

國立交通大學

電信工程研究所

碩士論文

基於疊代解碼效能的可變長度碼設計

A Variable-Length Coder Design Based on the
Performance of Iterative Decoding

研究生：戴玲玲

指導教授：張文輝 博士

中華民國九十九年六月

基於疊代解碼效能的可變長度碼設計

**A Variable-Length Coder Design Based on the Performance of Iterative
Decoding**

研究生：戴玲玲

Student : Ling-Ling Dai

指導教授：張文輝

Advisor : Wen-Whei Chang

國立交通大學

電信工程研究所



Submitted to Institute of Communication Engineering
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 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

基於疊代解碼效能的可變長度碼設計

學生：戴玲玲

指導教授：張文輝 博士

國立交通大學電信工程研究所碩士班



本論文主要探討可變長度碼在疊代訊源通道解碼機制上的設計與應用。首先針對可變長度碼在位元層級上的籬柵結構，推導其軟性訊源解碼演算法，並將其整合運用於疊代訊源通道解碼機制。主要是針對具特定機率分佈的訊源，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。另外探討各種不同的可變長度編碼方式，並以額外資訊轉換圖加以理論分析。最後提出基於額外資訊轉換特性的可變長度碼設計，實驗證實能有效對抗通道雜訊干擾，進而提昇系統的強健性。

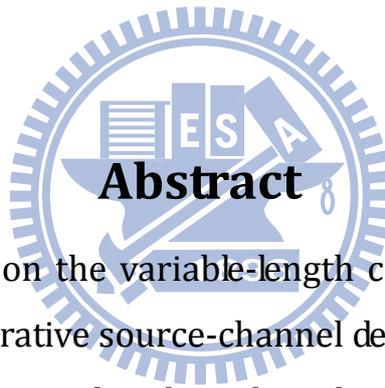
A Variable-Length Coder Design Based on the Performance of Iterative Decoding

Student: Ling-Ling Dai

Advisor: Dr. Wen-Whei Chang

Institute of Communication Engineering

National Chiao Tung University



This thesis focuses on the variable-length code (VLC) design using the performance of iterative source-channel decoder (ISCD). Our work starts with a soft-bit source decoding algorithm based on the bit-based trellis. According to the source distribution, VLC can exploit the residual redundancy between source-encoder indexes to enhance the robustness of the ISCD system. Then we analyze the performance of different VLC schemes, in terms of the extrinsic information transfer chart (EXIT chart). To further improve the reliability of ISCD system, we propose a new approach to VLC based on the EXIT chart. Simulation results show that the proposed VLC scheme can improve the error robustness of ISCD system.

致謝

本篇論文得以順利完成，首先要由衷地感謝我的指導教授張文輝老師，由於老師的細心教導與耐心指正，讓我瞭解到做研究時所應該持有的態度與精神，使我受益良多。此外也特別感謝實驗室的學長、同學以及學弟妹，無論是在研究或課業上都非常願意共同討論，並熱心地協助我解決許多問題。同時也要感謝在準備進入研究所以及就學期間一路支持並鼓勵我的朋友們。最後，僅將此論文獻給我的父母及家人。



目錄

中文摘要	i
英文摘要	ii
致謝	iii
目錄	iv
圖目錄	vi
表目錄	viii
第一章 緒論	1
第二章 可變長度碼之軟性訊源解碼機制	6
2.1 訊源解碼架構	7
2.2 位元層級籬柵圖	8
2.3 位元後驗機率推導	11
2.3.1 前向機率遞迴計算	11
2.3.2 後向機率遞迴計算	13
2.4 索引錯誤率之計算	15
第三章 可變長度碼之疊代訊源通道解碼機制	19

3.1	系統架構與流程.....	20
3.2	通道解碼演算法.....	23
3.3	解碼輸出之額外訊息.....	25
第四章	雙向性可變長度碼之最佳化設計.....	28
4.1	雙向性可變長度碼.....	29
4.1.1	基於最小平均長度的設計.....	31
4.1.2	基於錯誤更正能力的設計.....	33
4.2	額外資訊轉換特性分析.....	36
4.3	索引指定最佳化設計.....	45
第五章	實驗模擬與結果分析.....	51
第六章	結論與未來展望.....	71
附錄 A	S-random 交錯器.....	73
	參考文獻.....	75



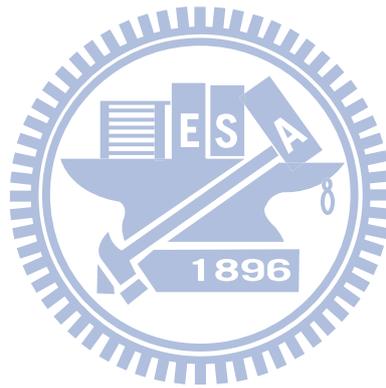
圖目錄

圖 2.1	可變長度碼之傳輸模型.....	8
圖 2.2	可變長度碼之樹狀圖與籬柵圖.....	9
圖 2.3	碼串狀態轉移與籬柵關係的示意圖.....	10
圖 2.4	已知 (N, K) 的索引路徑圖.....	17
圖 3.1	系統傳輸端架構圖.....	20
圖 3.2	系統接收端架構圖.....	21
圖 4.1	四位元碼字樹狀圖.....	29
圖 4.2	軟性輸入軟性輸出解碼器.....	37
圖 4.3	通道解碼器架構圖.....	41
圖 4.4	通道解碼器之額外資訊轉換圖.....	42
圖 4.5	訊源解碼器架構圖.....	43
圖 4.6	訊源解碼器之額外資訊轉換圖.....	44
圖 4.7	RVLC 的 EXIT Chart.....	47
圖 4.8	可變長度碼之二位元置換演算法.....	48
圖 5.1	不同可變長度碼之解碼效能比較.....	52
圖 5.2	Huffman Code 之 EXIT Chart.....	55
圖 5.3	RVLC ₁ 之 EXIT Chart.....	56
圖 5.4	RVLC ₂ 之 EXIT Chart.....	56

