lambda_1 P_e(T) + \lambda_2 R(T) \\ t_j^m \in S,\ 1 \leq j \leq L_m - 1,\ m = 0, 1. \end{gathered} \tag{4.17}$$

4.2.2 Solution Algorithm for Optimal VJWSQ Design

In this section we present a globally optimal algorithm for the problem (4.17). To this end, we model the problem as an MWP problem in a WDAG G'. The graph model is more complex than in the fixed-rate scenario. Let $G' = (V', E', w')$. The set of vertices is $V' = \{abc | 0 \leq a \leq b \leq c \leq N, a < c\}$. A vertex abc corresponds to a possible triple of consecutive thresholds. Namely s_a and s_c are two consecutive thresholds in one quantizer and s_b is a threshold of the other quantizer. The edge set is $E' = \{(abc, bcd) | 0 \leq a \leq b \leq c \leq d \leq N, a < d\}$. Using the information of one vertex, we could get the reconstruction value of cell $C_{ac} \triangleq [s_a, s_c)$

$$\gamma(a, c) \triangleq \frac{\int_{s_a}^{s_c} sp(s)ds}{\int_{s_a}^{s_c} p(s)ds} \tag{4.18}$$

Denote $E^* = E' \setminus \{(000, 00a), (aNN, NNN) : 1 \leq a \leq N-1\}$. In any edge $(abc, bcd) \in E^*$, there are two cells involved, C_{ac} and C_{bd}. We only include the distortion and rate of C_{bd} in the weight of the edge. For this, first define

$$P(C_{bd}) \triangleq \int_{s_b}^{s_d} p(s)ds \tag{4.19}$$

$$d(C_{bd}) \triangleq \int_{s_b}^{s_d} [s - \gamma(b,d)]^2 p(s)ds \tag{4.20}$$

$$r(C_{bd}) \triangleq -P(C_{bd}) \log_2 P(C_{bd}) \tag{4.21}$$

$P(C_{bd})$ is the probability that host signal s lies in cell C_{bd}. $d(C_{bd})$ is the distortion of cell C_{bd}. $r(C_{bd})$ is the rate of cell C_{bd}.

Substituting (4.18) into (4.20), we could get a simplified expression

$$\begin{aligned} d(C_{bd}) &= \int_{s_b}^{s_d} [s^2 + \gamma(b,d)^2 - 2\gamma(b,d)s]p(s)ds \\ &= \int_{s_b}^{s_d} s^2 p(s)ds + \gamma(b,d)^2 \int_{s_b}^{s_d} p(s)ds - 2\gamma(b,d) \int_{s_b}^{s_d} sp(s)ds \\ &= \int_{s_b}^{s_d} s^2 p(s)ds + \gamma(b,d)^2 \int_{s_b}^{s_d} p(s)ds - 2\gamma(b,d)[\gamma(b,d) \int_{s_b}^{s_d} p(s)ds] \\ &= \int_{s_b}^{s_d} s^2 p(s)ds - \gamma(b,d)^2 \int_{s_b}^{s_d} p(s)ds \end{aligned} \tag{4.22}$$

For every host signal that lies in the quantization cell C_{ac}, the conditional bit error probability of the received watermarked signal $\hat{y}$ is composed by two parts. The first part is when $\hat{y}$ smaller than the sent reconstruction value, we call that left part. The other one is when $\hat{y}$ greater than the sent reconstruction value, we call that right part.

Further, we could define probability of error for the edge as

$$P_e((abc, bcd)) \triangleq P(C_{ac})P_e^r(C_{ac}) + P(C_{bd})P_e^l(C_{bd}) \tag{4.23}$$

where $P_e^r(C_{ac})$ is the right half of conditional bit error probability given the host signal lies in the quantization cell C_{ac}. Similarly, $P_e^l(C_{bd})$ is the left half of conditional bit error probability given the host signal lies in the quantization cell C_{bd}. To reduce the computational complexity, we estimate them as

$$P_e^r(C_{ac}) = P_e^l(C_{bd}) \triangleq Q(\frac{\gamma(b,d) - \gamma(a,c)}{2\sigma_n}) \tag{4.24}$$

An example is illustated in Figure (4.4). You could see which area will be counted when calculating $P_e^r(C_{ac})$ and $P_e^l(C_{bd})$. Since AWGN is symmetrical about the center and the distance between start point of integration and center is the same for $\gamma(a,c)$ and $\gamma(b,d)$, $P_e^r(C_{ac})$ equals to $P_e^l(C_{bd})$. The estimated probability of error is larger than real one. In other words, we use the upper bound of P_e for one edge.

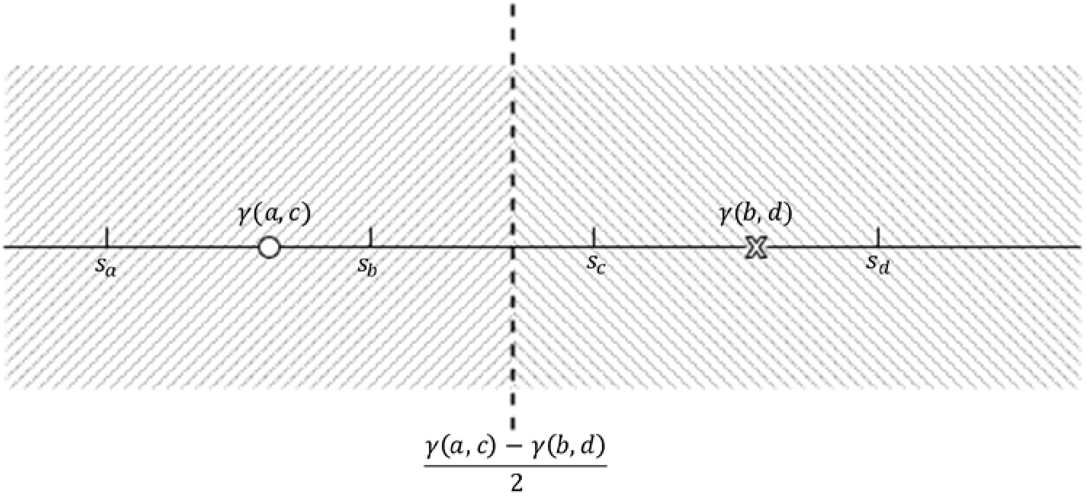


Figure 4.4: An example of half of conditional bit error probability for one edge(abc, bcd). If one watermarked point $\gamma(b,d)$ is transmitted and the received point lies in the area that filled with left slash, it will cause a left half conditional error. The probability of points lies in that area is $P_e^l(C_{bd})$. The area that filled with right slash is the same when calculating $P_e^r(C_{ac})$

For each edge in E^*, we assign the weight as

$$w(abc, bcd) = d(C_{bd}) + \lambda_1 r(C_{bd}) + \lambda_2 P_e((abc, bcd)) \tag{4.25}$$

