

國立交通大學

電信工程研究所

碩士論文

可變長度碼的疊代訊源通道解碼機制

Iterative Source-Channel Decoding for

Variable-Length Codes

研究生：徐子凡

指導教授：張文輝 博士

中華民國一百年六月

# 可變長度碼的疊代訊源通道解碼機制

## Iterative Source-Channel Decoding for Variable-Length Codes

研 究 生：徐子凡

Student：Tzu-Fan Hsu

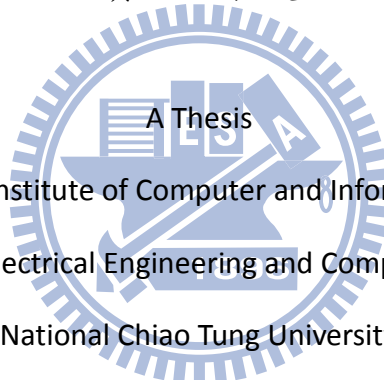
指 導 教 授：張文輝

Advisor：Wen-Whei Chang

國 立 交 通 大 學

電 信 工 程 研 究 所

碩 士 論 文



Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Communication Engineering

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一年六月

# 可變長度碼的疊代訊源通道解碼機制

學生：徐子凡

指導教授：張文輝 博士

國立交通大學電信工程研究所

## 中文摘要

本論文主要探討基於可變長度碼之索引層級疊代訊源通道解碼機制，及其在心電訊號壓縮的應用。傳統的通道解碼演算法受限於其位元層級處理模式，與基於索引層級而推導的訊源解碼演算法存在著相容性問題。有鑑於此，我們首先利用可變長度碼在索引層級上的分段籬柵結構，結合通道編碼器之狀態轉移以建構由二維狀態組成的籬柵圖。接著依據此籬柵圖推導其軟性通道解碼演算法，並將其整合運用於疊代解碼機制。主要是針對具特定機率分佈的訊號源，利用其可變長度編碼所得的殘餘冗息，再基於渦輪碼原則在疊代解碼過程中相互交換訊源及通道解碼器輸出的額外訊息。實驗模擬証實，索引層級的疊代解碼機制能有效提升系統的強健性。

# Iterative Source-Channel Decoding for Variable-Length Codes

Student: Tzu-Fan Hsu

Advisor: Dr. Wen-Whei Chang

Institute of Communications Engineering

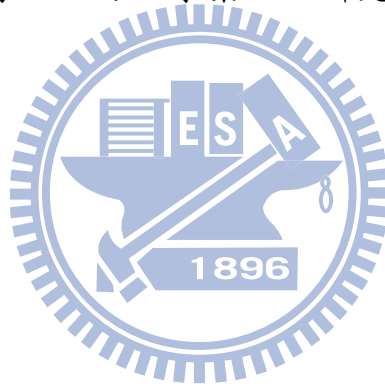
National Chiao Tung University

## Abstract

Transmission of convolutionally encoded source codec parameters over noisy channels can benefit from the use of iterative source-channel decoding (ISCD). This thesis focuses on the design of a symbol-based ISCD for variable-length code (VLC). Our work starts with a sectionalized code trellis whose two-dimensional states are created by combining the symbol-based VLC trellis and the state transition of a channel encoder. We first derive a soft-output channel decoding algorithm that integrates the residual source into the ISCD algorithm. Based the turbo principle, ISCD exchanges the extrinsic information resulting from the source decoder and the channel decoder to improve the decoding performance. Simulation results are presented for Gauss-Markov sources and ECG signals which demonstrate the error-resilience capabilities of symbol-based iterative decoding.

## 誌謝

首先我要感謝我的指導教授張文輝老師對我細心的指導，讓我學習到正確的研究方法及態度，得以在這兩年的研究生涯中順利的完成我的研究及論文。並且除了課業上的指導之外，同時也關心我們的生活，以及對於我的未來及人生給了非常多的建議和經驗的分享。另外也要感謝實驗室的同學及學長，在課業及研究上相互的討論及鼓勵，並幫助我解決了許多問題。最後要感謝我的家人在這兩年來對我的支持與鼓勵，讓我能夠專心致力於學業以及研究。

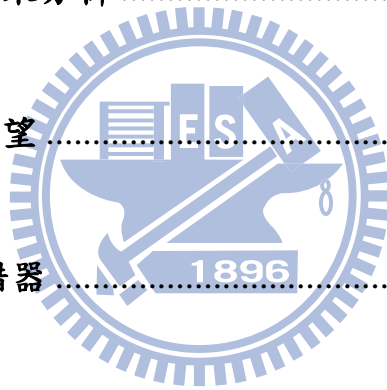


# 目錄

中文摘要 .....	i
英文摘要 .....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	vii
第一章 緒論 .....	1
第二章 傳輸系統架構 .....	6
2.1 系統傳輸模型.....	6
2.2 可變長度訊源編碼.....	8
2.2.1 霍夫曼編碼.....	9
2.2.2 碼字層級籬柵圖.....	10
2.3 迴旋通道編碼.....	11
2.4 疊代解碼效能評估.....	13
第三章 碼字層級之軟性通道解碼機制 .....	15
3.1 迴旋碼之碼字層級籬柵圖 .....	16



3.2 訊息符號後驗機率的推導 .....	18
<b>第四章 疊代訊源通道解碼機制 .....</b>	<b>22</b>
4.1 軟性訊源解碼.....	23
4.2 疊代解碼之流程與額外資訊 .....	27
4.2.1 通道解碼額外資訊.....	27
4.2.2 訊源解碼額外資訊.....	29
<b>第五章 實驗模擬與結果分析 .....</b>	<b>33</b>
<b>第六章 結論與未來展望 .....</b>	<b>52</b>
<b>附錄 A S-random 交錯器 .....</b>	<b>54</b>
<b>參考文獻 .....</b>	<b>56</b>



# 圖目錄

圖 2.1 系統傳輸模型 .....	6
圖 2.2 可變長度碼之籬柵圖( $T=5, N=10$ ) .....	11
圖 2.3 通道迴旋碼之位元層級籬柵圖 .....	12
圖 3.1 迴旋碼之籬柵圖 .....	17
圖 4.1 系統接收端架構圖 .....	32
圖 5.1 軟性訊源解碼效能 .....	35
圖 5.2 ISCD2 的疊代解碼效能( $\rho=0.8$ ) .....	38
圖 5.3 位元與碼字層級解碼效能之比較( $\rho=0.8$ ) .....	39
圖 5.4 不同訊源疊代解碼的重建訊號比( $\rho=0.8, \rho=0.95$ ) .....	40
圖 5.5 MIT-BIH 資料庫編號 100 之相關圖 .....	46
圖 5.6 重建訊號( $k=2$ ) .....	47
圖 5.7 重建訊號( $k=4$ ) .....	48
圖 5.8 不同通道訊雜比下之重建誤差 .....	49



# 表目錄

表 5.1 不同向量維度之可變長度碼書 .....	44
表 5.2 心電圖之平均解碼效能 .....	44



# 第一章 緒論

近年來科技進步，行動多媒體通訊的相關應用已成為關鍵服務，例如無線上網、線上觀看影片…等等。由於多媒體資料包括音樂、影像、語音…等資料量都相當龐大，故需要完善的壓縮機制使資料量得以縮減，進而減少硬體空間以及網路頻寬的資源浪費，降低成本並使傳輸更有效率。針對這樣的議題，現今的多媒體資料壓縮技術，一般會將可變長度碼(variable-length code, VLC)作為編碼機制的標準選項。因此可變長度碼之設計及其在無線網路強健性傳輸，將是一個重要的研究議題。

所謂的可變長度碼，是基於具特定機率分佈的訊號源，針對不同發生機率的資料使用不同長度的碼字加以編碼。為了減少碼字的平均位元長度，機率較大的資料使用長度較短的碼字，相對機率較小的資料則使用長度較長的碼字加以編碼。針對不同的資料型態及傳輸目的而有不同的編碼演算法，其中較著名的編碼方式包括霍夫曼碼(Huffman code)、LZ編碼(Lempel-Ziv coding)和算術碼(Arithmetic code)。這些編碼方式確實可以有效達到資料壓縮目的，卻也因為碼字長度非固定，而衍生出如何具體實現其解碼同步的重要議題。除此之外，受到雜訊干擾造成的位元錯誤，在碼字解碼過程中將造成連續

性的誤判及索引失序問題，使得系統的解碼效益嚴重受損。因此理想的可變長度碼，不僅需要具有維持同步的編碼設計，也需要強健的解碼機制以減少位元錯誤，進一步使系統達到更好的效能。

在傳統的通訊系統設計中，訊源編碼和通道編碼 兩個系統區塊往往是分開考量而獨立設計的。其中訊源編碼專注於資料壓縮，而通道編碼則致力於資料保護。這樣的設計概念源自於沈農(Shannon)的消息理論，其相關理論是在個別的編碼器規劃時，假設另一個編碼器已完成最佳化設計。然而實際的通訊環境，會因為訊號延遲及系統複雜度的各種限制而造成其具體實現上的困難。根據前人實驗觀察，訊源經訊源編碼器處理後，量化索引序列之間存在著某種型的非均勻機率分佈或隱含相關性，而這些殘留的資訊稱為殘餘冗息(residual redundancy)。若能妥善運用，即可在不需提升通道頻寬的情形下，提供通道編碼一些事前資訊(a priori information)以改善解碼運作的正確性。而這些在接收端將訊源解碼器和通道環境之效應一併納入系統設計考量的研究，一般稱為合併訊源通道解碼(joint source-channel decoding, JSCD)。基於殘餘冗息的合併訊源通道解碼又可大致分為三個種類：錯誤隱藏(error concealment)、訊源控制通道解碼機制(source-controlled channel decoding, SCCD)，以及疊代訊源通道解碼機制(iterative source-channel decoding, ISCD)。在錯誤隱藏部分，殘餘冗

息讓訊源解碼器得以隱藏通道解碼器所無法消除的殘餘錯誤。這樣的解碼觀念在[1]被提出，並稱之為軟性訊源解碼機制(soft-bit source decoding, SBSDD)。而在[2]所提出的訊源控制通道解碼機制，則試圖將殘餘冗息整合於通道解碼機制過程中。有別於傳統的通道解碼機制，訊源控制通道解碼機制的輸入端也包括了事前機率，大多來自於事先統計訊源分佈的資訊。而疊代訊源通道解碼機制則合併了以上兩者的優點，比起單獨使用其中任何一項都達到更好的效果。

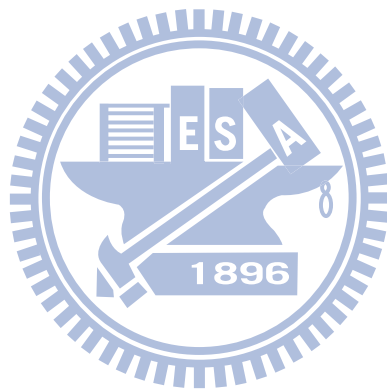
渦輪碼(Turbo Code)[3][4]是通道解碼研究的重大突破。其傳送端是由兩個迴旋編碼器串聯且配合中間串聯的交錯器所組成，而接收端則是將個別解碼器輸出的額外訊息(extrinsic information)，相互交換作為另一個解碼器的事前訊息，以期加強索引判斷的可靠度。這樣的疊代架構，使渦輪解碼機制經證實能接近Shannon理論且具實用價值的編碼技術。而疊代訊源解碼機制參考了渦輪碼解碼的疊代解碼架構，將通道解碼器與訊源解碼器作連結，前者利用迴旋碼的錯誤檢查能力得到額外訊息，而後者則利用訊源的殘餘冗息作為加強解碼能力的依據。相較於訊源編碼和通道編碼分開考量的傳統設計模式，基於渦輪原理的疊代訊源解碼機制可以有效改善訊源的重建品質。

如前所述，可變長度碼是針對具特定機率分佈的訊源資料使用長度不同的碼字加以編碼，而這樣的訊號源及編碼方式使殘餘冗息能

提供更有利的資訊。因此疊代訊源解碼機制不僅廣泛應用於固定長度碼，也被應用在可變長度碼的傳輸系統。前人研究[5]與[6]針對不相關的訊號源，設計出可變長度碼在索引層級的籬柵圖(trellis)，使其得以運用BCJR演算法計算碼字的後驗機率。這種將訊源解碼的軟性輸出運用在疊代解碼的模式，在[7]與[8]中進一步延伸應用在一階高斯馬可夫訊號源，而訊號源之間強烈的相關性使得殘餘冗息更具有解碼效益，疊代解碼器運用這樣的資訊能達到更顯著的效果。然而在[7]與[8]的疊代解碼機制中，只有訊源解碼運作於索引層級上，而通道解碼則是基於位元層級運作。因此在疊代訊源通道解碼的過程中，有助於解碼效能之額外訊息在兩個解碼器之間相互交換時，需要經過額外的索引層級與位元層級間的轉換，而這個轉換破壞了一個索引內位元間的相關性，因此降低了疊代解碼的效能。有鑑於此，我們提出將通道解碼器運作提昇至索引層級之疊代訊源通道解碼機制，並參考[5]與[6]之籬柵圖建構索引層級通道解碼所需的籬柵圖，而此疊代解碼演算法之推導則源自於固定長度碼之碼字層級疊代解碼機制[9]。

本文首先在第二章介紹系統整體之傳輸架構與傳輸通道模型，並說明將如何評估解碼效能。第三章介紹碼字層級之軟性通道解碼機制及其籬柵圖之設計。第四章針對整體的疊代訊源通道解碼機制進行推導，並說明其解碼步驟。第五章針對一階高斯馬可夫訊號及心電圖訊

號，分別提供實驗數據及分析結果。最後第六章則提供結論與未來展望。



## 第二章 傳輸系統架構

基於無線感測網路內頻寬有限與雜訊干擾的考量，我們在傳送端針對訊號源引入可變長度編碼以及通道編碼處理，而在接收端則是運用渦輪原則的循環疊代模式以逐次累積有助於解碼的事前資訊直到系統效能收斂。在傳輸系統的輸入端，我們初步採用一階高斯馬可夫訊號作為訊號源，配合不同的相關係數來模擬各種應用的訊號源(如視訊、音訊、心電圖...等)。本章節將介紹整體系統的架構，並描述傳輸端各個模組的運作方式及功能。第一節說明系統的傳輸模型。第二節介紹可變長度碼及其籬柵圖。第三節介紹通道迴旋碼及其籬柵圖。最後一節則探討可變長度碼之疊代解碼效能評估。

### 2.1 系統傳輸模型

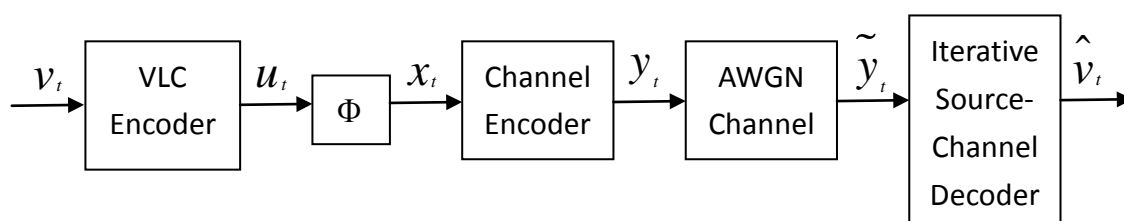


圖 2.1 系統傳輸模型

系統的傳輸模型如圖 2.1 所示，傳輸以封包(packet)為單位，一個封包  $v_1^T = [v_1, v_2, \dots, v_T]$  由  $T$  個具關聯性的符號(symbol)  $v_i$  所組成，其中

