

國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

聯合訊源通道編碼技術用於

無線近身網路之心電訊號壓縮

**Joint Source-Channel Coding for Electrocardiogram
(ECG) Compression on Wireless Body Area Networks**

學生：莊憲榮

指導教授：張錫嘉教授

中華民國一〇二年九月

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

聯合訊源通道編碼技術用於
無線近身網路之心電訊號壓縮

**Joint Source-Channel Coding for Electrocardiogram
(ECG) Compression on Wireless Body Area Networks**

學生：莊憲榮

指導教授：張錫嘉教授

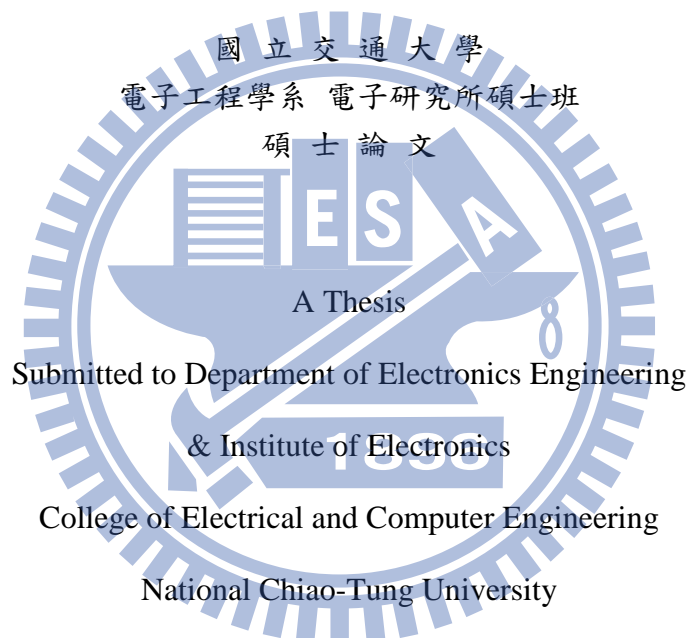
中華民國一〇二年九月

聯合訊源通道編碼技術用於無線近身網路之心電訊號壓縮

Joint Source-Channel Coding for Electrocardiogram (ECG) Compression
on Wireless Body Area Networks

研究生：莊憲榮
指導教授：張錫嘉 博士

Student : Hsien-Jung Chuang
Advisor : Dr. Hsie-Chia Chang



In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Electronics Engineering

Sep. 2013

Hsinchu, Taiwan, Republic of China

中華民國一〇二年九月

聯合訊源通道編碼技術用於 無線近身網路之心電訊號壓縮

學生：莊憲榮

指導教授：張錫嘉 教授

國立交通大學

電子工程學系 電子研究所碩士班

摘要

在無線近身網路中，心電訊號壓縮是一項用來監測心臟相關疾病的重要技術。由於以往的心電壓縮技術著重於高壓縮率以及高速傳輸，在編碼端有比較高的複雜度，並不適用於無線近身網路。據此，本論文採用向量量化和索引指定方法來設計低複雜度的訊號壓縮器。此外，我們在解碼端考慮了通道效應，設計了迴旋編碼器提供資料保護並採用聯合訊源通道解碼的架構，搭配所提新的事前機率初始方法，不但達到良好的訊號重建品質，也提升了解碼速度。在 MIT-BIH 資料庫模擬結果，相較於其他心電傳輸技術，本論文所採用的架構能以較低複雜度之壓縮器，搭配迴旋碼達成良好重建品質並提高 1.5 倍收斂速度。

Joint Source-Channel Coding for Electrocardiogram (ECG) Compression on Wireless Body Area Networks

Student : Hsien-Jung Chuang

Advisor : Dr. Hsie-Chia Chang

Department of Electronics Engineering
Institute of Electronics
National Chiao Tung University

ABSTRACT

A wireless body area network (WBAN) for ECG compression is a promising approach for monitoring cardiac disease. Based on percentage root-mean-square difference (PRD) requirements, many ECG compression techniques have been introduced to achieve high compression ratio (CR) but may not be suitable in WBAN due to their higher encoding complexity. In this thesis, the *joint source-channel coding scheme* is proposed to not only adopt vector quantization and index assignment as the ECG compressor, but apply channel decoding technique to ensure robust ECG transmission. Note that the encoder parameters can be modified to obtain better reconstructing ECG signals at fixed CR. In addition, a new *a priori knowledge initialization algorithm* is presented to provide 1.5 times converge speed in contrast to original iterative source-channel decoding algorithm. After simulating with the MIT-BIH arrhythmia database, the results show that the proposed schemes with low encoding complexity has higher reconstruction quality than previous works.

誌 謝

三年的碩班生涯，有了家人以及朋友的陪同，使我能在這條路上走得開心順利，良好的實驗室環境，最佳的實驗室資源，這些不管在我研究上或者生活上都扮演著重要腳色。

首先要感謝指導教授張錫嘉老師，提供了研究上的建議，財力上的支援，日常生活上的關心，及自由的研究環境，讓我在這三年的研究能夠完成。接著要感謝在業界工作的建青學長，在我實習時總是不吝指教，學長做事的態度為我做了最好的楷模，在我研究上，更是常提供一針見血的建議。還有帶我的實驗室學長振揚，雖然我們研究方向上的不同，學長還是會就研究上的不足，提供實際建議，有問必答，讓我在通道編碼這領域有更多的了解。最後要謝謝 OCEAN 以及 OASIS 團隊，你們的熱心指點及協助，讓我能在這三年快樂的完成學業。

最後，我要感謝我的家人，謝謝我的爸爸和媽媽，對我的精神鼓勵以及財力上的支援，讓我能順利完成碩士論文，最後要感謝口試委員，給予我的寶貴意見。

中華民國 一〇二年 九月
莊憲榮



Contents

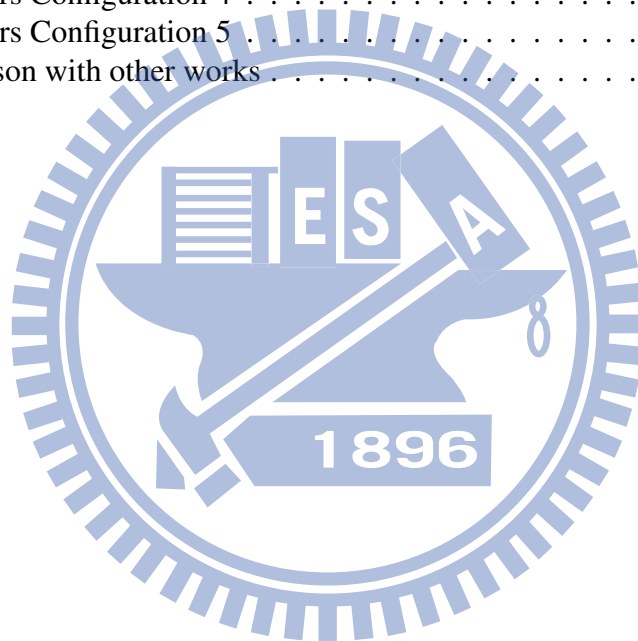
1	Introduction	1
1.1	Background	1
1.2	Motivation	2
1.3	Thesis Organization	3
2	Digital communication system	4
2.1	Source Encoder	5
2.2	Channel Encoder	9
2.3	Soft Iterative Decoding	12
2.3.1	Extrinsic Information	13
2.3.2	Exchange of Extrinsic Information	15
2.4	Soft-Output Channel Decoding	18
2.5	Softbit Source Decoding	22
3	Joint Source-Channel Coding Scheme	26
3.1	System Overview	27
3.2	Vector Quantizer with LBG Algorithm	29
3.3	Index Assignment with Pseudo-Gray Code	32
3.4	Iterative Source-Channel Decoding (ISCD) Algorithm	37
3.4.1	BCJR Algorithm for Channel Decoding	39
3.4.2	Softbit Source Decoding	46
3.5	Proposed A Priori Knowledge Initialization	48
4	Simulation Results	51
4.1	Conventional ISCD Performance	52
4.2	Different Encoder Configurations Analysis	54
4.2.1	Vector Dimension	54
4.2.2	Index Assignment	56
4.2.3	Channel Encoder	57
4.3	Proposed ISCD Performance	58
4.4	Comparison of Different ECG Compression Techniques	60
5	Conclusion and Future Work	62
5.1	Conclusion	62
5.2	Future Work	63
	Bibliography	64

List of Figures

1.1	Wireless body area network system	3
2.1	Communication system diagram	4
2.2	Source encoding flow	5
2.3	Quantization flow	6
2.4	Block diagram of recursive convolutional encoder	10
2.5	Block diagram of two equivalent (2,1,2) convolutional codes	11
2.6	State diagram of the (2,1,2) recursive systematic convolutional codes	12
2.7	Graph representation of the extrinsic and the intrinsic probabilities	13
2.8	Graph representation of the extrinsic exchanging between two vertices	15
2.9	The bipartite graph for the (7,4) Hamming code	19
2.10	Conventional parameter decoding by hard decision	22
2.11	Softbit source decoding by parameter estimation	23
3.1	Block diagram of the transmitter	27
3.2	Block diagram of the receiver	29
3.3	1-bit vector quantizer design with LBG algorithm	33
3.4	2-bit vector quantizer design with LBG algorithm	33
3.5	Block diagram of VQ on noisy channel	34
3.6	Flow chart of the binary switch algorithm	38
4.1	Parameter SNR of conventional ISCD	53
4.2	γ of conventional ISCD	54
4.3	Comparison of different vector dimensions	55
4.4	Comparison of different index assignments	56
4.5	Comparison of different RSCs	58
4.6	Comparison of conventional and proposed ISCD	59

List of Tables

2.1	Quantization Table	7
2.2	Index assignment table	9
4.1	Parameters Configuration 1	52
4.2	Parameters Configuration 2	55
4.3	Parameters Configuration 3	56
4.4	Parameters Configuration 4	58
4.5	Parameters Configuration 5	59
4.6	Comparison with other works	61



Chapter 1

Introduction

For elderly people and those at the risk of various heart diseases, medical service providers invest in remote ECG monitoring systems to take good care of patients by tracking of their health states. Fig.1.1 [1] shows a wireless body area network (WBAN) system for ECG transmission. The system, which includes sensors and a wireless transmitter, needs to be as small as possible for better user experience. However, these kind of remote systems are difficult to design. There are many restrictions like power consumption and limited resources. Since a large percent of power is consumed up by the transmitter, so we conduct a research to reduce the transmitted data size as well as ensure data correctness over noisy wireless channels. Therefore, the compressor and the protector are added to the transmit side for the data compression and correction. In this thesis, we investigate low complexity algorithms to achieve better compression and protection. Simulation results will be provided to support our findings.

1.1 Background

Generally, data compression can be categorized into loss and lossless methods. For ECG data compression, the loss type has been applied for high compression ratio (CR) and preserve clinical information. ECG data compression can be divided into three groups. The first groups are direct data compression. They base their detection of redundancies on analysis in time domain. Examples include turning point (TP) [2], amplitude zone time epoch coding

(AZTEC) [3], coordinate reduction time encoding system (CORTES) [4], the delta algorithm and the Fan algorithm [5]. The second groups are transformation methods [6]. They convert the time domain signal to the frequency or other domains. They mainly utilize the spectral distribution analysis for redundancies. Example include Fourier transform, Karhunen-Loeve transform (KLT), Walsh transform, and the discrete cosine transform (DCT), and the wavelet transform [7]. The third groups are parameter extraction techniques. They extract the characteristic and parameters of the signal. The extracted parameters are subsequently used for classification based on a priori knowledge of the signal. Examples include the peak picking method [8], the linear prediction method [9], the neural network method [10].

1.2 Motivation

Although there already exists many compression techniques, due to the application of WBAN application, which means limited electronic resources, only simple computations can be done. Additionally, these techniques have been introduced purely achieve the high CR on some PRD requirement. However, in WBAN application, the comprehensive lossy compression on noisy channel should be considered.

In this thesis, we introduce a joint source-channel coding methodology to enhance the compression quality with error correctability in WBAN. The approach is based on the vector quantization. Under different channel conditions, the resulting vector dimensions are used respectively to gain a decoding performance. Vector quantization and convolutional coding plus some simple puncturing in the transmitter, may have the chance to meet the limitation of computation complexity for ECG compression. Furthermore, we proposed an efficient algorithm in the decoding procedure. The algorithm provides more accurate source information in bad channel environment and improves the convergence in ISCD.

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2, we introduce the modern digital systems and discuss several source and channel coding techniques. Chapter 3 will first identify the system notations. Then, the algorithms of each units and the proposed initialization are provided. Chapter 4 gives the simulation results and the comparison with other related works. Finally, conclusions and future works are given in Chapter 5.

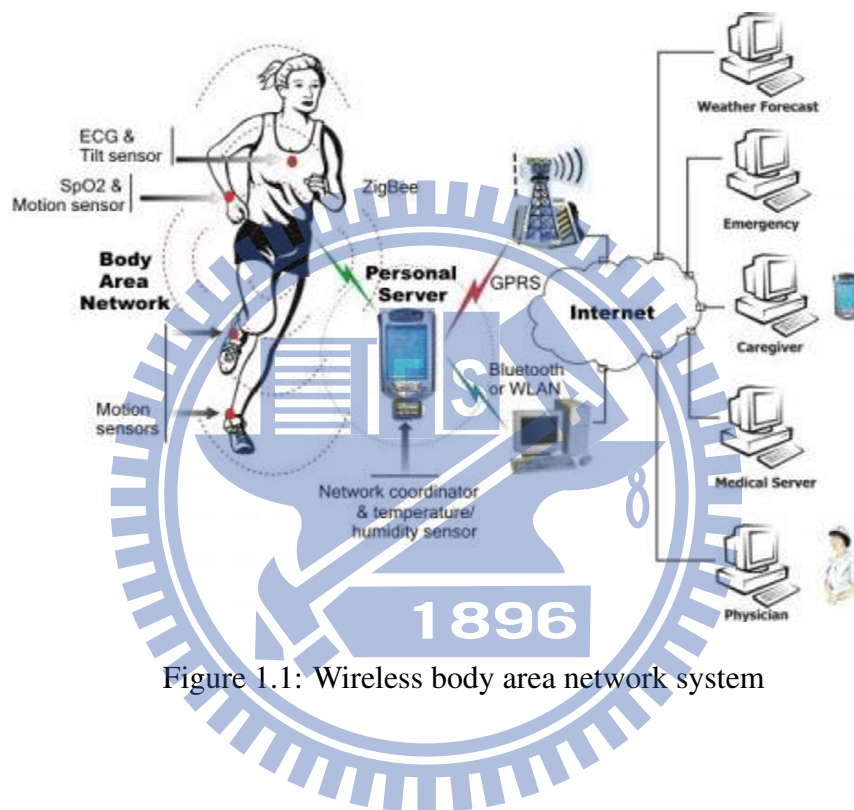


Figure 1.1: Wireless body area network system

Chapter 2

Digital communication system

While sending digital data from a source to a sink, there are several procedures to ensure reliable transmission. The elementary function of digital communication system can be depicted as Fig.2.1.

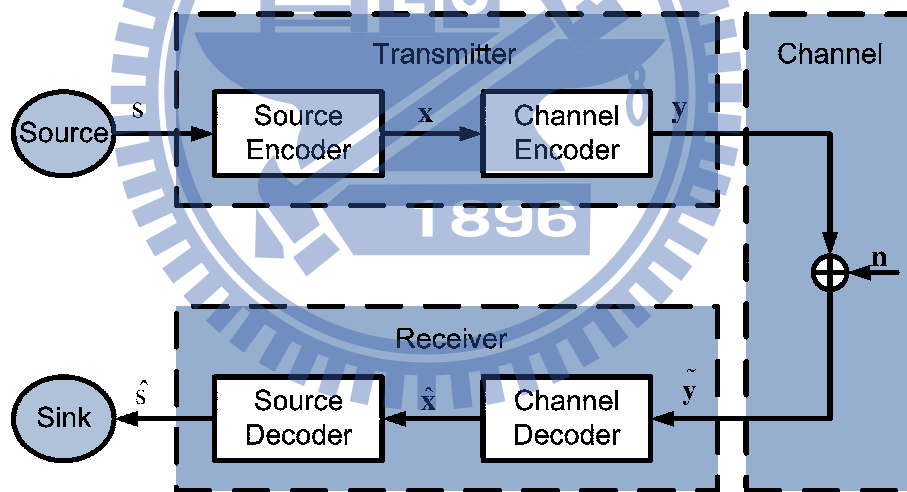


Figure 2.1: Five elements (gray unit) of digital communication system

The information source generates a signal s . The transmitter encodes a time segment of s by a code sequence y . Generally, the goal of the encoder mapping is to minimize the undesirable effects of channel noise on the signal. The actual procedure of transmitter can be separate into two parts, called source and channel encoding. The source encoder performs the compression by using a code sequence x to represent s . The channel encoder inserts some artificial redundancy, known as parity check symbols, to minimize the transmission error. Some errors could

be detected and/or corrected. The code sequence \mathbf{x} is expanded to a code sequence \mathbf{y} . The third element is transmission channel. The noise sequence \mathbf{n} adds to \mathbf{y} . The receiver exploits the received sequence $\tilde{\mathbf{y}}$ in order to recover the source s . The reconstructed signal \hat{s} is sent to the sink.

2.1 Source Encoder

The goal of source coding is to remove redundancy from source, also known as data compression or bit-rate reduction. The ECG signal is continuous in time and magnitude. The digitalize and quantize process adds some artificial noise to the ECG signal. However, the distortion of reconstruction signal is tolerably accepted.

The encoding process can be separated into three steps. See Fig.2.2.



Figure 2.2: Three basic steps of source encoding procedure

First, the source s is decorrelated to \mathbf{v} . Secondly, quantized to particular parameter set \mathbf{u} , in the quantizer codebook \mathbf{U} ($\mathbf{u} \in \mathbf{U}$). Finally, the quantized set \mathbf{u} is one-to-one mapping into the set of bit pattern \mathbf{x} .

Decorrelation is the main part to remove redundancy of the source data. This process is generally lossless and invertible. In order to decorrelate the source signal, there are two types of the signal correlation.

- **Autocorrelation**

Autocorrelation is the similarity between a given signal and its lagged version. In the other words, the correlation with a signal between two different time series is autocorrelation. It is also referred to as lagged correlation or serial correlation. The autocorrelation of white noise signal will have a peak at the origin and will be zero otherwise. The au-

to correlation of periodic signal will have a peak at the origin and repeat with the same period.

- **Cross-correlation**

Cross-correlation is the similarity between two different signals. The decorrelation algorithm must be applied to multiple recordings of the same signal generated in different transmitting channels, such as the multichannel ECG signal. The noise and distortion in different channels are independent, but they have the same source statistic distribution. Distributed source coding(DSC) [11] is a way to solve the cross-correlation problem, and it is used in multimedia or video compression. One of the DSC properties is encoder complexity shifted to decoder.