For the edges $(000, 00a)$ and (aNN, NNN), the weight is defined as follows

$$w(000, 00a) = d(C_{0a}) + \lambda_1 r(C_{0a}) \tag{4.26}$$

$$w(aNN, NNN) = 0. \tag{4.27}$$

Every path from (000) to (NNN) corresponds to a pair of quantizers (Q^0, Q^1) of a VJWSQ. Specifically, if the path has an even number of vertices $2L + 2$, then the path can be represented as the sequence of vertices $000, 00a_1, 0a_1b_1, a_1b_1a_2, b_1a_2b_2, ...,$ $a_{L-1}b_{L-1}N, b_{L-1}NN, NNN$. The corresponding VJWSQ has the thresholds $t_i^0 = a_i$ and $t_i^1 = b_i$, for all $1 \leq i \leq L - 1$. An example of this case is shown in Fig.4.5. If the path has an odd number of vertices $2L+3$, then the path can be represented as the sequence of vertices $000, 00a_1, 0a_1b_1, a_1b_1a_2, b_1a_2b_2, ..., a_{L-1}b_{L-1}a_L, b_{L-1}a_LN, a_LNN, NNN$. The corresponding VJWSQ has the thresholds $t_i^0 = b_i$, $1 \leq i \leq L - 1$ and $t_i^1 = a_i$, for all $1 \leq i \leq L$. An example of this case is shown in Fig.4.6. Clearly, this correspondence is one-to-one. Additionally, the weight of the path equals the value of the cost function in (4.17). It follows that the problem (4.17) is equivalent to the MWP problem from the source node 000 to the final node NNN in the WDAG G'.

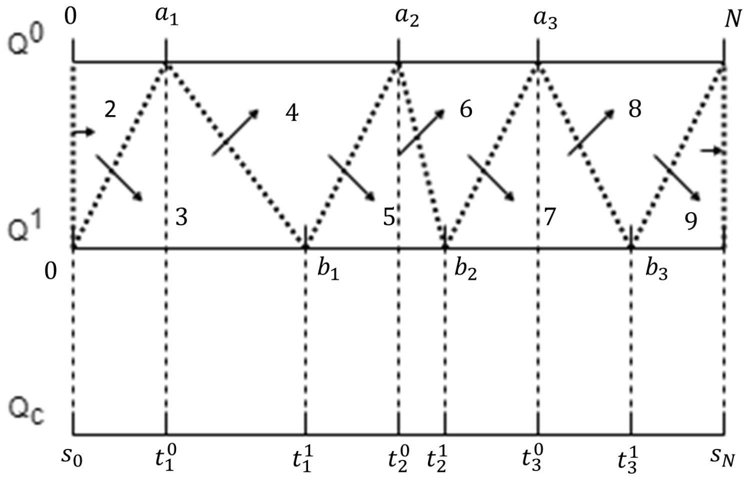


Figure 4.5: An example of a path with even number of vertices in the WDAG G', $L = 4$. The thresholds $\{t_1^0, t_1^1, t_2^0, t_2^1, t_3^0, t_3^1\}$ of the corresponding VJWSQ system are marked. A node is represented by a triangle, e.g node $(b_1a_2b_2)$ is represented by the triangle $b_1a_2b_2$. The sequence numbers of vertices are shown in triangles. Particularly, the first node (000) and last node (NNN) are two vertical lines and are not marked in the figure. An edge is represented by an arrow.

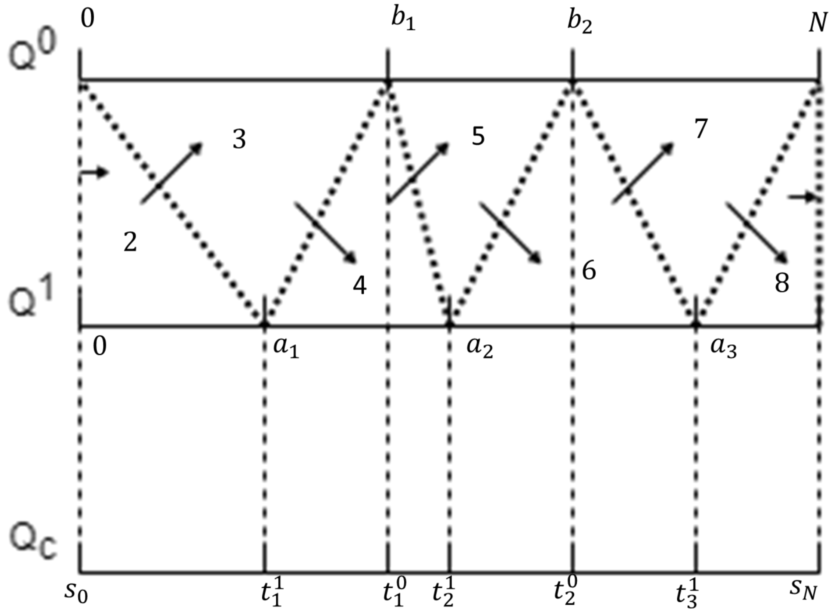


Figure 4.6: An example of a path with odd number of vertices in the WDAG G', $L = 3$. The thresholds $\{t_1^1, t_1^0, t_2^1, t_2^0, t_3^1\}$ of the corresponding VJWSQ system are marked. A node is represented by a triangle, e.g node $(b_1 a_2 b_2)$ is represented by the triangle $b_1 a_2 b_2$. The sequence numbers of vertices are shown in triangles. Particularly, the first node (000) and last node (NNN) are two vertical lines and are not marked in the figure. An edge is represented by an arrow.

The pseudocode of the solution algorithm is presented as Algorithm 2. The weight for all possible first two edges are initialized at the beginning of algorithm. The total weight for one possible path are calculated by adding up the weight for the optimal path to node $(ijk) \in E'$ and weight for the last two edges.

Algorithm 2 Solution for single-source MWP problem in G'

```
Let $s$ be an 3D array of size $(N+1)^3$, which hold the total weight of the MWP from node 000 to any other node. Initialize $s[000] = 0$, all other $s[abc] = \infty$.
Let $t$ be an 3D array of size $(N+1)^3$, which hold the last visited node of the MWP from node 000 to any node. All elements are initialized to 000.
Calculate weight for all possible first two edges and store:
for b=1 to N-1 do
    for c=b+1 to N do
        Let $w_1$ be the weight of the sum of first two edges
        $w_1 = w(000, 00b) + w(00b, 0bc)$;
        $s[0bc] \leftarrow w_1$;
        $t[0bc] \leftarrow 00b$;
    end for
end for
Calculate total weight for all paths without last two edges
for a=1 to N-2 do
    for b=a+1 to N-1 do
        for c=b+1 to N do
            for u=0 to a-1 do
                Let $w_2$ be the weight of the edge from $uab$ to $abc$;
                Find MWP to $abc$ :
                if $s[abc] > s[uab] + w_2$ then
                    $s[abc] \leftarrow s[uab] + w_2$;
                    $t[abc] \leftarrow uab$;
                end if
            end for
        end for
    end for
end for
Calculate total weight for all paths
for a=1 to N-1 do
    for u=0 to a-1 do
        Let $w_3$ be the weight of the sum of last two edges
        $w_3 = w(uaN, aNN) + w(aNN, NNN)$
        Find MWP to $NNN$ :
        if $s[NNN] > s[uaN] + w_3$ then
            $s[NNN] \leftarrow s[uaN] + w_3$;
            $t[aNN] \leftarrow uaN$;
            $t[NNN] \leftarrow aNN$;
        end if
    end for
end for
```