圖 5.5 可變長度碼之 EXIT Chart 比較 (CSNR=3dB).....	57
圖 5.6 Huffman Code 之 BER.....	57
圖 5.7 RVLC ₁ 之 BER.....	58
圖 5.8 RVLC ₂ 之 BER.....	58
圖 5.9 可變長度碼之 BER 比較.....	59
圖 5.10 Huffman Code 之 SER.....	60
圖 5.11 RVLC ₁ 之 SER.....	60
圖 5.12 RVLC ₂ 之 SER.....	61
圖 5.13 可變長度碼之 SER 比較.....	61
圖 5.14 RVLC ₃ 之 EXIT Chart	65
圖 5.15 RVLC ₂ 與 RVLC ₃ 之 EXIT Chart 比較.....	66
圖 5.16 RVLC ₃ 之 BER.....	66
圖 5.17 RVLC ₃ 之 SER.....	67
圖 5.18 RVLC ₂ 與 RVLC ₃ 之 BER 比較.....	67
圖 5.19 RVLC ₂ 與 RVLC ₃ 之 SER 比較.....	68

表目錄

表 4.1 第三層碼字之可用候選碼字表.....	31
表 4.2 索引指定對照表.....	49
表 5.1 不同可變長度碼之碼書.....	70



第一章 緒論

近年來科技進步，行動多媒體通訊的相關運用與互動將成為關鍵服務，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、影音…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及解碼機制，將是一個重要的研究議題。

所謂可變長度碼，是基於具特定機率分佈的訊源，針對不同發生機率的資料使用不同長度的碼字加以編碼。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，其中較有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)…等。這此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，在索引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，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架構規劃時，假設另一個編碼器已完成最佳化設計，然而真實的通訊環境，會因為訊號的延遲及系統的複雜度各種限制而造成其具體實現上的困難。根據前人實驗觀察，訊號源經訊源編碼器處理後，位元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，即可在不需要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) 以改善解碼工作的正確性。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架構設計考量的研究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)。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[3]被提出，並稱為軟性訊源解碼機制(soft-bit source decoding, SBSB)。而在[4]所提出的訊源控制通道解碼機制，則試圖將殘餘冗息運用在通道解碼機制中。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。這樣的事前機率大多來自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，比起單獨使用其中一項進一步達到更好的效果。

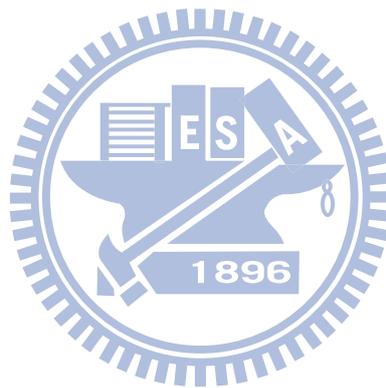
渦輪碼[5][6]是通道解碼機制的重大突破。其傳送端是由兩個迴旋編碼器串聯且配合中間串聯的交錯器所組成，而接收端則是將個別解碼器輸出的額外訊息，相互交換作為另一個解碼器的事前訊息，以期加強位元判斷的可靠度。這樣的疊代架構，使渦輪解碼機制經證實能接近 Shannon 理論且具實用價值的編碼技術。而疊代訊源解碼機制運用了渦輪解碼器的疊代架構，將通道解碼器與訊源解碼器作連結，前者利用迴旋碼的錯誤檢查能力得到冗息，而後者則利用訊源殘餘冗息作為加強解碼能力的依據。相較於訊源編碼和通道編碼分開考量的傳統的通訊系統設計，疊代訊源解碼機制因運用渦輪解碼原理而在估測訊源符號的品質上有相當大的改進。

如前所述，可變長度碼是針對具特定機率分佈的來源資料使用長度不同的碼字加以編碼，而這樣的訊號來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但被認定為強健的解碼機制，也被運用為可變長度碼的傳輸系統。[7]與[8] 針對不相關的訊號源解碼做實驗，並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 演算法計算軟性後驗機率。這種將與訊源符號相關的軟性輸出運用在疊代解碼上的方式，在[9]與[10]中延伸使用在一階高斯-馬可夫訊號源，而訊源之間的強烈的相關性使殘餘冗息更具有解碼效益，解碼器運用這樣的資訊達到更顯著的效果。

然而解碼效能除了解碼器的設計及演算法的強健度之外，可變長度碼本身的設計也具有相當的影響力。可變長度碼因為每個碼字長度不同，位元錯誤將引起同步失序問題並嚴重影響索引錯誤。早期提出的前置碼(Prefix code)觀念，可以在無位元錯誤的情形下實現正確的前向索引解碼。然而在受到雜訊干擾而發生少數位元誤判的情形下，仍會造成索引嚴重的連續錯誤。由此可知，可變長度碼可經由設計更具有同步的鑑別能力，盡可能的減少位元錯誤，即可進一步使同步失序問題得以改善。可變長度碼因碼字長度具有彈性，因此基於目的不同而有各種不同的設計方式。除了解碼器的解碼演算法之外，我們透過可變長度碼的設計，讓整體疊代系統更能有效地利用殘餘冗息，不但使系統效能得以改善，更能減少位元錯誤造成同步失序的索引連續錯誤。

本文將首先在第二章介紹可變長度碼基於位元層級籬柵圖的解碼架構及其相關推導，並提出一項索引解碼的方式。第三章則針對整體的疊代訊源通道解碼機制做推導，並說明其解碼步驟。在第四章介紹雙向性可變長度碼的特性及編碼方式，再說明分析疊代解碼機制效能的額外資訊轉換圖，利用這樣的分析方式加以設計雙向性可變長度碼，以期系統有最佳的效能。最後則提供實驗數據及分析結果。