Quantization is the process of mapping a large set v of values to a small set u . This procedure is irreversible and introduces distortion. We can simply decompose the quantization process to three part. See Fig.2.3.



Figure 2.3: Quantization procedure on noiseless channel

Quantization usually involves encoder and decoder. An encoding or forward quantization part is represented by "E", which maps an input value to quantization index. And a decoder or backward quantization part is represented by "D", which maps the index to quantization value. Two basic type of quantization will be discussed below.

- **Scalar quantization**

Scalar quantization is the most common type of quantization. It is a process to map a scalar input value to output value output. Table 2.1 shows an example of uniform scalar quantization.

Table 2.1: Quantization Table

Input value interval	Quantization index	Quantized value
$(-\infty, -3.5)$	0	-4
$[-3.5, -2.5)$	1	-3
$[-2.5, -1.5)$	2	-2
$[-1.5, -0.5)$	3	-1
$[-0.5, 0.5)$	4	0
$[0.5, 1.5)$	5	1
$[1.5, 2.5)$	6	2
$[2.5, 3.5)$	7	3
$[3.5, \infty)$	8	4

The quantization maps an infinite range of values into nine integers, which needs 4 bit to represent. We call this quantizer a 4-bit scalar quantizer. The uniform quantization means the quantized values are equally spaced. When an analog signal is uniformly sampled and uniformly quantized, the resulting digital representation is called pulse-code modulation. It is commonly used in digital communication systems, such as speech, audio, and video. The opposite of uniform quantization is nonuniform quantization. The origin of nonuniform quantization aim at reducing the quantization error. The nonuniform quantizer design is an optimization problem: Finding a good quantization that minimized the quantization error. A good approach is Lloyd-Max algorithm [12] [13]. The optimal quantizer must satisfy the centroid condition and nearest neighbour condition proposed by [12] [13]. So, the optimal quantizer is also referred to as Lloyd-Max quantizer.

- **Vector quantization**

Vector quantization design algorithm is first proposed by Linde, Buzo, and Gray(LBG) [14]. The LBG algorithm is often used to design the codebook in vector quantization. This algorithm is also referred to k-means algorithm or generalized Lloyd's algorithm. Vector quantization is originally used for lossy compression. It encodes values form multidimensional vector space to a finite set of vector. The index of the vector is sent instead of the quantized values, this achieves more compression. Vector quantization is also used to lossy data correlation. If some dimensions are losses, it can be recovered by available dimensions through the nearest group. Compared with scalar quantization,

vector quantization results in lower distortion when the source is correlated, and more flexibility in bit rate when the source is independent. But the codebook design of vector quantization requires large computation complexity, and it is closely related to compression quality.

Index assignment is the final step of source encoding the quantization index to a particular bit pattern. The following three index assignment is important for iterative source-channel decoding and suitable for scalar quantization.

- **Natural binary**

If the bit pattern encodes the quantization indexes to binary patterns in increasing order, the mapping is called natural binary code. The leftmost bit is most significant bit which usually cause high distortion due to the channel noise. The rightmost bit is least significant bit causes small distortion.

- **Folded binary**

The folded binary bit mapping is also referred to as sign-magnitude code. The leftmost bit is the sign bit and other bits represent the magnitude. If the smallest amplitude of index appears more than other indexes, the transmission error does less distortion than the natural binary code does.

- **Reflected binary**

The reflected binary bit mapping is also known as Gray code. The term reflected binary code is first introduced by Frank Gray in 1947 [15]. Gray code maps every two successive quantization indexes to bit streams with one differ bit. When single transmission error occurs more than multiple transmission errors, Gray code results in smaller distortion than other mappings.

Table 2.2 shows different index assignments after 8-level quantization.

The index assignment techniques mentioned above are commonly used in digital transmission systems. However, these techniques are restricted to scalar quantization. In vector quantization, we do not know which quantization indexes are successive or rearrange these indexes in

Table 2.2: Index assignment table

	Index							
	0	1	2	3	4	5	6	7
Index assignments	bit pattern							
Natural binary	000	001	010	011	100	101	110	111
Folded binary	111	110	101	100	000	001	010	011
Reflected binary	000	001	011	010	110	111	101	100

increasing order. Pseudo Gray code is first proposed by Kenneth Zeger and Allen Gersho [16], which minimizes the distortion operated on noisy channel for vector quantization. The further discussion about pseudo Gray code will be listed in chap 3.

2.2 Channel Encoder

Channel coding adds some artificial redundancy to the information bits after source encoding. The redundancy helps the receiver to correct the error caused by channel noise. Channel coding can detect and correct transmission errors. There are two types of channel codes.

- **Linear block codes**

An (n, k) binary block code is a one-to-one mapping of k bit information sequence \mathbf{x} and n bit codeword \mathbf{y} . A binary block code is linear if and only if the module-2 sum of two codewords is also a codeword. The encoding procedure of linear block code can be realized in terms of matrix operation $\mathbf{y} = \mathbf{xG}$, where \mathbf{x} , the information sequence, is a k bit row vector, and \mathbf{G} , the generator matrix, is a k by n matrix. Every generator matrix \mathbf{G} can be converted to a generator matrix in row-echelon form. Then, by column operations, every generator matrix can be further converted to a generator matrix \mathbf{G}' denoting as $[\mathbf{I} \ \mathbf{P}]$ which is concatenated by a k by k identity \mathbf{I} and a k by $n - k$ matrix \mathbf{P} . We call this the systematic form of a generator matrix, and we call the code is systematic if the information sequence \mathbf{x} itself is part of codeword, $\mathbf{y}' = \mathbf{xG}' = (\mathbf{x}, \mathbf{r})$. The redundant bits \mathbf{r} contains $n - k$ bits, known as parity check bits. Different coefficient setting of \mathbf{G} generate different linear block codes. Famous examples of linear block codes are Hamming code, Reed-Solomon code, Hadamard code, and LDPC code.

- **Convolutional codes**

Convolutional codes was first proposed by Elias [17] in 1955. Compared to linear block codes, the convolutional codes contains memories. The encoder saves the previous information symbols in several memory elements. Combining both previous symbols and current symbol to generate the convolutional codewords. For an (n,k,m) convolutional code, every k -tuple information symbol will be encoded to an n -tuple codeword symbol by the encoder with memory order m . The error correction performance is determined by the memory order m and the code rate $R = k/n$.

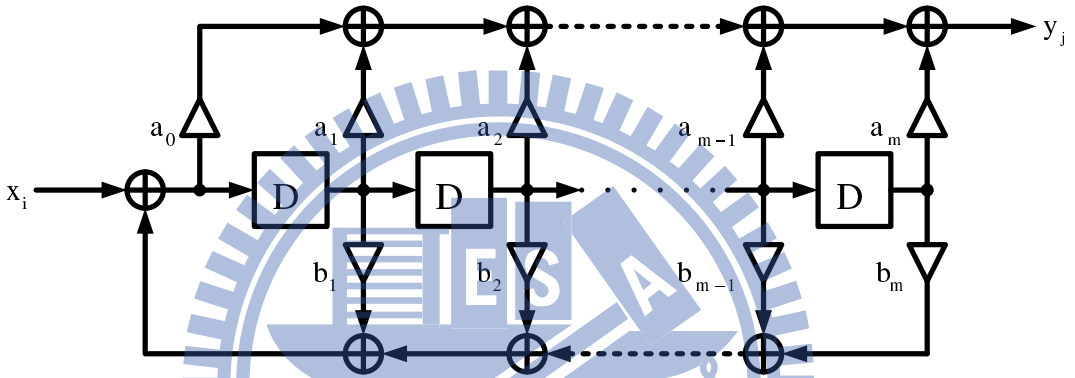


Figure 2.4: Block diagram of recursive convolutional encoder

Fig.2.4 demonstrates the structure of recursive convolutional encoder. The x_i is the i -th bit of the input symbol, $\mathbf{x} = (x_0, x_1, \dots, x_i, \dots, x_{k-1})$. The y_j is the j -th bit of the output symbol, $\mathbf{y} = (y_0, y_1, \dots, y_j, \dots, y_{n-1})$. In this figure, \mathbf{D} denotes a one tap delay operator. Besides, $a_0 \sim a_m$ and $b_1 \sim b_m$ determine the connections of circuit, each of them is either 0 or 1. We can use a fractional function $g_{i,j}(D)$ as (2.1) to express this encoder realization.

$$g_{i,j}(D) = \frac{a_0 + a_1D + \dots + a_mD^m}{1 + b_1D + b_2D^2 + \dots + b_mD^m} \quad (2.1)$$

Similar to linear block codes, the entire set of $k \times n$ generator polynomials $g_{i,j}(D)$ can be put together into a k by n generator matrix, given by

$$\mathbf{G}(D) = \begin{bmatrix} g_{0,0}(D) & g_{0,1}(D) & \cdots & g_{0,n-1}(D) \\ g_{1,0}(D) & g_{1,1}(D) & \cdots & g_{1,n-1}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0}(D) & g_{k-1,1}(D) & \cdots & g_{k-1,n-1}(D) \end{bmatrix} \quad (2.2)$$

Moreover apply some matrix operation on the generator matrix $\mathbf{G}(D)$ generally gets a systematic form $\mathbf{G}'(D)$. Here we use the (2,1,2) convolutional encoder as an example. The Fig.2.5(a) is a convolutional encoder with $\mathbf{G}(D) = [1 + D + D^2, 1 + D^2]$. The Fig. 2.5(b) is the equivalent recursive systematic convolutional encoder with $\mathbf{G}'(D) = [1, \frac{1+D^2}{1+D+D^2}]$.

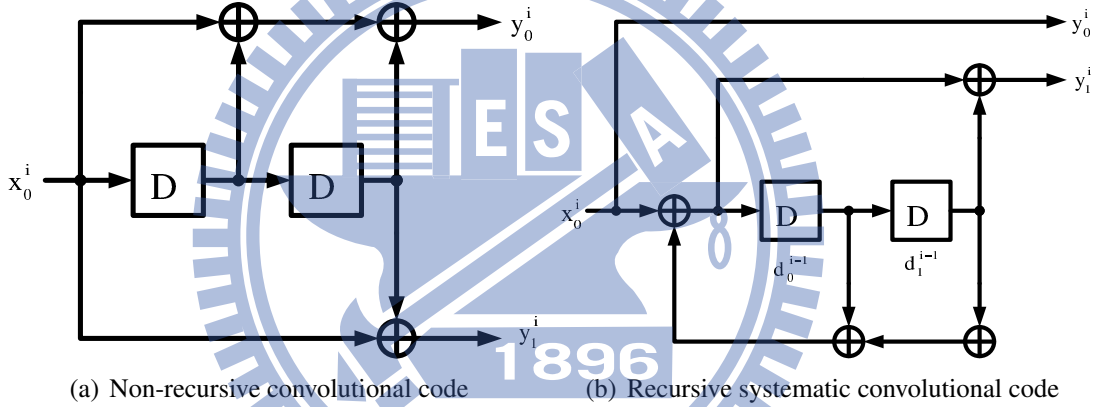


Figure 2.5: Block diagram of two equivalent (2,1,2) convolutional codes

Besides the matrix description of convolutional codes, a convolutional codes can be described in a state diagram. The state diagram comprises encoder states and state transition. The states are the contents of all delay elements. If the the memory order of encoder is m , the total number of states is 2^m . The state transition is caused by the input data. Here we use the (2,1,2) encoder (Fig.2.5(b)) as an example.

The previous content of memory elements are represented as d_0^{i-1} and d_1^{i-1} . All combinations of (d_0^{i-1}, d_1^{i-1}) represent previous encoder states noted as S_{i-1} . The ellipses of Fig.2.6 are the encoder states. The branches labelled with $y_0^i y_1^i$ indicate the states transi-

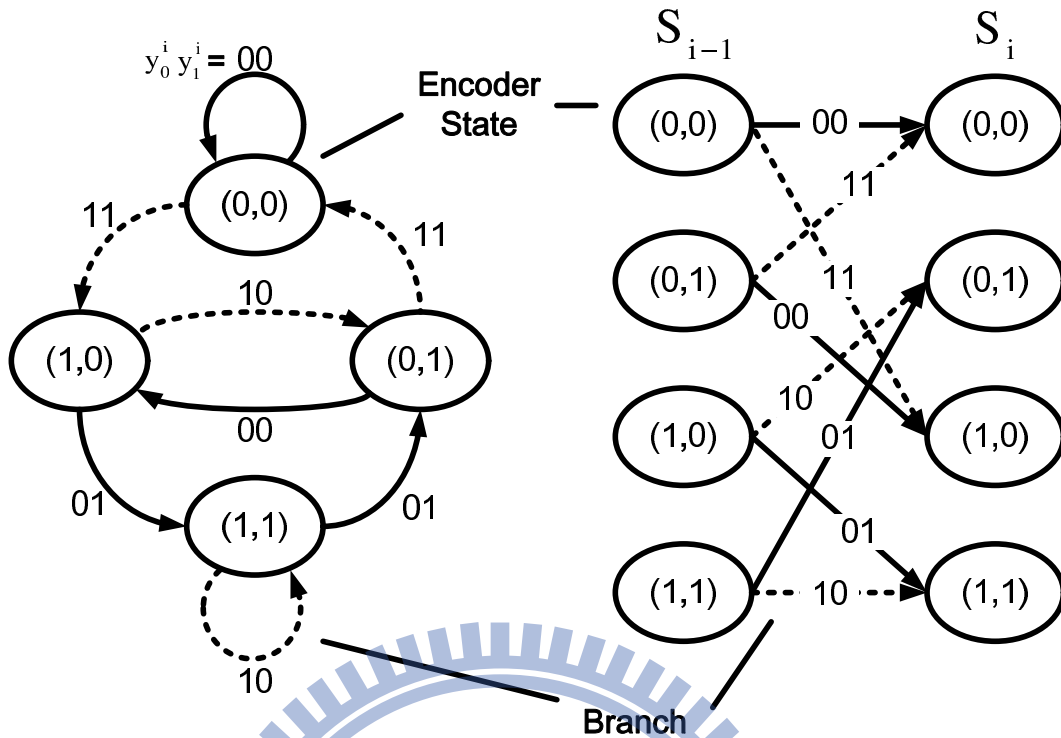


Figure 2.6: State diagram of the (2,1,2) recursive systematic convolutional codes

tions with the encoder i -th output $y_0^i y_1^i$. The solid branches and the dash branches indicate the i -th input of encoder $x_0^i = 0$ and $x_0^i = 1$ respectively.

The key advantage of convolutional codes over linear block codes is the usage of memory. Because of the memory usage, several consecutive codewords are mutually dependent. When the burst error happens, the decoder utilize the past code sequence to improve the error correcting capabilities. In contrast, the decoding process of the linear block codes is restricted to the decision of the current received codeword.

2.3 Soft Iterative Decoding

Since Shannon introduced the concept of the channel capacity and the noisy channel coding theorem in 1948, the goal of communication field has been to reach the Shannon limit with reasonable computation complexity and signal delay. Many coded modulation has been proposed such as linear block codes and convolutional codes to find the compromise between

the hardware cost and coding performance. However, all these approaches remained a wide gap between the Shannon limit and the performance of these coded modulation systems. The breakthrough was made when the TURBO code [18] was proposed in 1993. At the transmitter site, the scheme consists of two recursive systematic convolutional encoders separated by an interleaver. The iterative decoding strategy was applied at the receiver part, which exchanged so-called extrinsic information between two decoders in each iteration. This strategy is similar to the belief propagation algorithm for low density parity check codes [19].

2.3.1 Extrinsic Information

The key element of belief propagation algorithm is extrinsic information. To introduce the concept of extrinsic information, we use the normal graph [20]. The graph contains the vertices and the edges. The constraints are denoted by vertices. The edges connecting two vertices are denoted by ordinary edges. The edges connecting only one vertex are left edges.

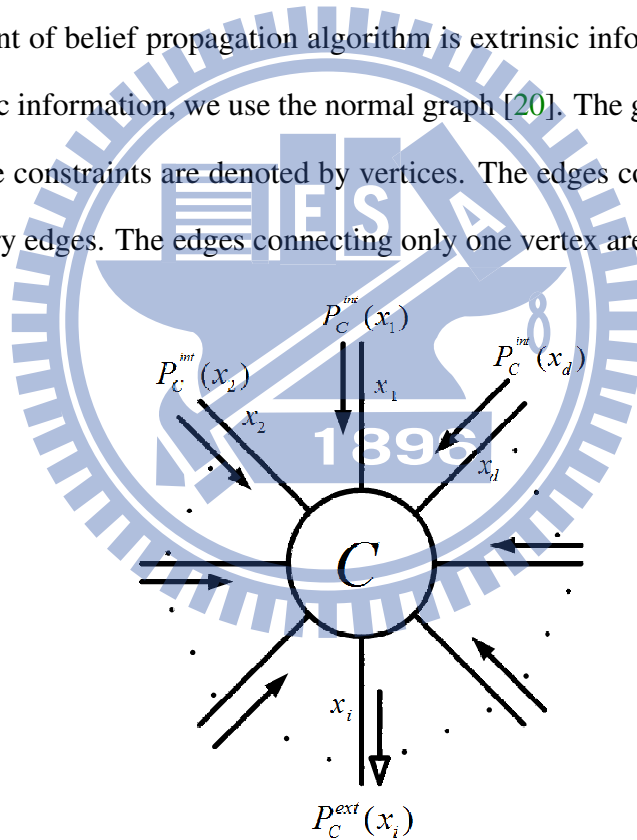


Figure 2.7: Graph representation of the extrinsic and the intrinsic probabilities

Fig.2.7 is a graph with one vertex and d left edges. There are d symbols, $x_1 \sim x_d$, and we assume they take values form the alphabet set A . We define a set \mathbf{X}_C^d which is a subspace of the n -dimensional vector space A^d ($\mathbf{X}_C^d \subset A^d$), and any d -tuple $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbf{X}_C^d$ will satisfy the constrain C . Now we consider the following conditional probability

$$P(x_i|C) \quad (2.3)$$

which is a posteriori probability (APP) of x_i under constraint C . According to the Bayes' theorem, we can expand (2.3) as

$$P(x_i|C) = \frac{P(C|x_i)P(x_i)}{P(C)} \quad (2.4)$$

The term $P(x_i)$ is the a priori probability and is also referred to the intrinsic probability for x_i , denoted by $P_C^{int}(x_i)$. The term $P(C|x_i)$ is termed the extrinsic probability with respect to C . The extrinsic probability is

$$\begin{aligned} P_C^{ext}(x_i) &= \rho_c \sum_{\substack{x_j, \forall j \neq i \\ \mathbf{x} \in \mathbf{X}_C^d}} P(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d) \\ &= \rho_c \sum_{\substack{x_j, \forall j \neq i \\ \mathbf{x} \in \mathbf{X}_C^d}} \prod_{j=1, j \neq i} P_C^{int}(x_j) \\ &= \rho_c P(C|x_i) \end{aligned} \quad (2.5)$$

we assume that the symbol variables x_1, x_2, \dots, x_d are independent, and ρ_c is a normalization constant. Consequently, the a posteriori probability in (2.4) can be written as

$$P_C^{post}(x_i) = P(x_i|C) = \rho_p P_C^{ext}(x_i) P_C^{int}(x_i) \quad (2.6)$$

where $\rho_p = (\rho_c P(C))^{-1}$. The log-likelihood ratio representation for (2.6) will be

$$L_C^{post}(x_i) = \ln \frac{P_C^{post}(x_i = 1)}{P_C^{post}(x_i = 0)} = \ln \frac{P_C^{ext}(x_i = 1)}{P_C^{ext}(x_i = 0)} + \ln \frac{P_C^{int}(x_i = 1)}{P_C^{int}(x_i = 0)} = L_C^{ext}(x_i) + L_C^{int}(x_i) \quad (2.7)$$

Which means the a posteriori probability can be decomposed into a priori knowledge and extrinsic information.