$t$  為時間索引。封包大小  $T$  在系統應用上可分為已知或未知兩種情形，通常多媒體傳輸中  $T$  為已知，本論文將僅針對  $T$  為已知的情況進行系統之設計。在傳輸端符號  $v_t$  依序經過  $M$  位元的量化後產生符號  $u_t = \lambda$ ，其中  $\lambda \in \mathcal{I} = \{0, \dots, 2^M - 1\}$ 。接著再經過可變長度編碼處理，利用前置碼  $C$  將符號  $u_t = \lambda$  對映至一可變長度的碼字，並可將其表示為位元向量  $c(\lambda) = [u_t(1), u_t(2), \dots, u_t(\ell(c(\lambda)))]$ ，其中  $\ell(c(\lambda))$  為此碼字長度。將這些位元向量串連起來可得到一總長度為  $N$  個位元的二位元序列  $\mathbf{w} = [w_1, w_2, \dots, w_N] = [c(u_1), c(u_2), \dots, c(u_T)]$ ，其中  $w_n \in \{0, 1\}$  為序列  $\mathbf{w}$  中的第  $n$  個位元。將一個封包中的  $T$  個符號  $u_1^T = [u_1, \dots, u_t, \dots, u_T]$  經過一碼字層級的交錯器 (interleaver)，而取得一重新置換的序列  $x_1^T = [x_1, \dots, x_t, \dots, x_T]$ ，其中  $x_t = \Phi(u_t)$ 。交錯器的功能在於讓訊源與通道兩解碼器輸入端的訊號互不相關，如此才能確保其疊代運算過程不致於有相同資訊重複使用之虞。接下來，將置換過後的序列  $x_1^T = [x_1, \dots, x_t, \dots, x_T]$  經過一編碼率為  $1/2$  的通道編碼器 (channel encoder)，得到輸出序列為  $y_1^T = [y_1, \dots, y_t, \dots, y_T] = [(x_1, z_1), \dots, (x_t, z_t), \dots, (x_T, z_T)]$ ，其中  $x_t$  及  $z_t$  分別為訊息符號與同位檢查符號。分別將符號  $x_t$  及  $z_t$  以位元向量的形式串連為長度為  $N$  的二位元序列  $\mathbf{b}^s = [b_1^s, b_2^s, \dots, b_N^s]$  與  $\mathbf{b}^p = [b_1^p, b_2^p, \dots, b_N^p]$ 。最後進行二位元鍵移調變 (Binary Phase Shift Keying, BPSK modulation)，再經由可加性白色高斯雜訊 (Additive White Gaussian Noise, AWGN) 通道傳送。在



接收端，針對收到的軟性輸出序列  $\tilde{y}_1^T = [\tilde{y}_1, \dots, \tilde{y}_t, \dots, \tilde{y}_T]$  進行訊源通道疊代解碼，得到其估算的符號序列  $\hat{v}_1^T = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T]$ 。

在接收端，收到的訊息及同位檢查位元序列分別為  $\tilde{\mathbf{b}}^s = [\tilde{b}_1^s, \tilde{b}_2^s, \dots, \tilde{b}_N^s]$  及  $\tilde{\mathbf{b}}^p = [\tilde{b}_1^p, \tilde{b}_2^p, \dots, \tilde{b}_N^p]$ ，提供了通道相關的資訊以協助解碼處理。假設通道為可加性白高斯(AWGN)的無記憶性通道，在通道雜訊比為  $\frac{E_b}{N_0}$  下且雜訊之變異數為  $\sigma_e^2 = N_0/2$ ，封包第  $n$  個位置的訊息位元通道資訊  $p(\tilde{b}_n^s | b_n^s)$  及同位檢查位元通道資訊  $p(\tilde{b}_n^p | b_n^p)$  可分別由下式計算：

$$p(\tilde{b}_n^s | b_n^s) = \frac{1}{\sqrt{2\pi\sigma_e}} \cdot \exp\left[-\frac{E_b}{N_0} (\tilde{b}_n^s - b_n^s)^2\right] \quad (2.1)$$

$$p(\tilde{b}_n^p | b_n^p) = \frac{1}{\sqrt{2\pi\sigma_e}} \cdot \exp\left[-\frac{E_b}{N_0} (\tilde{b}_n^p - b_n^p)^2\right] \quad (2.2)$$

## 2.2 可變長度訊源編碼

在訊源編碼的相關研究中，可變長度碼 (variable-length code, VLC) 與固定長度碼 (fixed-length code) 有其不同適用的應用領域。針對具有特定機率分佈的訊號源，使用可變長度碼在資料壓縮層次有較好的效果，然而在解碼過程中因通道雜訊所衍生的同步失序議題則尚待解決。在前人的研究文獻中，指出此議題包含了位元同步及索引同步兩類，前者隨著通訊系統的進步已有效克服，然而位元錯誤所引起

的索引錯誤以及索引同步失序的問題仍是複雜而有待解決的。在解碼的過程中，若能運用訊號特性設計其狀態(state)轉移關係，再配合籬柵(trellis)結構即可發展更為可靠的解碼演算法。以下我們將依據此概念，探討可變長度碼與其籬柵(trellis)結構。

### 2.2.1 霍夫曼編碼

可變長度碼的基本理念為符號的碼字長度反映其所攜帶的訊息量，設計方法為讓較常出現的符號對應到長度較短的碼字，以期平均傳輸位元量可以近似理論熵值(entropy)， $H = \sum_{\lambda \in \mathcal{I}} P(\lambda) \cdot \frac{1}{\log_2(P(\lambda))}$ ，其中  $P(\lambda)$  為索引值  $\lambda$  的發生機率。本論文使用的霍夫曼編碼(Huffman coding)，其設計符合前置碼條件(prefix condition)，才能在解碼時避免索引判定的混淆，得以進行正確的索引同步解碼。首先，經過事前訓練取得不同索引的機率分佈  $P(\lambda)$ ，接著依據以下的演算流程產生一組可變長度的霍夫曼碼：

1. 將索引機率由大至小依序排列，並將機率最小的兩個索引分別標記上 0 及 1。
2. 將這兩個索引視為新的一群，其機率為兩者之和，並以此機率安排至新的順序。
3. 重複執行程序，直到僅餘兩個狀態。最後由右至左取得每個索引

所對應的碼字。

### 2.2.2 碼字層級籬柵圖

在使用可變長度碼的情況之下，當傳輸發生錯誤時，由於索引同步失序，即使位元解碼出現的錯誤很少，亦可能造成嚴重的索引誤判。在這種情況之下，收到序列  $\tilde{y}_1^T$  中對應索引  $u_i = \lambda$  的碼字  $c(\lambda)$  在  $t$  時刻的位元位置可能將無法準確地定位。因此，我們定義在  $t$  時刻所有可能的位元位置  $g_t = n \in \mathcal{N}_t$  為碼字  $c(\lambda)$  在  $t$  時刻的起始狀態。我們可以利用一個適當地對應至可變長度碼的籬柵圖來實現。籬柵圖的狀態定義為  $g_t = n$ ，其中  $g_t$  為時刻  $t$  的狀態，而  $g_t = n \in \mathcal{N}_t$  表示在編碼位元序列  $\mathbf{w}$  中的位元位置。以下舉一個簡單的例子：

一組可變長度碼為  $C = \{c(0) = [1], c(1) = [0, 1], c(2) = [0, 0, 0], c(3) = [0, 0, 1]\}$ ，其對應的碼長分別為  $l(c(0)) = 1$ 、 $l(c(1)) = 2$ 、 $l(c(2)) = 3$  及  $l(c(3)) = 3$ 。假設封包的索引值數目  $T=5$ ，封包位元長度  $N=10$ ，即可將所有可能的索引組合如圖2.2繪出。

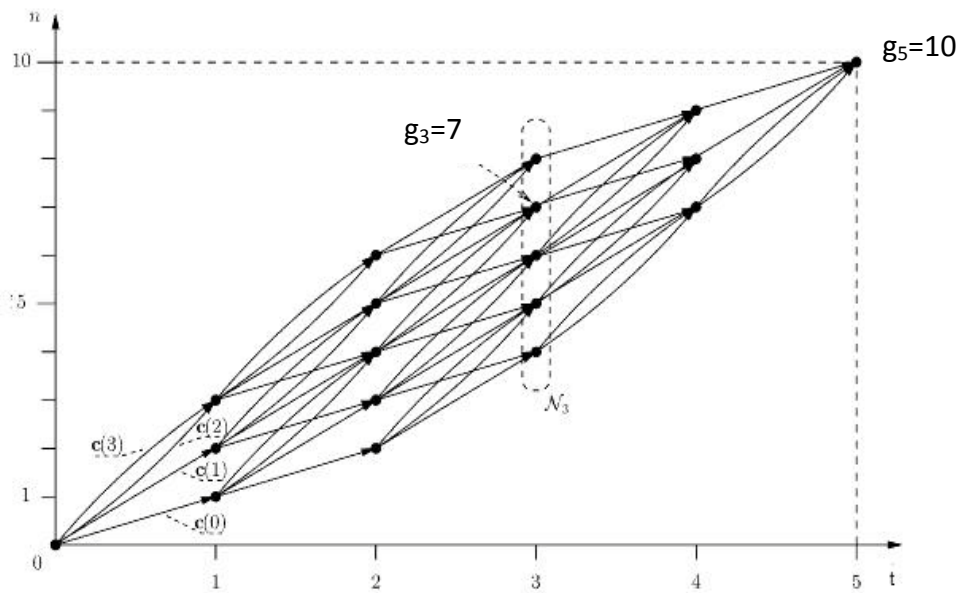


圖2.2 可變長度碼之籬柵圖(T=5, N=10)

從圖2.2可看出，索引  $u_t = \lambda$  會使得狀態  $g_{t-1} = n_1$  轉移至狀態  $g_t = n_2$ ，且對應至碼字  $c(\lambda)$ ，其碼長度為  $l(c(\lambda)) = n_2 - n_1$ 。假使一組碼書中有兩個以上相同碼長的碼字，則在籬柵圖上狀態  $g_{t-1} = n_1$  和狀態  $g_t = n_2$  之間會存在若干平行的狀態轉移。此外，圖2.2還顯示出由於我們的傳輸系統以封包為單位傳輸，在每個封包長度為T個符號且N個位元的條件之下，會造成籬柵圖隨時間依序呈現發散、穩定及收斂三種階段。

### 2.3 迴旋通道編碼

為了保護要傳輸的量化資料，在系統中引入通道編碼器(channel encoder)以降低通道雜訊的干擾。本論文使用的是一編碼率為 1/2 的遞迴系統迴旋(recursive systematic convolutional, RSC)碼。所謂的系

統碼(systematic code)即為在通道編碼的過程中，會產生與其輸入端相同的訊息位元以及依照通道編碼演算法所計算出的同位檢查位元。在段碼(block code)中，碼率為  $k/m$  的編碼器接收了  $k$  個訊息位元然後產生  $m$  位元的碼字，因此編碼過程是以一個區塊為單位進行，很顯然地在編碼器裡必須有足夠空間儲存全部的訊息片段以便產生碼字。但在實際操作上訊息位元是以串列的方式輸入，預設大的空間儲存是既不經濟也不實際的，在這種情形下使用迴旋碼是相當好的選擇，因為迴旋碼編碼器的編碼過程是透過不斷的與輸入訊息運算來產生碼字，且可以利用籬柵圖(trellis diagram)來呈現輸入訊息、輸出訊息以及編碼器中暫存器狀態的轉移關係。以圖 2.3 為例，此圖為一編碼率為  $1/2$  的迴旋編碼器之位元層級籬柵圖。

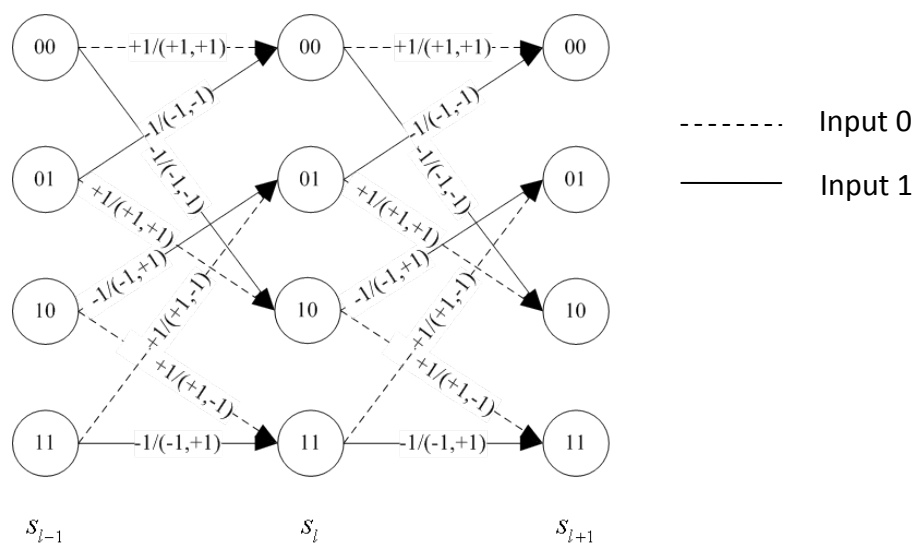


圖 2.3 通道迴旋碼之位元層級籬柵圖

若進一步將  $l(c(x_i))$  段的位元層級籬柵圖合併為一碼字層級的籬柵圖，便可從此碼字層級的籬柵圖看出從某個迴旋編碼器的起始狀態  $s_{i-1}$ ，經由索引  $x_i$  的輸入，而產生的輸出碼字  $y_i = (x_i, z_i)$  以及迴旋編碼器轉移後的狀態  $s_i$ 。利用此迴旋編碼器之碼字層級籬柵圖，結合圖 2.2 的可變長度碼之籬柵圖將可提供通道解碼的依據，其解碼過程將在第三章詳述。

## 2.4 疊代解碼效能評估

基於索引同步的考量，接收端需要知道一個傳送封包的索引數目  $T$  以及總位元長度  $N$ ，將符合此條件的所有可能索引值組合列舉出來，再從這些組合中依據不同的估算機制找出最佳的索引解。執行方式則是利用如圖 2.2 的籬柵圖進行訊源符號的解碼估算，即能在封包之索引值數目及總位元長度的限制條件下，找到一組最佳的索引組合。

訊源疊代通道解碼器的主要任務是計算每個索引值的後驗機率  $P(u_i = \lambda | \tilde{y}_1^T)$ ，並依據將解碼輸出端的重建訊號  $\hat{v}_1^T = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T]$  的均方 (mean square, MS) 誤差或符號錯誤率 (symbol error rate, SER) 最小化的原則，最佳化地估算訊源符號  $v_i$ 。以下我們將分別考慮最小均方誤差 (minimum mean square error, MMSE) 及最大後驗機率 (maximum a posteriori, MAP) 兩種估算機制。

所謂的最小均方誤差估算旨在將原始符號  $v_t$  與重建符號  $\hat{v}_t$  的平  
均誤差最小化，如下式所示：

$$\hat{v}_t^{(MMSE)} = \min_{\hat{v}_t} E \left\{ (v_t - \hat{v}_t)^2 \mid \tilde{y}_1^T \right\} \quad (2.3)$$

如此即相當於將重建訊雜比(reconstruction signal-to-noise ratio, RSNR)  
最大化，而

$$RSNR = \frac{\sum_t v_t^2}{\sum_t (v_t - \hat{v}_t)^2} \quad (2.4)$$

由(2.3)式可推得其估算為

$$\hat{v}_t^{(MMSE)} = \sum_{\lambda=0}^{2^M-1} v_q(\lambda) \cdot P(u_t = \lambda \mid \tilde{y}_1^T) \quad (2.5)$$

其中， $v_q(\lambda)$  為量化表(quantization table)中對應至索引  $\lambda$  的量化值。  
由於此估算法可以將重建訊號的訊雜比最大化，因此特別適用於多媒  
體通訊。

另一種常用估算方法則是最大後驗機率估算法，可在符號層級上  
解碼並且將符號錯誤機率最小化，其估算如下式所示：

$$\begin{aligned} \hat{v}_t^{(MAP)} &= v_q(u_t = \lambda_{map}) \\ \lambda_{map} &= \max_{\lambda} P(u_t = \lambda \mid \tilde{y}_1^T) \end{aligned} \quad (2.6)$$

有別於訊雜比，符號錯誤率只比較量化索引的原始值與重建值的差異，  
並不考慮重建符號的誤差大小，其計算方式為

$$SER = \frac{1}{T} \sum_{t=1}^T (v_t \neq \hat{v}_t^{(MAP)}) \quad (2.7)$$

### 第三章 碼字層級之軟性通道解碼機制

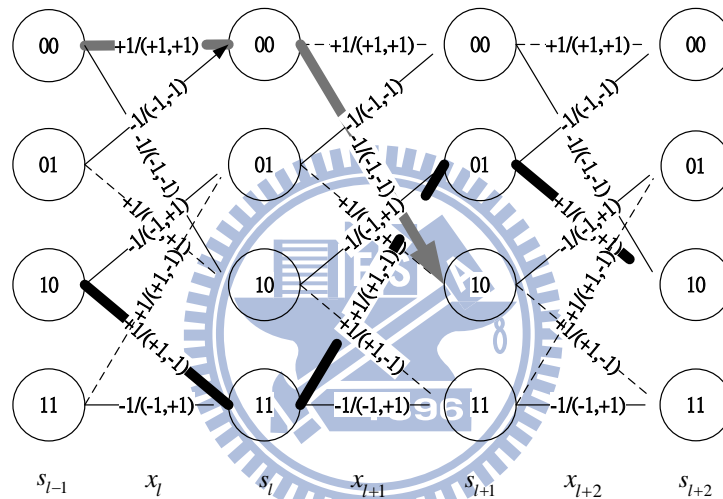
配合傳輸端之使用通道編碼器，一個軟性輸出的通道解碼器不僅輸出硬性判定的重建符號，也能夠提供可靠訊息給後續的訊源解碼程序以增進系統的強健效能。普遍使用的 BCJR 演算法，是針對段碼及迴旋碼，並基於籬柵結構而推導的最大後驗機率解碼演算法。在[10]中，提出一個基於位元層級籬柵圖之前向-後向(forward-backward)的遞迴運算，其籬柵圖中的每個狀態都會有兩條分支由其分出，而每條分支代表單一的符號位元。若進一步將此籬柵圖依碼字的位元長度分段合併，則產生的籬柵結構有助於引入訊源解碼端回傳之參數導向(parameter-oriented)的額外資訊[11],[12]。為了具體實現此通道解碼運算，我們提出一種改良式的 BCJR 演算法，主要構想是將收到的碼字位元序列以碼字為單位依序處理，並基於碼字層級而計算每個符號的後驗機率。有別於傳統的 BCJR 演算法是逐次解碼單一位元，我們所提出的演算法是將碼字索引視為一個非二位元的符號來執行通道解碼處理。本章節將說明如何修改 BCJR 演算法，並配合分段(sectionalized)的碼字層級籬柵圖進行軟性通道解碼流程。第一節介紹通道解碼所需的碼字層級籬柵圖，第二節則詳細說明如何逐步推導不同索引之後驗機率。