第二章 可變長度碼之軟性訊源解碼機制

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。在早期的研究文獻中，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。本章節將依據此概念，逐步探討可變長度碼與其解碼系統的設計。第一節說明系統的傳輸結構，第二節則介紹可變長度碼及其樹狀圖(code tree)和籬柵圖(code trellis)，第三節將推導訊源解碼之軟性位元輸出，最後一節則探討可變長度碼在索引錯誤率的估算模式。

2.1 訊源編碼架構

系統的傳輸模型如圖 2.1 所示。傳輸以封包(packet)為單位，一個封包內由 K 個具關聯性的符號(symbol)所組成，標示為 $\underline{U}=[U_1, U_2, \dots, U_K]$ 。其中封包大小 K 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 K 為已知，本論文將僅針對 K 為已知的情況進行系統之設計。在傳送端第 k 時間的符號 U_k ，經過 M 位元的量化(quantization)後得到索引(index)值 $I_k = \lambda \in \mathcal{I}$ ，其中 $\mathcal{I} = \{0, 1, \dots, 2^{M-1}\}$ 為量化的索引集合，且假設事先訓練可得其索引機率分佈 $P(\lambda)$ 。令 $\mathcal{C} = \{C(0), C(1), \dots, C(2^{M-1})\}$ 為一組可變長度的碼書(code book)， $c(\cdot)$ 為針對該組碼書 \mathcal{C} 由索引值對應至該索引碼字之函數，而 $l(\cdot)$ 為一函數計算出該碼字之長度。索引值 I_k 利用一前置碼(prefix code) \mathcal{C} 執行可變長度編碼後，得到位元向量 $c(\lambda) = C(I_k = \lambda)$ ，其碼字長度為 $l(c(\lambda))$ 。針對一組具 K 個符號的封包而言，經過可變長度編碼器可輸出總長度為 N 的二位元字串 $\underline{b} = [c(I_1), c(I_2), \dots, c(I_K)] = [b_1, b_2, \dots, b_N]$ ，其中第 n 個位元表示為 $b_n \in \{0, 1\}$ 。經過附加性白高斯雜訊(Additive White Gaussian Noise, AWGN)通道(channel)後，得到雜訊干擾的實數串 $\underline{\hat{b}} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_N]$ ，再依據最大後驗機率(maximum a posteriori probability)原則進行訊源解碼以得到其估算的符號串 $\underline{\hat{U}}$ 。

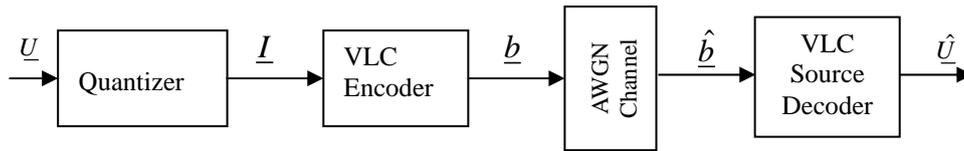


圖 2.1 可變長度碼之傳輸模型

2.2 位元層級籬柵圖

可變長度碼的設計概念是讓較常出現的量化索引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 λ 的發生機率。眾人皆

知的霍夫曼編碼(Huffman coding)方式即為一範例，其設計需滿足前置碼條件(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。針對任一組前置碼 \mathcal{C} ，都存在其對應的碼字樹狀圖(code tree)。為了提供後續的解碼與分析使用，以下將介紹可變長度碼的樹狀圖與籬柵圖(trellis diagram)，並定義其籬柵符號及狀態轉移方式。

可變長度的編碼模式可以表示為一個有限狀態機(finite-state machine)，所有可能的輸出序列 $\underline{b} = [b_1, b_2, \dots, b_N]$ 則可使用籬柵圖加以描述。一種較為常用的狀態轉移定義是源自於編碼樹狀圖，由已知的可變長度碼書利用碼字樹狀圖表示，並依序標示狀態代號 s_0, \dots, s_{N_s-1} ，其中 N_s 為狀態總數。 s_0 為起始狀態，且必須標示在樹狀圖之根節點(root node)上，並設計每個葉節點(leaf node)最終都將回歸至狀態

s_0 。由樹狀圖決定狀態轉移方式後，將所有的節點(node)排列並繪出狀態轉移路徑即可得到籬柵圖。由於二位元字串 $\underline{b}=[b_1, b_2, \dots, b_N]$ 所形成序列中，每一個位元位置都有一組籬柵圖，經排列後可以發現樹狀圖特性造成每一個葉節點都成為下一個碼字的根節點。定義 S_{n-1} 為第 n 時間位元 b_n 的前一刻狀態， S_n 為後一刻狀態，則狀態 S_{n-1} 轉移至狀態 S_n 的路徑將由 $b_n \in \{0,1\}$ 決定。

以下列舉一組符合前置碼條件之可變長度碼為例，
 $C = \{c(0)=[00], c(1)=[11], c(2)=[011], c(3)=[100], c(4)=[0101], c(5)=[1010]\}$