The precomputation can be done the same as in FJWSQ. When calculate the distortion defined in (4.22) for an edge, the cumulative moments can be used by utilizing the following relations

$$\int_{s_{start}}^{s_end} s^2 p(s)ds - \gamma(start, end)^2 \int_{s_{start}}^{s_{end}} p(s)ds \tag{4.28}$$

$$\int_{s_{start}}^{s_{end}} s^k p(s)ds = \int_{-\infty}^{s_{end}} s^k p(s)ds - \int_{-\infty}^{s_{start}} s^k p(s)ds,\ k = 0, 2. \tag{4.29}$$

The back-tracking algorithm need to be changed to satisfy node with 3 points. In Algorithm 2, t is an array that holds the last visited node of the MWP from node 000 to any node. We get the result thresholdss from t. Since the total number of cells L_m is not defined, we keep getting the end points of each quantizer in the reverse order until we get the first end point. The way we get the result end points is shown in Algorithm 3.

Algorithm 3 Solution to get result end points

```
Let $h_{0b}$ be an array that holds all end points of one side quantizer in the reverse order. i.e. $h_{0b}[1]$ holds the index of the last end point of $Q^0$.
Let $h_{1b}$ be an array that holds all end points of the other side quantizer in the reverse order.
Let $pos$ be a variable that tracks how many elements have been stored in $h_{0b}$ and $h_{1b}$.
Let $L_0$ be the total number of cells in $Q^0$. Let $L_1$ be the total number of cells in $Q^1$.
Get the end points in reverse order:
$h_{1b}[1] = t[NNN](1)$;
$h_{0b}[1] = t[h_{1b}[1]NN](1)$;
$h_{1b}[2] = t[h_{0b}[1]h_{1b}[1]N](1)$;
$pos \leftarrow 2$
while $h_{1b}[pos] > 0$ do
    $h_{0b}[pos] = t[h_{1b}[pos]h_{0b}[pos-1]h_{1b}[pos-1]]$;
    $pos \leftarrow pos + 1$;
    $h_{1b}[pos] = t[h_{0b}[pos-1]h_{1b}[pos-1]h_{0b}[pos-2]]$;
end while
Check if two quantizers have different total number of cells:
if $h_{0b}[pos-1] \neq 0$ then
    The first end point of $Q^0$ is not $-\infty$. It means two quantizers have the same length.
    $L_0 \leftarrow pos - 1$;
    $L_1 \leftarrow pos - 1$;
    Delete $h_{1b}[pos]$ since it is 0.
    Sort $h_{0b}$ and $h_{1b}$, we get $T^0$ and $T^1$.
else
    The first end point of $Q^0$ is $-\infty$. It means $Q^1$ has one more cell than $Q^0$.
    $L_0 \leftarrow pos - 2$;
    $L_1 \leftarrow pos - 1$;
    Delete $h_{1b}[pos]$ and $h_{0b}[pos-1]$ since they are 0.
    Sort $h_{0b}$ and $h_{1b}$, we get $T^0$ and $T^1$.
end if
```

Our goal is to solve problem (4.15), then we solve problem (4.16) with various λ_1 and λ_2. Increase λ_2 if $P_e \geq P_{e0}$. Increase λ_1 if $R \geq R_c$. Otherwise, decrease λ_2 to get lower distortion and acceptable probability of error or decrease λ_1 to get lower probability of error. Bisection search can be used to choose appropriate λ_1 and λ_2.

5 Simulation Result and Analysis

Having described the algorithms for designing optimum JWSQ systems, this section assesses the performance of two algorithms and compares result with [4] and [5]. We denote JWSQ in [4] as FJWSQ and our algorithm as Proposed-FJWSQ. Likewise, JWSQ in [5] is VJWSQ and our algorithm is Proposed-VJWSQ. We construct the source alphabet S by discretizing a continuous Gaussian source with zero mean and unit variance.

Distortion is measured by squared error distortion and minimum distance decoder is used. The attack channel is assumed as an AWGN channel with variance σ_n^2. Extended set of all possible reconstruction/end points S [5] is obtained by partitioning $[-5, 5]$ into 1000 or 500 equally length segments union two points $s_0 = -\infty$, $s_N = \infty$. In total, $N = 1003$ for fixed-rate scenario, $N = 503$ for variable-rate scenario.

In this chapter, we first discuss the results for FJWSQ system design problem in subsection 5.1. The results for VJWSQ system design probelm is presented in subsection 5.2. In each of these 2 cases, we plot curves in terms of decoding bit error probability P_e versus distortion noise ratio(DNR), which is the same as plots in [4] and [5]. DNR is defined as

$$DNR = 10 \log_{10} \frac{D(S,X)}{\sigma_n^2} \tag{5.1}$$

We use $\sigma_n^2 = \{0.019, 0.006, 0.0019, 0.0012, 0.00095\}$ in our experiments. The picked σ_n^2 lead to $DNR = \{0, 5, 10, 12, 13\}$ to compare with results in [4] and [5] for fixed-rate and variable-rate respectively.

To analyze the performance of JWSQ and Proposed-JWSQ systems in more per-

[5] S represents reconstruction points while finding optimum fixed-rate quantizers. It represents end points while finding optimum variable-rate quantizers.

spective, we implemented JWSQ using Matlab. Particularlly, function $linprog(f, A, b_{simp})$ in Matlab to solve linear programming problem is used. Both algorithms are running under the same environment.

5.1 Discussion of FJWSQ Results

In this section, we present the experimental result for FJWSQ design problem. We solve the constraint optimization problem in (4.3) for $\Delta_{min} \in \{0.1 : 0.05 : 0.22, 0.215 : 0.0005 : 0.2245\}$. We pick the smallest Δ_{min} which meet the P_e constraint for all σ_n^2.

The initial value of reconstruction points for FJWSQ system are set as

$$Y^0 = [-3.0872, -2.4177, -1.8737, -1.3594, -0.8554, -0.3554, 0.1434, 0.6425, 1.1440, 1.6516, 2.1782, 2.8202]$$

$$Y^1 = [-2.7865, -2.1412, -1.6144, -1.1065, -0.6051, -0.1060, 0.3928, 0.8928, 1.3966, 1.9108, 2.4555, 3.1218]$$

By analyzing the result of each iteration of FJWSQ system, we can see that the distortion is decreasing and P_e is increasing with iteration increase. Fig. 5.1 and Fig. 5.2 shows an example of the change of distortion D and bit error probability P_e with iterations for FJWSQ system. Gaussian source is used and $\sigma_n^2 = 0.006$. That is the reason we set this initial value. The distortion is 0.0533 if we partition $[-5, 5]$ into $2L + 1$ equally length segments as the initial value. The distortion for this reconstruction value set is 0.219. It is closer to our target 0.0190, which means less running time. It takes about 900 iterations from equally partition to this initial value.