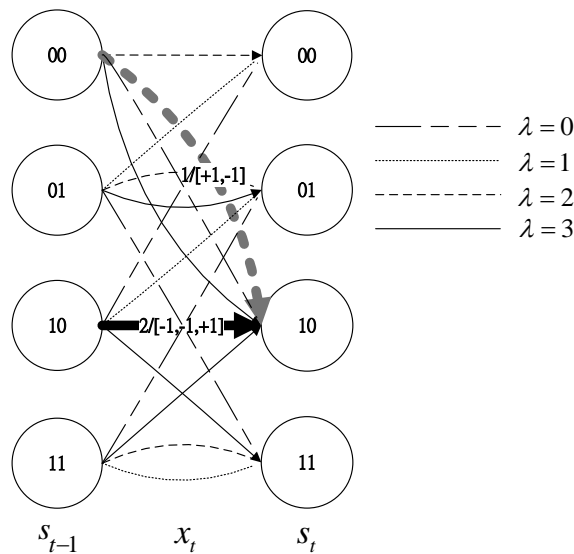
### 3.1 迴旋碼之碼字層級籬柵圖

為了執行碼字層級的軟性通道解碼，經由將碼字位元序列剖析為  $\ell(c(x_i))$  個位元的符號，也就是將  $\ell(c(x_i))$  段原始位元層級的籬柵圖合併為一個碼字層級的分段籬柵圖。在前人的研究中，此分段籬柵圖已經證實可以提升固定長度碼的通道解碼效能，但若要將此概念延伸應用於可變長度碼則面臨諸多挑戰。在這種可變長度分支的情形下，將不同的路徑輸入到一個狀態中必須考慮從收到的序列中截取不同的位元數目，因而與固定長度碼的作法迥然不同[13]。以下舉例說明：考慮一組可變長度碼  $C = \{c(0) = [1], c(1) = [0, 1], c(2) = [0, 0, 0], c(3) = [0, 0, 1]\}$  以及一編碼率為  $1/2$  且編碼生成矩陣為  $G(D) = \begin{bmatrix} 1 & -(1+D^2) \\ & (1+D+D^2) \end{bmatrix}$  的迴旋碼。在圖 3.1(a) 所示的位元層級籬柵圖中，淺色的路徑表示輸入符號  $x_i = 1$  且碼字為  $c(1) = [0, 1]$ ，其位元狀態轉移為  $s_{i-1} = 00 \rightarrow s_i = 00 \rightarrow s_{i+1} = 10$ ，而其輸出的同位檢查位元可由其路徑所經過的分支而得為  $[+1, -1]$ 。而深色的路徑則表示輸入符號  $x_i = 2$  且碼字為  $c(2) = [0, 0, 0]$ ，其位元狀態轉移為  $s_{i-1} = 10 \rightarrow s_i = 11 \rightarrow s_{i+1} = 01 \rightarrow s_{i+2} = 10$ ，而輸出的同位檢查位元依序為  $[-1, -1, +1]$ 。以此類推，我們可以從位元層級籬柵圖中找出從不同起始狀態因輸入不同符號  $x_i$  所產生的路徑，並將該路徑所屬之分支合併產生一條在碼字層級分段籬柵圖的分支。在分段籬柵圖 3.1(b) 中，顯示了所有從位元層級籬柵圖中的路徑經合併產生的分支，分支上所

標記的為通道編碼器的輸入符號及輸出的同位檢查位元向量，即  $x_i / [z_i(1), z_i(2), \dots, z_i(l(c(x_i)))]$ 。在此我們注意到，若使用固定長度的碼字，合併後的分段籬柵圖中的每個狀態  $s_i$  都會有  $2^M$  個分支由其分出及併入。而若為可變長度的碼字，由於每個符號都可能輸入至任何狀態下的通道編碼器，因此每個狀態  $s_i$  還是會有  $2^M$  個分支由其分出，但進入到狀態  $s_i$  的分支數量則會依使用的前置碼  $C$  之碼字組合而定。



(a) 位元層級



(b) 碼字層級

圖 3.1 迴旋碼之籬柵圖 (a)位元層級 (b)碼字層級

由圖 3.1(b)可看出，輸入不同長度的碼字可能會到達相同的狀態  $s_t$ ，但無法判定輸入碼字的長度及其在  $t$  時刻的位元位置  $g_t = n$ 。有鑒於此，我們將一維的狀態  $s_t$  擴展至二維的狀態  $\sigma_t = (s_t, g_t) \in \mathbf{S} \times \mathcal{N}_t$ ，其中  $\mathbf{S} = \{0, 1, \dots, 2^\kappa - 1\}$  為一個帶有  $\kappa$  階暫存器的迴旋編碼器中  $2^\kappa$  個可能的狀態集合，而  $\mathcal{N}_t$  則在 2.2.2 節中定義。此概念之具體實現方式，是將可變長度碼之籬柵圖(圖 2.2)及迴旋碼之碼字層級分段籬柵圖(圖 3.1(b))合併，以作為通道解碼的依據。將籬柵結構如此定義之後，每個籬柵圖中的分支上都有其對應之特定符號  $x_t = \lambda$  的後驗機率。為了方便說明起見，我們稱此籬柵圖產生了一個由狀態轉移函式  $F_\sigma(x_t, \sigma_{t-1})$  及輸出函式  $F_p(x_t, \sigma_{t-1})$  所定義的有限狀態機(finite-state machine)。由此概念而知，由狀態  $\sigma_{t-1}$  轉移至狀態  $\sigma_t = F_\sigma(x_t, \sigma_{t-1})$  的分支所對應的碼字符號組合可記為  $y_t = \{x_t, z_t\}$ ，其中  $x_t$  為訊息符號， $z_t = F_p(x_t, \sigma_{t-1})$  則為一個給定狀態  $\sigma_{t-1}$  下輸出的同位檢查符號。

### 3.2 訊息符號後驗機率的推導

針對迴旋碼使用改良式的 BCJR 演算法，可實現軟性輸入及軟性輸出(soft-input soft-output, SISO)的通道解碼模組。為了使訊息符號更具可靠性，我們將觀察整體接收的封包，並依據上節介紹的碼字層級籬柵結構，推導前向-後向遞迴公式以計算傳送符號之後驗機率。

主要是利用在接收端所收到經過通道干擾的實數序列

$\tilde{y}_1^T = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_T\} = \{(\tilde{x}_1, \tilde{z}_1), (\tilde{x}_2, \tilde{z}_2), \dots, (\tilde{x}_T, \tilde{z}_T)\}$ ，配合籬柵結構以推導每一

個訊息符號值  $x_t = \lambda$  之後驗機率  $P_{CD}(x_t = \lambda | \tilde{y}_1^T)$ 。考慮籬柵圖中的二維狀

態  $\sigma_t$ ，可將其後驗機率寫成：

$$P_{CD}(x_t = \lambda | \tilde{y}_1^T) = \sum_{\sigma_t} P_{CD}(x_t = \lambda, \sigma_t | \tilde{y}_1^T) \quad (3.1)$$

其中  $P_{CD}(x_t = \lambda, \sigma_t | \tilde{y}_1^T)$  可經由貝氏定理分解為：

$$\begin{aligned} P_{CD}(x_t = \lambda, \sigma_t | \tilde{y}_1^T) &= C \cdot P_{CD}(x_t = \lambda, \sigma_t, \tilde{y}_1^T) \\ &= C \cdot \alpha_t^x(\lambda, \sigma_t) \cdot \beta_t^x(\lambda, \sigma_t) \end{aligned} \quad (3.2)$$

其中  $\alpha_t^x(\lambda, \sigma_t) = P(x_t = \lambda, \sigma_t, \tilde{y}_1^t)$  為前向機率， $\beta_t^x(\lambda, \sigma_t) = P(\tilde{y}_{t+1}^T | x_t = \lambda, \sigma_t, \tilde{y}_1^t)$  為

後向機率，而  $C = 1/P(\tilde{y}_1^T)$  為正規化因子。利用訊息符號的一階馬可夫

模型以及通道無記憶性的假設，其前向遞迴及後向遞迴可分別以下列

式子表示為：

$$\begin{aligned} \alpha_t^x(\lambda, \sigma_t) &= \sum_{\sigma_{t-1}} \sum_q P(x_t = \lambda, \sigma_t, x_{t-1} = q, \sigma_{t-1}, \tilde{y}_t, \tilde{y}_1^{t-1}) \\ &= \sum_{\sigma_{t-1}} \sum_q P(x_{t-1} = q, \sigma_{t-1}, \tilde{y}_1^{t-1}) \cdot P(x_t = \lambda, \sigma_t, \tilde{y}_t | x_{t-1} = q, \sigma_{t-1}, \tilde{y}_1^{t-1}) \\ &= \sum_{\sigma_{t-1}} \sum_q \alpha_{t-1}^x(q, \sigma_{t-1}) \cdot \gamma_{\lambda, q}^x(\tilde{y}_t, \sigma_t, \sigma_{t-1}) \end{aligned} \quad (3.3)$$

$$\begin{aligned}
\beta_t^x(\lambda, \sigma_t) &= \sum_{\sigma_{t+1}} \sum_q P(x_{t+1} = q, \sigma_{t+1}, \tilde{y}_{t+1}, \tilde{y}_{t+2}^T | x_t = \lambda, \sigma_t, \tilde{y}_1^t) \\
&= \sum_{\sigma_{t+1}} \sum_q P(\tilde{y}_{t+2}^T | x_{t+1} = q, \sigma_{t+1}, \tilde{y}_{t+1}, x_t = \lambda, \sigma_t, \tilde{y}_1^t) \\
&\quad \cdot P(x_{t+1} = q, \sigma_{t+1}, \tilde{y}_{t+1} | x_t = \lambda, \sigma_t, \tilde{y}_1^t) \\
&= \sum_{\sigma_{t+1}} \sum_q P(\tilde{y}_{t+2}^T | x_{t+1} = q, \sigma_{t+1}, \tilde{y}_1^{t+1}) \\
&\quad \cdot P(x_{t+1} = q, \sigma_{t+1}, \tilde{y}_{t+1} | x_t = \lambda, \sigma_t, \tilde{y}_1^t) \\
&= \sum_{\sigma_{t+1}} \sum_q \beta_{t+1}^x(q, \sigma_{t+1}) \cdot \gamma_{q,\lambda}^x(\tilde{y}_{t+1}, \sigma_{t+1}, \sigma_t)
\end{aligned} \tag{3.4}$$

其中在前向遞迴的式子(3.3)之

$$\begin{aligned}
\gamma_{\lambda,q}^x(\tilde{y}_t, \sigma_t, \sigma_{t-1}) &= P(x_t = \lambda, \sigma_t = (s_t, g_t), \tilde{y}_t | x_{t-1} = q, \sigma_{t-1} = (s_{t-1}, g_{t-1}), \tilde{y}_1^{t-1}) \\
&= P(s_t | x_t = \lambda, g_t, \tilde{y}_t, x_{t-1} = q, \sigma_{t-1}, \tilde{y}_1^{t-1}) \\
&\quad \cdot P(\tilde{y}_t | x_t = \lambda, g_t, x_{t-1} = q, \sigma_{t-1}, \tilde{y}_1^{t-1}) \\
&\quad \cdot P(x_t = \lambda, g_t | x_{t-1} = q, \sigma_{t-1}, \tilde{y}_1^{t-1}) \\
&= P(s_t | x_t = \lambda, s_{t-1}) \cdot P(\tilde{y}_t | x_t = \lambda, g_t, \sigma_{t-1}) \\
&\quad \cdot P(x_t = \lambda, g_t | x_{t-1} = q, g_{t-1})
\end{aligned} \tag{3.5}$$

為了應用符號解碼模式來消滅傳輸錯誤，將籬柵圖上每條路徑的支路量度(branch metric)  $\gamma_{\lambda,q}^x(\tilde{y}_t, \sigma_t, \sigma_{t-1})$  適當地表述其關鍵是系統設計必須考慮到許多額外的因子，例如碼字層級籬柵結構及可加性白色高斯雜訊特性。首先，基於前一節所介紹的通道解碼籬柵圖， $P(s_t | x_t = \lambda, s_{t-1})$  將取決於訊息符號  $x_t = \lambda$  是否合法地將籬柵圖中的狀態  $s_{t-1}$  連結至  $s_t$ 。若能夠連結， $P(s_t | x_t = \lambda, s_{t-1}) = 1$ ，反之則  $P(s_t | x_t = \lambda, s_{t-1}) = 0$ 。

接著，(3.5)式中的第二項為通道資訊(channel information)，對於可加性白色高斯雜訊通道而言，可分解成與訊息符號及與同位檢查符號相關之通道資訊，如下式所示：

$$\begin{aligned} P(\tilde{y}_t | x_t = \lambda, g_t, \sigma_{t-1}) &= P(\tilde{x}_t | x_t = \lambda, g_t, \sigma_{t-1}) \cdot P(\tilde{z}_t | z_t = F_p(x_t = \lambda, \sigma_{t-1}), g_t) \\ &= P(\tilde{x}_t | x_t = \lambda, g_t) \cdot P(\tilde{z}_t | z_t = F_p(x_t = \lambda, \sigma_{t-1}), g_t) \end{aligned} \quad (3.6)$$

至於接收到的訊息符號及同位檢查符號之條件機率，可分別由下列計算式而得：

$$\begin{aligned} P(\tilde{x}_t | x_t = \lambda, g_t) &= \prod_{m=1}^{\ell(c(\lambda))} P(\tilde{b}_{g_{t-1}+m}^s | x_t(m)) \\ &= \left( \frac{1}{\sqrt{2\pi\sigma_e}} \right)^{\ell(c(\lambda))} \cdot \exp\left[ -\frac{E_b}{N_0} \sum_{m=1}^{\ell(c(\lambda))} (\tilde{b}_{g_{t-1}+m}^s - x_t(m))^2 \right] \end{aligned} \quad (3.7)$$

$$\begin{aligned} P(\tilde{z}_t | z_t = \lambda, g_t) &= \prod_{m=1}^{\ell(c(\lambda))} P(\tilde{b}_{g_{t-1}+m}^p | z_t(m)) \\ &= \left( \frac{1}{\sqrt{2\pi\sigma_e}} \right)^{\ell(c(\lambda))} \cdot \exp\left[ -\frac{E_b}{N_0} \sum_{m=1}^{\ell(c(\lambda))} (\tilde{b}_{g_{t-1}+m}^p - z_t(m))^2 \right] \end{aligned} \quad (3.8)$$

其中  $g_{t-1} = g_t - \ell(c(\lambda))$ 。最後，由於  $x_t = \Phi(u_t)$ ，碼字間的相關性經由交錯器處理而消失，(3.5)式中的第三項應被設為零階相關性的機率  $P(x_t = \lambda, g_t)$ 。經由上述之推導，將可被逐項計算得出  $\gamma_{\lambda,q}^x(\tilde{y}_t, \sigma_t, \sigma_{t-1})$ ，再將其分別代入(3.3)及(3.4)式求出前向機率  $\alpha_t^x(\lambda, \sigma_t)$  及後向機率  $\beta_t^x(\lambda, \sigma_t)$ ，再經由(3.1)式進而求出每一個訊息符號值  $x_t = \lambda$  之後驗機率  $P_{CD}(x_t = \lambda | \tilde{y}_1^T)$ ，最後利用 2.4 節所介紹的效能評估機制，即可求得通道解碼輸出之符號錯誤率與重建訊雜比。