圖 2.2 繪出該組碼字所對應之樹狀圖及籬柵圖。

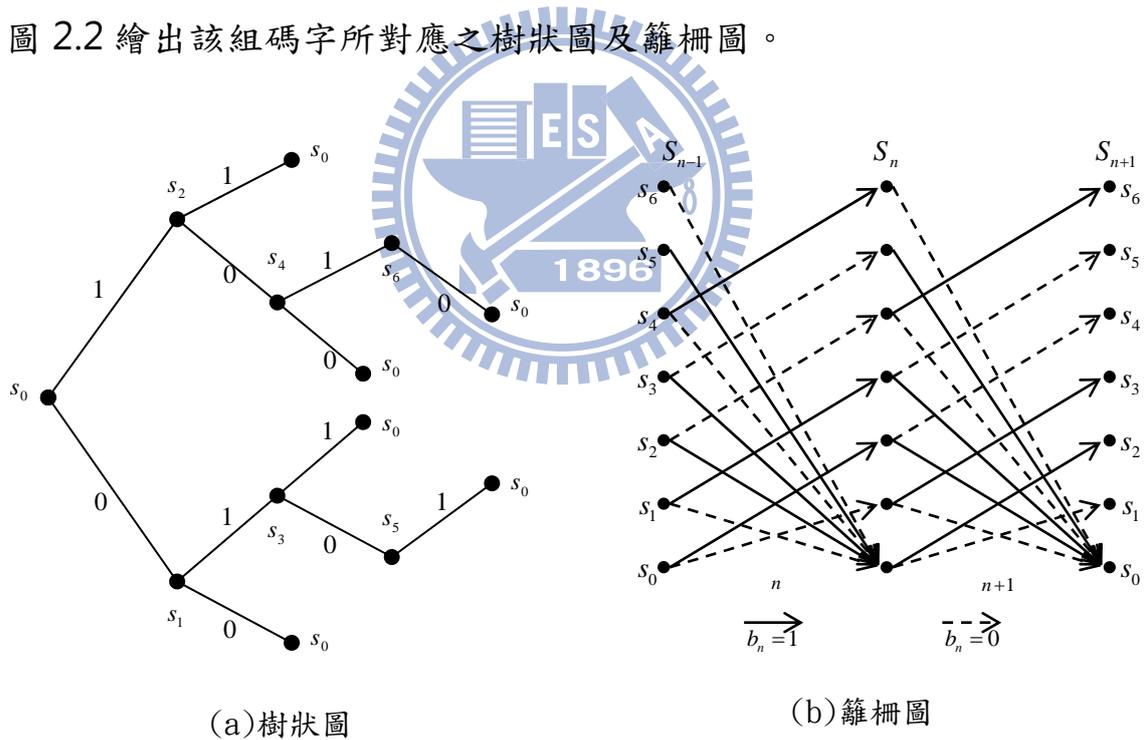


圖 2.2 可變長度碼之(a)樹狀圖與(b)籬柵圖

令樹狀圖所有的 N_s 個狀態集合為 $S = \{s_0, \dots, s_{N_s-1}\}$ ，接下來定義兩個集合 \mathcal{P}_σ 及 $\mathcal{P}_{\sigma_1, \sigma_2, i}$ ，以表現可變長度碼樹狀圖與籬柵圖之間的特殊關聯性。首先我們考慮時刻 n 時的狀態 $S_n = \sigma$ ， $\sigma \in S$ 會與某些碼字的特

定字首 $c_{[s_0, \dots, \sigma]}$ 有關，而此字首 $c_{[s_0, \dots, \sigma]}$ 所對應之狀態序列為 $[S_v, \dots, S_n] = [s_0, \dots, \sigma]$ ，其中依據籬柵圖可回溯此碼字根節點會落在時刻 v 處，即 $v = n - 1 - l(c_{[s_0, \dots, \sigma]})$ 。依此我們定義集合 $\mathcal{P}_\sigma = \{\lambda \in \mathcal{I} \mid c_{[s_0, \dots, \sigma]} \text{ is the prefix of } c(\lambda)\}$ ，即所有字首符合 $c_{[s_0, \dots, \sigma]}$ 的索引形成的集合。針對 $S_n = \sigma = s_0$ 而有 $c_{[s_0]}$ 為空集合的情形，我們定義 $\mathcal{P}_{s_0} = \mathcal{I}$ 。另外定義因輸入 $b_n = i$ 而使狀態自 $S_{n-1} = \sigma_1$ 轉移至 $S_n = \sigma_2$ 的集合為 $\mathcal{P}_{\sigma_1, \sigma_2, i} = \{\lambda \in \mathcal{I} \mid c_{[s_0, \dots, \sigma_1, \sigma_2]} = [c_{[s_0, \dots, \sigma_1]}, i] \text{ is the prefix of } c(\lambda)\}$ 其中 $\sigma_1, \sigma_2 \in S$ ， $b_n = i \in \{0, 1\}$ 。

舉圖 2.2 的可變長度碼為例，其依位元序列 b 而排列的籬柵圖如圖 2.3 所示。狀態總數 $N_s = 7$ ，故狀態集合 $S = \{s_0, \dots, s_6\}$ 。符合狀態序列由 s_0 逐步到 s_4 的字首為 $c_{[s_0, \dots, s_4]} = [10]$ 子集合 $\mathcal{P}_{s_2} = \{1, 3, 5\}$ 、符合因輸入位元 $b_n = 0$ 而使狀態由 s_2 轉移至 s_4 的子集合 $\mathcal{P}_{s_2, s_4, 0} = \{3, 5\}$ 。