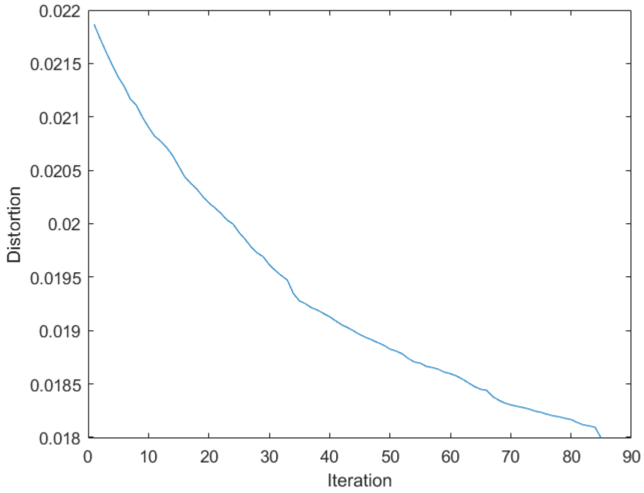


Figure 5.1: Change of distortion with iterations for FJWSQ system. Gaussian source is used,$\sigma_n^2 = 0.006$.

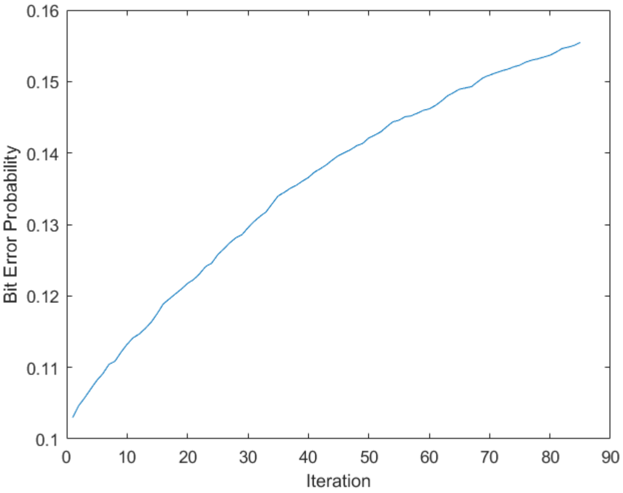


Figure 5.2: Change of bit error probability with iterations for FJWSQ system. Gaussian source is used, $\sigma_n^2 = 0.006$.

Fig 5.3 demonstrates the results of FJWSQ and Proposed-FJWSQ systems using Gaussian source when the codebook size $L = 12$ and the encoding distortion $D_1 = 0.01909$ for both systems.

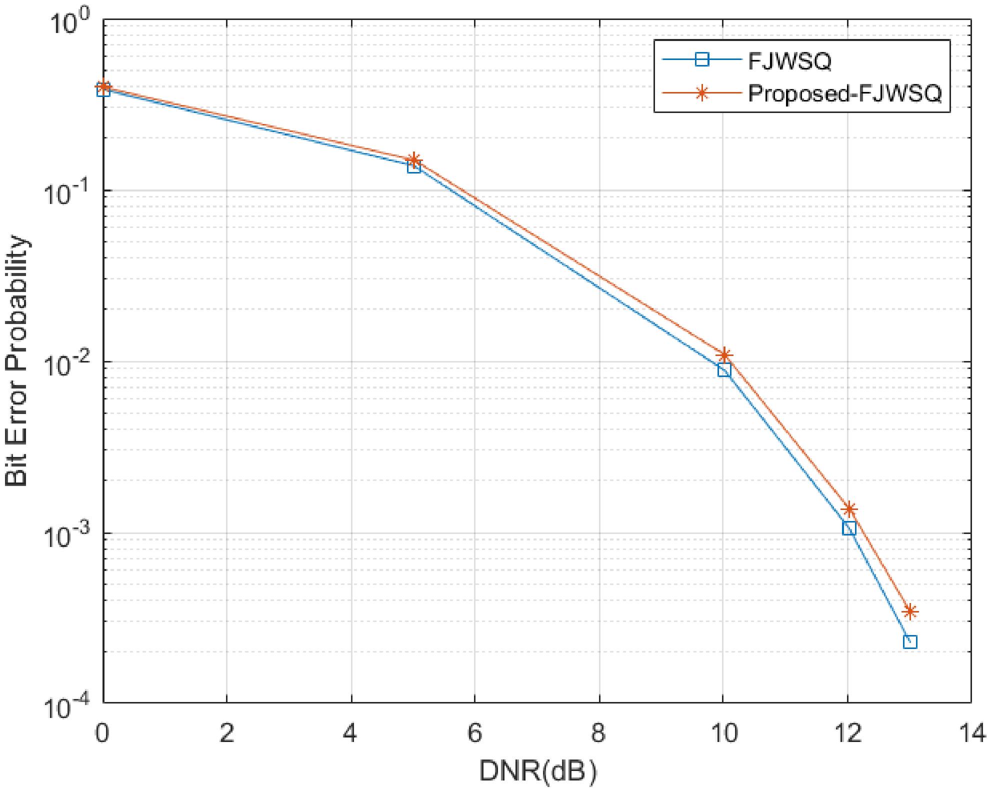


Figure 5.3: Comparisons of performance for FJWSQ and Proposed-FJWSQ systems. Gaussian source is used. $L = 12$, $D_1 = 0.01909$

Table 1 shows the running time of FJWSQ and Proposed-FJWSQ systems. The unit for running time is second. The total number of iterations that used in FJWSQ to get final result is also included.

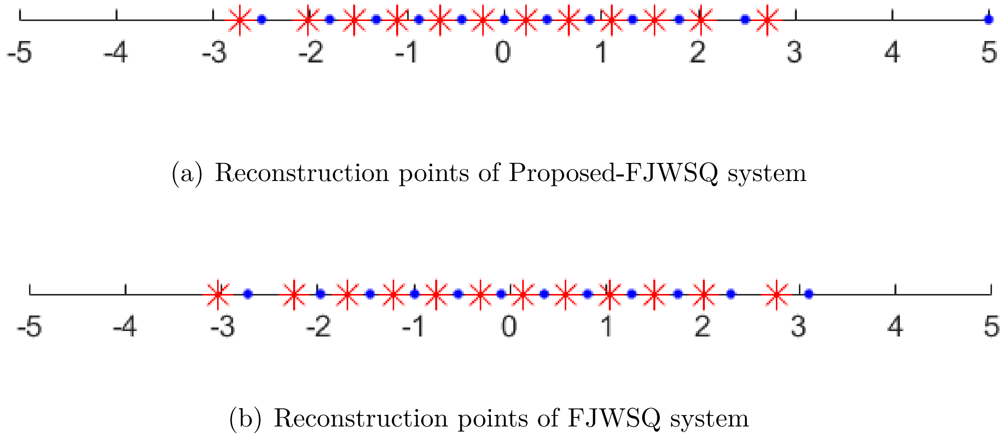
Table 1: Running time of FJWSQ and Proposed-FJWSQ systems for Gaussian source. Unit is second. And Total number of iterations for FJWSQ system