## 第四章 疊代訊源通道解碼機制

渦輪碼(turbo code)是 1993 年發明的一種高效能的前向錯誤更正(forward error correction, FEC)碼，同時也是首次經證實能接近 Shannon 理論且具實用價值的編碼技術。渦輪解碼架構是由兩個相同的通道解碼器配合中間串聯的交錯器所組成，藉由各別解碼器輸出的額外訊息(extrinsic information)交互疊代來加強估算符號的可靠度。而疊代訊源通道解碼機制(Iterative source-channel decoding, ISCD)則是一種類似渦輪解碼的系統，其核心模組是由通道與訊源兩種不同的解碼器所組成。在疊代解碼過程中，通道解碼器與訊源解碼器互相交換額外訊息，其訊息分別源自於迴旋碼的錯誤檢查能力與訊源殘餘冗息(residual redundancy)。

在本章我們將探討可變長度碼的疊代訊源通道解碼設計，主要是因為可變長度碼是針對具有特定機率分佈的訊號源，故碼字間所存在的殘餘冗息將會是解碼過程中值得運用的重要資訊。在前人的研究中，主要的解碼過程運作於位元層級上，但訊源解碼本身卻是實現在碼字層級上。這使得在解碼過程中位元層級與碼字層級的資訊交換時，必須經過額外的位元與碼字機率之間的轉換，因此破壞了一個碼字內位元間的相關性，降低了疊代解碼的效能。有鑑於此，如第三章所述，

我們將二進位迴旋碼的通道解碼運作建立在碼字層級之上，進而推行出碼字層級的疊代解碼機制。為了達到這個目標，我們必須發展出改良式的 BCJR 演算法及其所對應的籬柵結構，使得碼字層級的資訊能夠被運用。而在推展新的演算法時所面臨到的問題也將在此章節逐一探討。

#### 4.1 軟性訊源解碼

對於可變長度碼及通道碼的結合，我們將殘餘的訊源冗息及添加的通道碼冗息經過類似渦輪碼的計算，使得在疊代解碼的過程中能得到效能上的提升。在疊代解碼系統中，含有兩個軟性輸入軟性輸出之解碼器。其中通道解碼器之目標在於處理收到的碼字序列  $\tilde{y}_1^T = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_T]$ ，並結合訊息符號之事前機率而計算出通道解碼所得的額外訊息。而訊源解碼器則計算訊源額外訊息，並將其經過交錯器後，在下一次的疊代中當作通道解碼端額外的事前資訊。將兩個解碼器的額外訊息不斷反覆疊代交換，直到可靠性之增益已無明顯效果。在最後一次的疊代之後，由訊源解碼端所產生的後驗機率將視為參數估算機制之輸入，最後經由一個反量化處理產生傳送訊號  $v_i$  的估算符號  $\hat{v}_i$ 。



訊源解碼的額外訊息之決定準則在[7]中被推導出，但我們提出進一步的調整，主要是在訊源解碼的過程中利用交錯區塊而允許  $T$  個符號的延遲。我們將依照定義的交錯區塊 (interleaving block) 大小決定長度  $T$ ，而交錯區塊大小  $T$  之意義為將訊號源符號序列中的每  $T$  個符號訂為一個區塊，而交錯處理只針對區塊內的  $T$  個符號。若解碼程序是基於一階馬可夫模型而運作，則符號間的相關性將會被納入解碼流程中所考慮的因子。接著，以時刻  $t$  為參考點，整個過去所收到的訊息符號序列  $\tilde{u}_1^t$  甚或是之後所收到的符號序列  $\tilde{u}_{t+1}^T$  都會被考慮為訊源解碼的依據。為了具體實現此概念，我們將提出一個前向-後向的演算法，來說明如何將過去及未來所收到的符號序列轉換為在疊代解碼過程中有用的額外資訊。

我們考慮一組長度為  $T$  的符號序列  $v_1^T$ ，每個符號  $v_i$  都經過量化而得  $M$  個位元的索引  $u_i = \lambda$ ，再執行可變長度編碼而得其碼字為  $c(u_i = \lambda)$ 。接著，將這  $T$  個碼字組合出位元長度為  $N$  的位元序列，所有滿足碼字長度為  $T$  且位元長度為  $N$  的序列組合可利用如圖 2.2 的可變長度碼之籬柵圖來表示。考慮籬柵圖中的狀態  $g_{t-1}$  及  $g_t$ ，對應於收到序列

$\tilde{y}_1^T = \{\tilde{u}_1^T, \tilde{z}_1^T\}$  之傳送符號  $u_t = \lambda$  的後驗機率為

$$P_{SD}(u_t = \lambda | \tilde{y}_1^T) = \sum_{g_t} \sum_{g_{t-1}} P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) \quad (4.1)$$

其中  $P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)$  經由貝氏定理可進一步分解為

$$P_S(g_{D1}, u_t = \lambda, g_t | \tilde{y}_1^T) = C \cdot \alpha_t^u(\lambda, g_t) \beta_t^u(\lambda, g_t) \cdot P(\tilde{z}_1^T | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \quad (4.2)$$

其中  $C = 1/P(\tilde{y}_1^T)$  為正規化因子，而前向機率  $\alpha_t^u(\lambda, g_t) = P(g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T)$ ，

後向機率  $\beta_t^u(\lambda, g_t) = P(\tilde{u}_{t+1}^T | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T)$ 。前向機率與後向機率又可分

別簡化為下列的遞迴公式：

$$\begin{aligned} \alpha_t^u(\lambda, g_t) &= \sum_{g_{t-2}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t-1} - g_{t-2}} P(g_{t-2}, u_{t-1} = q, g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \\ &= \sum_{g_{t-2}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t-1} - g_{t-2}} P(u_t = \lambda, g_t, \tilde{u}_t | g_{t-2}, u_{t-1} = q, g_{t-1}, \tilde{u}_1^{t-1}) \\ &\quad \cdot P(g_{t-2}, u_{t-1} = q, g_{t-1}, \tilde{u}_1^{t-1}) \end{aligned} \quad (4.3)$$

$$\begin{aligned} &\cdot P(g_{t-2}, u_{t-1} = q, g_{t-1}, \tilde{u}_1^{t-1}) \\ &= \sum_{g_{t-2}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t-1} - g_{t-2}} \gamma_{\lambda, q}^u(u_t, g_t, g_{t-2}) \cdot \alpha_{t-1}^u(q, g_{t-1}) \\ \beta_t^u(\lambda, g_t) &= \sum_{g_{t+1}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t+1} - g_t} P(\tilde{u}_{t+2}^T, \tilde{u}_{t+1}, g_{t+1}, u_{t+1} = q | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \\ &= \sum_{g_{t+1}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t+1} - g_t} P(\tilde{u}_{t+2}^T | \tilde{u}_1^{t+1}, g_{t+1}, u_{t+1} = q, g_{t-1}, u_t = \lambda, g_t) \\ &\quad \cdot P(\tilde{u}_{t+1}, g_{t+1}, u_{t+1} = q | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \\ &= \sum_{g_{t+1}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t+1} - g_t} P(\tilde{u}_{t+2}^T | \tilde{u}_1^{t+1}, g_{t+1}, u_{t+1} = q, g_t) \\ &\quad \cdot P(\tilde{u}_{t+1}, g_{t+1}, u_{t+1} = q | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \\ &= \sum_{g_{t+1}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t+1} - g_t} \gamma_{q, \lambda}^u(u_{t+1}, g_{t+1}, g_{t-1}) \cdot \beta_{t+1}^u(q, g_{t+1}) \end{aligned} \quad (4.4)$$

其中(4.3)式之

$$\begin{aligned} \gamma_{\lambda, q}^u(u_t, g_t, g_{t-2}) &= P(u_t = \lambda, g_t, \tilde{u}_t | g_{t-2}, u_{t-1} = q, g_{t-1}, \tilde{u}_1^{t-1}) \\ &= P(\tilde{u}_t | u_t = \lambda, g_t, u_{t-1} = q, g_{t-1}, g_{t-2}, \tilde{u}_1^{t-1}) \\ &\quad \cdot P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2}, \tilde{u}_1^{t-1}) \\ &= P(\tilde{u}_t | u_t = \lambda, g_t) \cdot P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2}) \end{aligned} \quad (4.5)$$

(4.5)式中的第一項 $P(\tilde{u}_t | u_t = \lambda, g_t)$ 為與訊息符號 $u_t$ 相關之通道訊息，可由如(3.7)式之計算方式而得。而第二項為一階馬可夫訊號之轉移機率，將其對應至可變長度碼之籬柵圖可得

$$P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2}) = \frac{1}{C(q, g_{t-1})} \begin{cases} P(u_t = \lambda | u_{t-1} = q), & \text{for } \ell(c(\lambda)) = g_t - g_{t-1} \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

其中正規化因子

$$C(q, g_{t-1}) = \sum_{g_t} \sum_{\lambda \in \mathcal{I}: \ell(c(\lambda)) = g_t - g_{t-1}} P(u_t = \lambda | u_{t-1} = q) \quad (4.7)$$

在疊代解碼的過程中，(4.5)式中的第二項可更新為 $P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2}) \cdot P_{CD}^{(extr)}(u_t = \lambda, g_t)$ 。值得注意的是，由(4.7)式的加總條件可看出，此正規化因子考慮到封包總位元長度 $N$ 及符號個數 $T$ 之限制，因此可變長度碼之籬柵圖在發散及收斂階段不是所有碼字 $\lambda = 0, \dots, 2^M - 1$ 所對應的狀態轉移 $g_{t-1}$ 至 $g_t = g_{t-1} + \ell(c(\lambda))$ 都會存在。經由上述公式之推導，最終可求得每一個訊息符號可能值 $u_t = \lambda$ 之後驗機率 $P(u_t = \lambda | \tilde{y}_1^T)$ 。下一節將說明如何利用後驗機率擷取訊源解碼之額外資訊，以及疊代解碼過程中，兩個解碼器所輸出的額外訊息如何交換運用。

## 4.2 疊代解碼之流程與額外資訊

為了取得疊代解碼所需的軟性輸出，以下將推導出其額外訊息部分，用以回傳給另一個解碼器使用。內容分為通道解碼器及訊源解碼器個別輸出的額外訊息。最後再總結整個疊代訊源通道解碼的流程。

### 4.2.1 通道解碼額外資訊

由 (3.1) 式通道解碼計算所得的訊息符號之後驗機率  $P_{CD}(x_t = \lambda | \tilde{y}_1^T)$ ，進一步正規化而計算考慮通道解碼籬柵圖狀態  $g_t$  及之後驗機率：

$$P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T) = \frac{1}{C'} \cdot P_{CD}(x_t = \lambda | \tilde{y}_1^T) \quad (4.8)$$

其中正規化因子

$$C' = \sum_q \sum_{\{g_t, g_{t-1} : g_t - g_{t-1} = \ell(c(q))\}} P_{CD}(x_t = q | \tilde{y}_1^T) \quad (4.9)$$

為了疊代解碼程序能使用通道解碼器的軟性輸出，此機率

$P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)$  必須經由貝氏定理分解為以下三項：事前機率

$P_a(x_t = \lambda, g_t)$ ，通道相關資訊  $P_c(x_t = \lambda, g_t)$  及通道解碼額外訊息

$P_{CD}^{[ext]}(x_t = \lambda, g_t)$ 。(4.8)式將被展開為

$$\begin{aligned}
P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T) &= P(g_{t-1}, x_t = \lambda, g_t, \tilde{x}_1^T, \tilde{z}_1^T) / P(\tilde{y}_1^T) \\
&= P(x_t = \lambda, g_{t-1}, g_t, \tilde{x}_1^T) \cdot \frac{P(\tilde{z}_1^T | x_t = \lambda, g_{t-1}, g_t, \tilde{x}_1^T)}{P(\tilde{y}_1^T)} \\
&= P(\tilde{x}_t | x_t = \lambda, g_{t-1}, g_t, \tilde{x}_1^{t-1}, \tilde{x}_{t+1}^T) \cdot P(x_t = \lambda, g_{t-1}, g_t) \\
&\quad \cdot P(\tilde{x}_1^{t-1}, \tilde{x}_{t+1}^T) \cdot \frac{P(\tilde{z}_1^T | x_t = \lambda, g_{t-1}, g_t, \tilde{x}_1^T)}{P(\tilde{y}_1^T)} \\
&= C \cdot P(\tilde{x}_t | x_t = \lambda, g_t) \cdot P(x_t = \lambda, g_t) \cdot P_{CD}^{[ext]}(x_t = \lambda, g_t)
\end{aligned} \tag{4.10}$$

其中  $C = 1/P(\tilde{y}_1^T)$  為正規化因子， $P(x_t = \lambda, g_t)$  定義為事前機率  $P_a(x_t = \lambda, g_t)$ ， $P(\tilde{x}_t | x_t = \lambda, g_t)$  定義為通道相關訊息  $P_c(x_t = \lambda, g_t)$ ，而通道解碼額外資訊

$$P_{CD}^{[ext]}(x_t = \lambda, g_t) = P(\tilde{z}_1^T | x_t = \lambda, g_{t-1}, g_t, \tilde{x}_1^T) \tag{4.11}$$

值得注意的是，(4.10) 式中的事前機率之初始值為  $P_a(x_t = \lambda, g_t) = P(x_t = \lambda, g_t)$ 。然而在疊代解碼的過程中，一般的作法是將其與來自訊源解碼器的額外資訊  $P_{SD}^{[ext]}(x_t = \lambda, g_t)$  整合再作為通道解碼端的事前機率。因此可將(4.10)式改寫為

$$\begin{aligned}
P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T) &= C \cdot P(\tilde{x}_t | x_t = \lambda, g_t) \cdot P(x_t = \lambda, g_t) \\
&\quad \cdot P_S^{[ext]}(x_t = \lambda, g_t) \cdot P_{CD}^{[ext]}(x_t = \lambda, g_t)
\end{aligned} \tag{4.12}$$

由(4.12)式可得，通道額外資訊將如下式所示：

$$P_{CD}^{[ext]}(x_t = \lambda, g_t) = \frac{P(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)}{C \cdot P_c(x_t = \lambda, g_t) \cdot P_a(x_t = \lambda, g_t) \cdot P_S^{[ext]}(x_t = \lambda, g_t)} \tag{4.13}$$

有別於針對固定長度碼所推導的疊代解碼演算法，此機率

$P_{CD}^{[ext]}(x_t = \lambda, g_t)$  無法直接進行反交錯處理而提供訊源解碼器的事前資訊。

為了計算反交錯處理的通道額外訊息  $P_{CD}^{[ext]}(u_t = \lambda, g_t)$ ，我們先將(3.1)式求得的後驗機率  $P_{CD}(x_t = \lambda | \tilde{y}_1^T)$  做反交錯處理產生  $P_{CD}(u_t = \lambda | \tilde{y}_1^T)$ ，再將其正規化而得

$$P_{CD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) = \frac{1}{C'} \cdot P_{CD}(u_t = \lambda | \tilde{y}_1^T) \quad (4.14)$$

其中正規化因子

$$C' = \sum_q \sum_{\{g_t, g_{t-1} : g_t - g_{t-1} = \ell(c(q))\}} P_{CD}(u_t = q | \tilde{y}_1^T) \quad (4.15)$$

參考(4.10)-(4.12)的說明，同理可推得

$$P_{CD}^{[ext]}(u_t = \lambda, g_t) = \frac{P_{CD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)}{C \cdot P_c(u_t = \lambda, g_t) \cdot P_a(u_t = \lambda, g_t) \cdot P_S^{[ext]}(u_t = \bar{D}\lambda, g_t)} \quad (4.16)$$

#### 4.2.2 訊源解碼額外資訊

訊源解碼額外資訊的計算步驟詳述如下：首先針對(4.2)式計算所得後驗機率  $P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)$ ，利用貝氏定理分解為三項資訊：通道相關資訊  $P_c(u_t = \lambda, g_t)$ ，通道額外資訊  $P_{CD}^{[ext]}(u_t = \lambda, g_t)$  以及訊源額外資訊  $P_S^{[ext]}(u_t = \lambda, g_t)$ 。為了求出這三項資訊，我們將(4.5)式代入(4.3)式，則(4.2)式可表示為

$$\begin{aligned} P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) \\ = C \cdot P_c(u_t = \lambda, g_t) \cdot P_S^{[ext]}(u_t = \bar{D}\lambda, g_t) \cdot P_{CD}^{[ext]}(u_t = \lambda, g_t) \end{aligned} \quad (4.17)$$