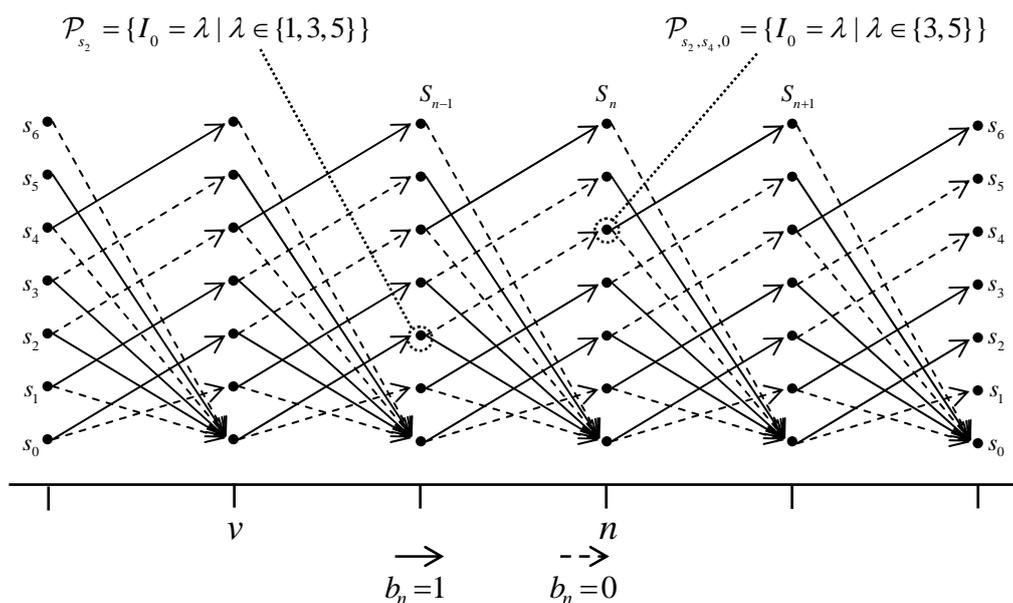


圖 2.3 碼串狀態轉移與籬柵關係的示意圖

2.3 位元後驗機率推導

本節將推導軟性輸入及軟性輸出(soft-input soft-output, SISO)的位元層級後驗機率 $P(b_n = i | \hat{\underline{b}})$ 。利用可變長度碼的籬柵圖，此後驗機率可表示成因輸入為 $b_n = i$ 而使狀態自 $S_{n-1} = \sigma_1$ 轉移至 $S_n = \sigma_2$ 的邊界機率，其計算公式為

$$P(b_n = i | \hat{\underline{b}}) = \sum_{\sigma_1 \in \mathcal{S}} \sum_{\sigma_2 \in \mathcal{S}} P(S_{n-1} = \sigma_1, b_n = i, S_n = \sigma_2 | \hat{\underline{b}}) \quad (2.1)$$

根據 BCJR 演算法[11]，我們可以利用機率連鎖律(Chain rule)將(2.1)式之狀態轉移機率分成三項：

$$P(S_{n-1} = \sigma_1, b_n = i, S_n = \sigma_2 | \hat{\underline{b}}) = \frac{1}{p(\hat{\underline{b}})} \cdot \underbrace{p(\hat{\underline{b}}_{n+1}^N | S_n = \sigma_2, \hat{\underline{b}}_1^n)}_{\beta_n(\sigma_2)} \cdot \underbrace{p(\hat{\underline{b}}_n, b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1, \hat{\underline{b}}_1^{n-1})}_{\gamma_n(i, \sigma_2, \sigma_1)} \cdot \underbrace{p(S_{n-1} = \sigma_1, \hat{\underline{b}}_1^{n-1})}_{\alpha_{n-1}(\sigma_1)} \quad (2.2)$$

其中 $\hat{\underline{b}}_x^y = [\hat{b}_x, \hat{b}_{x+1}, \dots, \hat{b}_y]$ 為由位置 x 至 y 受到通道干擾的資料串。 α 、 β 項分別為前向(forward)及後向(backward)機率。

2.3.1 前向機率遞迴計算

根據前人研究[11]，(2.2)式中的前向機率 α 值可由遞迴計算而得：

$$\alpha_n(\sigma_2) = \sum_{\sigma_1 \in \mathcal{S}} \sum_{i=0}^1 \gamma_n(i, \sigma_2, \sigma_1) \alpha_{n-1}(\sigma_1) \quad (2.3)$$

其中 $\gamma_n(i, \sigma_2, \sigma_1)$ 可利用機率的鏈鎖律推導如下：

$$\gamma_n(i, \sigma_2, \sigma_1) = p(\hat{b}_n | b_n = i) \cdot P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1, \hat{b}_1^{n-1}) \quad (2.4)$$

(2.4)式中等號右邊的第一項為通道資訊(Channel information)。第二項說明了在已知 \hat{b}_1^{n-1} 且前一刻狀態為 $S_{n-1} = \sigma_1$ 的情形下，狀態因輸入 $b_n = i$ 而轉移至 $S_n = \sigma_2$ 的機率。該轉移機率可利用子集合 \mathcal{P}_{σ_1} 和 $\mathcal{P}_{\sigma_1, \sigma_2, i}$ 所涵蓋的索引機率表示。這些子集合定義了共通起始於根節點 $S_v = s_0$ 的字首 $c_{[s_0, \dots, \sigma_1]}$ ，其中 $v = n-1 - l(c_{[s_0, \dots, \sigma_1]})$ 即當前根節點的位置。

若目前考慮的索引值為 I_0 ，該項即可改寫成：

$$P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1, \hat{b}_1^{n-1}) = P(I_0 \in \mathcal{P}_{\sigma_1, \sigma_2, i} | I_0 \in \mathcal{P}_{\sigma_1}, S_v = s_0, \hat{b}_1^{n-1}) \quad (2.5)$$

在(2.5)式中，籬柵狀態由 $S_{n-1} = \sigma_1$ 轉移至 $S_n = \sigma_2$ 表示了一組上升(ascending)型籬柵路徑，即 $\mathcal{P}_{\sigma_1, \sigma_2, i} \subset \mathcal{P}_{\sigma_1}$ 。利用此特性，我們可推導：

$$P(I_0 \in \mathcal{P}_{\sigma_1, \sigma_2, i} | I_0 \in \mathcal{P}_{\sigma_1}, S_v = s_0, \hat{b}_1^{n-1}) = \frac{P(I_0 \in \mathcal{P}_{\sigma_1, \sigma_2, i} | S_v = s_0, \hat{b}_1^{n-1})}{P(I_0 \in \mathcal{P}_{\sigma_1} | S_v = s_0, \hat{b}_1^{n-1})} \quad (2.6)$$