σ_n^2	0.019	0.006	0.0019	0.0012	0.00095
time(FJWSQ)	2201	2546	4396	2002	3540
time(Proposed-FJWSQ)	15	16	26	27	27
# iterations(FJWSQ)	36	42	108	105	96

We can see that the FJWSQ system performs slightly better than our Proposed-FJWSQ system. However, the running time varies a lot. Proposed-FJWSQ system saves 99% of the time. Time is an important measurement standard. Specifically, Proposed-FJWSQ system's running time is even shorter than one iteration of FJWSQ system.

Another advantage is that the running time can be controlled by setting the different values to L and N in Proposed-FJWSQ system. In other words, decreasing the total number of edges or length of the source alphabet leads to shorter running time since the time complexity of Proposed-FJWSQ is $O(LN^3)$. However, the FJWSQ system terminates only when the constraints are unsatisfied, which is uncontrollable. Although the time for each iteration is similar, the solution of linear programming problem for each iteration is unpredictable.

Fig. 5.4 illustrates the reconstruction values of result quantizers when $\sigma_n^2 = 0.00095$ and Table 2 lists the values of reconstruction points for these two systems. We observed that the smallest $\varDelta_{min}$ picked for all σ_n^2 are the same 0.205, which lead to the same quantizers of Proposed-FJWSQ system. Also, the result $\varDelta_i$ of Proposed-FJWSQ system satisfies $\varDelta_i \geq \varDelta_{min}$.



(a) Reconstruction points of Proposed-FJWSQ system

(b) Reconstruction points of FJWSQ system

Figure 5.4: Reconstruction points of FJWSQ and Proposed-FJWSQ systems. Red stars are reconstruction points for Q^0, blue circles are reconstruction points for Q^1

Table 2: Reconstruction values of result quantizers of JWSQ and Proposed-FJWSQ systems when $\sigma_n^2 = 0.00095$

	FJWSQ
Y^0	[-3.05,-2.24,-1.7,-1.22,-0.77,-0.32,0.13,0.57,1.03,1.49,2.02,2.76]
Y^1	[-2.73,-1.97,-1.46,-0.99,-0.54,-0.1,0.35,0.8,1.26,1.74,2.29,3.1]
	Proposed-FJWSQ
Y^0	[-2.72,-2.02,-1.54,-1.1,-0.66,-0.22,0.22,0.66,1.1,1.54,2.02,2.7]
Y^1	[-2.5,-1.8,-1.32,-0.88,-0.44,0,0.44,0.88,1.32,1.8,2.48,4.99]

5.2 Discussion of VJWSQ Results

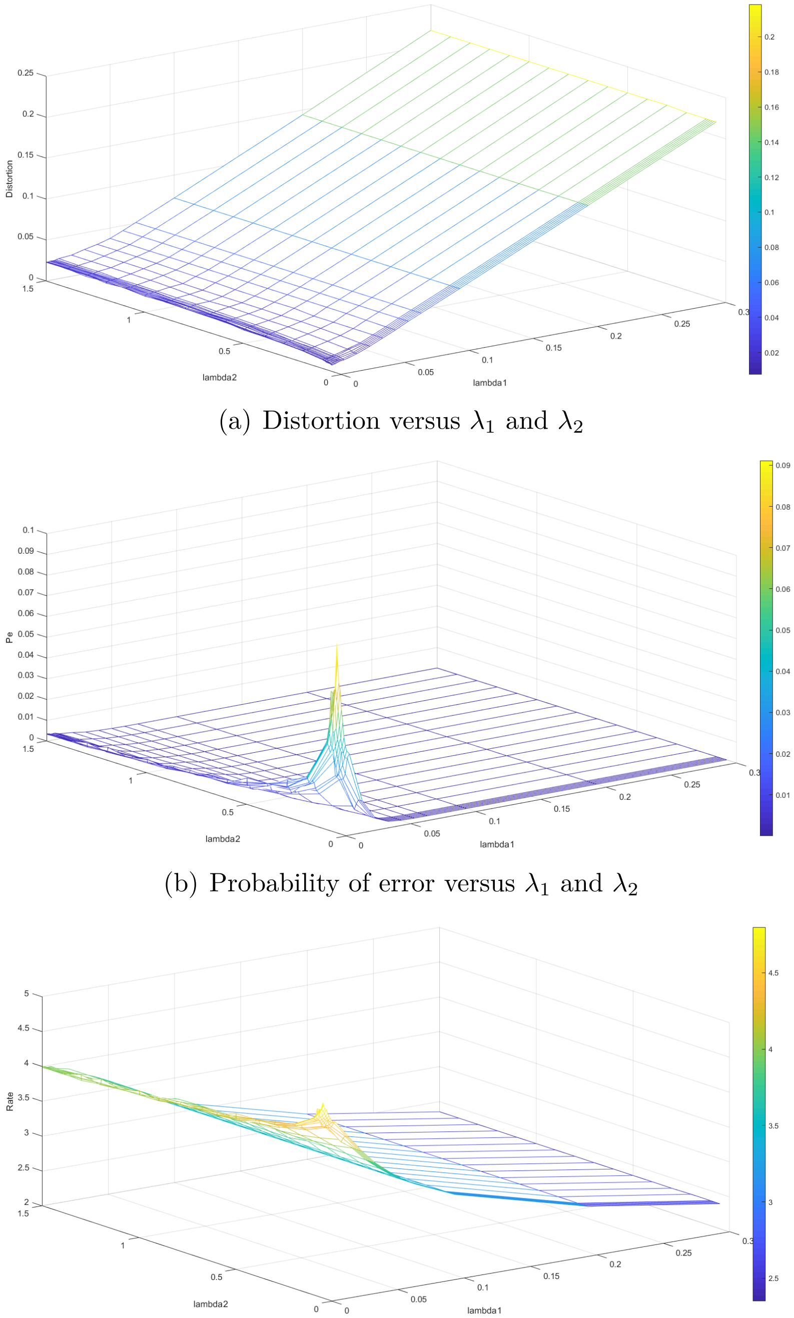
In this section, we present the experimental result of VJWSQ design problem. We first discuss the influence of λ_1 and λ_2 on the results in subsection 5.2.1. Then the comparison with VJWSQ system is presented in subsection 5.2.2. We also compare with Proposed-FJWSQ in subsection 5.2.3.