其中

$$P_c(u_t = \lambda, g_t) = P(\tilde{u}_t | u_t = \lambda, g_t) \quad (4.18)$$

$$P_S^{[extr]}(u_t = \lambda, g_t) = \beta_t^u(\lambda, g_t) \sum_{g_{t-2}} \sum_{q \in \mathcal{I}: \ell(c(q)) = g_{t-1} - g_{t-2}} P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2})$$

$$P_{SD}^{[extr]}(u_t = \lambda, g_t) = \frac{P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T) \cdot \alpha_{t-1}^u(q, g_{t-1})}{C \cdot P_c(u_t = \lambda, g_t) \cdot P_{CD}^{[extr]}(u_t = \lambda, g_t)} \quad (4.19)$$

$$P_{CD}^{[extr]}(u_t = \lambda, g_t) = P(\tilde{z}_1^T | g_{t-1}, u_t = \lambda, g_t, \tilde{u}_1^T) \quad (4.20)$$

由(4.17)式可得，在下次疊代中要回傳給通道解碼器的訊源額外資訊為

$$P_{SD}^{[extr]}(u_t = \lambda, g_t) = \frac{P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)}{C \cdot P_c(u_t = \lambda, g_t) \cdot P_{CD}^{[extr]}(u_t = \lambda, g_t)} \quad (4.21)$$

其中  $P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)$  為可變長度籬柵圖中每個訊源符號的後驗機率，其計算方式為將(4.5)式中  $\gamma_{\lambda, q}^u(u_t, g_t, g_{t-2})$  的事前資訊更改為  $P(u_t = \lambda, g_t | u_{t-1} = q, g_{t-1}, g_{t-2}) \cdot P_{CD}^{[extr]}(u_t = \lambda, g_t)$ ，依序帶回公式(4.3)、(4.4)及(4.2)而得。至於交錯後的訊源額外訊息  $P_{SD}^{[extr]}(u_t = \lambda, g_t)$  之計算方式，則是將  $P_{SD}(g_{t-1}, u_t = \lambda, g_t | \tilde{y}_1^T)$  代入(4.1)式而得  $P_{SD}(u_t = \lambda | \tilde{y}_1^T)$ ，經由交錯處理產生  $P_{SD}(x_t = \lambda | \tilde{y}_1^T)$  再計算

$$P_{SD}^{[extr]}(x_t = \lambda, g_t) = \frac{P_{SD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)}{C \cdot P_c(x_t = \lambda, g_t) \cdot P_{CD}^{[extr]}(x_t = \lambda, g_t)} \quad (4.22)$$

$$\text{其中 } P_{SD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T) = C' \cdot P(x_t = \lambda | \tilde{y}_1^T) \quad (4.23)$$

$$C' = 1 / \sum_{q \in \{g_t, g_{t-1}: g_t - g_{t-1} = \ell(c(q))\}} P(x_t = q | \tilde{y}_1^T) \quad (4.24)$$

最後，總結我們所提出的基於可變長度碼之籬柵圖的碼字層級疊代訊源通道解碼演算法，其接收端架構及解碼流程如圖 4.1 所示，而疊代解碼步驟如下所述：

1. 初始化:

設定疊代次數為  $\text{iteration}=0$ ，而所有符號之訊源解碼額外訊息初始機率值設為  $P_{SD}^{[ext]}(x_t = \lambda, g_t) = 1$ 。

2. 在通道解碼模組部分: 受通道雜訊干擾的接收符號序列  $\tilde{y}_1^T = \{\tilde{x}_1^T, \tilde{z}_1^T\}$

進入接收端，它提供了通道訊息  $P(\tilde{x}_t | x_t = \lambda, g_t)$  與  $P(\tilde{z}_t | x_t = \lambda, g_t)$ 。針對每一個符號，將事先訓練所得的符號機率  $P(x_t = \lambda, g_t)$  及訊源解碼輸出的額外資訊  $P_{SD}^{[ext]}(x_t = \lambda, g_t)$  整合視為通道解碼器之事前訊息，再經由(3.1)式計算通道解碼之後驗機率  $P_{CD}(x_t = \lambda | \tilde{y}_1^T)$ 。接者，考慮籬柵圖狀態的符號後驗機率  $P_{CD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)$  可由(4.8)式計算而得。至於經過反交錯器後的通道解碼額外資訊  $P_{CD}^{[ext]}(u_t = \lambda, g_t)$ ，可由(4.16)式計算而得，並傳送至訊源解碼器作為事前訊息。

3. 在訊源解碼模組部分: 解碼器的輸入端為接收到的符號序列

$\tilde{y}_1^T = \{\tilde{u}_1^T, \tilde{z}_1^T\}$ ，將通道解碼端的額外資訊  $P_{CD}^{[ext]}(u_t = \lambda, g_t)$  代入至(4.1)-(4.5)式，計算而得訊源解碼之後驗機率  $P_{SD}(u_t = \lambda | \tilde{y}_1^T)$ 。接著將此後驗機率經過交錯器後得  $P_{SD}(x_t = \lambda | \tilde{y}_1^T)$ ，再由(4.23)式求得考慮籬柵圖狀態的符號後驗機率  $P_{SD}(g_{t-1}, x_t = \lambda, g_t | \tilde{y}_1^T)$ 。至於經過交錯器後的訊源解碼額外資訊  $P_{SD}^{[ext]}(x_t = \lambda, g_t)$ ，則可由(4.22)式計算而得，並回傳至通道解碼器作為事前訊息。



4. 增加疊代次數  $\text{iteration} \leftarrow \text{iteration} + 1$ 。若已完成預設的疊代次數，則接著執行步驟 5，否則將回到步驟 2 繼續解碼程序。
5. 利用步驟 3 計算的符號後驗機率  $P_{SD}(u_t = \lambda | \tilde{y}_1^T)$ ，分別經由最大後驗機率及最小均方誤差原則分別估算出重建符號  $\hat{v}_t^{(\text{MAP})}$  及  $\hat{v}_t^{(\text{MMSE})}$ ，再用於計算其效能評估所需的符號錯誤率及重建訊雜比。

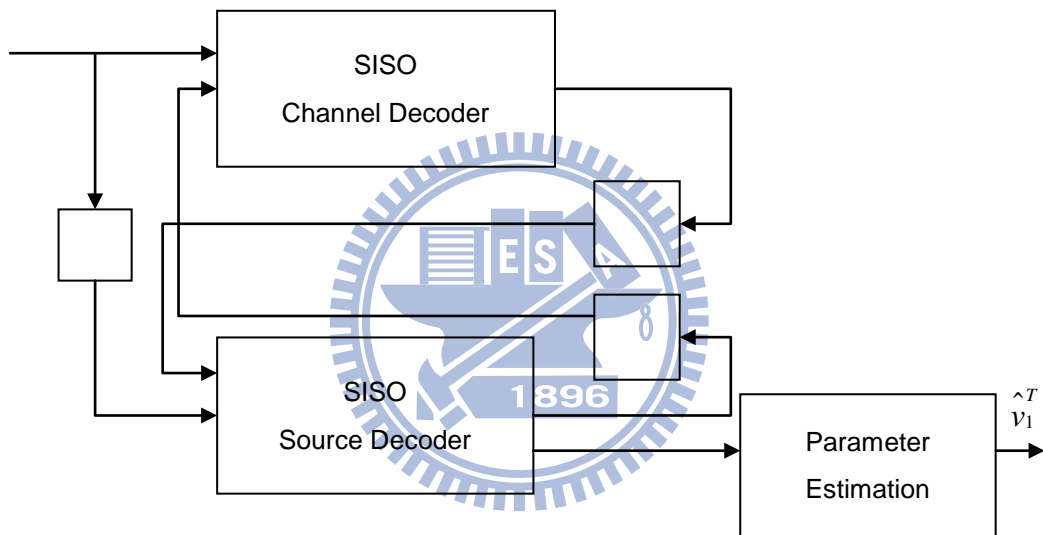


圖 4.1 系統接收端架構圖

## 第五章 實驗模擬與結果分析

在前面的章節已經詳細地介紹了傳輸架構及其編解碼演算法。針對具有特定機率分佈的訊號源，可變長度碼的碼字間所存在的殘餘冗息可運用在疊代解碼過程中。為了進一步驗證其效能，我們將製作模擬系統的傳送端與接收端，計算其符號錯誤率及重建訊雜比。

本章分為三項實驗，首先，我們將比較由改良式BCJR演算法與傳統BCJR演算法所推導出的軟性訊源解碼機制之解碼效能。實驗二將模擬第四章所介紹的索引層級疊代訊源通道解碼機制，並比較其與位元層級疊代解碼機制[7]之解碼效能。最後有別於前兩項實驗是針對一階高斯馬可夫訊號源，實驗三則探討輸入訊號為心電圖(Electrocardiography, ECG)之解碼效能。在此實驗中，除了利用符號錯誤率及PRD等客觀的效能評估標準，也將經過解碼還原的心電圖呈現，用以主觀地判斷重建之心電圖與原始心電圖之相似程度。

### [實驗一] 可變長度碼之軟性訊源解碼機制

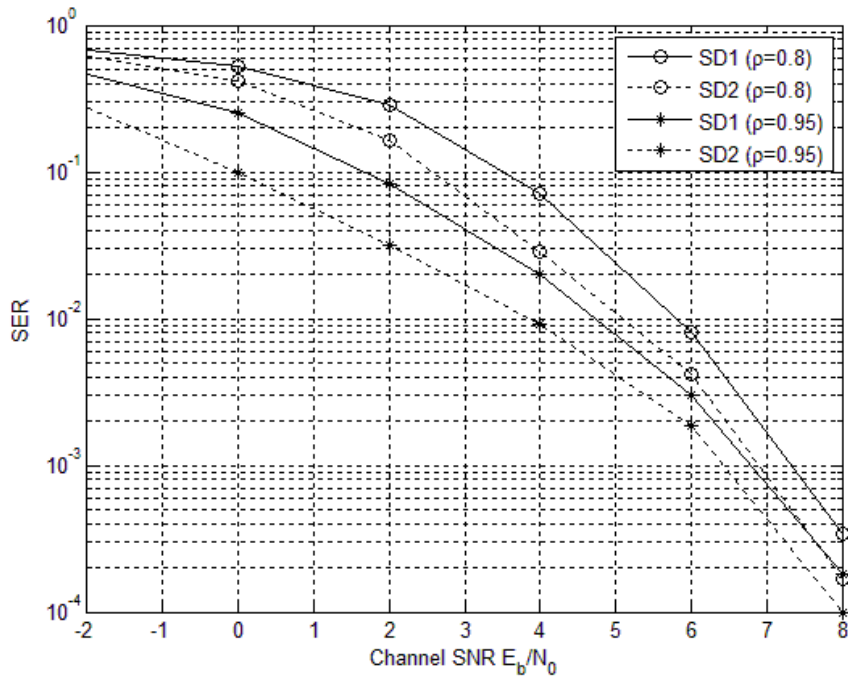
目的:探討不同訊源解碼演算法之效能

模擬環境:在這個實驗中，我們只考慮量化索引序列內含的訊源冗息之錯誤更正能力，因此圖2.1之傳輸端將暫時不考慮使用交錯器及通道編碼器。輸入符號 $v_i$ 採用一階高斯馬可夫訊號源，其變異數為 $\sigma_v^2=1$ ，

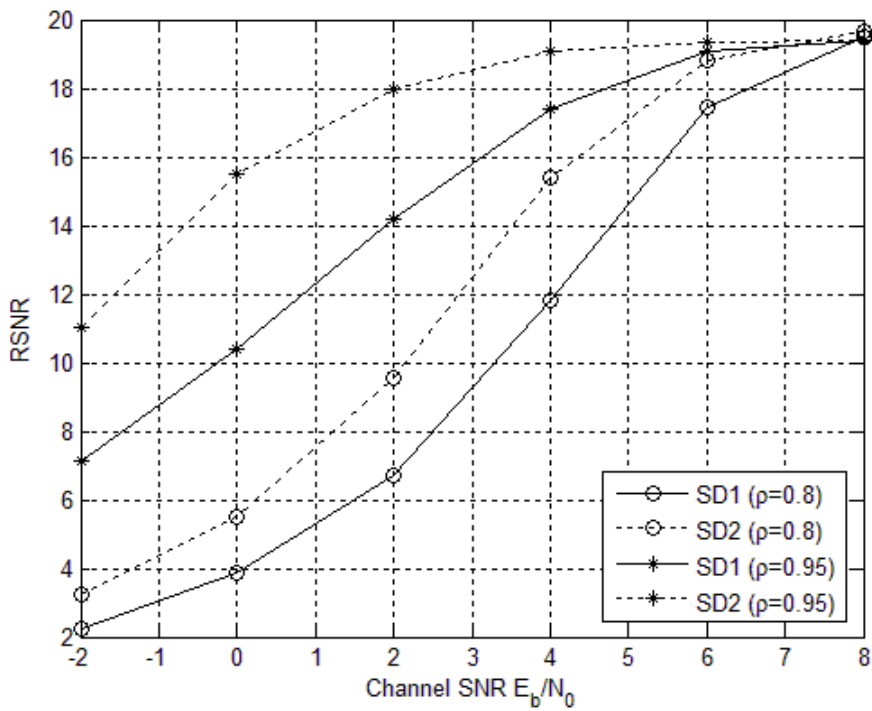
且考慮兩種相關係數  $\rho=0.8$  及  $\rho=0.95$ ，用以模擬相關性程度不同之訊號源。訊號源關係式表示為  $v_i = \rho \cdot v_{i-1} + n_i$ ，其中  $n_i$  為平均值為零的可加性白色高斯雜訊。

每一次模擬皆以封包  $v_1^T = [v_1, v_2, \dots, v_T]$  為單位傳送，每一個封包含有  $T=100$  個符號，而每個符號都經由  $M=4$  個位元的純量量化器處理後得到 16 階的索引序列  $u_1^T = [u_1, \dots, u_i, \dots, u_T]$ 。接著根據霍夫曼碼將封包所需的索引依序作可變長度編碼，將所有碼字串連可得位元序列  $w = [w_1, w_2, \dots, w_N] = [c(u_1), c(u_2), \dots, c(u_T)]$ 。由於採用了可變長度碼，故每個封包所輸出的位元總長度  $N$  不盡相同。最後以二位元相位鍵移(BPSK)調變之後，經由可加性白色高斯雜訊(AWGN)通道傳送。接收端則比較兩種軟性輸出訊源解碼機制之效能，其中SD1為前人提出之訊源解碼機制[7]，SD2為我們在4.1節所提出的訊源解碼機制。

實驗結果：



(a) 符號錯誤率



(b) 重建雜訊比

圖5.1 軟性訊源解碼效能

討論:本實驗模擬 4.2 節之訊源解碼架構，並將其解碼效能與傳統的

BCJR 演算法作比較，相關因子係數為  $\rho=0.8$  及  $\rho=0.95$  之輸入訊號源所得結果如圖 5.1 所示。圖 5.1(a) 之縱軸為由最大後驗機率(MAP)估算原則所得的符號錯誤率，橫軸為通道訊雜比，符號錯誤率會隨著通道環境變好而降低。圖 5.1(b) 之縱軸為由最小均方誤差 (MMSE) 估算原則所得的重建訊雜比，橫軸為通道訊雜比，重建訊雜比會隨著通道環境變好而上昇直到訊雜比之最大上限值。首先由圖中可看出，由於兩種訊源解碼機制皆運用到符號間之一階相關性，因此符號錯誤率以及重建訊雜比在  $\rho=0.95$  時之解碼效能都較  $\rho=0.8$  來得佳。就固定相關係數而言，由圖中可看出 SD2 在符號錯誤率及重建訊雜比都有較好的表現，由此驗證了 SD2 確實較 SD1 更有能力整合訊源冗息之一階相關性，而得到較多的解碼效能增益。進一步觀察在相關係數為  $\rho=0.95$  時，SD2 與 SD1 之解碼效能差異皆較  $\rho=0.8$  時顯著，再次驗證了兩種訊源解碼機制在運用訊號源冗息的差異。