2.3.2 Exchange of Extrinsic Information

The iterative soft decoding relies on the exchange of extrinsic information. We use Fig. 2.8 to explain the exchange of extrinsic.

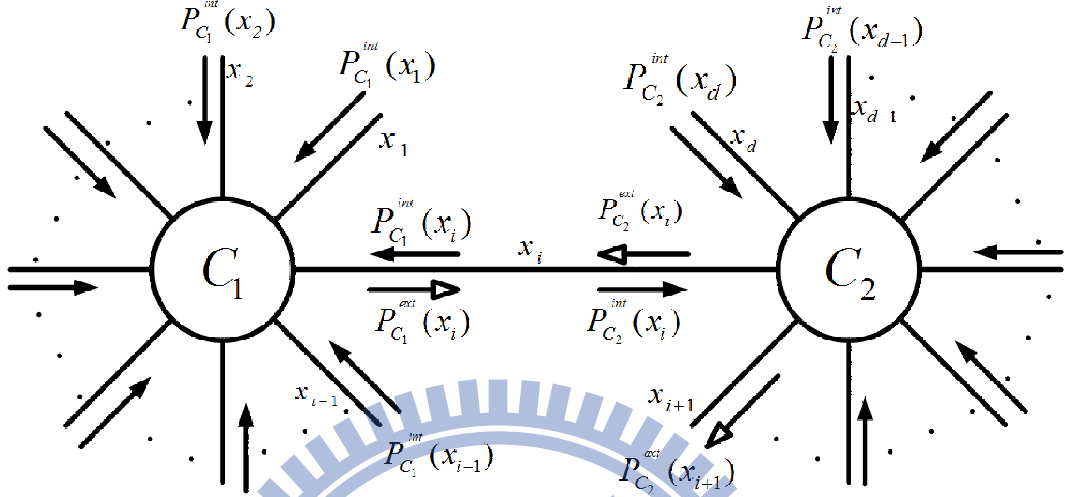


Figure 2.8: Graph representation of the extrinsic exchanging between two vertices

This graph contains two vertices, C_1 and C_2 . The vertex C_1 has $i - 1$ left edges, corresponding to the symbols $x_1 \sim x_{i-1}$. The vertex C_2 has $d - i$ left edges, corresponding to the symbols $x_{i+1} \sim x_d$. The symbol x_i is on the ordinary edge connecting two vertices, C_1 and C_2 . We also define two constraint sets $\mathbf{X}_{C_1}^i$ and $\mathbf{X}_{C_2}^{d-i+1}$ such that any $\mathbf{x}_1 = (x_1, x_2, \dots, x_i) \in \mathbf{X}_{C_1}^i$ and $\mathbf{x}_2 = (x_i, x_{i+1}, \dots, x_d) \in \mathbf{X}_{C_2}^{d-i+1}$ will satisfy C_1 and C_2 respectively. Now we consider the extrinsic probability of the symbol x_{i+1} under the constraint C_2 . Like the single vertex situation, the extrinsic information can be expressed as

$$P_{C_2}^{ext}(x_{i+1}) = \rho_2 \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P_{C_2}^{int}(x_i) \prod_{j=i+2}^d P_{C_2}^{int}(x_j) = \rho_2 P(C_2 | x_{i+1}) \quad (2.8)$$

according to (2.5). However, this from contains the intrinsic probability $P_{C_2}^{int}(x_i)$ on the ordinary edge. We consider C_1 further.

$$P_{C_1, C_2}^{ext}(x_{i+1}) = \rho_C P(C_1, C_2 | x_{i+1}) \quad (2.9)$$

According to (2.5), the conditional probability can be decomposed as

$$\begin{aligned}
P(C_1, C_2|x_{i+1}) &= \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P(C_1, C_2, x_i, x_{i+2}, \dots, x_d|x_{i+1}) \\
&= \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P(C_1, C_2, x_i, x_{i+1}, x_{i+2}, \dots, x_d|x_{i+1}) \\
&= \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P(C_1, C_2, \mathbf{x}_2|x_{i+1}) \\
&= \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P(C_2|C_1, \mathbf{x}_2)P(C_1, \mathbf{x}_2|x_{i+1})
\end{aligned} \tag{2.10}$$

while the term $P(C_2|C_1, \mathbf{x}_2) = P(C_2|\mathbf{x}_2) = 1$, for $x_{i+1} \in \mathbf{X}_{C_2}^{d-i+1}$, because \mathbf{x}_2 contains x_i .

Continue derive from (2.10)

$$\begin{aligned}
P(C_1, C_2, x_i, x_{i+2}, \dots, x_d|x_{i+1}) &= P(C_1, \mathbf{x}_2|x_{i+1}) \\
&= P(C_1|\mathbf{x}_2)P(\mathbf{x}_2|x_{i+1}) \\
&= P(C_1|x_i)P(x_i)P(x_{i+2}) \cdots P(x_d) \\
&= (\rho_1)^{-1} P_{C_1}^{ext}(x_i) P_{C_1, C_2}^{int}(x_i) \prod_{j=i+2}^d P_{C_2}^{int}(x_j)
\end{aligned} \tag{2.11}$$

We combine the derivations from (2.9), (2.10), and (2.11).

The extrinsic information is

$$\begin{aligned}
P_{C_1, C_2}^{ext}(x_{i+1}) &= \rho_C \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} (\rho_1)^{-1} P_{C_1}^{ext}(x_i) P_{C_1, C_2}^{int}(x_i) \prod_{j=i+2}^d P_{C_2}^{int}(x_j) \\
&= \rho_{C'} \sum_{\substack{\mathbf{x}_2 \setminus x_{i+1} \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i+1}}} P_{C_1}^{ext}(x_i) \prod_{j=i+2}^d P_{C_2}^{int}(x_j)
\end{aligned} \tag{2.12}$$

where $\rho_{C'} = \rho_C/(\rho_1|\mathbf{A}|)$, the a priori probability $P_{C_1, C_2}^{int}(x_i)$ can be initialized to $1/|\mathbf{A}|$. We

can find that if the extrinsic probability $P_{C_1}^{ext}(x_i)$ is available and

$$P_{C_2}^{int}(x_i) = P_{C_1}^{ext}(x_i) \quad (2.13)$$

we can evaluate all the extrinsic informations from symbol x_{i+1} to symbol x_d . Similarly, if we want to compute the extrinsic informations of $x_1 \sim x_{i-1}$, we must calculate the extrinsic probability $P_{C_2}^{ext}(x_i)$ first. And set

$$P_{C_1}^{int}(x_i) = P_{C_2}^{ext}(x_i) \quad (2.14)$$

The process of (2.13) and (2.14) is the exchange of the information, widely used in soft iterative decoding. The message passing of ordinary edge can be simply represented by

$$\mu_{C_1 \rightarrow C_2}(x_i) = P_{C_1}^{ext}(x_i) = \rho_1 \sum_{\substack{\mathbf{x}_1 \setminus x_i \\ \mathbf{x}_1 \in \mathbf{X}_{C_1}^{i-1}}} \prod_{j=1}^{i-1} P_{C_1}^{int}(x_j) \quad (2.15)$$

$$\mu_{C_2 \rightarrow C_1}(x_i) = P_{C_2}^{ext}(x_i) = \rho_2 \sum_{\substack{\mathbf{x}_2 \setminus x_i \\ \mathbf{x}_2 \in \mathbf{X}_{C_2}^{d-i}}} \prod_{j=i+1}^d P_{C_2}^{int}(x_j) \quad (2.16)$$

Meanwhile, the a posteriori probability of x_i under two constrains according to (2.5) and (2.6) can be written as

$$\begin{aligned} P_{C_1, C_2}^{post}(x_i) &= P(x_i | C_1, C_2) \\ &= \rho_{p'} P_{C_1, C_2}^{ext}(x_i) P_{C_1, C_2}^{int}(x_i) \\ &= \rho_{p'} \rho_{c'} P(C_1, C_2 | x_i) P_{C_1, C_2}^{int}(x_i) \\ &= \rho_{p'} \rho_{c'} P(C_1 | x_i) P(C_2 | x_i) P_{C_1, C_2}^{int}(x_i) \\ &= \frac{\rho_{p'} \rho_{c'}}{\rho_{c_1} \rho_{c_2}} P_{C_1}^{ext}(x_i) P_{C_2}^{ext}(x_i) P_{C_1, C_2}^{int}(x_i) \end{aligned} \quad (2.17)$$

The fourth equality holds because of the Markov chain [21], $P(C_1, C_2 | x_i) = P(C_1 | x_i) P(C_2 | x_i)$. The log-likelihood ratio representation for (2.17) will be

$$L_{C_1, C_2}^{post}(x_i) = \ln \frac{P_{C_1, C_2}^{post}(x_i = 1)}{P_{C_1, C_2}^{post}(x_i = 0)} = L_{C_1}^{ext}(x_i) + L_{C_2}^{ext}(x_i) + L_{C_1, C_2}^{int}(x_i) \quad (2.18)$$

2.4 Soft-Output Channel Decoding

The iterative soft decoding is based on the iterative exchange of extrinsic information about the data bit x_i . In order to decode iteratively, the extrinsic information of x_i after channel decoding $P_{CD}^{ext}(x_i)$ is needed. Here, the soft-output channel decoding for linear binary block codes and convolutional codes will be introduced.

- **Soft decoding of linear binary block codes**

An (n, k) linear block code can also be specified by an $n - k$ by k matrix \mathbf{H} . The \mathbf{y} is an n -tuple codeword if and only if $\mathbf{y} \cdot \mathbf{H}^T = 0$. The matrix \mathbf{H} is called a parity check matrix. In the following parity check matrix of (7,4) Hamming code, for example,

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

and

$$(y_0, y_1, y_2, y_3, y_4, y_5, y_6) \cdot \mathbf{H}^T = (0, 0, 0) \quad (2.20)$$

where $(y_0, y_1, y_2, y_3, y_4, y_5, y_6) = \mathbf{y}$. The three parity check equations of (7,4) Hamming codes are

$$\begin{aligned} y_0 + y_1 + y_3 + y_4 &= 0 \\ y_0 + y_2 + y_3 + y_5 &= 0 \\ y_1 + y_2 + y_3 + y_6 &= 0 \end{aligned} \quad (2.21)$$

these three equation can be represented by the bipartite graph [22].

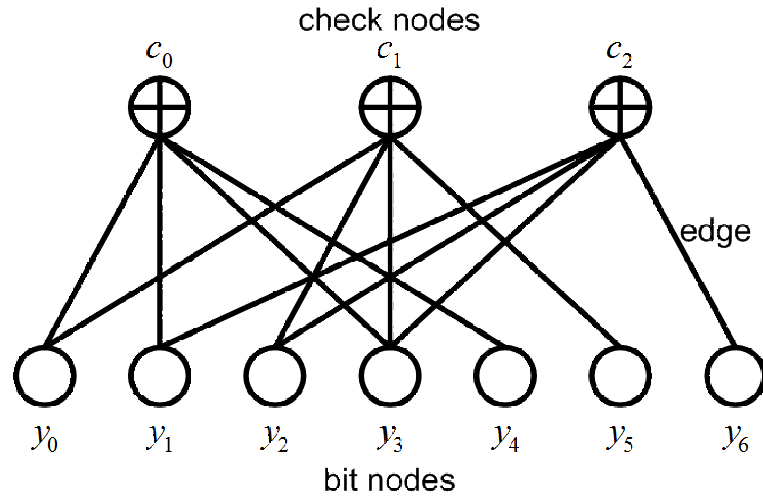


Figure 2.9: The bipartite graph for the (7,4) Hamming code

The bipartite graph Fig.2.9 contains 7 bit nodes and 3 check nodes. The number of edges connected to each check node which is corresponding to the each column weight of \mathbf{H} . Here, we define two types of set, $\mathbf{B}(i)$ and $\mathbf{C}(j)$. The set $\mathbf{B}(i)$ denotes all the bit node indexes connecting to check node c_i . The set $\mathbf{C}(j)$ denotes all the check node indexes connecting to check node y_j . So, we have

$$\begin{aligned} \mathbf{B}(0) &= \{0, 1, 3, 4\} \\ \mathbf{B}(1) &= \{0, 2, 3, 5\} \\ \mathbf{B}(2) &= \{1, 2, 3, 6\} \end{aligned} \tag{2.22}$$

and

$$\begin{aligned} \mathbf{C}(0) &= \{0, 1\} \\ &\vdots \\ \mathbf{C}(6) &= \{2\} \end{aligned} \tag{2.23}$$

Now we consider a (n,k) linear binary block code, the corresponding \mathbf{H} is an $n - k$ by n matrix. There are $n - k$ parity check equations, $c_0 \sim c_{n-k}$. The extrinsic information of

y_i satisfy c_l is denoted as $P_{c_l}^{ext}(y_i)$.

According to (2.5)

$$\begin{aligned} P_{c_l}^{ext}(y_i) &= P(c_l|y_i) \\ &= P\left(\sum_{\substack{j \neq i \\ j \in B(l)}} y_j = 0\right) \end{aligned} \quad (2.24)$$

In binary case, [23] have already derived a formula

$$\begin{aligned} P_{c_l}^{ext}(y_i = 0) &= \frac{1}{2} \left[1 + \prod_{\substack{j \neq i \\ j \in B(l)}} (1 - 2P^{int}(y_j = 1)) \right] \\ P_{c_l}^{ext}(y_i = 1) &= \frac{1}{2} \left[1 - \prod_{\substack{j \neq i \\ j \in B(l)}} (1 - 2P^{int}(y_j = 1)) \right] \end{aligned} \quad (2.25)$$

In log-likelihood representation is

$$\begin{aligned} L_{c_l}^{ext}(y_i) &= \ln \frac{P_{c_l}^{ext}(y_i = 0)}{P_{c_l}^{ext}(y_i = 1)} \\ &= \ln \frac{1 + \prod_{\substack{j \neq i \\ j \in B(l)}} (1 - 2P^{int}(y_j = 1))}{1 - \prod_{\substack{j \neq i \\ j \in B(l)}} (1 - 2P^{int}(y_j = 1))} \\ &= \ln \frac{1 + \prod_{\substack{j \neq i \\ j \in B(l)}} \tanh\left(\frac{L^{int}(y_j)}{2}\right)}{1 - \prod_{\substack{j \neq i \\ j \in B(l)}} \tanh\left(\frac{L^{int}(y_j)}{2}\right)} \\ &= 2 \tanh^{-1}\left(\prod_{\substack{j \neq i \\ j \in B(l)}} \tanh\left(\frac{L^{int}(y_j)}{2}\right)\right) \end{aligned} \quad (2.26)$$

Finally, the extrinsic information of y_i satisfy $c_0 \sim c_{n-k}$ is

$$\begin{aligned} L_{c_0 \sim c_{n-k}}^{ext}(y_i) &= \sum_{l \in C(i)} L_{c_l}^{ext}(y_i) \\ &= 2 \sum_{l \in C(i)} \tanh^{-1}\left(\prod_{\substack{j \neq i \\ j \in B(l)}} \tanh\left(\frac{L^{int}(y_j)}{2}\right)\right) \end{aligned} \quad (2.27)$$

If the systematic encoder of (n, k) linear binary block code is applied, the extrinsic information of x_i after channel decoding $L_{CD}^{ext}(x_i)$ is

$$L_{CD}^{ext}(x_i) = L_{c_0 \sim c_{n-k}}^{ext}(y_i) \quad (2.28)$$

According to (2.7), the soft-output of channel decoder, also known as a posteriori knowledge, is

$$L_{CD}^{post}(x_i) = L^{int}(x_i) + L_{CD}^{ext}(x_i) \quad (2.29)$$

- **Soft decoding of convolutional codes**

There are several algorithms to decode convolutional codes, hard decoding algorithms are including Viterbi algorithm [24], and sequential decoding algorithm [25]. The Viterbi algorithm is maximum likelihood (ML) decoding and is highly parallelizable, suitable for high speed application. The sequential decoding algorithm usually used when the memory order m is large. Both of them gives the most likely codeword. Moreover the maximum a posteriori (MAP) decoding algorithm [26] and the soft output Viterbi algorithm (SOVA) [27] are utilized in soft iterative decoding.

The MAP decoding algorithm was first introduced by L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv in 1974, also named BCJR algorithm. The BCJR decoder minimize the bit error rate of convolutional codes and provides maximum a posteriori probabilities for each data bit x_i which are generated from the mutual dependence of the codeword sequence \mathbf{y} and received codeword sequence $\tilde{\mathbf{y}}$. Although the BCJR algorithm is optimal algorithm however the large computation complexity remains a issue in practical application. The more detail about BCJR algorithm will be given in Chapter 3 explicitly.

The SOVA algorithm is a modified version of Viterbi algorithm, and J. Hagenauer introduced several versions to approach low computation complexity of soft output trellis decoding. Differ from Viterbi algorithm, the SOVA modifies the path metric with a priori knowledge of input sequence to produce the reliability of output sequence. The modi-

fied metric calculation of SOVA uses Euclidean distance instead of Hamming distance. In comparison to BCJR algorithm, the SOVA is a suboptimal approximation, and some quality degrades due to approximation.

2.5 Softbit Source Decoding

Source decoding is performed by inverting each step of encoding procedure, reversing the index assignment, quantization, and decorrelation, see Fig.2.2. The conventional parameter decoding procedure is shown in Fig.2.10.

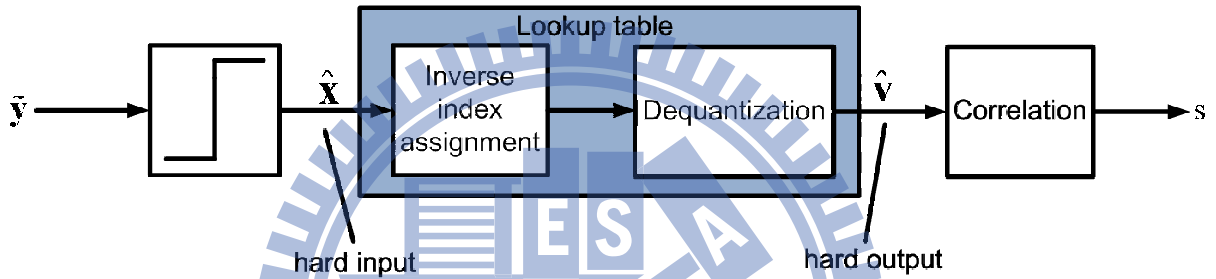


Figure 2.10: Conventional parameter decoding by hard decision

In conventional parameter decoding, the sign of received sequence \tilde{y} decides the sequence \hat{x} . The inverse index assignment and dequantization step merge into lookup table. The bit pattern \hat{x} maps into the corresponding reproduced value/vector \hat{v} in codebook $\mathbf{U}(\hat{v} \in \mathbf{U})$ by a lookup table. Finally, the eliminate redundancy is reinserted, and the reconstruction signal \hat{s} is obtained.