5.2.1 Influence of λ_1 and λ_2 On the Proposed-VJWSQ Results

Since there are two factors that affect the Proposed-VJWSQ system's final result, λ_1 and λ_2, we first find out how λ_1 and λ_2 influence the final result. By setting different ranges of values for λ_1 and λ_2, we plot decoding probability of error P_e, encoding distortion D, rate R versus λ_1 and λ_2 in Fig. 5.5. Specifically, the values of λ_1 used are $\{0.0005 : 0.0001 : 0.0008, 0.001 : 0.001 : 0.008, 0.01 : 0.01 : 0.07, 0.1 : 0.1 : 0.3\}$, the values of λ_2 used are $\{0.05 : 0.01 : 0.09, 0.1 : 0.1 : 1.5\}$. We pick Gaussian source with zero mean and unit variance and AWGN channel with $\sigma_n^2 = 0.0019$.

We could see that the major changes of P_e happen around 0. A tighter range of λ_1 and λ_2 can be set to see more details. New range of λ_1 is $\{0.0005 : 0.0001 : 0.0008, 0.001 : 0.001 : 0.008\}$, new range of λ_2 is $\{0.05 : 0.01 : 0.15\}$. With these new ranges, we have 3D plots that shown in Fig. 5.6. It is obvious that λ_1 is the dominant part since a small change in λ_1 has a great impact on the final result. For all of the

σ_n^2 in $\{0.019, 0.006, 0.0019, 0.0012, 0.00095\}$, we can set smaller interval of λ_1 to get better results.

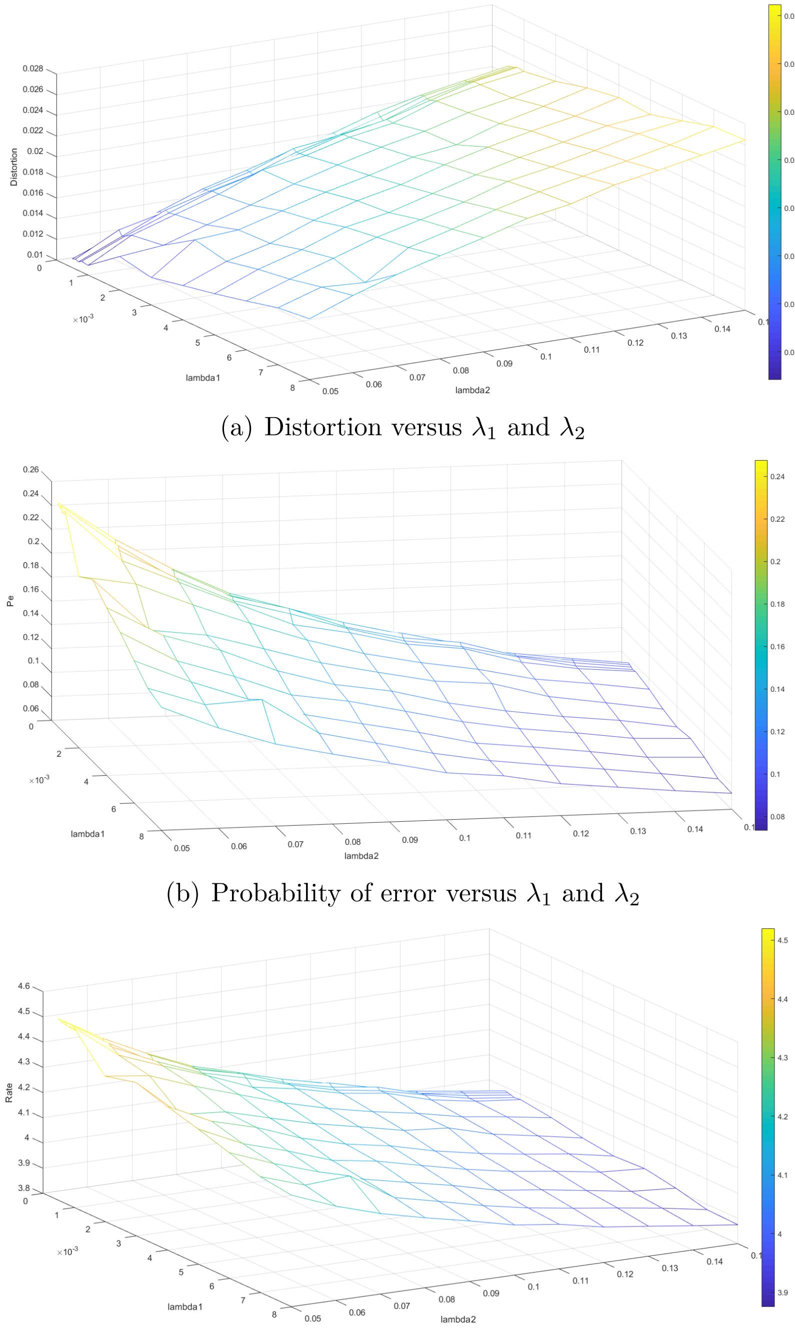


(a) Distortion versus λ_1 and λ_2

(b) Probability of error versus λ_1 and λ_2

(c) Rate versus λ_1 and λ_2

Figure 5.5: 3D plots show the effect of λ_1 and λ_2 on D,R and P_e. λ_1 is picked in the range $[0.0005, 0.3]$, λ_2 is picked in the range $[0.05, 1.5]$



(a) Distortion versus λ_1 and λ_2

(b) Probability of error versus λ_1 and λ_2

(c) Rate versus λ_1 and λ_2

Figure 5.6: 3D plots show the effect of λ_1 and λ_2 on D,R and P_e. λ_1 is picked in the range $[0.0005, 0.008]$, λ_2 is picked in the range $[0.05, 0.15]$

5.2.2 Comparison with VJWSQ System

The initial value of reconstruction set in VJWSQ system is obtained by partitioning $[-5, 5]$ into $2L + 1$ equally length segments, where $L = 12$. We solve the constraint problem (4.16) with $\lambda_1 = \{0.0005 : 0.0001 : 0.0008, 0.001 : 0.001 : 0.008\}$ and $\lambda_2 = \{0.05 : 0.01 : 0.15, 0.2 : 0.2 : 1, 1.5 : 0.25 : 3.5\}$. Results of VJWSQ system are obtained from their figure in [5]. Fig. 5.7 shows the comparison of VJWSQ and Proposed-VJWSQ systems. We achieve the same distortion $D_2 = 0.01909$ and smaller rate $R_0 = 4.127$. In their case, $R_0 = 4.15$.