## [實驗二] 可變長度碼之疊代訊源通道解碼機制

目的: 模擬位元層級與碼字層級疊代解碼機制，並比較系統的錯誤率以及重建訊號訊雜比。

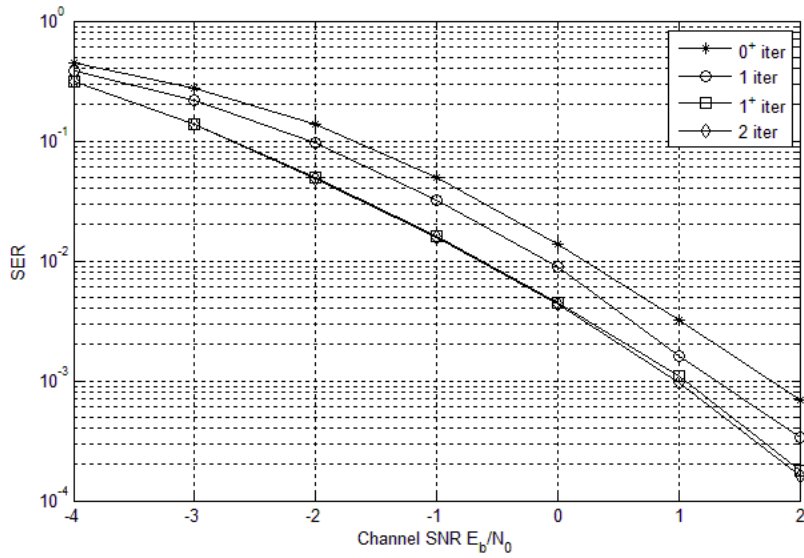
模擬環境: 考慮如圖 2.1 之傳輸架構，輸入符號  $v_t$  採用一階高斯馬可夫訊號源，其變異數為  $\sigma_v^2=1$ ，且考慮兩種相關係數  $\rho=0.8$  及  $\rho=0.95$ ，

用以模擬相關性程度不同之訊號源。訊號源關係式表示為  $v_t = \rho \cdot v_{t-1} + n_t$ ，

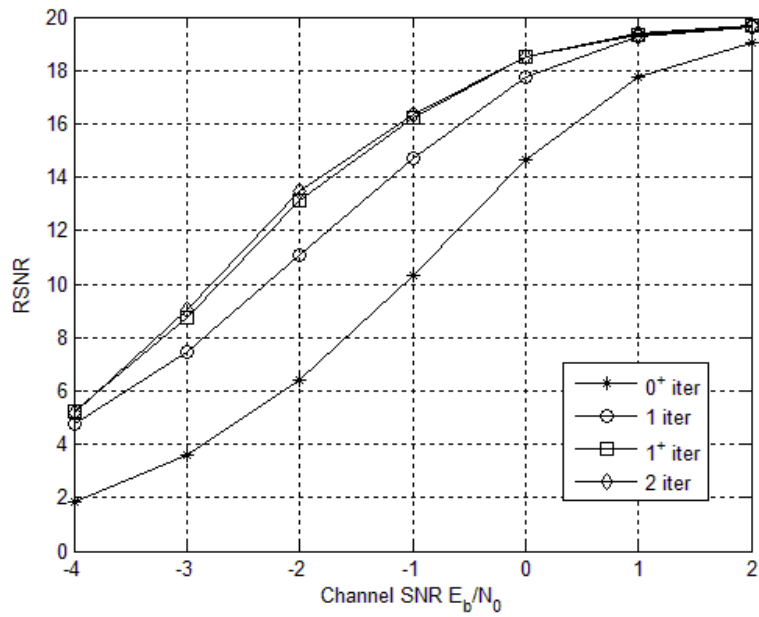
其中  $n_i$  為平均值為零的可加性白色高斯雜訊。每一次模擬皆以封包  $v_i^T = [v_1, v_2, \dots, v_T]$  為單位傳送，每一個封包含有  $T=25$  個符號，而每個符號都經由  $M=4$  個位元的純量量化器處理後得到 16 階的索引序列  $u_i^T = [u_1, \dots, u_i, \dots, u_T]$ ，接著根據霍夫曼碼將封包所屬的索引依序作可變長度編碼，得到碼字序列。接著將編碼完成的封包輸入碼字層級的 S-random 交錯器，且設定  $S = \frac{T}{2}$ ，交錯器之詳細流程於附錄 A 詳述。接下來，將置換過後的序列  $x_i^T = [x_1, \dots, x_i, \dots, x_T]$  經過一編碼率為  $1/2$  的遞迴系統性迴旋編碼器，而編碼生成矩陣為  $G(D) = \begin{bmatrix} 1 & \frac{(1+D^2)}{(1+D+D^2)} \end{bmatrix}$ ，其輸出序列為  $y_i^T = [y_1, \dots, y_i, \dots, y_T] = [(x_1, z_1), \dots, (x_i, z_i), \dots, (x_T, z_T)]$ ，其中  $x_i$  及  $z_i$  分別為訊息符號與同位檢查符號。最後將通道編碼器的輸出封包  $y_i^T$  以二位元鍵移調變後，再經由可加性白色高斯雜訊通道傳送。

本實驗將模擬位元層級及碼字層級兩種疊代解碼機制，前者 ISCD1 使用實驗一的 SD1 訊源解碼器及位元層級通道解碼器，而後者 ISCD2 則使用實驗一的 SD2 訊源解碼器及碼字層級通道解碼器。實驗結果顯示 ISCD1 在疊代次數  $\text{iteration}=3$  時達到效能收斂，而 ISCD2 則在  $\text{iteration}=2$  時收斂。碼字層級解碼機制 ISCD2 之接收端如圖 4.1 所示，其中通道解碼演算法如第三章所述，訊源解碼演算法如 4.1 節所推導，其疊代解碼流程及額外資訊之交換則如 4.2 節所說明。在多次疊代完成後，分別計算符號錯誤率以及重建訊雜比。

實驗結果：

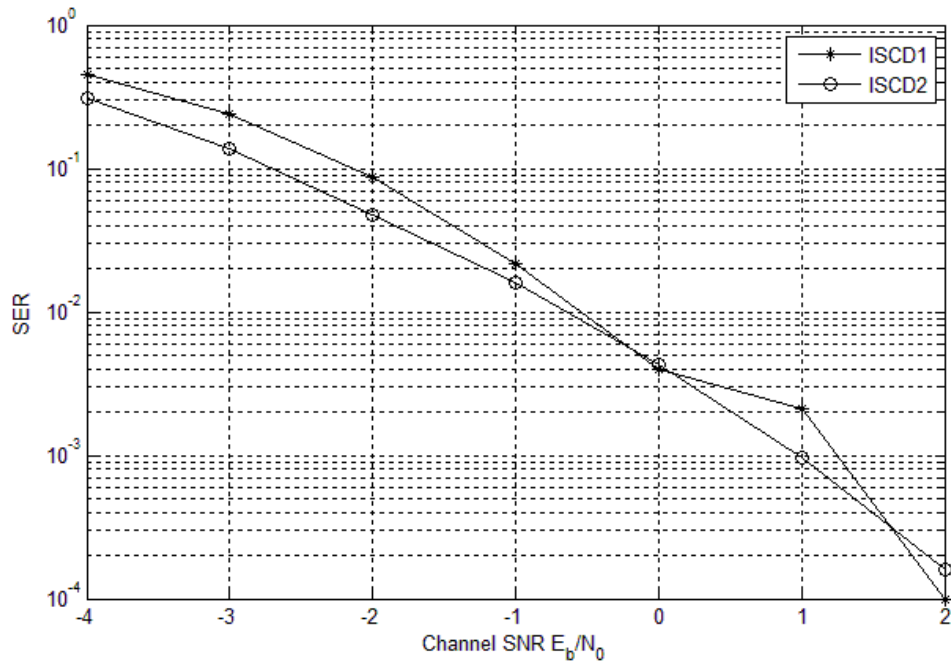


(a) 符號錯誤率

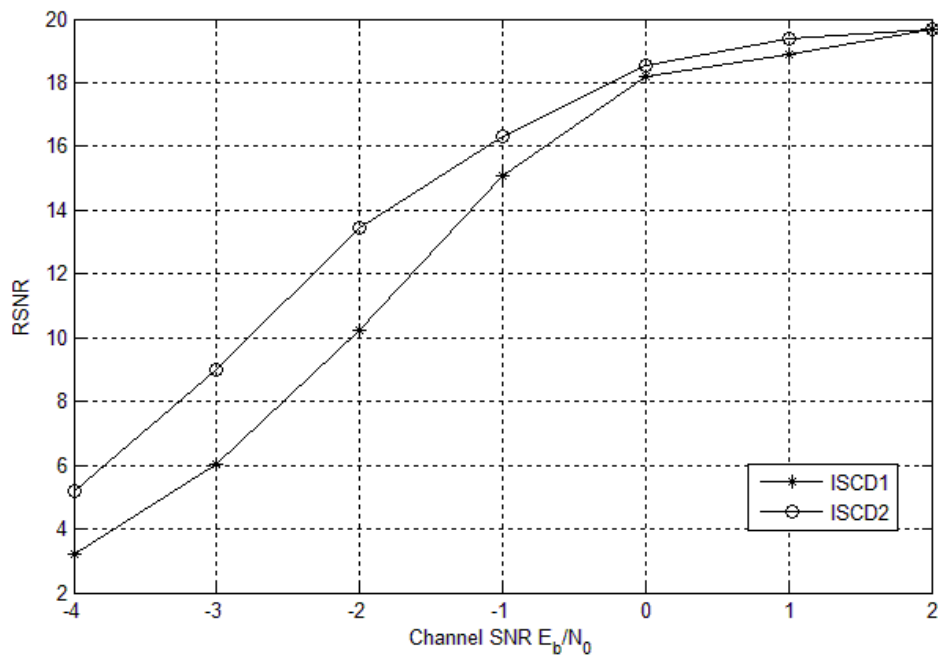


(b) 重建訊雜比

圖5.2 ISCD2的疊代解碼效能( $\rho=0.8$ )



(a) 符號錯誤率



(b) 重建訊雜比

圖5.3 位元與碼字層級疊代解碼效能之比較( $\rho=0.8$ )



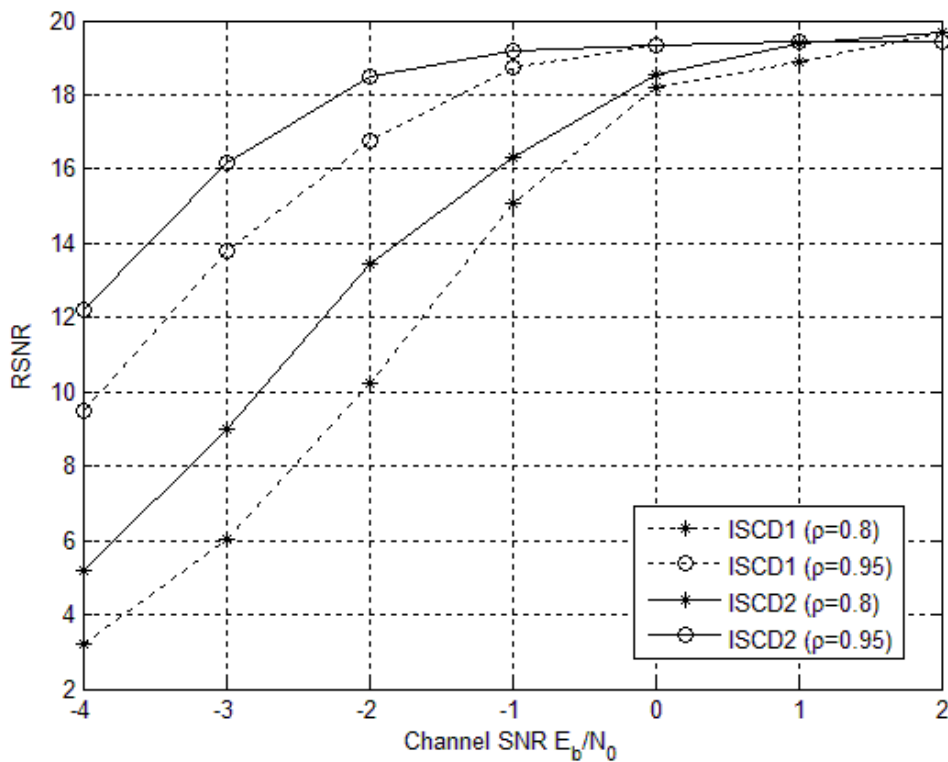


圖5.4 不同訊源疊代解碼的重建訊雜比( $\rho=0.8, \rho=0.95$ )

討論：首先模擬第四章之碼字層級疊代訊源通道解碼演算法 ISCD2，圖 5.2 為輸入訊號源之相關係數為  $\rho=0.8$  時之模擬結果。其中圖 5.2(a) 之縱軸為由最大後驗機率估算原則所計算的符號錯誤率，橫軸為通道訊雜比，而圖 5.2(b) 之縱軸為由最小均方誤差估算原則所計算的重建訊雜比，橫軸為通道訊雜比。解碼機制在 ISCD2 兩次疊代後達到效能收斂，疊代過程依序為圖中之  $0^+$  iter、1 iter、 $1^+$  iter 及 2 iter，其中  $0^+$  iter 表示直接經過一次通道解碼，1 iter 表示將通道解碼輸出之額外訊息整合於訊源解碼器之結果， $1^+$  iter 為將在 1 iter 時訊

源解碼輸出之額外訊息再回授至通道解碼器時進行通道解碼，最後 2 iter 為將 1<sup>+</sup> iter 所產生之通道解碼額外訊息整合於訊源解碼器之結果。從圖中可看出解碼效能會隨著疊代次數之上昇而提升，且 1<sup>+</sup> iter 與 2 iter 的效能相近表示此疊代解碼機制在第 2 次疊代時，已運用到足夠的額外資訊而達到解碼效能之收斂。

圖 5.3 為兩種疊代解碼機制 ISCD1 及 ISCD2 之比較，訊號源相關係數為  $\rho=0.8$ 。由圖中可看出 ISCD2 之解碼效能較 ISCD1 佳，其原因有二。首先在實驗一中驗證了 SD2 的解碼效能較 SD1 佳，再者如第三章所述，位元層級之解碼機制 ISCD1 在疊代解碼過程時，碼字與位元層級間的機率轉換破壞了符號內位元間的相關性。最後由圖 5.4 可看出，由於疊代解碼效能取決於訊號源之一階馬可夫相關性，因此輸入訊號之相關係數為  $\rho=0.95$  時之解碼效能會較  $\rho=0.8$  時佳。

### [實驗三] 心電訊號之疊代訊源通道解碼效能

目的：探討經過不同前置處理的心電圖訊號在本論文架構下之解碼效能

模擬環境：考慮如圖 2.1 之傳輸架構，本實驗採用實地量測的心電圖做為訊號源，接收端則採用我們提出的疊代解碼架構 ISCD2，並設定疊代次數為 2 次。心電圖是評估心臟循環功能的主要工具，有助於診

斷病患的心律不整症狀，如心跳速率、規則性、心臟電脈衝的發源地及傳導路徑的異常。可靠的心電訊號編碼機制對於遠距醫療及居家健康照護相當重要。有鑑於此，本實驗引用美國MIT-BIH (Beth Israel Deaconess Medical Center)公開提供的心電圖資料庫。此資料庫之單筆心電訊號平均紀錄時間為30分鐘，訊號取樣頻率為360Hz，每個取樣點以11個位元做純量量化。為兼顧壓縮率及重建品質，同時縮短系統模擬所需之時間，我們以有限狀態向量量化(finite-state vector quantization, FSVQ)[14]針對心電訊號做前置處理。相較於傳統的向量量化，具有記憶性的FSVQ可利用前一編碼區塊的資訊幫助下一區塊的運作，其犧牲了些微的重建品質，但具有更高的壓縮能力。所謂的FSVQ，是依預設的狀態數目 $K$ 事先訓練一組能涵蓋心電訊號特性的初始碼書(initial codebook)，並將所有的訓練向量依此初始碼書分羣，再針對各羣訓練出其 $M$ 位元的狀態碼書(state codebook)。不同的狀態碼書間則以下次狀態函數(next state function)定義其連結關係。執行FSVQ量化處理時，先將輸入訊號分割為 $k$ 維度的向量，再依序與目前的狀態碼書比對選擇其中最近似的碼字，並輸出該碼字的索引值 $v_i$ ，同時依下次狀態函數決定下一組狀態碼書。取得量化索引後即可執行本論文所述之編碼流程，編碼所需之可變長度碼依訓練訊號之機率分佈而產生，重建訊號則以經2次疊代解碼後所得最大後驗機率之索引

值直接查詢目前狀態碼書取得，同時依下次狀態函數決定下一次解碼後索引所對應的狀態碼書。心電圖重建訊號品質之評估標準為其PRD (Percent Root mean square Difference)，定義如下：

$$PRD = \sqrt{\frac{\sum_{i=1}^L [x_{ori}(i) - x_{rec}(i)]^2}{\sum_{i=1}^L x_{ori}(i)^2}} \times 100 \quad (5.1)$$