Considering high compression rate situation, the source parameter codec is very sensitive to channel noise, we better use the reliability information for all receiver stage. In 2001, softbit source decoding [28] was introduced to error concealment in speech and audio signal. Now we discuss the softbit source decoding scheme in Fig.2.11.

Generally, the softbit source decoding is composed by utilization of source statistics and parameter estimation. The source statistics can be measured in transmitter cite. After using source statistics, a posteriori knowledge $P(\mathbf{x}|\tilde{y})$ will be used in parameter estimation, and the extrinsic

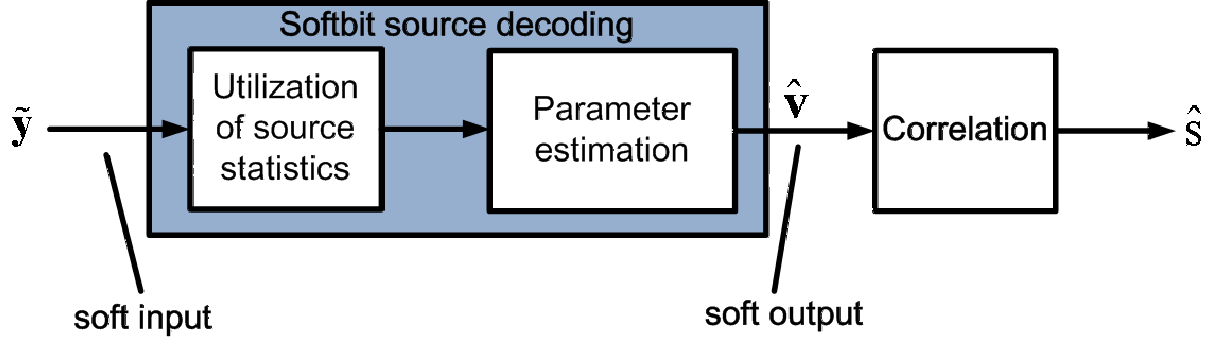


Figure 2.11: Softbit source decoding by parameter estimation

information of x_i noted by $P_{SBSD}^{ext}(x_i)$ will be used in iterative soft decoding. In contrast to conventional parameter decoding, softbit source decoding performs parameter estimation instead of lookup table to reconstruct $\hat{\mathbf{v}}, \hat{\mathbf{v}} \in \mathbb{R}$.

The first part of softbit source decoding is utilization of source statistics, which will produce a posteriori knowledge and extrinsic information.

According to (2.4), the a posteriori knowledge $P(\mathbf{x}|\tilde{\mathbf{y}})$ can be written as

$$P(\mathbf{x}|\tilde{\mathbf{y}}) = \frac{P(\tilde{\mathbf{y}}|\mathbf{x})P(\mathbf{x})}{P(\tilde{\mathbf{y}})} \quad (2.30)$$

and the extrinsic information $P_{SBSD}^{ext}(x_i)$ can be expressed as

$$P_{SBSD}^{ext}(x_i) = \sum_{\mathbf{x}} P(\mathbf{x}) \prod_{\substack{x_j \neq x_i \\ x_j \in \mathbf{x}}} P_{SBSD}^{int}(x_j) \quad (2.31)$$

The term $P(\mathbf{x})$ in (2.31) and (2.31) is the a priori information of index \mathbf{x} , which is dependent on different source types. Now we consider $P(\mathbf{x})$ in three source types.

The index-level a priori probability of uniformly distributed parameter is

$$\begin{aligned} P(\mathbf{x}) &= P(\mathbf{u}|\mathbf{U}) \\ &= \frac{1}{|\mathbf{U}|} \end{aligned} \quad (2.32)$$

The first equality holds because of the one-to-one mapping of \mathbf{x} and \mathbf{u} . The second equality

holds because of the equal probably codeword appearance of codebook \mathbf{U} , and the codeword number of codebook \mathbf{U} is $|\mathbf{U}|$.

The index-level a priori probability of non-uniformly distributed parameter is

$$P(\mathbf{x}) = P(\mathbf{u}|\mathbf{U}) \quad (2.33)$$

(2.33) indicates that the index-level a priori probability $P(\mathbf{x})$ of a particular source codec \mathbf{u} is independent from past or future parameters.

If a source have memory from one past parameter, we call this source a 1st order Markov source. Now, we consider a source with the 1st order Markov property.

The index-level a priori probability of 1st order Markov source is

$$\begin{aligned} P(\mathbf{x}_t) &= P(\mathbf{x}_t|\mathbf{x}_{t-1})P(\mathbf{x}_{t-1}) \\ &= P(\mathbf{u}_t|\mathbf{u}_{t-1})P(\mathbf{u}_{t-1}|\mathbf{U}) \end{aligned} \quad (2.34)$$

Where the subscribe $t, t - 1$ is the time index. (2.34) shows that the current index-level a priori probability $P(\mathbf{x}_t)$ is composed of the past index-level a priori probability $P(\mathbf{x}_{t-1})$ and the index transition probability $P(\mathbf{x}_t|\mathbf{x}_{t-1})$. The accuracy of $P(\mathbf{x}_t)$ depends on the level we know about the source and the computation effort we pay. Higher order Markov model of the source usually gets a better $P(\mathbf{x}_t)$. To compromise the accuracy $P(\mathbf{x}_t)$ and the computation effort, the 1st Markov model of source is considered. For unknown distribution source, we evaluate the index a priori probability and index transition probability by adopting training process.

The second part of softbit source decoding is parameter estimation. According to estimation theory [29], the parameter estimation is more beneficial to decode while comparing the conventional hard decision. The goal of parameter estimation is to minimize the overall distortion $E[\|\mathbf{v} - \hat{\mathbf{v}}\|^2]$. The optimal estimation rule for a single codec parameters satisfies the minimum mean squared error estimation rule [29].

Minimum mean squared error (MMSE) estimation:

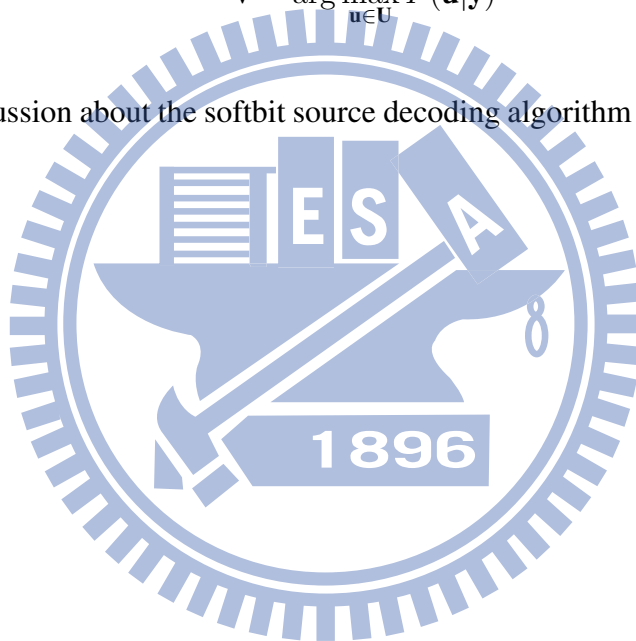
$$\hat{\mathbf{v}} = \sum_{\mathbf{u} \in \mathcal{U}} \mathbf{u} \cdot P(\mathbf{u}|\tilde{\mathbf{y}}) \quad (2.35)$$

However, the squared error criterion is not appropriate in all situation [29]. Sometimes, we want to minimize the error rate in estimations. For this, the maximum a posteriori is proposed.

Maximum a posteriori (MAP) estimation:

$$\hat{\mathbf{v}} = \arg \max_{\mathbf{u} \in \mathcal{U}} P(\mathbf{u}|\tilde{\mathbf{y}}) \quad (2.36)$$

The detail discussion about the softbit source decoding algorithm will be derived in Chapter 3.



Chapter 3

Joint Source-Channel Coding Scheme

In 1948, Shannon presented the *source-channel separation theorem*. However, the design and development of the perfect source and channel coding techniques is complicated because of the hardware cost and infinite signal delay. The soft iterative decoding has originally been efficient decoding technique for channel codes. The soft iterative decoding has already been reviewed in Section 2.3. In addition to the original application of decoding channel codes, its outstanding efficiency made the soft iterative decoding also popular for iterative source-channel decoding [28]. Unlike conventional system, an interleaver is added between the source encoder and channel encoder to generate the uncorrelated channel encoder input. Similarly, the de-interleaver is added in receiver to recover the order of original source. Additionally, the decoder with feedback scheme takes advantage of extrinsic information exchanging. For unknown data distribution, the measurement of source statistics at transmitter site is needed to softbit source decoder. Combining the quantization and the measurement of source statistics, we use vector quantization technique. For high transmission rate requirement, the channel encoder is convolutional encoder with puncture scheme.

At first, we will define the detail mathematical notations of overall system. Starting from the transmitter, the process of encoders will be described. At the receiver part, we will discuss two decoders separately.

3.1 System Overview

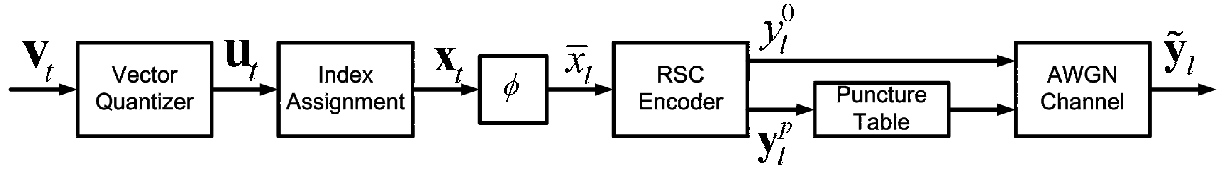


Figure 3.1: Block diagram of the transmitter

Consider the block diagram of the transmitter in Fig.3.1 At time instant t , we have d real valued source sequence \mathbf{v}_t as follows:

$$\mathbf{v}_t = \{v_t(1), v_t(2), \dots, v_t(d)\}$$

which is quantized by a d -dimension vector quantizer produces \mathbf{u}_t as follows:

$$\mathbf{u}_t = \{u_t(1), u_t(2), \dots, u_t(d)\}$$

and bit mapped to an M -bit sequence \mathbf{x}_t as follows:

$$\mathbf{x}_t = \{x_t(1), x_t(2), \dots, x_t(m), \dots, x_t(M)\}$$

where $x_t(m) \in \{+1, -1\}$, we assume all bits are pre-modulated using Binary Phase Shift Keying (BPSK).

Define the set:

$$\mathbf{X}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$$

which is a set containing T sequences, are passed through a interleaver ϕ to generate an uncorrelated bit pattern $\bar{\mathbf{x}}$:

$$\bar{\mathbf{x}} = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_l, \dots, \bar{x}_L\}$$

where $L = MT$ is the size of the interleaver. Each bit \bar{x}_l corresponds to a specific bit $x_t(m)$:

$$\bar{x}_l = x_t(m), \begin{cases} m = 1, 2, \dots, M \\ t = 1, 2, \dots, T \end{cases} \quad (3.1)$$

The interleaved bit pattern $\bar{\mathbf{x}}$ is the information bits for channel encoder. We use the $(n,1)$ recursive systematic convolutional (RSC) code as the mother code. Thus, each input bit \bar{x}_l will generate n output bits \mathbf{y}_l , which can be separated by a systematic bit y_l^0 and parity check bits \mathbf{y}_l^p :

$$\mathbf{y}_l = \{y_l^0, y_l^1, \dots, y_l^{n-1}\} = \{y_l^0, \mathbf{y}_l^p\} = \{\bar{x}_l, \mathbf{y}_l^p\} \quad (3.2)$$

Define the puncture set \mathbf{P} :

$$\mathbf{P} = \{(i, j) | \text{The parity check bit } y_j^i \text{ is eliminated before transmission.}\}$$

The output bits after convolutional encoder will be punctured to fewer bits $\mathbf{Y}_1^{L'}$:

$$\mathbf{Y}_1^{L'} = \{y_j^i | i \in \{0, \dots, n-1\}, j \in \{1, \dots, L'\}, (i, j) \notin \mathbf{P}\}$$

The length L' includes the information length L and the termination length for the RSC encoder. The punctured output $\mathbf{Y}_1^{L'}$ is transmitted through the channel. If the noise of channel in Fig.3.1 is considered to be additive white Gaussian noise (AWGN) with a mean of zero, a variance of $\sigma^2 = N_0/2$ with BPSK modulated, then the channel transition probability can be characterized by the conditional probability density function (PDF):

$$P(\tilde{y}_j^i | y_j^i) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{(\tilde{y}_j^i - y_j^i)^2}{2\sigma^2}\right], (i, j) \notin \mathbf{P} \\ \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{(-y_j^i)^2}{2\sigma^2}\right], (i, j) \in \mathbf{P} \end{cases} \quad (3.3)$$

The second equality holds by assuming the bit "0" and bit "1" is equal probable, so we let the value zero be a received soft value.

$\{\mathbf{v}|Q(\mathbf{v}) = \mathbf{u}(i)\}$ of the input vectors mapping into the i -th codeword.

Now, let $\mathbf{V} = \{V(1), V(2), \dots, V(d)\}$ be a real random vector with a cumulative distribution function $F(\mathbf{v}) = P(V(i) \leq v(i); i = 1, 2, \dots, d)$.

The expected distortion of a quantizer Q with the codebook \mathbf{U} and the partition \mathbf{P} applied to the random \mathbf{V} is shown as follows.

$$D(Q) = D(\{\mathbf{U}, \mathbf{S}\}) = E[d(\mathbf{V}, Q(\mathbf{V}))] \quad (3.6)$$

Where $d(\mathbf{V}, Q(\mathbf{V}))$ is a distortion measure function, the most common one is squared error distortion.

$$d(\mathbf{V}, Q(\mathbf{V})) = \|\mathbf{V} - Q(\mathbf{V})\|^2 \quad (3.7)$$

A quantizer is said to be optimal if it minimizes the expected distortion, and it must satisfy two conditions. Given an optimal quantizer Q with the codebook $\mathbf{U} = \{\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(2^M - 1)\}$ and the partition $\mathbf{S} = \{S_i | i = 0, 1, \dots, 2^M - 1\}$.

The partition \mathbf{S} must satisfy the *nearest neighbour condition*.

$$S_i = \{\mathbf{V} | d(\mathbf{V}, \mathbf{u}(i)) \leq d(\mathbf{V}, \mathbf{u}(j)), \forall j\} \quad (3.8)$$

Associated with each S_i , is a nearest neighbor region called *Voronoi region* [], and it is defined by:

$$\begin{aligned} \bigcup_{i=0}^{2^M-1} S_i &= \mathbb{R}^d \\ \bigcap_{i=0}^{2^M-1} S_i &= \emptyset \end{aligned} \quad (3.9)$$

On the other hand, the codebook \mathbf{U} must satisfy the *centroid condition*.

$$\mathbf{u}(i) = \frac{1}{|S_i|} \sum_{\mathbf{v} \in S_i} \mathbf{v} \quad (3.10)$$

Algorithm 1 shows the LBG algorithm for design an M -bit vector quantizer with a training

sequence.

Algorithm 1: M -bit quantization algorithm

Input: A training sequence $\{\mathbf{v}_j | j = 0, \dots, n - 1\}$, an M -bit initial codebook \mathbf{U}_0 , and distortion threshold ϵ

Output: The optimal codebook \mathbf{U}_m

1 **Initialization:**

2 $m = 0$

3 $D_{-1} = \infty$

4 **Partition:**

5 Compute the minimum distortion partition \mathbf{S} of the training sequence satisfies *nearest neighbour condition*.

6

7 Compute the average distortion $D_m = \frac{1}{n} \sum_{j=0}^{n-1} \min_{\mathbf{u} \in \mathbf{U}_m} \|\mathbf{v}_j - \mathbf{u}\|^2$.

8 **if** $(D_{m-1} - D_m)/D_m \leq \epsilon$ **then**

9 **Stop:**

10 Return \mathbf{U}_m .

11 **end**

12 **Codebook Updating:**

13 Compute the optimal codebook \mathbf{U}_{m+1} for \mathbf{S} satisfies *centroid condition*.

14 $m = m + 1$, **GOTO Partition**

Because the LBG algorithm is local optimal, the initialization of codebook is significantly important. There are several ways to choose \mathbf{U}_0 . One method is to choose 2^M input vector as the codeword. The second method we use is splitting method, and the corresponding algorithm is shown in **Algorithm 2**.

Here we use the 2-dimensional space with 4096 zero mean and unit variance training sequence for example [30]. In Fig.3.3 and Fig.3.4, the input training vectors are marked with dots, the codewords are marked with circles, and the Voronoi regions are separated with boundary lines. Fig.3.3(a) shows the 2-bit quantizer design with the 1-bit initial codebook. We perform 1-bit quantization algorithm, and the 1-bit optimal codebook(Fig.3.3(b)) is produced. The 1-bit optimal codebook splits as the 2-bit initial codebook(Fig.3.4(a)). Finally, we perform the 2-bit quantization algorithm, the 2-bit optimal codebook(Fig.3.4(b)) will be produced.

The quantizer design algorithm for unknown distribution source needs a training sequence, and the training sequence must be large enough at least 1000 times the number of codewords. Except for generating the optimal codebook, the training sequence also measures the source

Algorithm 2: Initial by splitting algorithm

Input: A training sequence $\{\mathbf{v}_j | j = 0, \dots, n - 1\}$, and perturbation vector ϵ

Output: The initial codebook $\mathbf{U}_0(2^M)$ for the M -bit quantization algorithm

1 Initialization:

2 $N = 0$

3 $\mathbf{U}_0(1) = E[X]$

4 Splitting:

5 The codebook $\mathbf{U}_0(2^N) = \{\mathbf{u}(i) | i = 0, \dots, 2^N - 1\}$, "split" each codeword $\mathbf{u}(i)$ into two close codeword $\mathbf{u}(i) + \epsilon$ and $\mathbf{u}(i) - \epsilon$.

6 The new codebook is $\mathbf{U}_0(2^{N+1}) = \{\mathbf{u}(i) + \epsilon, \mathbf{u}(i) - \epsilon | i = 0, \dots, 2^N - 1\}$.