在(2.6)式中的條件機率在 \hat{b}_1^{n-1} 已知的情形下，可利用起始自 $S_v = s_0$ 的碼字之邊界機率表示。我們發現該式分子與分母僅不同於 I_0 所屬的子集合，以下以 \mathcal{P} 表示其任一子集合：

$$P(I_0 \in \mathcal{P} | S_v = s_0, \hat{b}_1^{n-1}) = \sum_{\lambda \in \mathcal{P}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^{n-1}) \quad (2.7)$$

其中的 $P(I_0 = \lambda | S_v = s_0, \hat{b}_1^{n-1})$ 可基於位元位置 v 再分解成：

$$P(I_0 = \lambda | S_v = s_0, \hat{b}_1^{n-1}) = \frac{1}{C} \cdot p(\hat{b}_{v+1}^{n-1} | c_{[s_0, \dots, \sigma_1]}) \cdot P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v) \quad (2.8)$$

(2.8)式右邊的第一項即為基於字首 $c_{[s_0, \dots, \sigma_1]}$ 計算 \hat{b}_{v+1}^{n-1} 的通道機率，第二項則為已知 \hat{b}_1^v 且 $S_v = s_0$ 的情形下， $c(I_0 = \lambda)$ 的條件機率。

將(2.6)至(2.8)式代入(2.5)式，將分子與分母共同的因式消除後，我們得到(2.4)式的轉移機率項：

$$P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1, \hat{b}_1^{n-1}) = \frac{1}{C(\sigma_1)} \sum_{\lambda \in \mathcal{P}_{\sigma_1, \sigma_2, i}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v) \quad (2.9)$$

其中 $C(\sigma_1) = \sum_{\lambda \in \mathcal{P}_{\sigma_1}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v)$ 為正規化常數 (normalization factor)。



2.3.2 後向機率遞迴計算

為了推導後項機率 β 值，我們再一次使用(2.5)式的推導模式，將籬柵狀態 $S_n = \sigma_2$ 以子集合 \mathcal{P}_{σ_2} 所包括的擁有相同字首 $c_{[s_0, \dots, \sigma_2]}$ 的索引值代替，即可得到：

$$\begin{aligned} \beta_n(\sigma_2) &= p(\hat{b}_{n+1}^N | S_n = \sigma_2, \hat{b}_1^n) \\ &= p(\hat{b}_{n+1}^N | I_0 \in \mathcal{P}_{\sigma_2}, S_v = s_0, \hat{b}_1^n) \end{aligned} \quad (2.10)$$

其中 $v = n-1-l(c_{[s_0, \dots, \sigma_2]})$ 為此時刻 n 所往前回朔的根節點 $S_v = s_0$ 位置。

利用鏈鎖律及邊界機率原理，假設現正考慮的索引為 $I_0 = \mu$ ，(2.10)

式可再被推導：

$$\beta_n(\sigma_2) = \frac{1}{C'(\sigma_2)} \sum_{\mu \in \mathcal{P}_{\sigma_2}} p(\hat{b}_{n+1}^N | I_0 = \mu, S_v = s_0) \cdot P(I_0 = \mu | S_v = s_0, \hat{b}_1^n) \quad (2.11)$$

$$\begin{aligned} \text{其中正規化常數 } C'(\sigma_2) &= P(I_0 \in \mathcal{P}_{\sigma_2} | S_v = s_0, \hat{b}_1^n) \\ &= \sum_{\lambda \in \mathcal{P}_{\sigma_2}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^n) \end{aligned}$$

注意(2.11)式第一項條件為 $I_0 = \mu, S_v = s_0$ ，故不再依賴 \hat{b}_1^n 。定義 $v' = v + l(c(\mu))$ 表示目前考慮的索引 $I_0 = \mu$ 其碼字 $c(\mu)$ 結束的位元位置、 $c_{[\sigma_2, \dots, s_0]}(\mu)$ 為碼字 $c(\mu)$ 由狀態 σ_2 逐步轉移至 s_0 之字尾。則針對 $\mu \in \mathcal{P}_{\sigma_2}$ ，(2.11)式中的 $p(\hat{b}_{n+1}^N | I_0 = \mu, S_v = s_0)$ 可寫成：

$$\begin{aligned} p(\hat{b}_{n+1}^N | I_0 = \mu, S_v = s_0) &= p(\hat{b}_{v'+1}^N | I_0 = \mu, S_v = s_0) \\ &\quad \cdot p(\hat{b}_{n+1}^{v'} | c_{[\sigma_2, \dots, s_0]}(\mu)) \end{aligned} \quad (2.12)$$

其中通道資訊 $p(\hat{b}_{n+1}^{v'} | c_{[\sigma_2, \dots, s_0]}(\mu))$ 包含了字尾 $c_{[\sigma_2, \dots, s_0]}(\mu)$ 在位元位置 $n+1$ 到 v' 通道輸出的資訊。基於通道無記憶特性，我們得到：

$$p(\hat{b}_{n+1}^{v'} | c_{[\sigma_2, \dots, s_0]}(\mu)) = \prod_{\eta=1}^{l(c_{[\sigma_2, \dots, s_0]}(\mu))} p(\hat{b}_{n+\eta} | c_\eta(\mu)) \quad (2.13)$$

其中 $c_\eta(\mu)$ 代表字尾 $c_{[\sigma_2, \dots, s_0]}(\mu)$ 的第 η 個位元，而 $p(\hat{b}_{n+\eta} | c_\eta(\mu))$ 則為該位元的通道資訊。將(2.11)至(2.13)式統合後得到

$$\begin{aligned} \beta_n(\sigma_2) &= \frac{1}{C''(\sigma_2)} \sum_{\mu \in \mathcal{P}_{\sigma_2}} \underbrace{p(\hat{b}_{v'+1}^N | I_0 = \mu, S_v = s_0)}_{=: B(v' | \mu, v)} \\ &\quad \cdot p(\hat{b}_{n+1}^{v'} | c_{[\sigma_2, \dots, s_0]}(\mu)) \cdot P(I_0 = \mu | S_v = s_0, \hat{b}_1^n) \end{aligned} \quad (2.14)$$