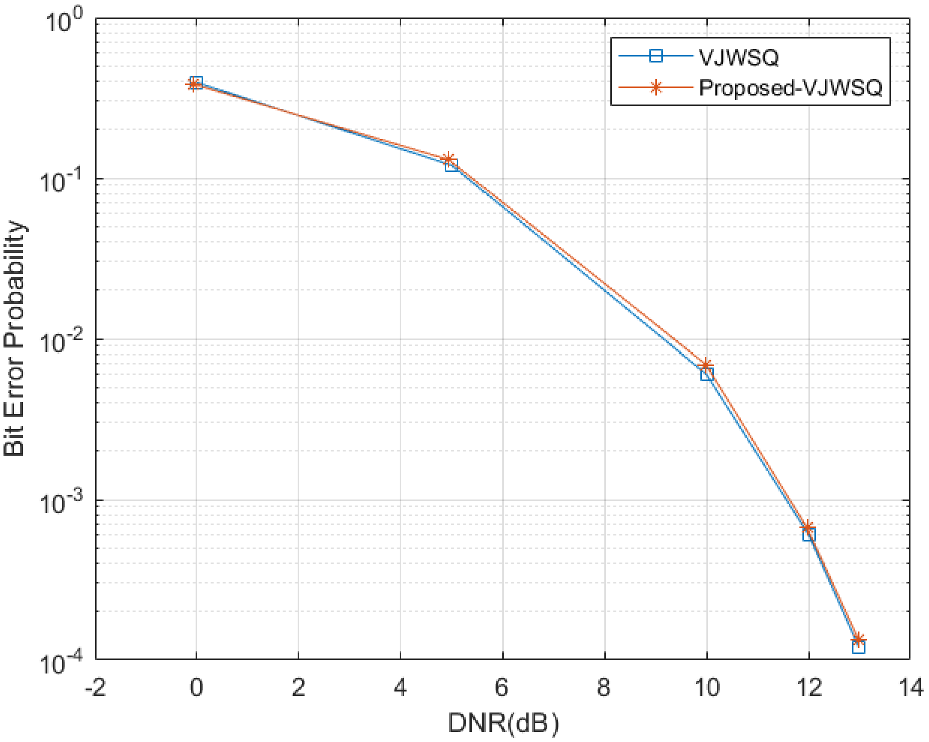


Figure 5.7: Comparisons of performance for binary VJWSQ and Proposed-VJWSQ systems. Gaussian source is used. $D_2 = 0.01909$, $R_0 = 4.15$

We can see that the results are similar. One thing to notice is that Proposed-VJWSQ system doesn't need to set constraints on the total number of cells, which means more flexibility. The total number of cells of Proposed-VJWSQ system is presented in Table 3. Particularly, when $\sigma_n^2 = 0.006$ and $\sigma_n^2 = 0.0019$, the total number of cells for two quantizers are different.

Table 3: Total number of cells of two quantizers Q_0 and Q_1 for Proposed-VJWSQ system

σ_n^2	0.019	0.006	0.0019	0.0012	0.0009
L_0	22	21	21	21	21
L_1	22	22	22	21	21

Also, the running time varies a lot. Althrough we didn't get the same result as in [5], we tried $\mu_1 = \{50, 100, 200, 400\}$ and $\mu_2 = \{0.001, 0.01, 0.1, 1, 10\}$ and stored the running time, D, Pe and R of each iteration. We formulate another type of comparison: within limited number of iterations of VJWSQ, which means less running time, how is the result of JWSQ system compared with Proposed-VJWSQ system. According to their analysis in [5], generally it takes 1000-3000 iterations to get the final result. Let iteration constraint $Iter_{up} = 200$, which is much smaller than total iterations. The comparison of D, Pe, R and running time are included in Table 5.2.2. We can see that within limited time, but still longer than our Proposed-VJWSQ system's running time, the result of VJWSQ system is worse than Proposed-VJWSQ system.

Table 4: Comparison of D, P_e, R and running time of VJWSQ and Proposed-VJWSQ systems. Iteration constraint on VJWSQ system is 200.

σ_n^2	0.019	0.006	0.0019	0.0012	0.00095
Proposed-VJWSQ system					
D	0.0225	0.0265	0.0260	0.0239	0.0291
P_e	0.347	0.0733	0.0016	0.000145	0.0000025
R	3.99	3.88	3.89	3.95	3.81
running time	29	30	39	57	50
VJWSQ system					
D	0.226	0.0296	0.0261	0.025	0.0292
P_e	0.368	0.0859	0.006	0.0075	0.024
R	4.08	3.88	3.97	4.09	3.98
running time	14056	6574	6692	15593	14457

Although Table 5.2.2 shows Proposed-VJWSQ system achieves better result. We need to admit that the result of VJWSQ system may not be the optimal result since Lagrangian multipliers μ_1 and μ_2 are picked from a finite set. It is possible that the pair of Lagrangian multipliers for optimal result is not included in our experimental set. However, the same situation also holds for our Proposed-VJWSQ system. In other words, Proposed-VJWSQ system can try more pairs of Lagrangian multipliers within the same amount of time. The possibility of getting the optimal result is higher for Proposed-VJWSQ system.

5.2.3 Comparison with Proposed-FJWSQ system

Fig. 5.8 illustrates the comparison of performance for binary Proposed-FJWSQ system and Proposed-VJWSQ system. Gaussian source with zero mean and unit variance is used. To have a fair comparison, both of two systems achieve encoding distortion $D_1 = D_2 = 0.01909$. Rate $R_0 = 4.127$ for Proposed-VJWSQ system and 4.358 for Proposed-FJWSQ system using formula (3.15) and (3.7) respectively. Data are obtained from Section 5.1 and subsection 5.2.2 for Proposed-FJWSQ system and

Proposed-VJWSQ system respectively.