7

8 $N = N + 1$

9 **if** $N = M$ **then**

10 **Stop:**

11 Return $\mathbf{U}_0(2^N)$.

12 **end**

13 **Run N -bit Quantization Algorithm:**

14 Compute the optimal codebook $\mathbf{U}_0(2^N)$.

15 **GOTO Splitting**

statistics. The index-level a priori knowledge $P(\mathbf{u}(i))$ can be measured by the partition $\mathbf{S} = \{S_i | i = 0, \dots, 2^M - 1\}$:

$$P(\mathbf{u}(i)) = \frac{|S_i|}{\sum_{j=0}^{2^M-1} |S_j|} \quad (3.11)$$

where $|S_i|$ denotes the number of elements in the set S_i . The index-level a priori knowledge $P(\mathbf{u}(i))$ assists soft iterative decoding at the receiver site.

3.3 Index Assignment with Pseudo-Gray Code

A pseudo-gray code is an assignment of M -bit binary indexes to 2^M point in a Euclidean space so that Hamming distance between two points corresponds closely to the Euclidean distance. A quantization for noisy channel is a long-standing problem. One approach is adding redundancy bits for channel coding. On the other hand, performance gain can be achieved by assign channel words to codeword of quantization. Review the quantization process referred in Fig.2.3. We now consider the case of a noisy channel. A block diagram depicting a noisy channel vector quantizer is shown in Fig.3.5.

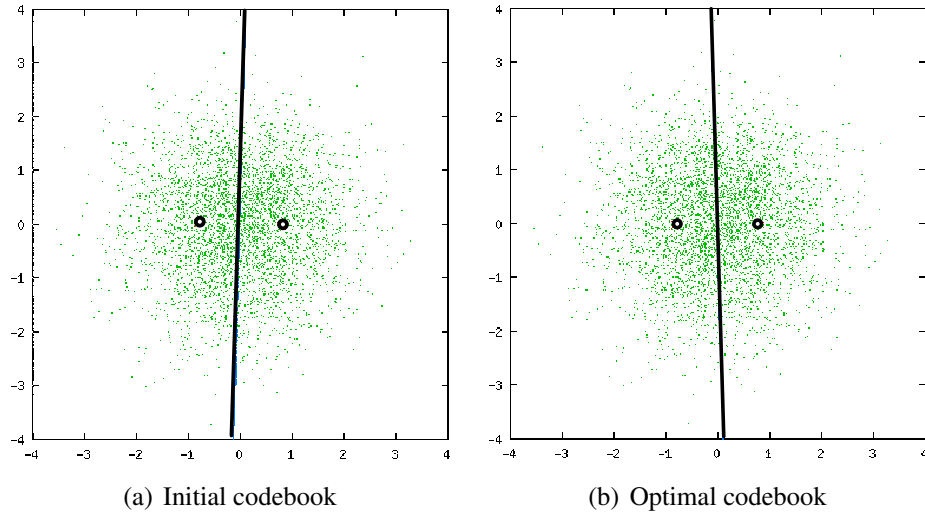


Figure 3.3: 1-bit vector quantizer design with LBG algorithm

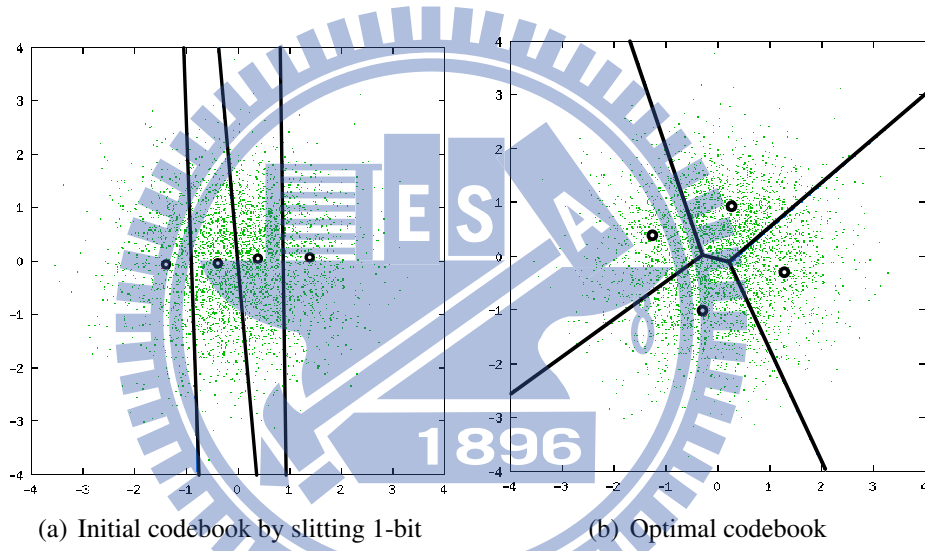


Figure 3.4: 2-bit vector quantizer design with LBG algorithm

In Fig.3.5, the encoder E maps random d -dimensional vector \mathbf{V} into a M -bit index i taking value from the set $\{0, 1\}^M$. The index assignment π is a permutation function, which maps i to another index of $\{0, 1\}^M$. If the index in the set $\{0, 1\}^M$ are transmitted across a noisy channel, the receiver site will generally receive different index in the set $\{0, 1\}^M$.

The channel can be represented by a mapping $\tau : \{0, 1\}^M \rightarrow \{0, 1\}^M$ given by (3.12).

$$\tau(i) = i \oplus \eta, i \in \{0, 1\}^M \quad (3.12)$$

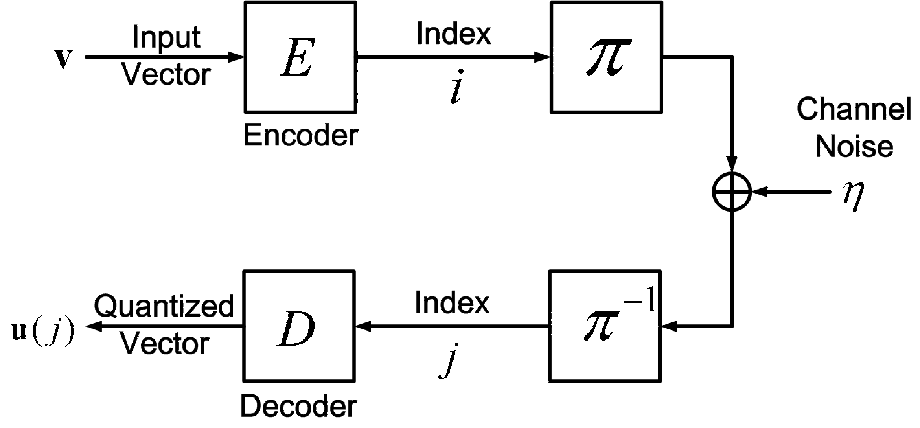


Figure 3.5: Block diagram of VQ on noisy channel

Where η is a random variable taking value from the set $\{0, 1\}^M$, and the \oplus is bitwise exclusive-or operation.

The π^{-1} is an inverse permutation function, and the decoder D maps the index j into the corresponding codeword $\mathbf{u}(j)$.

An M -bit d -dimensional noisy channel vector quantizer, Q_π , maps \mathbb{R}^d to the set of codeword $\mathbf{U} = \{\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(2^M - 1)\}$ given by

$$Q_\pi = D \circ \pi^{-1} \circ \tau \circ \pi \circ E \quad (3.13)$$

According to (3.6), the expected distortion of a noisy channel quantizer, Q_π , applied to the random vector \mathbf{V} is

$$\begin{aligned} D(Q_\pi) &= E[d(\mathbf{V}, Q_\pi(\mathbf{V}))] \\ &= E[d(\mathbf{V}, \mathbf{u}(j))] \end{aligned} \quad (3.14)$$

If the squared error distortion is used as error measure function, (3.14) can be written as

$$\begin{aligned}
D(Q_\pi) &= E[d(\mathbf{V}, \mathbf{u}(j))] \\
&= E \|\mathbf{V} - \mathbf{u}(j)\|^2 \\
&= E \|\mathbf{V} - \mathbf{u}(i)\|^2 + E \|\mathbf{u}(i) - \mathbf{u}(j)\|^2
\end{aligned} \tag{3.15}$$

The third equality holds because of the optimal noisy channel vector quantizer satisfy the centroid condition. Due to the quantization error on noiseless channel, the quantity $E \|\mathbf{V} - \mathbf{u}(i)\|^2$ is independent of the assignment of codeword. To minimize $D(Q_\pi)$, is to find the permutation π , such that

$$\arg \min_{\pi} E \|\mathbf{u}(i) - \mathbf{u}(j)\|^2 \tag{3.16}$$

For the sake of convenience, we define the D_π is the expect distortion depends on the permutation π .

$$D_\pi = E \|\mathbf{u}(i) - \mathbf{u}(j)\|^2 \tag{3.17}$$

In fact, D_π can be further derived as

$$\begin{aligned}
D_\pi &= E \|\mathbf{u}(i) - \mathbf{u}(j)\|^2 \\
&= \sum_{i=0}^{2^M-1} P(\mathbf{u}(i)) \sum_{j=0}^{2^M-1} P(\mathbf{u}(j)) \|\mathbf{u}(i) - \mathbf{u}(j)\|^2 \\
&= \sum_{i=0}^{2^M-1} P(\mathbf{u}(i)) \sum_{t=0}^{2^M-1} P(\eta = t) \|\mathbf{u}(i) - \mathbf{u}(\pi^{-1}(\pi(i) \oplus t))\|^2
\end{aligned} \tag{3.18}$$

The notation t is a particular noise sequence, $t \in \{0, 1\}^M$. In any memoryless binary symmetric channel, the probability $P(\eta = t)$ only depends on the weight of t , which can be written as

$$P(\eta = t) = q(W(t)). \tag{3.19}$$

Where $W(\bullet)$ is the weight function, and $q(\bullet)$ is channel dependent function representing the probability of error bits.

For each index $q \in \{0, 1\}^M$ and each integer m with $0 \leq m \leq M$, define the m th-neighbour set of q as

$$N^m(q) = \left\{ r \mid r \in \{0, 1\}^M, H(q, r) = m \right\} \quad (3.20)$$

where $H(\bullet, \bullet)$ is the Hamming distance function.

Use the property of (3.19) and definition of (3.20), continue derive (3.18)

$$\begin{aligned} D_\pi &= \sum_{i=0}^{2^M-1} P(\mathbf{u}(i)) \sum_{m=0}^M q(m) \sum_{z \in N^m(0)} \|\mathbf{u}(i) - \mathbf{u}(\pi^{-1}(\pi(i) \oplus z))\|^2 \\ &= \sum_{i=0}^{2^M-1} \sum_{m=0}^M q(m) P(\mathbf{u}(i)) \sum_{w \in N^m(i)} \|\mathbf{u}(i) - \mathbf{u}(\pi^{-1}(w))\|^2 \end{aligned} \quad (3.21)$$

Define the m -th cost of $\mathbf{u}(i)$, with respect to the permutation function π as

$$C_\pi^{(m)}(\mathbf{u}(i)) = P(\mathbf{u}(i)) \sum_{w \in N^m(i)} \|\mathbf{u}(i) - \mathbf{u}(\pi^{-1}(w))\|^2 \quad (3.22)$$

Define the cost function of $\mathbf{u}(i)$, with respect to the permutation function π as

$$C_\pi(\mathbf{u}(i)) = \sum_{m=0}^M q(m) C_\pi^{(m)}(\mathbf{u}(i)) \quad (3.23)$$

When the channel error probability is sufficiently small, which means $q(1) \gg q(2), q(3), \dots, q(M)$.

We neglect the effect of the multiple error by assuming $q(m) = 0$, when $m \geq 2$.

With this assumption, we simplify (3.23) and (3.21) as

$$C_\pi(\mathbf{u}(i)) \approx q(1) C_\pi^{(1)}(\mathbf{u}(i)) \quad (3.24)$$

and

$$D_\pi \approx q(1) \sum_{i=0}^{2^M-1} C_\pi^{(1)}(\mathbf{u}(i)) \quad (3.25)$$

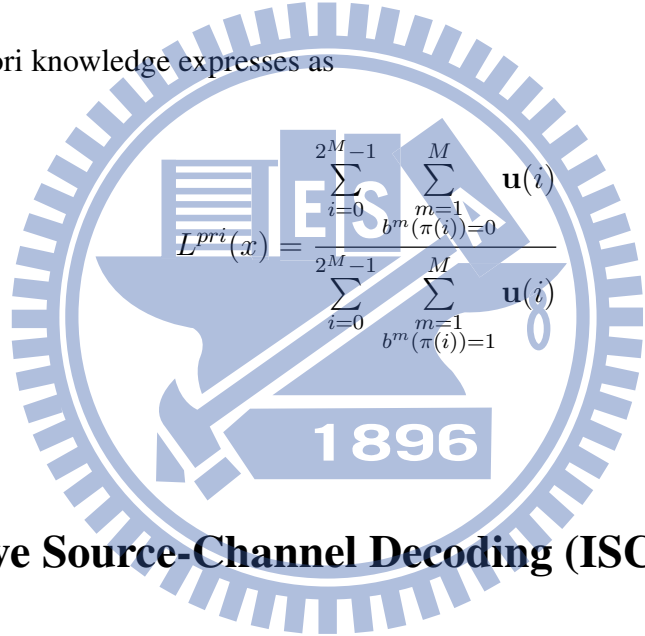
To minimize D_π in this case, the permutation function should be

$$\arg \min_{\pi} \sum_{i=0}^{2^M-1} C_{\pi}^{(1)}(\mathbf{u}(i)) \quad (3.26)$$

The pseudo-Gray code on a vector quantization codebook is performed by binary switch algorithm. The flow chart of the binary switch algorithm is shown in Fig.3.6. Because of the one-to-one mapping and the bit-level a priori knowledge can be calculated with the index-level a priori knowledge.

For each index $q \in \{0, 1\}^M$ and integer m with $1 \leq m \leq M$, define the m -th bit of sequence q as $b^m(q) \in \{0, 1\}$.

The bit-level a priori knowledge expresses as

$$L^{pri}(x) = \frac{\sum_{i=0}^{2^M-1} \sum_{m=1}^M \mathbf{u}(i) b^m(\pi(i)=0)}{\sum_{i=0}^{2^M-1} \sum_{m=1}^M \mathbf{u}(i) b^m(\pi(i)=1)} \quad (3.27)$$


3.4 Iterative Source-Channel Decoding (ISCD) Algorithm

The algorithm of the ISCD scheme can be divided into several steps as follows:

1. Initialization

- set iteration counter to $i = 0$, and the exit condition i_{max} .
- set extrinsic information of source decoding to $L_{SBS D}^{ext}(\bar{x}_l) = 0$.
- specify the a priori knowledge $L^{pri}(\bar{x}_l)$ and $\ln P_{SBS D}^{ext}(\mathbf{x}_t | \mathbf{x}_{t-1})$.

2. Map all received sequence to channel-related L -values $L_{ch}(\bar{x}_l)$ with $l = 1, 2, \dots, L$.

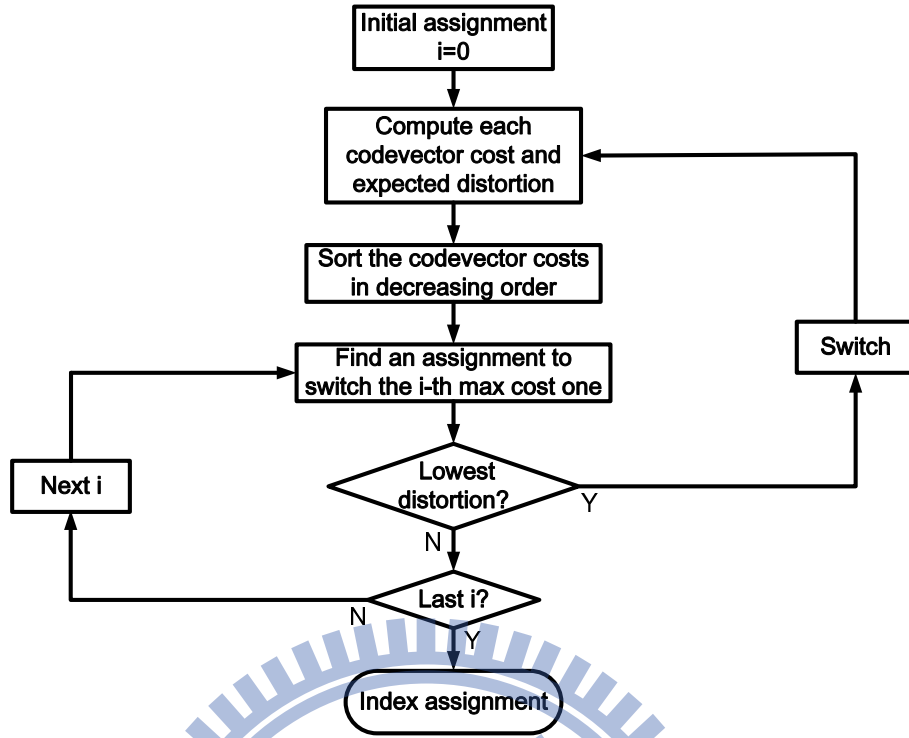


Figure 3.6: Flow chart of the binary switch algorithm

3. Update the channel a priori knowledge:

$$L_{CD}^{pri}(\bar{x}_l) = L^{pri}(\bar{x}_l) + L_{SBSD}^{ext}(\bar{x}_l) \quad (3.28)$$

4. Perform BCJR algorithm

generate the extrinsic information $L_{CD}^{ext}(\bar{x}_l)$ and the a posteriori knowledge $L^{post}(\bar{x}_l)$.

5. Deinterleave

6. Update the source a priori knowledge:

$$L_{SBSD}^{pri}(\mathbf{x}_t | \mathbf{x}_{t-1}) = L^{pri}(\mathbf{x}_t | \mathbf{x}_{t-1}) + \sum_{\substack{m=1 \\ \mathbf{x}_t=i}} L_{CD}^{ext}(x_t(m)) \quad (3.29)$$

$$L_{SBSD}^{pri}(x_t(\lambda)) = L^{pri}(x_t(\lambda)) \quad (3.30)$$

7. Perform SBSB algorithm

generate the extrinsic information $L_{SBS D}^{ext}(x_t(\lambda))$ and the a posteriori knowledge $L^{post}(x_t(\lambda))$.

8. Increase iteration counter $i = i + 1$
9. If the exit condition $i = i_{max}$ is reached, then continue, else interleave, proceed with Step 3.
10. Compute the index a posteriori probability $P^{post}(\mathbf{x}_t)$

first transform log domain to probability domain

$$P^{post}(x_t(\lambda) = +1) = \frac{e^{L^{post}(x_t(\lambda))}}{1 + e^{L^{post}(x_t(\lambda))}} \quad (3.31)$$

ignore the mutual dependences in \mathbf{x}_t

$$P^{post}(\mathbf{x}_t) = \prod_{\lambda=1}^M P^{post}(x_t(\lambda)) \quad (3.32)$$

11. Perform MMSE estimation to estimate source codec parameter $\hat{\mathbf{v}}_t$

$$\hat{\mathbf{v}}_t = \sum_{i=0}^{2^M-1} \mathbf{u}(i) P^{post}(\mathbf{x}_t = i) \quad (3.33)$$

where the $\mathbf{u}(i)$ is the i -th codeword in the codebook.