其中 $C''(\sigma_2) = P(I_0 \in \mathcal{P}_{\sigma_2} | S_v = s_0, \hat{b}_1^v)$ ，可由計算 $P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v)$ 之邊界機率而得。至於(2.14)式中的 $p(\hat{b}_{v'+1}^N | I_0 = \mu, S_v = s_0)$ 項，是已知 $I_0 = \mu$ 且 $S_v = s_0$ 條件下， $\hat{b}_{v'+1}^N$ 的發生機率，可視為索引 $I_0 = \mu$ 對應的碼字 $c(\mu)$ 結束時所在根節點的後向機率 β 值。

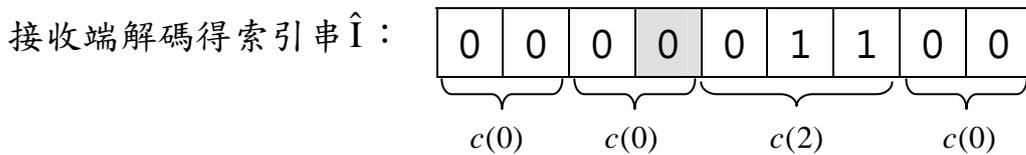
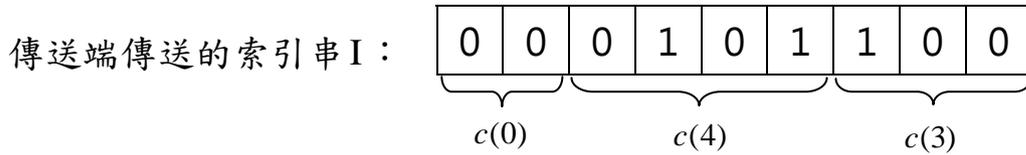
2.4 索引錯誤率之計算

基於上述方式執行 BCJR 演算法，我們可以計算已知長度的封包每個位元的後驗機率 $P(b_n = i | \hat{b})$ 。最後，將後驗機率作為軟性輸入值 (soft-input value) 輸入至估算機制，並根據最大後驗機率 (Maximum a posteriori estimation, MAP) 演算法估算封包的索引值 \hat{I} 。所謂的最大後驗機率演算法是利用 $\underline{b}_{MAP} = \arg \left\{ \max_{\underline{b}} \prod_{n=1}^N P(b_n | \underline{b}) \right\}$ ，在位元層級企圖找到一組字串 \underline{b}_{MAP} 使其整體累積的後驗機率為最大。為了找出 \underline{b}_{MAP} ，傳統作法是採用軟性輸出維特比演算法 (soft-output Viterbi algorithm)，找出一組後驗機率最大的合法碼字串。其權值 (metric) 計算方式為

$$M_n(\sigma_2) = \max \left\{ P(b_n = i | \hat{b}) \cdot P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1) \cdot M_{n-1}(\sigma_1) \right\} \quad (2.15)$$

若籬柵圖中狀態 $S_{n-1} = \sigma_1$ 與 $S_n = \sigma_2$ 可由 $b_n = i$ 連結起來，則 $P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1) = 1$ ，反之則 $P(b_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1) = 0$ 。

利用上述方式，我們可以得到最大後驗機率的合法碼字串。但在實踐上卻遇到嚴重的同步問題，以下使用圖 2.2 的一組可變長度碼為例：



傳送端傳送的内容為三個索引值 $\mathbf{I}=\{0,4,3\}$ 、相符的二位元碼串為 $\underline{b}=\{000101100\}$ 。通過通道雜訊干擾後，接收端解碼時將第 4 位置的位元誤判為 0，而估算出二位元碼串為 $\hat{\underline{b}}=\{000001100\}$ ，依此解出的合法索引值 $\hat{\mathbf{I}}=\{0,0,2,0\}$ 。

由上述例子可以明顯的發現，位元解碼若出現錯誤，即使很少，亦可能造成嚴重的索引誤判。不但如此，例子中的 \mathbf{I} 與 $\hat{\mathbf{I}}$ 的索引數目不同，無法計算其索引錯誤率(symbol error rate, SER)。

為了解決上述問題，接收端需要知道一個傳送封包含有幾個索引值。利用封包的索引值數目 K 以及封包位元長度 N ，將符合此條件的所有可能索引值組合列舉出來，再由這些組合中找出後驗機率最大的一個組合作為索引解，即 $\underline{I}_{MAP} = \arg \left\{ \max_{\underline{I}} \prod_{k=1}^K P(I_k | \hat{\underline{b}}) \right\}$ ，且 $P(I_k = \lambda | \hat{\underline{b}}) = \prod_{b_n \in \lambda} P(b_n | \hat{\underline{b}})$ 。以下舉一個簡單的例子：

一組可變長度碼為 $C_2 = \{c(0)=[0], c(1)=[10], c(2)=[01], c(3)=[111]\}$ ，其對應的碼長分別為 $l(c(0))=1$ 、 $l(c(1))=l(c(2))=2$ 、 $l(c(3))=3$ 。假設封包的索引值數目 $K=5$ ，封包位元長度 $N=10$ ，即可將所有可能的索引組合如圖 2.4 繪出。