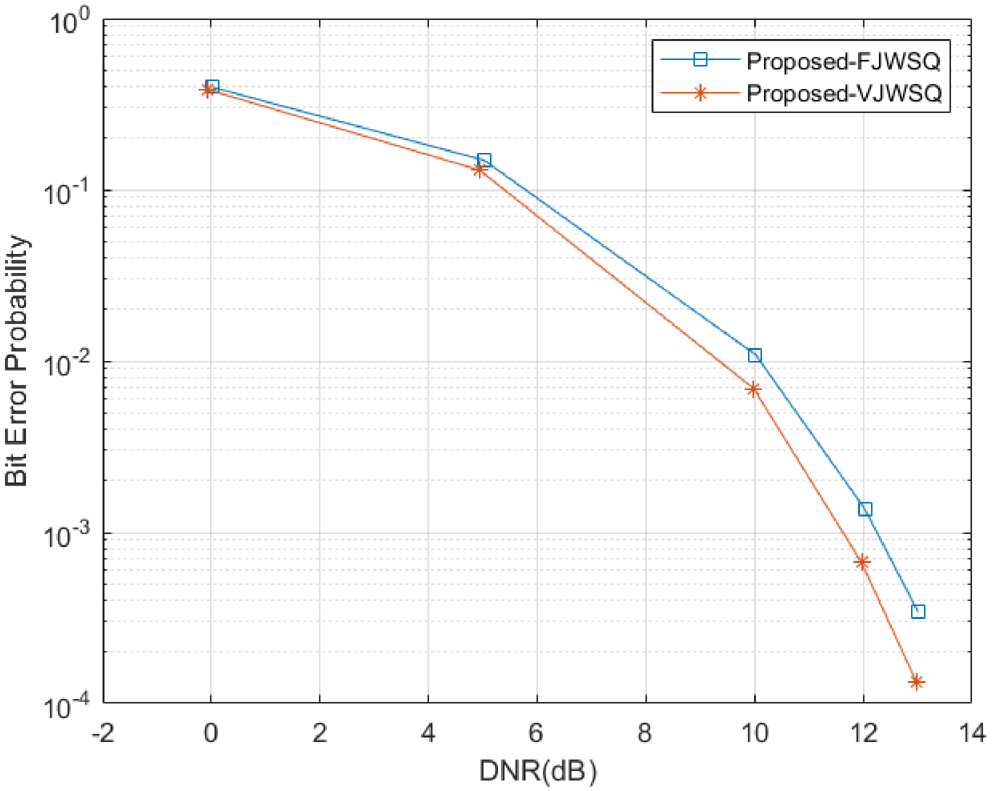


Figure 5.8: Comparisons of performance for binary Proposed-FJWSQ system and Proposed-VJWSQ system. Gaussian source with zero mean and unit variance is used. $D_1 = D_2 = 0.01909$, $R_0 = 4.127$

We can see that the performance of Proposed-VJWSQ system is better than performance of Proposed-FJWSQ. It is obvious that the smaller the σ_n^2 is, the greater the difference of P_e between two systems. In particular, the difference is 0.000213 when $\sigma_n^2 = 0.00095$, which improves 60% of result P_e.

6 Conclusion and Future Work

This work proposes faster solution algorithms for the optimal design of binary joint watermarking and scalar quantization(JWSQ) systems with fixed rate and variable rate. Each system is composed of two quantizers with disjoint codebooks. Both problems are initially formulated as constrained optimization problems targeted at minimizing the distortion under constraints on the rate and on the probability of error.

The first design is for the binary joint watermarking and scalar quantization system with fixed rate, which is a faster scheme than in [4]. We focus on finding the optimal codebooks that minimize the distortion under a constraint on the decoding bit error probability. The latter constraint is further converted to a constraint on the minimum distance between codebooks. The proposed solution algorithm is based on modeling the problem using a weighted directed acyclic graph (WDAG). For this, we assume that the source alphabet is generated by discretizing a finite length interval into equal-length segments. We show that there is a one-to-one correspondence between the paths from the source node to the final node with $2L + 2$ edges and the pairs of quantizers' codebooks of size L each. Finding the single-source minimum-weight path (MWP) in WDAG with $2L + 2$ edge is equivalent to finding the optimal codebooks. The time complexity of the solution algorithm is $O(LN^3)$, where L is the length of each codebook and N is the size of the source alphabet. The experimental results show that our algorithm is much faster than Wu and Yang's algorithm [4] while the performance is close.

The other design is for the binary joint watermarking and scalar quantization system with variable rate, which improves the performance in [5] with more flexibility. We design the reconstruction points under the constraints of decoding bit error probability and rate. By constructing the WDAG with information from two quantizers, our proposed algorithm solves the single-source MWP problem in a WDAG. The improved algorithm achieves a lower rate while having the same distortion and probability of error than in [5]. Meanwhile, it has a significant advantage in terms of running time and does not has any constraint on the number of cells of the two quantizers.

We tried to replicate the simulation results of variable-rate quantizers of [5], but did not succeed due to insufficient time. More experiments are needed in order to obtain the exact simulation results, in order to have a more fair comparison. Also, applying the algorithm to images would provide a better illustration of comparison. Furthermore, another interesting aspect to study is the asymptotical performance analysis of the proposed schemes.

Other future directions in the area of joint watermarking and scalar quantization worth investigating are pointed out next.

- It is interesting to see if using vector quantization could help to get better performance than scalar quantization in JWSQ systems.
- We only consider embedding one-bit watermark into the source signal. The scenario of embedding multiple bits is also an interesting area.

References

[1] S. Dumitrescu, H. Wu, ”Optimal Two-Description Scalar Quantizer Design”, Algorithmica 41, 269–287 (2005).

[2] B. Chen and G. W. Wornell, ”Quantization index modulation: a class of provably good methods for digital watermarking and information embedding”, 2000 IEEE International Symposium on Information Theory (Cat. No.00CH37060), Sorrento, Italy, 2000.

[3] L. Guillemot and J. -. Moureaux, ”Indexing Lattice Vectors in a Joint Watermarking and Compression Scheme”, 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, 2006.

[4] G. Wu and E. Yang, ”Joint watermarking and compression using scalar quantization for maximizing robustness in the presence of additive Gaussian attacks”, in IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 834-844, Feb. 2005.

[5] Y. Zhou and E. Yang, ”Joint robust watermarking and compression using variable-rate scalar quantization”, 2009 11th Canadian Workshop on Information Theory, Ottawa, ON, 2009, pp. 183-186.

[6] P. Moulin and J. A. O’Sullivan, ”Information-theoretic analysis of information hiding”, in IEEE Transactions on Information Theory, vol. 49, no. 3, pp. 563-593, March 2003, doi: 10.1109/TIT.2002.808134.

[7] S. I. Gel’fand and M. S. Pinsker. ”Coding for channel with random parameters”, Probl. Control Inf. Theory, vol. 9, no. 1, pp. 19-31, 1980.

[8] P. Moulin and J. A. O'Sullivan, ”Information-Theoretic Analysis of Information Hiding”, IEEE Transactions Information Theory, vol. 49, pp. 563-593, March2003.

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

[10] D. Karakos and A. Papamarcou, ”A Relationship Between Quantization and Watermarking Rates in the Presence of Additive Gaussian Attacks”, IEEE Transactions Information Theory, vol. 49, pp. 1970-1982, August 2003.

[11] Cox, Ingemar & Miller, Matthew & Bloom, Jeffrey & Fridrich, Jessica & Kalker, Ton. (2007). Digital Watermarking and Steganography. 10.1016/B978-0-12-372585-1.X5001-3.

[12] Liehua Me and G. R. Arce, ”A class of authentication digital watermarks for secure multimedia communication”, IEEE Transactions on Image Processing, vol. 10, no. 11, pp. 1754-1764, Nov. 2001

[13] J. Lacy, S. R. Quackenbush, A. R. Reibman, D. Shur and J. H. Snyder, ”On combining watermarking with perceptual coding”, Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181), 1998, pp. 3725-3728 vol.6

[14] Houng-Jyh Wang and C. -. Jay Kuo, ”An integrated progressive image coding and watermark system”, Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181), 1998, pp. 3721-3724 vol.6

[15] A. Maor and N. Merhav, ”On joint information embedding and lossy compression”, IEEE Transactions on Information Theory, vol. 51, no. 8, pp. 2998-3008, Aug. 2005

[16] A. V. Trushkin, ”Sufficient conditions for uniqueness of a locally optimal quantizer for a class of convex error weighting functions”, IEEE Trans. Inform. Th., vol. 28, no. 2, pp. 187-198, March 1982.