3.4.1 BCJR Algorithm for Channel Decoding

To derive the BCJR algorithm, we need to make some assumptions about data transmission. These assumptions are given in Section 3.1. Considering the received data sequence form channel, the BCJR algorithm can generate the a posteriori probability of each transmitted symbol as

$$P(\bar{x}_l | \tilde{\mathbf{Y}}_1^{L'}) \quad (3.34)$$

The APP is further used to compute the log-likelihood ratio

$$L^{post}(\bar{x}_l) = \ln \frac{P(\bar{x}_l = +1 | \tilde{\mathbf{Y}}_1^{L'})}{P(\bar{x}_l = -1 | \tilde{\mathbf{Y}}_1^{L'})} \quad (3.35)$$

The LLR in (3.35) can be rewrite as

$$\begin{aligned} L^{post}(\bar{x}_l) &= \ln \frac{P(\bar{x}_l = +1, \tilde{\mathbf{Y}}_1^{L'}) / \tilde{\mathbf{Y}}_1^{L'}}{P(\bar{x}_l = -1, \tilde{\mathbf{Y}}_1^{L'}) / \tilde{\mathbf{Y}}_1^{L'}} \\ &= \ln \frac{P(\bar{x}_l = +1, \tilde{\mathbf{Y}}_1^{L'})}{P(\bar{x}_l = -1, \tilde{\mathbf{Y}}_1^{L'})} \end{aligned} \quad (3.36)$$

The $P(\bar{x}_l = +1)$ and $P(\bar{x}_l = -1)$ have their own state transitions in the trellis diagram, so we have the equation

$$P(\bar{x}_l) = \sum_{(S_{l-1}, S_l)} P(\bar{x}_l, S_{l-1}, S_l) \quad (3.37)$$

Note that (S_{l-1}, S_l) represents the state transition from S_{l-1} to S_l .

With (3.37), the LLR can be modified to

$$\begin{aligned} L^{post}(\bar{x}_l) &= \ln \frac{\sum_{(S_{l-1}, S_l)} P(\bar{x}_l = +1, S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'})}{\sum_{(S_{l-1}, S_l)} P(\bar{x}_l = -1, S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'})} \\ &= \ln \frac{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'})}{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'})} \end{aligned} \quad (3.38)$$

Note that \mathbf{B}_l^+ is the state index set of branches caused by input $\bar{x}_l = +1$ between S_{l-1} and S_l , and the \mathbf{B}_l^- is the state index set of branches caused by input $\bar{x}_l = -1$ between S_{l-1} and S_l .

The joint probability $P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'})$ can be decomposed as (3.39) with Bayes's rule.

$$P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'}) = P(S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \times P(S_l, \tilde{y}_l | S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \times P(\tilde{\mathbf{Y}}_{l+1}^{L'} | S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{l-1}, \tilde{y}_l) \quad (3.39)$$

We can simplify these two conditional probabilities by the Markov process property of trellis diagram. We rewrite (3.39) as (3.40)

$$P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^{L'}) = P(S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \times P(S_l, \tilde{y}_l | S_{l-1}) \times P(\tilde{\mathbf{Y}}_{l+1}^{L'} | S_l) \quad (3.40)$$

Now we define three functions:

$$\alpha(S_l) = \ln P(S_l, \tilde{\mathbf{Y}}_1^l) \quad (3.41)$$

$$\gamma(S_{l-1}, S_l) = \ln P(S_l, \tilde{\mathbf{y}}_l | S_{l-1}) \quad (3.42)$$

$$\beta(S_l) = \ln P(\tilde{\mathbf{Y}}_{l+1}^l | S_l) \quad (3.43)$$

where $\alpha(S_{l-1})$ is *forward metric*, $\gamma(S_{l-1}, S_l)$ is *branch metric*, and $\beta(S_l)$ is *backward metric*. With these definitions, (3.40) can be written as

$$P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^l) = \exp(\alpha(S_{l-1})) \times \exp(\gamma(S_{l-1}, S_l)) \times \exp(\beta(S_l)) \quad (3.44)$$

We substitute (3.44) for (3.38), the LLR will be

$$L^{post}(\tilde{x}_l) = \ln \sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} \exp(\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)) - \ln \sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} \exp(\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)) \quad (3.45)$$

Moreover, The definition of $\alpha(S_l)$ can be extended as

$$\begin{aligned} \exp(\alpha(S_l)) &= P(S_l, \tilde{\mathbf{Y}}_1^l) \\ &= \sum_{S_{l-1}} P(S_{l-1}, S_l, \tilde{\mathbf{Y}}_1^l) \\ &= \sum_{S_{l-1}} P(S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) P(S_l, \tilde{\mathbf{y}}_l | S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \\ &= \sum_{S_{l-1}} P(S_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) P(S_l, \tilde{\mathbf{y}}_l | S_{l-1}) \\ &= \sum_{S_{l-1}} \exp(\alpha(S_{l-1})) \times \exp(\gamma(S_{l-1}, S_l)) \end{aligned} \quad (3.46)$$

Then we apply natural logarithm both sides in (3.47).

$$\alpha(S_l) = \ln \sum_{S_{l-1}} \exp(\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l)) \quad (3.47)$$

The above equation is a forward recursion for $\alpha(S_l)$, such recursive method needs an initial condition as

$$\alpha(S_0) = \begin{cases} 0, S_0 = S(0) \\ -\infty, S_0 \neq S(0) \end{cases} \quad (3.48)$$

We make similar extension to $\beta(S_l)$.

$$\begin{aligned} \exp(\beta(S_l)) &= P(\tilde{\mathbf{Y}}_{l+1}^{L'} | S_l) \\ &= \sum_{S_{l+1}} P(S_{l+1}, \tilde{\mathbf{Y}}_{l+1}^{L'} | S_l) \\ &= \sum_{S_{l+1}} P(S_{l+1}, \tilde{\mathbf{y}}_{l+1} | S_l) P(\tilde{\mathbf{Y}}_{l+2}^{L'} | S_l, S_{l+1}, \tilde{\mathbf{y}}_{l+1}) \\ &= \sum_{S_{l+1}} P(S_{l+1}, \tilde{\mathbf{y}}_{l+1} | S_l) P(\tilde{\mathbf{Y}}_{l+2}^{L'} | S_{l+1}) \\ &= \sum_{S_{l+1}} \exp(\gamma(S_l, S_{l+1})) \times \exp(\beta(S_{l+1})) \end{aligned} \quad (3.49)$$

Then we apply natural logarithm both sides in (3.50).

$$\beta(S_l) = \ln \sum_{S_{l+1}} \exp(\gamma(S_l, S_{l+1}) + \beta(S_{l+1})) \quad (3.50)$$

The above equation is a backward recursion for $\beta(S_l)$, we also need an initial condition as

$$\beta(S_{L'}) = \begin{cases} 0, S_{L'} = S(0) \\ -\infty, S_{L'} \neq S(0) \end{cases} \quad (3.51)$$

Additionally, the branch metric in (3.42) can be

$$\begin{aligned}
\exp(\gamma(S_{l-1}, S_l)) &= P(S_l, \tilde{\mathbf{y}}_l | S_{l-1}) \\
&= \frac{P(S_{l-1}, S_l, \tilde{\mathbf{y}}_l)}{P(S_{l-1})} \\
&= \frac{P(S_{l-1}, S_l)}{P(S_{l-1})} \times \frac{P(S_{l-1}, S_l, \tilde{\mathbf{y}}_l)}{P(S_{l-1}, S_l)} \\
&= P(S_l | S_{l-1}) P(\tilde{\mathbf{y}}_l | S_{l-1}, S_l) \\
&= P(\bar{x}_l) P(\tilde{\mathbf{y}}_l | \mathbf{y}_l) \\
&= P(\bar{x}_l) P(\tilde{y}_l^0 | \bar{x}_l) P(\tilde{\mathbf{y}}_l^p | \mathbf{y}_l^p)
\end{aligned} \tag{3.52}$$

To find the $P(\bar{x}_l)$, we need the a priori information for channel decoding represented by

$$L_{CD}^{pri}(\bar{x}_l) = \ln \frac{P(\bar{x}_l = +1)}{P(\bar{x}_l = -1)} \tag{3.53}$$

and the a priori probability will be

$$P(\bar{x}_l = \pm 1) = \frac{e^{\pm L_{CD}^{pri}(\bar{x}_l)}}{1 + e^{\pm L_{CD}^{pri}(\bar{x}_l)}} = \frac{e^{-L_{CD}^{pri}(\bar{x}_l)/2}}{1 + e^{-L_{CD}^{pri}(\bar{x}_l)}} \times e^{\bar{x}_l L_{CD}^{pri}(\bar{x}_l)/2} = A_t e^{\bar{x}_l L_{CD}^{pri}(\bar{x}_l)/2} \tag{3.54}$$

where A_t is independent of the value \bar{x}_l . With (3.4), the $P(\tilde{y}_l^0 | \bar{x}_l)$ can be modified to

$$\begin{aligned}
P(\tilde{y}_l^0 | \bar{x}_l) &= \frac{1}{\sqrt{2\pi}\sigma} \times e^{-\frac{(\tilde{y}_l^0 - \bar{x}_l)^2}{2\sigma^2}} \\
&= \frac{1}{\sqrt{2\pi}\sigma} \times e^{-\frac{(\tilde{y}_l^0)^2 + (\bar{x}_l)^2}{2\sigma^2}} \times e^{\frac{\tilde{y}_l^0 \times \bar{x}_l}{\sigma^2}} \\
&= B_t \times e^{\frac{L_c \times \tilde{y}_l^0 \times \bar{x}_l}{2}}
\end{aligned} \tag{3.55}$$

and with (3.5), $P(\tilde{\mathbf{y}}_l^p | \mathbf{y}_l^p)$ become as

$$\begin{aligned}
P(\tilde{\mathbf{y}}_l^p | \mathbf{y}_l^p) &= \prod_{i=1}^{n-1} \frac{1}{\sqrt{2\pi}\sigma} \times e^{-\frac{(\tilde{y}_l^i - y_l^i)^2}{2\sigma^2}} \\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{n-1} \times e^{-\frac{\sum_{i=1}^{n-1} [(\tilde{y}_l^i)^2 + (y_l^i)^2]}{2\sigma^2}} \times e^{\frac{\sum_{i=1}^{n-1} [\tilde{y}_l^i \times y_l^i]}{\sigma^2}} \\
&= C_t \times e^{\frac{L_c \sum_{i=1}^{n-1} [\tilde{y}_l^i \times y_l^i]}{2}}
\end{aligned} \tag{3.56}$$

Here we do zero adding as de-puncture step, which means $\tilde{y}_l^i = 0, (i, l) \in \mathbf{P}$. Because \tilde{y}_l^i is the same for all state transitions and $y_l^i = \pm 1$, the B_t and C_t are constants. Additionally, the channel reliability value L_c is $2/\sigma^2$, and will be $4E_s/N_0$ for the AWGN channel [23]. As a result, we will find that A_t , B_t , and C_t will be cancelled out in the LLR of (3.45). Finally, we get

$$\gamma(S_{l-1}, S_l) = \frac{1}{2} \bar{x}_l L_{CD}^{pri}(\bar{x}_l) + \frac{1}{2} L_c \tilde{y}_l^0 \bar{x}_l + \frac{1}{2} L_c \sum_{i=1}^{n-1} (\tilde{y}_l^i \times y_l^i) \tag{3.57}$$

By substituting (3.57) for the $\gamma(S_{l-1}, S_l)$ of (3.38), we have

$$\begin{aligned}
L^{post}(\bar{x}_l) &= \ln \frac{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} e^{\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)}}{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} e^{\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)}} \\
&= \ln \frac{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} e^{\frac{1}{2}(L_{CD}^{pri}(\bar{x}_l) + L_c \tilde{y}_l^0)} e^{\alpha(S_{l-1}) + \frac{L_c}{2} \sum_{i=1}^{n-1} \tilde{y}_l^i \times y_l^i + \beta(S_l)}}{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} e^{\frac{1}{2}(-L_{CD}^{pri}(\bar{x}_l) - L_c \tilde{y}_l^0)} e^{\alpha(S_{l-1}) + \frac{L_c}{2} \sum_{i=1}^{n-1} \tilde{y}_l^i \times y_l^i + \beta(S_l)}} \\
&= L_{CD}^{pri}(\bar{x}_l) + L_c \tilde{y}_l^0 + \ln \frac{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} e^{\alpha(S_{l-1}) + \frac{L_c}{2} \sum_{i=1}^{n-1} \tilde{y}_l^i \times y_l^i + \beta(S_l)}}{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} e^{\alpha(S_{l-1}) + \frac{L_c}{2} \sum_{i=1}^{n-1} \tilde{y}_l^i \times y_l^i + \beta(S_l)}} \tag{3.58} \\
&= L_{CD}^{pri}(\bar{x}_l) + L_c \tilde{y}_l^0 + \ln \frac{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} e^{\alpha(S_{l-1}) + \gamma^{ext}(S_{l-1}, S_l) + \beta(S_l)}}{\sum_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} e^{\alpha(S_{l-1}) + \gamma^{ext}(S_{l-1}, S_l) + \beta(S_l)}} \\
&= L_{CD}^{pri}(\bar{x}_l) + L_{ch}(\bar{x}_l) + L_{CD}^{ext}(\bar{x}_l)
\end{aligned}$$

(3.58) shows that the $L^{post}(\bar{x}_l)$ can be decomposed to three terms, the a priori knowledge $L_{CD}^{pri}(\bar{x}_l)$, the channel-related L -value $L_{ch}(\bar{x}_l)$, and the extrinsic information generated by channel decoding $L_{CD}^{ext}(\bar{x}_l)$.

The BCJR algorithm is often approximated to Log-MAP or Max-Log-MAP algorithm [31].

Considering Jacobian function [32]

$$\ln(e^{x_1} + e^{x_2}) \triangleq \max^*(e^{x_1}, e^{x_2}) = \max(e^{x_1}, e^{x_2}) + \ln(1 + e^{-|x_1 - x_2|}) \tag{3.59}$$

and its extension

$$\ln(e^{x_1} + e^{x_2} + e^{x_3} + \dots + e^{x_n}) = \max^*(\max^*(\dots \max^*(e^{x_1}, e^{x_2}), e^{x_3}), \dots, e^{x_n}) \tag{3.60}$$

In (3.59), if the logarithm term is small enough, we can drop it. The \max^* operation would be replaced by normal max operation.

$$\max^*(e^{x_1}, e^{x_2}) \approx \max(e^{x_1}, e^{x_2}) \quad (3.61)$$

With max function,(3.47) and (3.50) can be expressed in a simpler form

$$\alpha(S_l) = \max_{S_{l-1}} [\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l)] \quad (3.62)$$

$$\beta(S_l) = \max_{S_{l+1}} [\gamma(S_l, S_{l+1}) + \beta(S_{l+1})] \quad (3.63)$$

Thus, the a posteriori LLR alters

$$L^{post}(\bar{x}_l) = \max_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} [\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)] - \max_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} [\alpha(S_{l-1}) + \gamma(S_{l-1}, S_l) + \beta(S_l)] \quad (3.64)$$

and the extrinsic LLR alters

$$L_{CD}^{ext}(\bar{x}_l) = \max_{(S_{l-1}, S_l) \in \mathbf{B}_l^+} [\alpha(S_{l-1}) + \gamma^{ext}(S_{l-1}, S_l) + \beta(S_l)] - \max_{(S_{l-1}, S_l) \in \mathbf{B}_l^-} [\alpha(S_{l-1}) + \gamma^{ext}(S_{l-1}, S_l) + \beta(S_l)] \quad (3.65)$$

In conclusion, the channel decoder performs BCJR algorithm with the channel-related L -value $L_{ch}(\bar{x}_l)$, $L_c \tilde{\mathbf{y}}_l^p$ and bit-level a priori information $L_{CD}^{pri}(\bar{x}_l)$ to generate bit-level a posteriori LLR $L^{post}(\bar{x}_l)$ and bit-level extrinsic information $L_{CD}^{ext}(\bar{x}_l)$.

3.4.2 Softbit Source Decoding

In this section, we will discuss the softbit source decoding (SBSD) algorithm [28], which exploit the 1st-order a priori knowledge source information. In this algorithm, the index transition a priori probability $P_{SBSD}^{pri}(\mathbf{x}_t | \mathbf{x}_{t-1})$ needs to be trained by training data at transmitter. The

index-level APP is derived as

$$\begin{aligned}
P(\mathbf{x}_t = i | \tilde{\mathbf{X}}_1^T) &= P(\mathbf{x}_t | \tilde{\mathbf{X}}_1^t) \\
&= P(\mathbf{x}_t | \tilde{\mathbf{X}}_1^{t-1}, \tilde{\mathbf{x}}_t) \\
&= C \cdot P(\tilde{\mathbf{x}}_t | \mathbf{x}_t = i) \sum_{j=0}^{2^M-1} P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) \cdot P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})
\end{aligned} \tag{3.66}$$

where

$$P(\tilde{\mathbf{x}}_t | \mathbf{x}_t = i) = \prod_{\substack{m=1 \\ \{\mathbf{x}_t=i\}}}^M P(\tilde{x}_t(m) | x_t(m)) \tag{3.67}$$

With (3.67), (3.66) becomes

$$P(\mathbf{x}_t = i | \tilde{\mathbf{X}}_1^T) = C \cdot \prod_{\substack{m=1 \\ \{\mathbf{x}_t=i\}}}^M P(\tilde{x}_t(m) | x_t(m)) \cdot \sum_{j=0}^{2^M-1} P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) \cdot P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1}) \tag{3.68}$$

Then, consider the marginal APP of specific bit $x_t(\lambda)$, which is the combination of all possible index-level APP.