其中  $x_{ori}(i)$  為原始訊號， $x_{rec}(i)$  為還原後訊號， $L$  則為訊號長度。本

實驗使用美國MIT-BIH心電圖資料庫編號100之心電訊號，共取

600,000個樣本點做為訓練訊號，並依指定之維度 $k$ 分割為向量。測試

用訊號則固定取2,500個心電訊號向量，並分為100個封包，通道訊雜

比設定為-4、-2、0、2dB。FSVQ之狀態數目及量化位元數目分別設

定為 $K=16$ 及 $M=4$ ，向量維度則以 $k=2、4$ 分別進行模擬。為了評估平

均效能，我們將程式模擬重複10次並求出平均之SNR(dB)、PRD及SER。

為了進一步提升重建品質，模擬結果乃利用增益正規化後之訊號進行

訓練碼書及量化[15]，並假設在傳送端與接收端此增益值為已知且未

做量化處理。

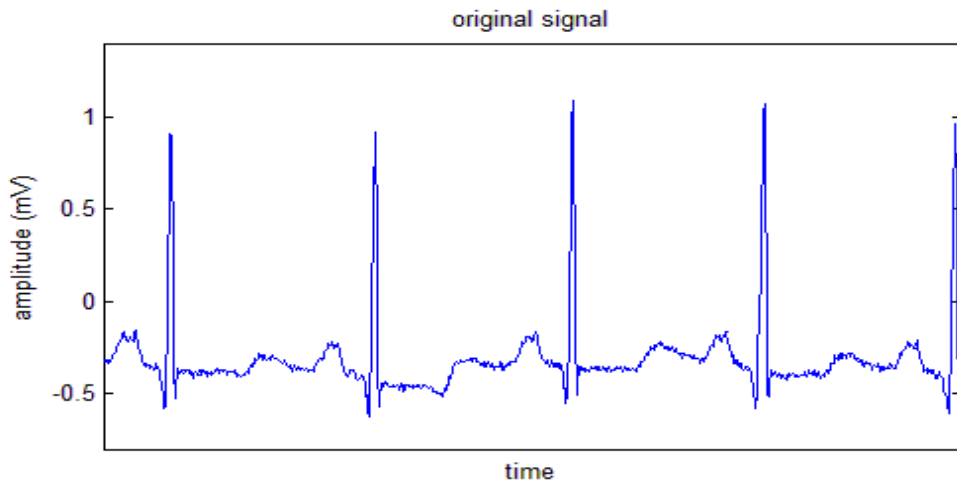
實驗結果：

索引值	$k = 2$		$k = 4$	
	索引機率	Huffman Code	索引機率	Huffman Code
1	0.02991	01111	0.02475	11001
2	0.01896	001101	0.02085	110000
3	0.04349	1011	0.00157	110001110
4	0.07598	0010	0.03977	1101
5	0.00796	0111010	0.00183	110001101
6	0.03404	00111	0.08167	0100
7	0.09590	110	0.00261	110001000
8	0.11795	100	0.10088	111
9	0.00422	0111011	0.00237	110001001
10	0.01892	011100	0.07976	0101
11	0.07261	0110	0.00205	110001100
12	0.09162	111	0.10829	011
13	0.02008	001100	0.00131	110001111
14	0.05645	1010	0.22289	10
15	0.13518	010	0.00462	11000101
16	0.17672	000	0.30478	00
	平均長度	3.59906	平均長度	2.87908
	entropy	3.56048	entropy	2.8469

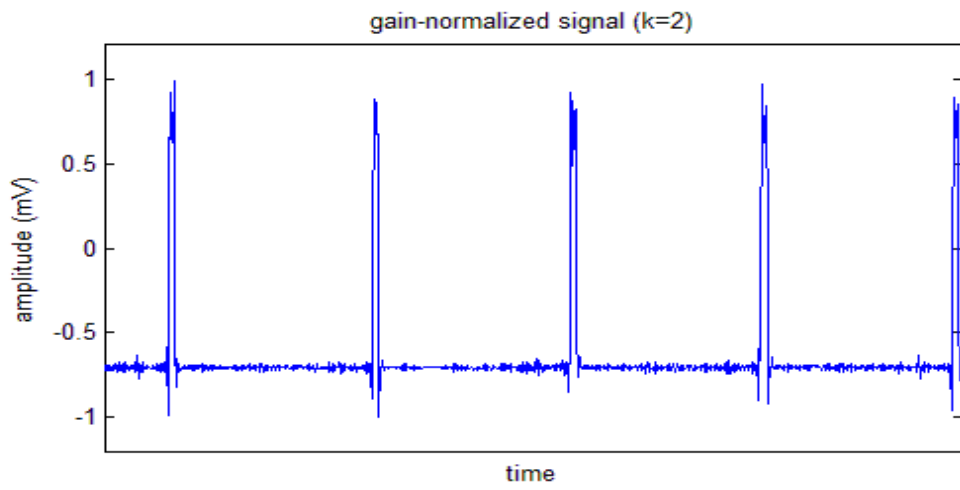
表5.1 不同向量維度之可變長度碼書

channel SNR(dB)	$k = 2$			$k = 4$		
	SNR(dB)	PRD	SER	SNR(dB)	PRD	SER
-4	3.2663	68.6567	0.5222	3.3078	68.3294	0.4940
-2	7.8723	40.4004	0.1398	10.0917	31.2908	0.0953
0	19.6819	10.3730	0.0102	27.4750	4.2291	0.0036
2	38.3637	1.2073	0.00052	33.1144	2.2094	0.00008
$\infty$ (FSVQ)	41.2024	0.8707	0	33.1149	2.2093	0

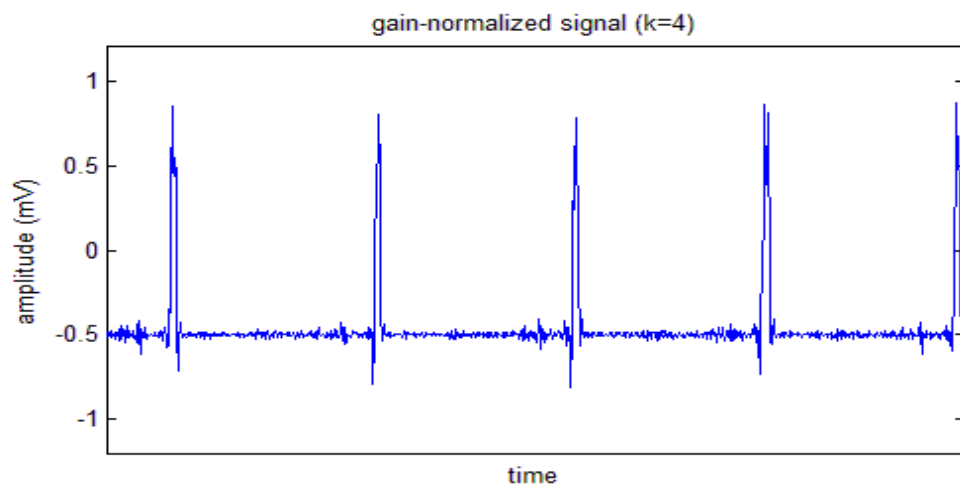
表5.2 心電圖之平均解碼效能



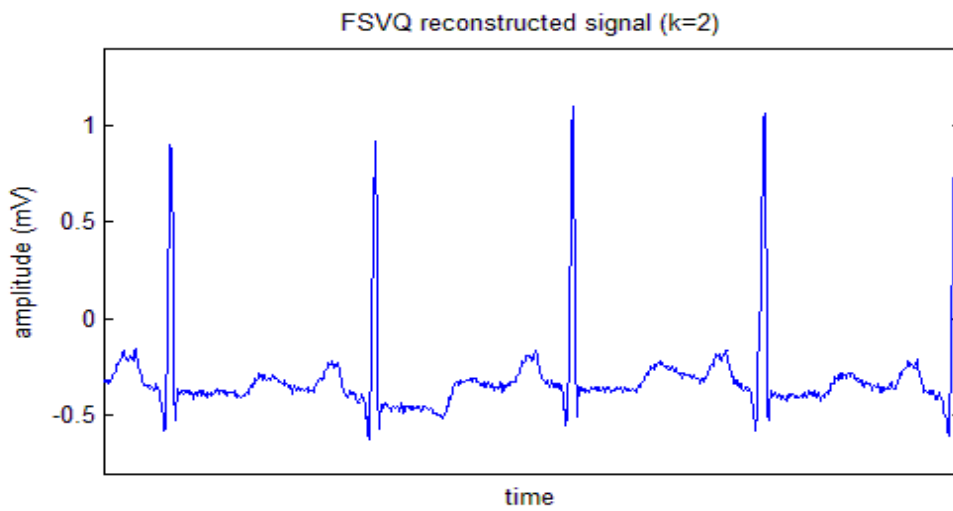
(a)



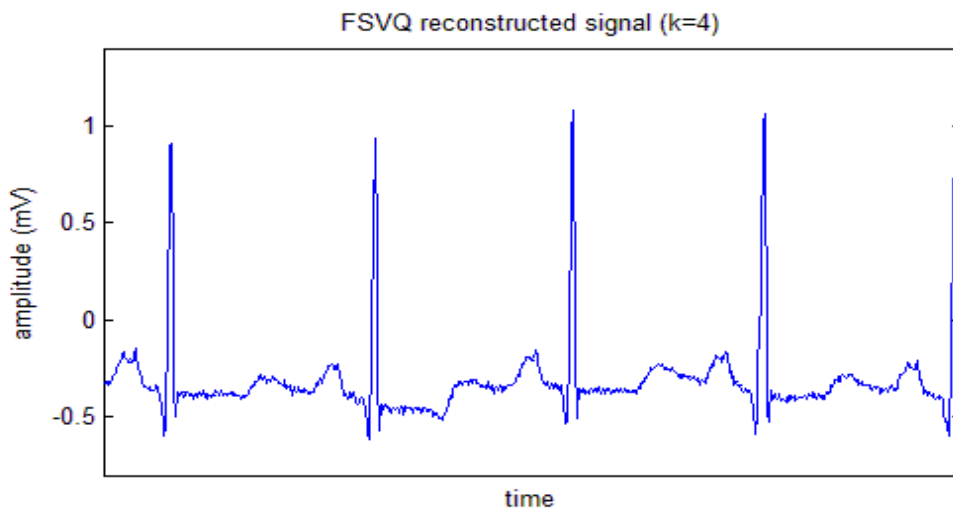
(b)



(c)

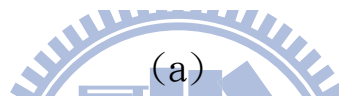
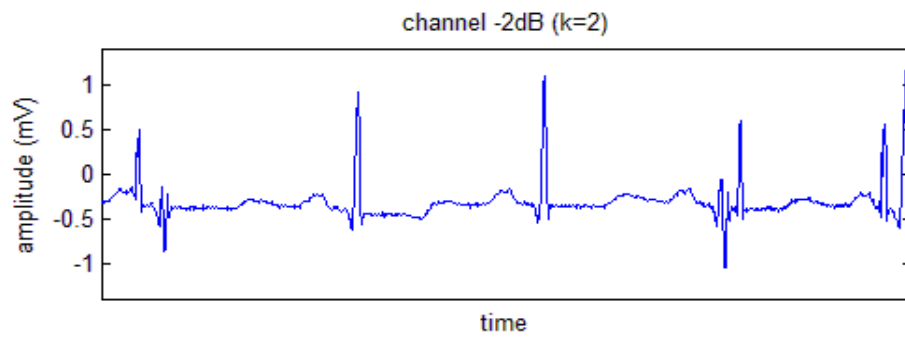
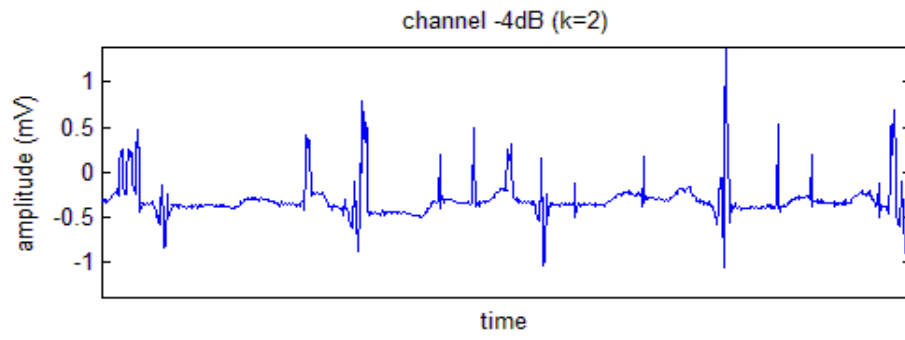


(d)

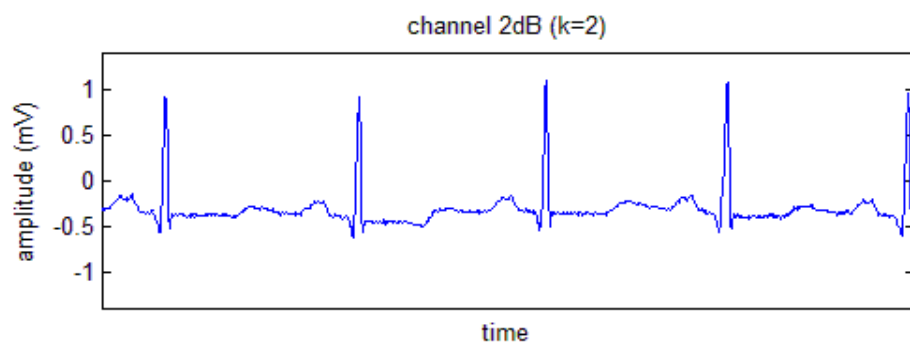
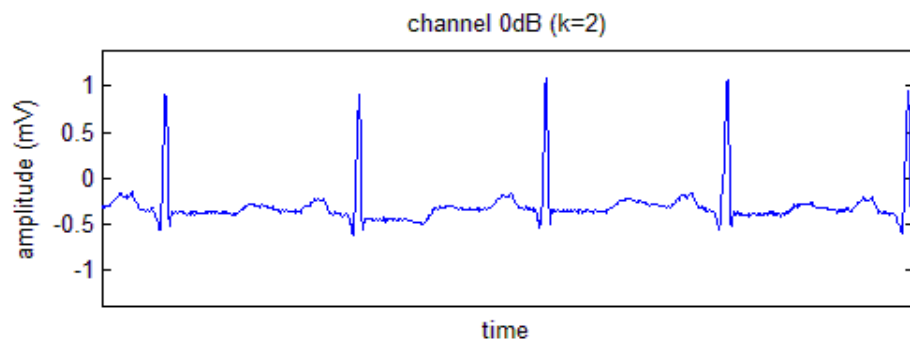


(e)

圖5.5 MIT-BIH資料庫編號100之相關圖 (a)原始心電訊號  
 (b)增益正規化訊號(k=2) (c)增益正規化訊號(k=4)  
 (c)重建訊號(FSVQ, k=2) (c)重建訊號(FSVQ, k=4)



(a)



(b)