Thus, the bit-level LLR can be expressed as

$$\begin{aligned}
L^{post}(x_t(\lambda)) &= L(x_t(\lambda) | \tilde{\mathbf{X}}_1^T) = \frac{\sum_{\mathbf{x}_t: x_t(\lambda)=+1} P(\mathbf{x}_t | \tilde{\mathbf{X}}_1^T)}{\sum_{\mathbf{x}_t: x_t(\lambda)=-1} P(\mathbf{x}_t | \tilde{\mathbf{X}}_1^T)} \\
&= \ln \frac{\sum_{\mathbf{x}_t: x_t(\lambda)=+1} \prod_{m=1}^M P(\tilde{x}_t(m) | x_t(m)) \sum_{j=0}^{2^M-1} P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})}{\sum_{\mathbf{x}_t: x_t(\lambda)=-1} \prod_{m=1}^M P(\tilde{x}_t(m) | x_t(m)) \sum_{j=0}^{2^M-1} P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})} \\
&= L_{ch}(x_t(\lambda)) \\
&\quad + \max_{\mathbf{x}_t: x_t(\lambda)=+1} \left[\sum_{\substack{m=1 \\ m \neq \lambda}}^M \frac{L_c \tilde{x}_t(m) x_t(m)}{2} + \max_j [\ln P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) + \ln P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})] \right] \\
&\quad - \max_{\mathbf{x}_t: x_t(\lambda)=-1} \left[\sum_{\substack{m=1 \\ m \neq \lambda}}^M \frac{L_c \tilde{x}_t(m) x_t(m)}{2} + \max_j [\ln P_{SBSD}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) + \ln P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})] \right]
\end{aligned} \tag{3.69}$$

Similar as channel decoding, the bit-level APP can be decomposed to three terms as

$$L^{post}(x_t(\lambda)) = L_{SBS D}^{pri}(x_t(\lambda)) + L_{ch}(x_t(\lambda)) + L_{SBS D}^{ext}(x_t(\lambda)) \quad (3.70)$$

By substitute (3.70) in (3.69), we get the extrinsic information generated by SBS D

$$\begin{aligned} & L_{SBS D}^{ext}(x_t(\lambda)) \\ = & \max_{\mathbf{x}_t: x_t(\lambda)=+1} \left[\sum_{\substack{m=1 \\ m \neq \lambda}}^M \frac{L_c \tilde{x}_t(m) x_t(m)}{2} + \max_j [\ln P_{SBS D}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) + \ln P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})] \right] \\ & - \max_{\mathbf{x}_t: x_t(\lambda)=-1} \left[\sum_{\substack{m=1 \\ m \neq \lambda}}^M \frac{L_c \tilde{x}_t(m) x_t(m)}{2} + \max_j [\ln P_{SBS D}^{pri}(\mathbf{x}_t = i | \mathbf{x}_{t-1} = j) + \ln P(\mathbf{x}_{t-1} = j | \tilde{\mathbf{X}}_1^{t-1})] \right] \\ & - L_{SBS D}^{pri}(x_t(\lambda)) \end{aligned} \quad (3.71)$$

In conclusion, the source decoder performs softbit source decoding algorithm with the de-interleaved channel-related L -value $L_{ch}(\tilde{x}_t(\lambda)|x_t(\lambda))$ and index-level a priori knowledge $\ln P_{SBS D}^{pri}(\mathbf{x}_t|\mathbf{x}_{t-1})$ together with bit-level a priori knowledge $L_{SBS D}^{pri}(x_t(\lambda))$ to generate bit-level a posteriori LLR $L^{post}(x_t(\lambda))$ and bit-level extrinsic information $L_{SBS D}^{ext}(x_t(\lambda))$.

3.5 Proposed A Priori Knowledge Initialization

The conventional ISCD exploits the properties of the source statistics and the channel characteristics. However, the decoding process in first half iteration, known as channel decoding, have no source information. Which means performance gain is significant large between first channel decoding and source decoding. The proposed initialization uses index-level a priori knowledge to calculate different bit-level a priori knowledge in first half iteration. The conventional bit-level a priori knowledge $L^{pri}(x)$ shown in (3.27) is a constant independent of x . In the first half iteration, the channel decoder utilizes the bit-level information of source $L_{CD}^{pri}(x)$,

know nothing about index-level of source. In later iterations, the a priori knowledge is updated with the source extrinsic information. The $L_{CD}^{pri}(x)$ now contains some source index information and is no longer a constant. The proposed a priori knowledge uses this concept by adding some index-level a priori knowledge in the first half iteration.

The proposed initialization is as follows

$$L_P^{pri}(\bar{x}_l) = L_P^{pri}(x_t(\lambda)) = \frac{\sum_{\substack{i=0 \\ b^\lambda(i)=0}}^{2^M-1} \mathbf{u}(i)}{\sum_{\substack{i=0 \\ b^\lambda(i)=1}}^{2^M-1} \mathbf{u}(i)} \quad (3.72)$$

The modified a priori knowledge is related to the different position in index-level. At the first half iteration, we set the a priori knowledge in channel decoding as proposed a priori initial

$$L_{CD}^{pri}(\bar{x}_l) = L_P^{pri}(\bar{x}_l) \quad (3.73)$$

According to (3.28), we have the modified extrinsic information $L_P^{ext}(\bar{x}_l)$ at the first half iteration

$$L_P^{ext}(\bar{x}_l) = L_P^{pri}(\bar{x}_l) - L_{CD}^{pri}(\bar{x}_l) \quad (3.74)$$

Finally, the only difference between proposed and conventional is the initialization of extrinsic information.

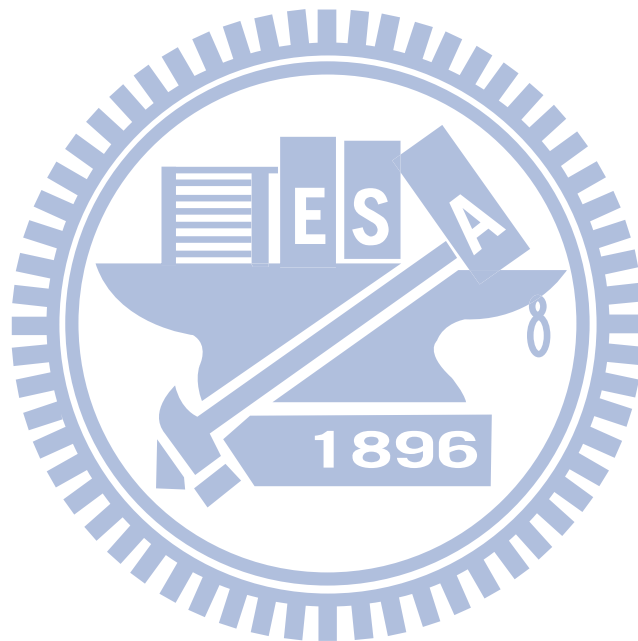
The proposed initialization of $L_{SBS D}^{ext}(\bar{x}_l)$ is

$$L_{SBS D}^{ext}(\bar{x}_l) = L_P^{ext}(\bar{x}_l) \quad (3.75)$$

which is used at the step 1. in the algorithm of the ISCD algorithm.

This initialization can be interpreted as the channel decoder not only uses the source information

in bit-level but also in index level. These information assist the channel decoder in good bias on received data. In addition, the proposed initialization does not require to train extra source statistics.



Chapter 4

Simulation Results

The ECG data in the MIT-BIH arrhythmia database [33] is tested in the proposed algorithm. The original ECG data have 11-bit resolution and the sampling frequency is equal to 360 Hz. The length considered for each ECG data is 650000 samples, corresponding to duration of 30 minutes. In the following result, we will focus on the CR equals to 11. That is, one ECG source sample only needs one bit transmitting through AWGN channel specified a known SNR, denoted as E_s/N_0 .

The organization of this chapter is given as follows:

First of all, the performance of the conventional ISCD will be given, including the decoding ability of channel and source decoder with channel condition. The upper bound of performance will be also considered. After that, we will compare different encoder configurations, such as different vector quantization dimensions, different index assignments, and different channel encoders. Additionally, we will use a better combination of these parameters as our encoder. Then, the conventional ISCD and the proposed ISCD will be compared. Finally, the performance comparison between other lossy ECG compressions will be shown.

The performance of ISCD system if measured by calculating the parameter SNR:

$$\text{Parameter SNR(dB)} = 10 \cdot \log_{10} \frac{E[v_t^2]}{E[(v_t - \hat{v}_t)^2]} \quad (4.1)$$

4.1 Conventional ISCD Performance

In order to explain how ISCD works, we use a particular parameters configuration shown in Table 4.1.

The performance of conventional ISCD over different channel conditions are shown in Fig.4.1.

Table 4.1: Parameters Configuration 1

Parameters	Values
Vector quantization dimension	4
Vector quantization bit	3
Index assignment	Initial
Interleaver size	18×20
Channel code rate	3/4
RSC memory	4
RSC generator	[23 35]

The effect of extrinsic information exchanging makes the parameter SNR increase as the iteration number increase, except for the 0⁺ iteration. The results of 0~3 iterations are shown. At the 0th, the calculation only contain the bit a priori knowledge and the channel-related information, ignore the parity bits and the index a priori knowledge. The upper bound of the conventional ISCD is about the performance of 3rd iteration, and the parameter SNR saturate at 39.16 dB. The largest performance gap between 0th and 0⁺ iterations is 5.74 dB when channel SNR equals to 3 dB. This means the protection of channel coding works well under moderate channel condition. The largest performance gap between 0⁺ and 1st iterations is 5.91 dB when channel SNR equals to -1 dB. The utilization of source statistics assists the source decoder to combat the heavy noisy channel.

Noticed that when the channel SNR is below -1 dB, the parameter SNR of 0⁺ iteration does not increase but decrease. Here we introduce another performance index γ , which is the cross-

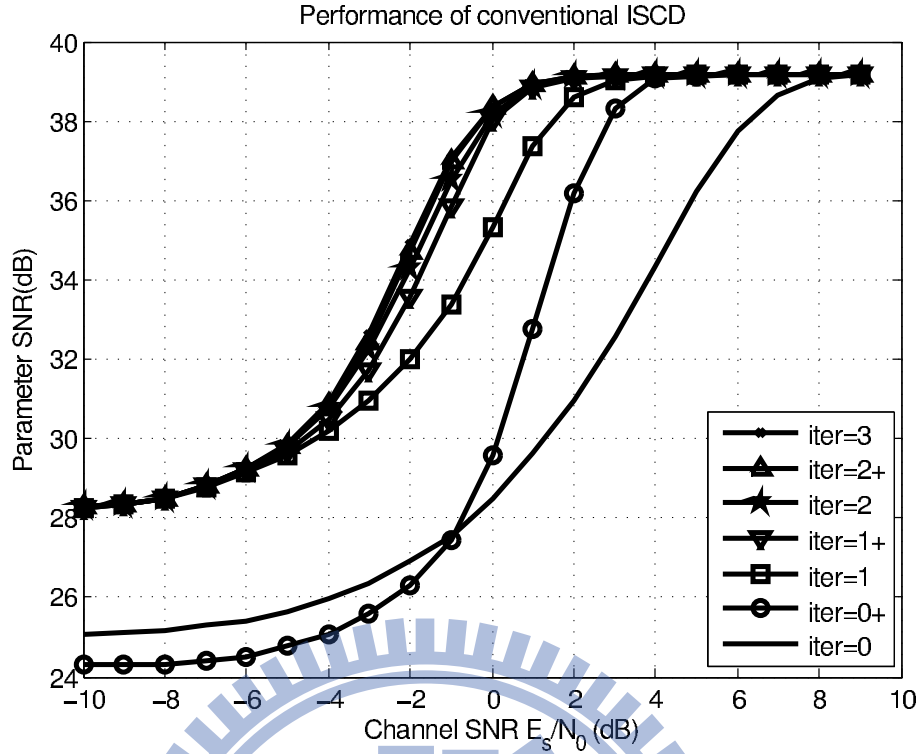


Figure 4.1: Parameter SNR of conventional ISCD

correlation of the transmitting sequence \mathbf{X}_1^T and the corresponding a posteriori LLR sequence $L^{post}(\mathbf{X}_1^T)$ given by

$$\gamma = E[\mathbf{X}_1^T \cdot L^{post}(\mathbf{X}_1^T)] \quad (4.2)$$

from another perspective, γ is the reliability of bit-level. Additionally, the parameter SNR can be seen as the reliability of parameter-level. The performance of γ versus channel SNR below -1dB is given in Fig.4.2. In Fig.4.2, γ increases at 0+th iteration when channel SNR is less than -1 dB. This effect indicates that the channel decoder decodes the received sequence from bit-level. The object of channel decoding is to detect or correct the transmission errors by minimizing the bit errors in $\hat{x}_t(\lambda)$. On the other hand, the object of source decoding is to conceal the transmission errors, to minimize the undesired effects on \hat{v}_t .

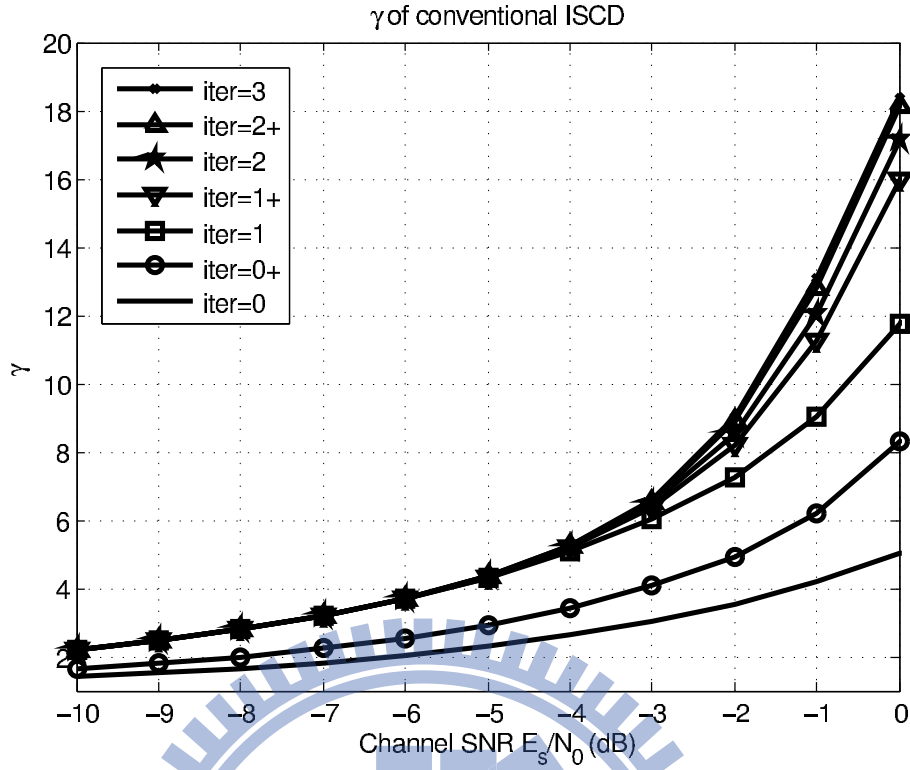


Figure 4.2: γ of conventional ISCD

4.2 Different Encoder Configurations Analysis

To find the best encoder configuration, we compare several parameters of encoder. The dimension of vector quantizer, the index assignments, and the RSC generators are compared in the following section.

4.2.1 Vector Dimension

To compare the performances of different vector dimensions, we use the parameters configuration in Table 4.2.

The channel code rate change in order to fix the same compression ratio. Fig.4.3 compare the performances of different vector dimensions at the 0th, 1st, 2nd iterations. The larger dimension of vector means the higher compression rate. Fig.4.3 shows that the large vector dimension outperforms the small vector dimension when channel SNR is low, and vice versa. At the 2nd iteration, the 6-dimensional encoder has best performance when channel SNR is less than -1.8

Table 4.2: Parameters Configuration 2

Parameters	Values
Vector quantization dimension	4,5,6
Vector quantization bit	3
Index assignment	Initial
Interleaver size	18×20
Channel code rate	$3/4, 3/5, 3/6$
RSC memory	4
RSC generator	[23 35]

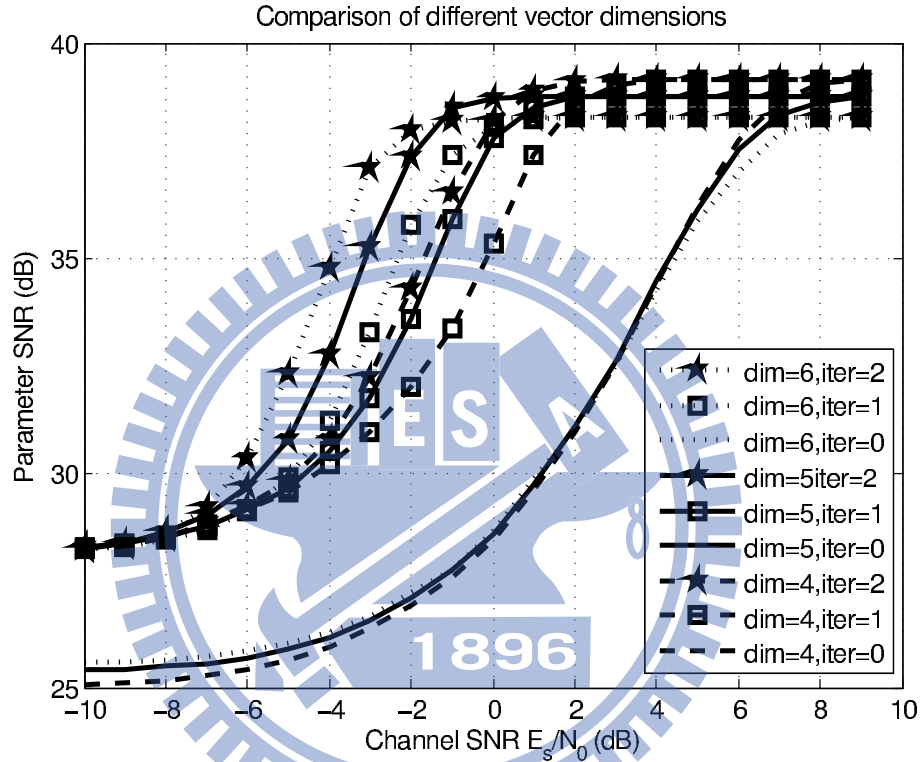


Figure 4.3: Comparison of different vector dimensions

dB, the 5-dimensional encoder works well when channel SNR is between -1.8 dB and 1 dB, the 4-dimensional encoder outperforms when channel SNR is larger than 1 dB. The saturate points of 4~6 vector dimension are 2 dB, 0 dB, and -2 dB respectively. The fact above shows that when we want to transmit the same data rate under heavy noisy channel condition, the best strategy is to compress more and to protect more. Strong data protection ensures the compressed data correctness transmitted through the noisy channel. When the channel condition is good, we do less compression of original data. The quantization is lossy compression which introduces irreversible distortion.

4.2.2 Index Assignment

To compare the performances of different index assignments, we use the parameters configuration in Table 4.3.