圖5.6 重建訊號(k=2) (a)Channel SNR=-4, -2dB

(b) Channel SNR=0, 2dB



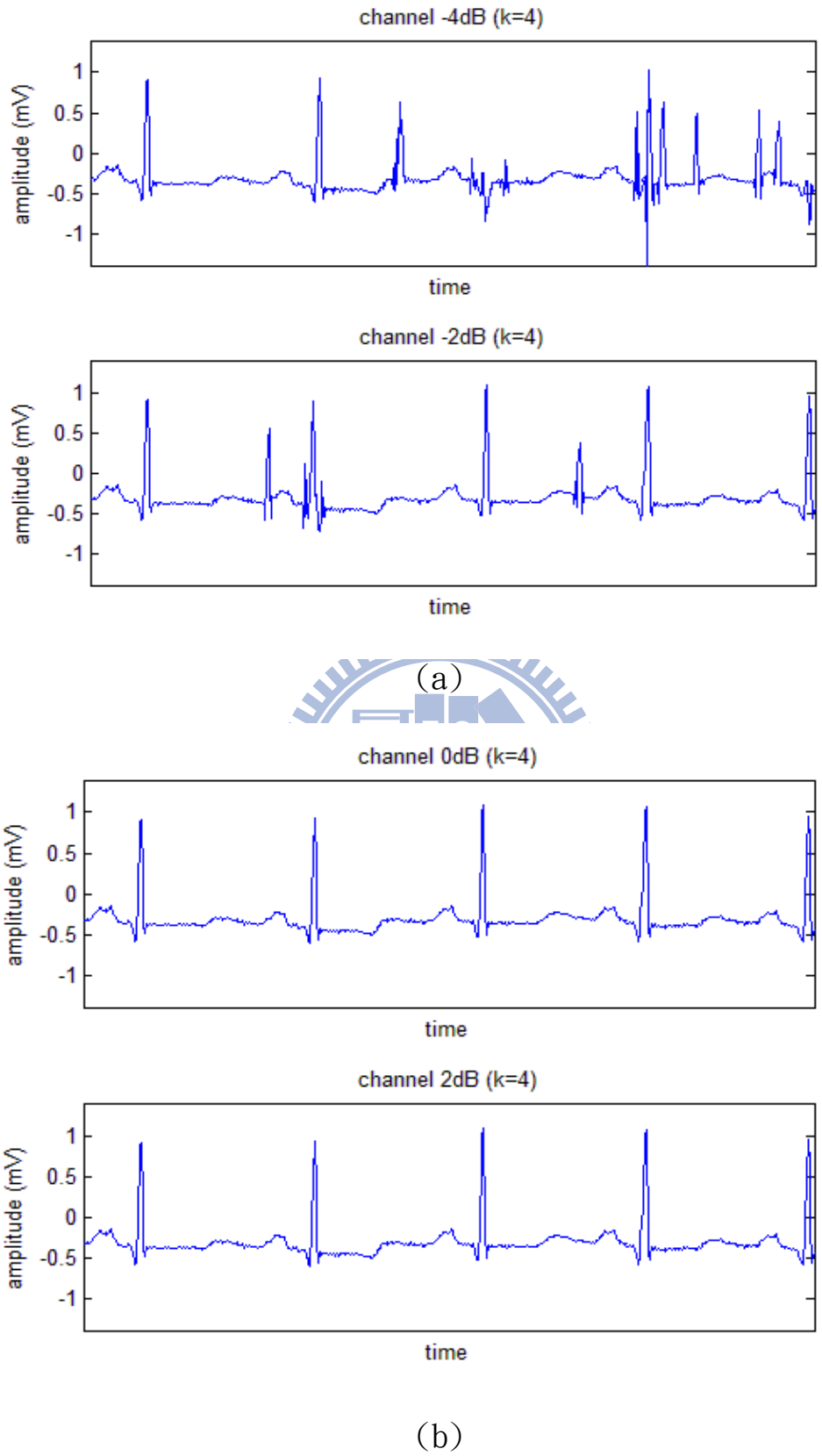
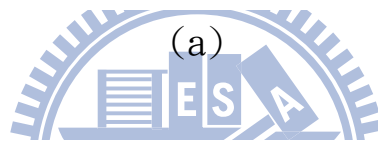
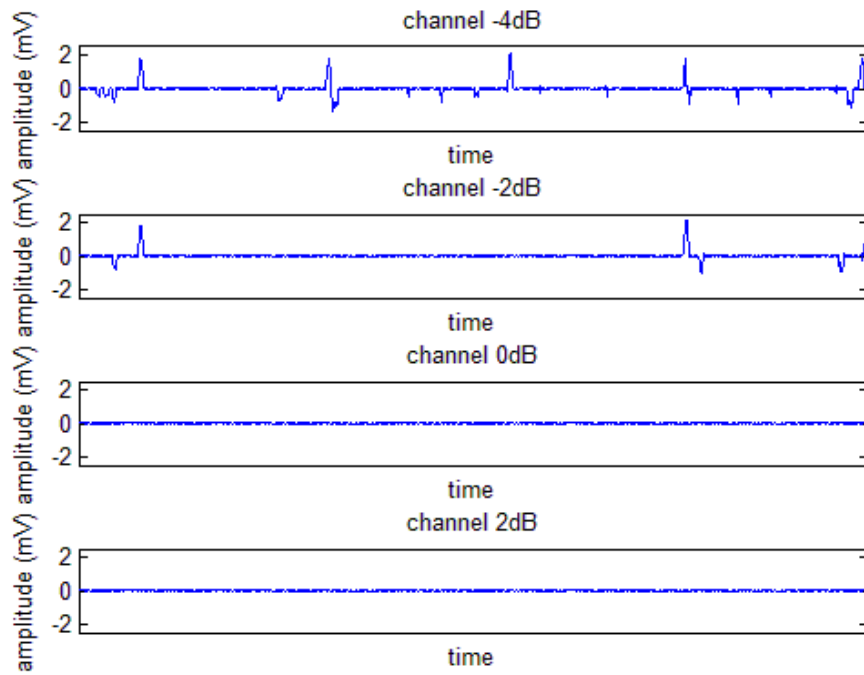
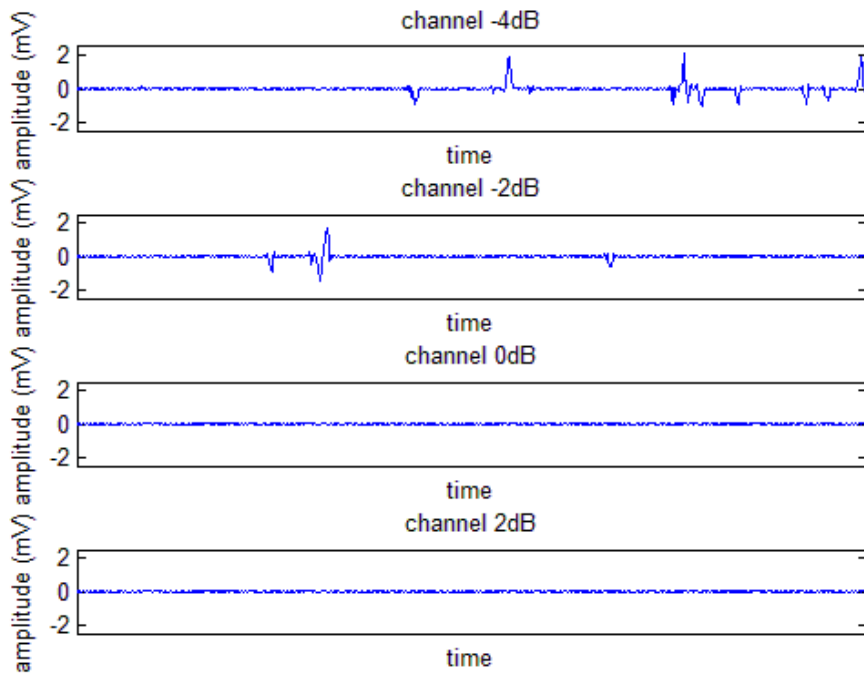


圖5.7 重建訊號(k=4) (a)Channel SNR=-4, -2dB

(b)Channel SNR=0, 2dB



(a)



(b)

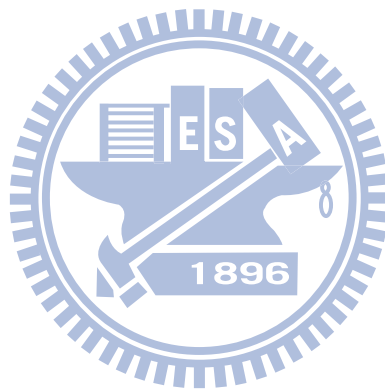
圖5.8 不同通道訊雜比下之重建誤差(a) $k=2$  (b) $k=4$

討論：表5.1列出不同維度之索引機率及其可變長度碼與平均長度，表5.2則列出在不同通道訊雜比下之平均解碼效能。由表5.1可看出 $k=4$ 時索引機率分佈較為集中在幾個索引，故其擁有較短的平均長度。而在進行模擬前，我們也分別觀察了向量維度 $k=2、4$ 之一階事前機率，發現 $k=4$ 時機率分佈也較為集中，如同前述實驗在相關性較高下的情況，因此其SER大部分較 $k=2$ 時低。得利於假設在傳送端與接收端有已知的增益值，不同向量維度下的PRD及SNR在通道訊雜比為0dB以上時皆有不錯的表現，尤其在通道訊雜比為2dB時，解碼還原訊號之PRD已相當接近僅以FSVQ處理下的結果。但此增益值已知的假設，未來需嘗試以後向調適(backward adaption)[16]或其他技術加以克服。

圖5.5採用顯示一段心電圖的原始訊號，及其在兩種向量維度 $k=2、4$ 時之增益正規化訊號及FSVQ重建訊號。由表5.2可發現心電訊號經FSVQ處理後所產生的誤差有限，而直接觀察重建波形也有相當高的保真度(fidelity)。

圖5.6及5.7分別顯示兩種向量維度 $k$ 的索引序列在不同通道訊雜比下之解碼還原訊號，圖5.8則顯示其重建誤差 $x_{ori}(i) - x_{rec}(i)$ 。比較圖5.5~5.7，可發現在通道訊雜比為0及2dB時，重建訊號與原始訊號非常相似；由圖5.8及表5.2，也可推論在此兩通道環境下，通道

雜訊造成的索引錯誤對重建誤差的影響有限。但在通道訊雜比為 $-2\text{dB}$ 時，因通道雜訊造成可變長度碼之索引解碼錯誤及判斷失序，效能開始滑落，重建訊號與原始訊號相比已有明顯的誤差，此現象在圖 5.6(a)、5.7(a)所示 $-4\text{dB}$ 時更為劇烈。同時我們也觀察到在 $-2$ 及 $-4\text{dB}$ 時會連續出現誤差較大的重建訊號，這是因為 FSVQ 具有記憶的特性使其容易在某部份索引產生解碼錯誤時，發生錯誤蔓延(error propagation)的現象。

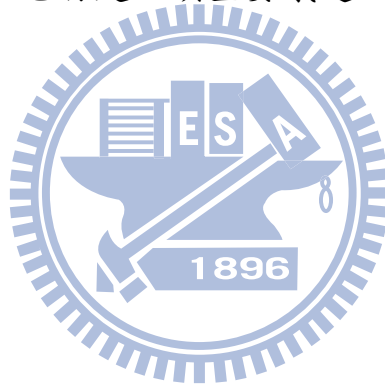


## 第六章 結論與未來展望

本論文主要探討基於可變長度碼之疊代訊源通道解碼機制的設計與應用。首先探討可變長度碼在碼字層級上的軟性通道解碼機制，並設計其解碼所需之二維狀態組成的籬柵結構，再運用 BCJR 演算法計算其後驗機率。此軟性輸出通道解碼機制可運用在疊代訊源通道解碼上，藉由訊源的機率分佈及可變長度碼的殘餘冗息特性，再根據渦輪原則相互交換訊源及通道解碼器輸出的額外訊息，可以有效改善系統的強健性。

現今行動通訊中，多媒體服務將成為趨勢，因此整合各種影音多媒體並經由無線通訊傳輸將成為熱門研究議題。而許多資料已有既定的壓縮編碼格式，例如 JPEG 為影像最為普遍的壓縮格式，一般使用符合前置條件(prefix condition)的霍夫曼碼作為其熵編碼(Entropy Coding)規格。其特色在於前向解碼時可符合索引同步的要求，且其平均傳輸位元量可以近似理論熵值。但霍夫曼碼對於位元錯誤相當敏感，很可能因為極少數的位元誤判而造成嚴重的連續性錯誤，在錯誤更正及疊代解碼上的效果非常有限。因此當疊代解碼系統運用在無線多媒體傳輸時，為了使訊號源間的殘餘冗息得以充分利用，需要考慮同時符合前置與後置條件的雙向性可變長度碼 (reversible VLC,

RVLC)。然而具有錯誤更正能力的雙向性可變長度碼設計仍是一個尚待解決的問題，因此未來應用在無線傳輸的系統中，壓縮量與更正能力的最佳化權衡是值得深入探討的議題。除此之外，我們在系統模擬中使用的通道環境僅考慮無記憶性的附加白高斯通道，並不符合真實多變的通道環境。因此在未來可望針對通道模擬環境加以改善，例如 Fading 通道…等等。問題是考慮更複雜通道的同時，將會使疊代解碼的的運算量大幅增加。因此如何克服龐大的運算，使得多媒體的互動式服務可以即時製作化也將是一項重要議題。



## 附錄A S-random 交錯器

在系統中，可變長度碼以一個具有T個符號所組成的封包作為單位來傳輸。而交錯器的使用目的在於希望能達到交錯後的資料相鄰之間的關聯性越小越好，因此我們選擇具良好交錯效果的S-random交錯器。其目標是要使交錯後的碼字序列像隨機選取的一樣，使相鄰的碼字間相關性最少。然而為了確保隨機選取的資料不會因巧合而發生高相關性的問題，因此設定一個數字S，使交錯後的碼字序列每個相鄰的位元都來自原始位置至少相差S以上的地方。

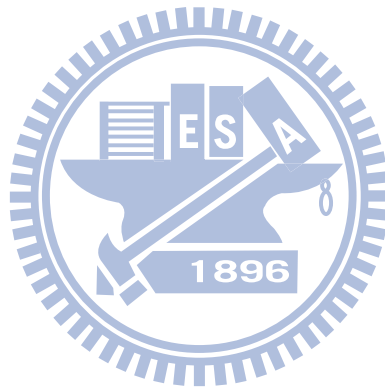
交錯器步驟：

1. 假設S為一常數。先空出與封包長度T一樣長的資料串。由於起始時沒有前面的參考值，所以第一筆資料將由隨機自原始資料串選取。假定我們隨機選取到原始位置為a作為交錯後的第一筆資料。注意取過的資料，就不能再取。
2. 接著再隨機選取一個數字。若這次我們選到的是原始位置為b的資料，計算與前一筆資料的原始距離  $|a-b|$  是否大於設定的S，若  $|a-b| < S$ ，則表示這相鄰的兩筆資料原始距離比我們假定的S還要來得小，所以現在不能選擇b作為下一筆資料，必需放棄。再重新隨機選取數字。
3. 若重新隨機選取的數字為c，且  $|a-c| \geq S$ ，則c可以被選做為下一

筆資料。

4. 以最後的一筆資料作為參考值 $a$ ，回到步驟2，以此類推，一直做到最後填滿為止。

在交錯器執行到尾聲，可能會出現一種情況，就是在最後剩下少數資料還沒被填入，但剩下還沒被挑進去的資料全部之間距離都小於 $S$ 的時候，若為慎重，需要全部重新挑過，挑到全部符合為止。至於 $S$ 值的決定，經前人研究發現，選擇 $S = \sqrt{\frac{T}{2}}$ 對於資料的交錯可以達到很好的效果。





## 參考文獻

- [1] T. Fingscheidt and P. Vary, "Softbit Speech Decoding: A New Approach to Error Concealment," *IEEE Trans. Speech Audio Processing.*, vol. 9, no.3, pp.240-251, 2001.
- [2] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449-2457, Sep. 1995.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding: Turbo Codes," in Proc. *IEEE International Conference on Communications (ICC)*, pp. 1064-1070, 1993.
- [4] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding And Decoding: Turbo-Codes," *IEEE Trans. on Commun.*, vol. 44, no. 10, pp. 1261-1271, 1996.
- [5] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in Proc. *3rd ITG Conf. Source Channel Coding*, Munich, Germany, Jan. 2000, pp. 111-116.
- [6] R. Bauer and J. Hagenauer, "Iterative source/channel -decoding using reversible variable length codes," in Proc. *IEEE Data Compression Conf.* , Snowbird, UT, Mar. 2000, pp. 93-102. 75
- [7] J. Kliewer and R. Thobaben, "Iterative Joint Source-Channel Decoding of Variable-Length Codes Using Residual Source Redundancy," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 919-929, 2005.
- [8] R. Thobaben and J. Kliewer, "Robust decoding of variable-length encoded Markov sources using a three-dimensional trellis," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 320-322, Jul. 2003.
- [9] Chun-Feng Wu and Wen-Whei Chang, "Iterative Source-Channel Decoding Using Symbol-Level Extrinsic Information," *IEICE Transactions on Communications*, Volume E93.B, Issue 12, pp. 3555-3563.

- [10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 287–294, Mar. 1974.
- [11] Y. Liu, S. Lin, and M. P. C. Fossorier, "MAP algorithms for decoding linear block codes based on sectionalized trellis diagrams," *IEEE Trans. Commun.*, vol.48, pp.577-587, April 2000.
- [12] M. Bingeman and A. K. Khandani, "Symbol-based turbo codes," *IEEE Communications Letters*, vol.3, pp.285-287, Oct.1999.
- [13] K. Sayood, H.H Otu and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Transaction on Communications*, vol.48, no.5, pp.787-794, May 2000.
- [14] J. Foster, R. M. Gray, M. Dunham, "Finite-state vector quantization for waveform coding", *IEEE Trans. on Info. Theory*, vol. IT-31, no. 3, May 1985.
- [15] Chia-Chun Sun and Shen-Chuan Tai, "Beat-Based ECG Compression Using "Gain-Shape Vector Quantization", *IEEE Trans. on Biomedical Engineering*, vol. 52, no. 11, Nov. 2005.
- [16] Juin-Hwey Chen and A. Gersho, "Gain-Adaptive Vector Quantization with Application to Speech Coding", *IEEE Trans on Comm.*, vol. COM-35, no. 9, Sep. 1987.