Table 4.3: Parameters Configuration 3

Parameters	Values
Vector quantization dimension	4
Vector quantization bit	3
Index assignment	Initial Pseudo-Gray Code Anti-Pseudo-Gray Code
Interleaver size	18×20
Channel code rate	3/4
RSC memory	4
RSC generator	[23 35]

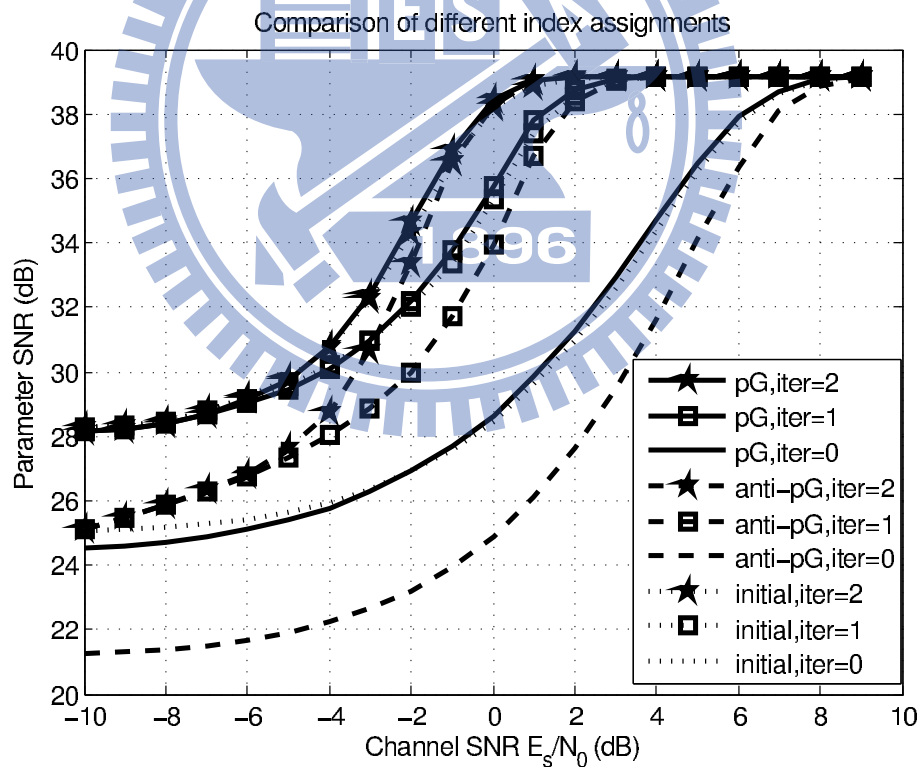


Figure 4.4: Comparison of different index assignments

The initial index assignment is the natural binary assignment of the codebook generated from the splitting method. In anti-pseudo-Gray code, binary indexes are assigned to codewords

that maximizes the average quantization distortion introduced in the reproduced source vectors. In (3.24), the term $C_{\pi}^{(1)}(\mathbf{u}(i))$ becomes larger when the anti-pseudo-Gray code uses. Consequently, the permutation function which maximize distortion in (3.25) is the anti-pseudo-Gray code. Fig.4.4 shows the performances of different index assignments at the 0th, 1st, and 2nd iterations. At the 2nd iteration, the performances of all assignments are saturated at 39.16 dB as $E_s/N_0 \geq 2$ dB. The performance of the anti-pseudo-Gray code drops as the channel SNR decreases. The distortion maximization of the anti-pseudo-Gray code reacts on the increase of $E[(v_t - \hat{v}_t)^2]$ in (4.1). At the 2nd iteration, the performance of the anti-pseudo-Gray code drops to 25.13 dB when channel condition is -10 dB. Interestingly, the performance of both initial and pseudo-Gray code assignments are similar. The largest performance gap occurs at the 1st iteration, the performance gain is 0.38 dB when channel SNR equals to -2 dB. However, the reason of similarity of two performances is the initial codebook generated from the spitting method. This method first clusters the source to two groups. These two groups will respectively assign bit "0" and "1" to represent, which means the source in the same group remain at least 1 bit unchanged. This concept is the same as the pseudo-Gray code. When we do pseudo-Gray code, we want to find a assignment of codeword Hamming distance close to Euclidean distance. From the point of view, we use the spitting method to assign the initial codebook.

4.2.3 Channel Encoder

To compare the performances of different channel encoders, we use the parameters configuration in Table 4.4.

The code rates of RSC are 1/2, 2/3, and 3/4 respectively [34]. We use the regular puncture technique to puncture RSC to code rate 3/4. In Fig.4.5, we can see the performance of these three encoders are slightly different. At the 2nd iteration, the performances of three RSCs are saturated at 39.16 dB as $E_s/N_0 \geq 2$ dB. The RSC with 3/4 code rate has a best performance in high channel SNR region. Compared to the RSC with 2/3 and 1/2 code rate, the performance gap occurs at the 2nd iteration, the performance gain are respectively 0.15 dB and 0.34 dB when $E_s/N_0 = 0$ dB. Because of the low computation complexity and the flexibility of multiple code

Table 4.4: Parameters Configuration 4

Parameters	Values
Vector quantization dimension	4
Vector quantization bit	3
Index assignment	Pseudo-Gray Code
Interleaver size	18×20
Channel code rate	$3/4$
RSC memory	4
RSC generator	$[23 \ 35]$ $[25, 23, 35]$ $[27, 31, 23, 35]$

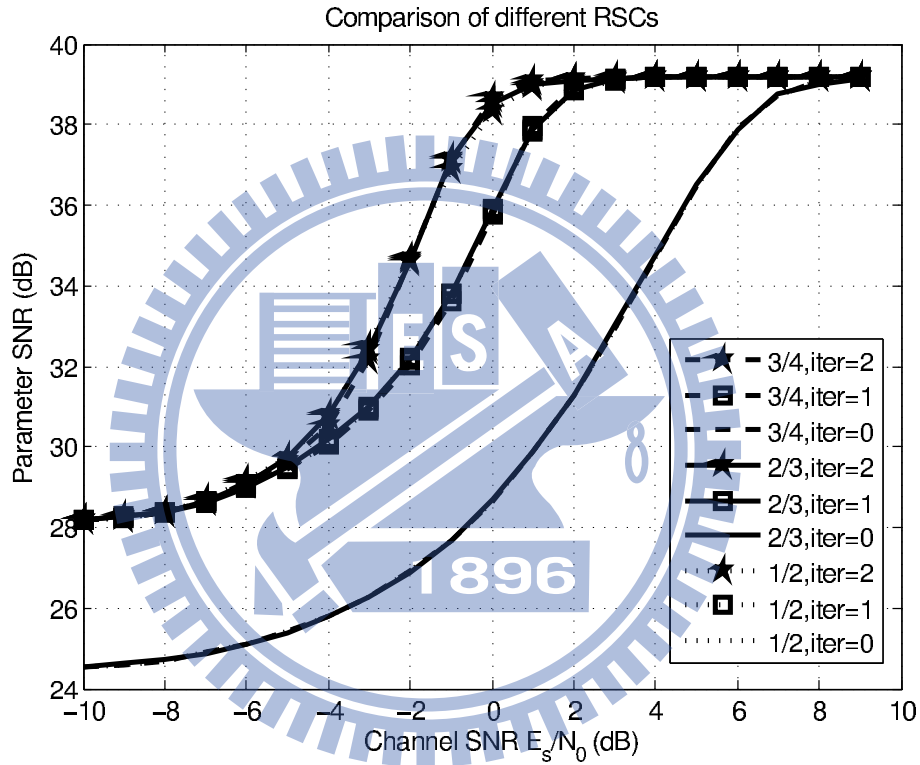


Figure 4.5: Comparison of different RSCs

rates, we still use the RSC with code rate $1/2$ as our mother code.

4.3 Proposed ISCD Performance

To compare the performances of conventional and proposed ISCD, we use the parameters configuration in Table 4.5.

In Fig.4.6, the highest performance gap occur when channel SNR is low. At the 0th iteration,

Table 4.5: Parameters Configuration 5

Parameters	Values
Vector quantization dimension	4
Vector quantization bit	3
Index assignment	Pseudo-Gray Code
Interleaver size	18×20
Channel code rate	$3/4$
RSC memory	4
RSC generator	[23 35]

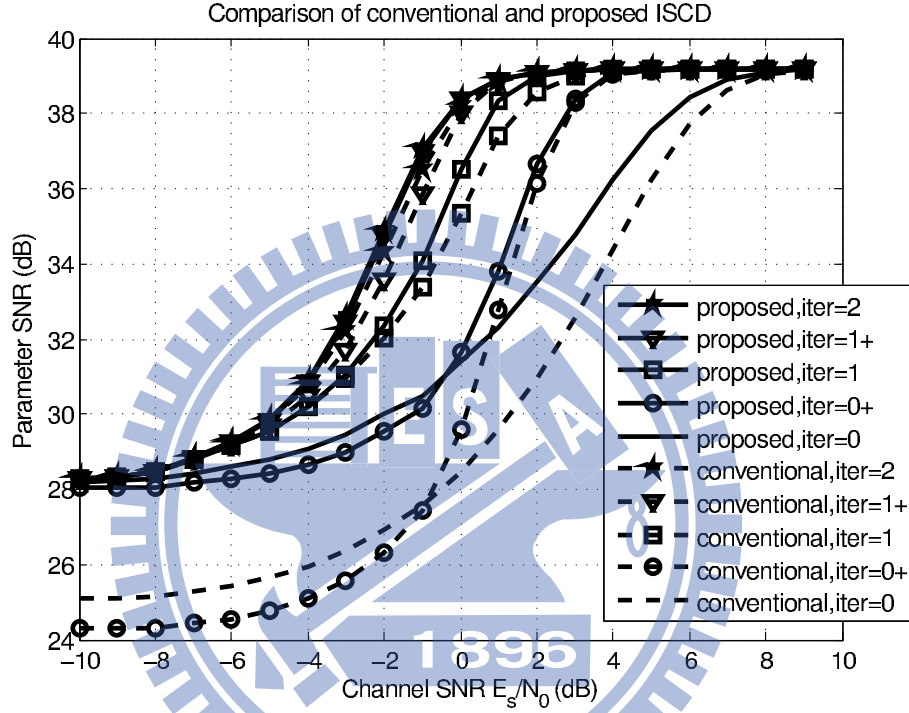


Figure 4.6: Comparison of conventional and proposed ISCD

the largest performance gain can be up to 3.16 dB while channel SNR equals -6 dB. At the 0th iteration, the largest performance gain is 3.76 dB when channel SNR is -8 dB. However, the performance gain drops to 1 dB at the 1st iteration under 0 dB channel condition. The proposed ISCD provides a ability to combat noisy channel environment. Because of the early utilization of source extrinsic information, we get much performance gain at the first two iterations. At the 1st iteration, the conventional ISCD uses the a priori probability in index-level. We still have 1 dB performance gain. Compared to conventional ISCD, the proposed ISCD performance saturates early at the 2^{ed} iteration. Both methods are saturated when $E_s/N_0 \geq 2$ dB.

4.4 Comparison of Different ECG Compression Techniques

In this section, we compare the proposed ISCD with other ECG compression techniques. Before comparison, we introduce the two performance parameters used to ECG compression.

The compression ratio (CR) is defined by:

$$CR = \frac{N_{in}}{N_{out}} \quad (4.3)$$

where N_{in} is the total bits of the original data, and N_{out} is the total bits after compression process.

The percent of root-mean-square difference (PRD) is represented as

$$PRD(\%) = 100 \times \sqrt{\frac{E[(v_t - \hat{v}_t)^2]}{E[v_t^2]}} \quad (4.4)$$

The CR is used to quantify the reduction in ECG data, and the PRD evaluate the distortion about the reconstruction ECG data. The comparison is shown in Table 4.6. [35] compressed ECG based on discrete cosine transform using the windowing techniques. Wavelet-based approaches used a recursive decomposition process in [36]. [37] performed the compressed sensing for ECG compression.

Table 4.6: Comparison with other works

	[35]	[36]		[37]		Proposed
Method	DCT	Wavelet		CS		VQ
CR	27.9	4	20	2	5	11
PRD	2.93	0.98	5.74	4	24	1.10
CR/PRD	9.52	4.08	3.48	0.5	0.21	10
Domain transformation	Y	Y	Y	Y	Y	N
Error correctability	N	N	N	N	N	Y

Compared to these works, we figure out that proposed method have the highest CR/PRD

ratio. Unlike modern lossy compression techniques, our compression process do not require domain transformation. The vector quantization simplifies the source encoding procedure. Furthermore, the proposed method has error correctability.



Chapter 5

Conclusion and Future Work

5.1 Conclusion

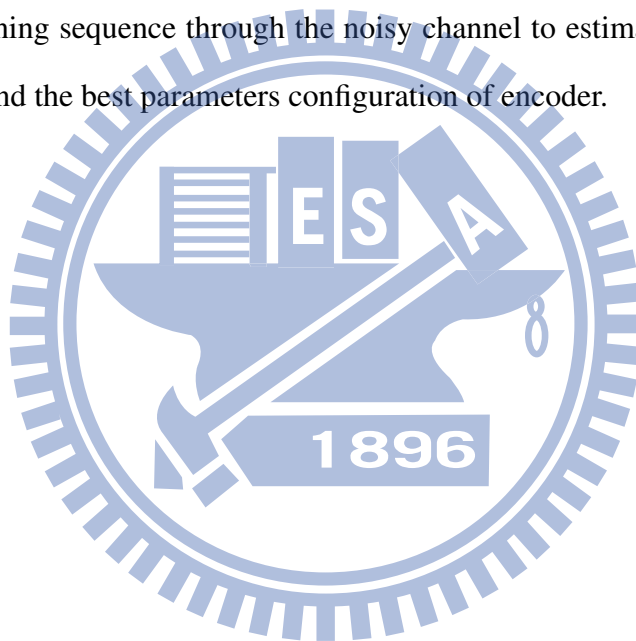
In this thesis, we first introduced the digital communication system. Then, we presented the detail derivation of a novel ISCD system with LBG, pseudo-Gray code, BCJR and SBSM algorithms. Based on the turbo-like decoding procedure, we proposed a initialization of a priori knowledge. The parameter SNR performances of conventional and proposed ISCDs were simulated and compared. The PRD versus CR performance of proposed ISCD and other ECG compression techniques are analyzed.

Simulation results show conventional ISCD, the channel decoding effect under low channel SNR, the splitting method for index assignment, and the best parameters configuration. Compared to conventional ISCD, the proposed ISCD uses the extrinsic information of source index early at 0th iteration. The parameter SNR performance enhances 3.76 dB at 0th iteration with CR equals to 11. Additionally, convergence has been improved while applying the proposed ISCD. Additionally, the proposed ISCD does not require to train extra source statistics. Compared to other ECG compression techniques, the proposed ISCD achieves the highest CR/PRD ratio, low cost of time domain computation, simple source encoding, and error correctability. We reduce complexity from transmitter part, and protect compressed ECG signal on a noisy channel.

5.2 Future Work

Future research directions of ISCD are listed as follows:

- Examine the ISCD performance under an MIMO, RF, or more realistic channel model.
- Examine the ISCD performance for different of interleaver and puncture tables
- Since the splitting method codebook construction introduces the most-significant-bit concept to vector quantizations. The unequal error protection of channel code should be considered.
- Pass the training sequence through the noisy channel to estimate the channel condition, in order to find the best parameters configuration of encoder.



Bibliography

- [1] [Online]. Available: http://www.wica.intec.ugent.be/files/images/WBAN.img_assist_custom.jpg
- [2] W. C. Mueller, "Arrhythmia detection program for an ambulatory ecg monitor," *Biomed. Sci. Instrum.*, vol. 14, pp. 81–85, 1978.
- [3] J. R. Cox, F. M. Nolle, H. A. Fozzard, and G. C. Oliver, "Aztec, a preprocessing program for real-time ecg rhythm analysis," *IEEE Transactions on Biomedical Engineering*, vol. BME-15, no. 2, pp. 128–129, 1968.
- [4] J. P. Abenstein, "Algorithms for real-time ambulatory ecg monitoring," *Biomed. Sci. Instrum.*, vol. 14, pp. 73–79, 1978.
- [5] R. Barr, S. M. Blanchard, and D. A. Dipersio, "Sapa-2 is the fan," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 5, pp. 337–337, 1985.
- [6] K. Rao and N. Ahmed, "Orthogonal transforms for digital signal processing," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '76.*, vol. 1, 1976, pp. 136–140.
- [7] Z. Lu, D. Y. Kim, and W. Pearlman, "Wavelet compression of ecg signals by the set partitioning in hierarchical trees algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 849–856, 2000.
- [8] R. McCaughern, A. Rosie, and F. Monds, "Asynchronous data compression techniques," *Proc. Purdue Centennial Year Symp. Information Process.*, vol. 2, pp. 525–531, 1969.
- [9] A. Al-Shrouf, M. Abo-Zahhad, and M. S. Ahmed, "A novel compression algorithm for electrocardiogram signals based on the linear prediction of the wavelet coefficients," *Digital signal. Process*, vol. 13, pp. 604–622, 2003.
- [10] A. Iwata, Y. Nagasaka, and N. Suzumura, "Data compression of the ecg using neural network for digital holter monitor," *IEEE Eng. Med. Biol. Mag.*, vol. 9, no. 3, pp. 53–57, 1990.
- [11] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [12] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [13] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, 1960.

- [14] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.
- [15] F. Gray, "Pulse code communication," Patent US 2 632 058, 3 17, 1953. [Online]. Available: <http://www.google.com/patents/US2632058>
- [16] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2147–2158, 1990.
- [17] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, vol. pt.4, pp. 37–47, 1955.
- [18] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *IEEE International Conference on Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record*, vol. 2, 1993, pp. 1064–1070 vol.2.
- [19] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [20] J. Forney, G.D., "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, 2001.
- [21] J. L. Fan., *Constrained coding and soft iterative decoding*. Netherlands: Kluwer Academic, 2001.
- [22] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [23] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [24] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [25] J. M. Wozencraft and R. B., *Sequential decoding*. Cambridge: MIT Press and John Wiley, 1961.
- [26] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
- [27] J. Hagenauer and P. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," in *IEEE Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989.*, 1989, pp. 1680–1686 vol.3.
- [28] M. Adrat, P. Vary, and J. Spittka, "Iterative source-channel decoder using extrinsic information from softbit-source decoding," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01).*, vol. 4, 2001, pp. 2653–2656 vol.4.
- [29] J. Melsa and D. Cohn., *Decision and Estimation Theory*. Ltd.: McGraw-Hill Kogakusha, 1978.

- [30] [Online]. Available: <http://www.data-compression.com/vqanim.shtml>.
- [31] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *IEEE International Conference on Communications, 1995. ICC '95 Seattle, 'Gateway to Globalization', 1995*, vol. 2, 1995, pp. 1009–1013 vol.2.
- [32] J. Erfanian, S. Pasupathy, and P. Gulak, "Reduced complexity symbol detectors with parallel structure for isi channels," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1661–1671, 1994.
- [33] A. L. Goldberger, L. Amaral, L. Glass, J. M. Hausdorff, P. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. (2000, Jun.) Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. @ONLINE. [Online]. Available: <http://circ.ahajournals.org/cgi/content/full/101/23/e215>
- [34] A. Graell i Amat, G. Montorsi, and S. Benedetto, "Design and decoding of optimal high-rate convolutional codes," *IEEE Transactions on Information Theory*, vol. 50, no. 5, pp. 867–881, 2004.
- [35] S. Lee, J. Kim, and J.-H. Lee, "A real-time ecg data compression and transmission algorithm for an e-health device," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2448–2455, 2011.
- [36] C.-T. Ku, K.-C. Hung, T.-C. Wu, and H.-S. Wang, "Wavelet-based ecg data compression system with linear quality control scheme," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 6, pp. 1399–1409, 2010.
- [37] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.