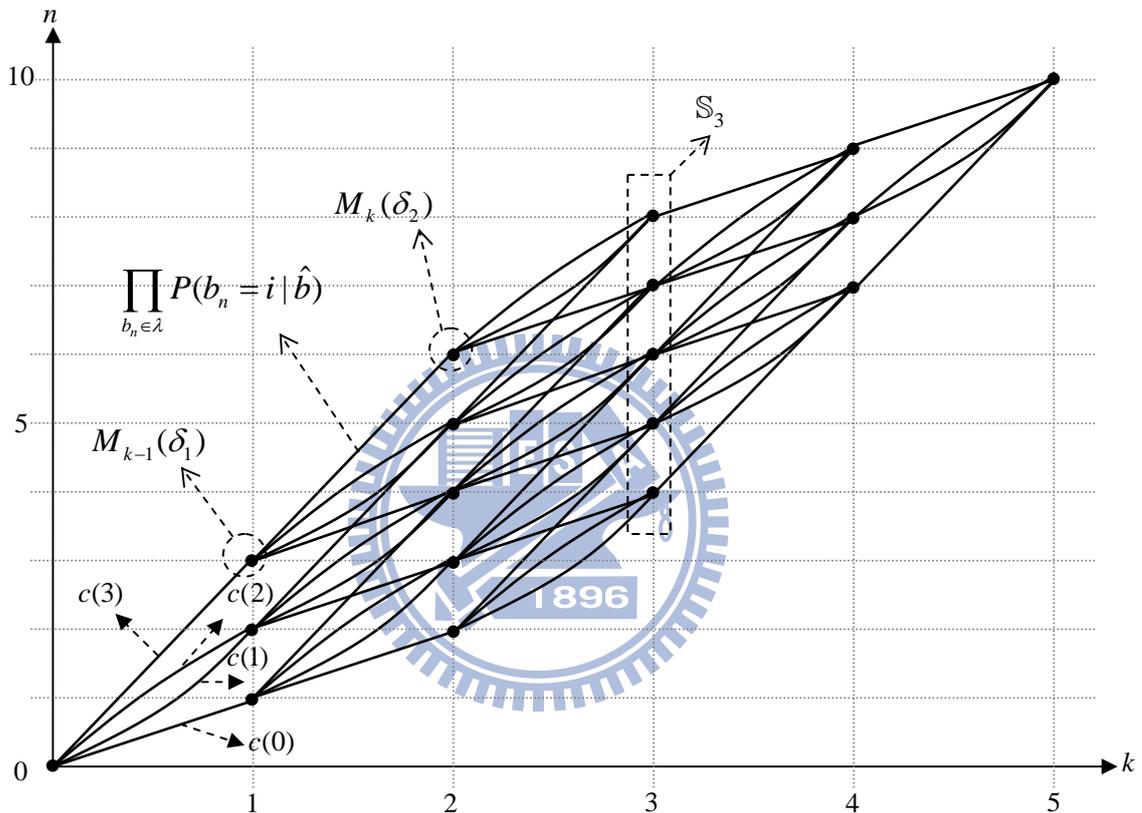


圖 2.4 已知 (N, K) 的索引路徑圖

圖 2.4 中橫軸 k 為索引數目，縱軸 n 為位元數目。利用此路徑圖在索引層級執行軟性輸出維特比演算法，即能在封包之索引數及封包位元長度的限制條件下，找出一組後驗機率最大的索引組合。有鑑於此我們修改(2.15)式如下：

$$M_k(\delta_2) = \max \left\{ \prod_{b_n \in \lambda} \left(P(b_n = i | \hat{b}) \right) \cdot P(I_k = \lambda, S_k = \delta_2 | S_{k-1} = \delta_1) \cdot M_{k-1}(\delta_1) \right\} \quad (2.16)$$

其中 S_k 為包含第 k 個索引值後所有位元長度的可能性， $\delta_2 \in S_k$ 、 $\delta_1 \in S_{k-1}$ 。例如圖 2.4 的 $S_3 = \{4, 5, 6, 7, 8\}$ ，即所有符合封包索引值數目 $K=5$ ，封包位元長度 $N=10$ 的可能性中，在包含第 3 個索引值後所有可能的位元長度。圖中若狀態 $S_{k-1} = \delta_1$ 與 $S_k = \delta_2$ 可由 $I_k = \lambda$ 連結起來，則 (2.16) 式中 $P(I_k = \lambda, S_k = \delta_2 | S_{k-1} = \delta_1) = 1$ ，反之則 $P(I_k = \lambda, S_k = \delta_2 | S_{k-1} = \delta_1) = 0$ 。

利用上述方式，我們可以從圖 2.4 的路徑圖，計算出符合封包索引值數目 K 、且封包位元長度為 N 之後驗機率最大的索引組合。



第三章 可變長度碼之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(interleaver)所組成，藉由各別解碼器輸出的額外訊息(extrinsic information)交互疊代來加強位元判斷的可靠度。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(Convolutional code)的錯誤檢查能力與訊源殘餘冗息。

在本章我們要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，故編碼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。此外，在上一章中我們了解到位元錯誤與位元同步對可變長度訊源解碼的影響是複雜且嚴重的。在通訊系統中，通道編碼技術雖可減少位元傳輸的錯誤，然而發展整合的疊代解碼機制將可以有效控制位元錯誤對於訊源解碼的影響程度。本章將先介紹疊代訊源通道編碼及解碼機制的整體流程，然後再逐一說明其額外資訊推導。

3.1 系統架構與流程

考慮傳輸端架構如圖 3.1 所示。傳輸資料封包由 K 個具關聯性的符號所組成 $\underline{U}=[U_1, U_2, \dots, U_K]$ 。封包經過量化後得到索引序列 $\underline{I}=[I_1, I_2, \dots, I_K]$ 。該封包索引值利用一前置碼 C 執行可變長度編碼後得到碼長為 N 的二位元字串 $\underline{b}=[c(I_1), c(I_2), \dots, c(I_K)]=[b_1, b_2, \dots, b_N]$ 。將此字串使用交錯器(interleaver)重新排序後，以 $\underline{b}'=[b'_1, b'_2, \dots, b'_N]$ 表示。 \underline{b}' 將被視為訊息位元(information bit)串，並代入一編碼率 $1/2$ 的通道編碼器(channel encoder)，其輸出為 $\underline{w}'=[w'_1, w'_2, \dots, w'_{\tilde{N}}]$ 。其中每一個 w'_n , $n \in \{1, \dots, \tilde{N}\}$ 都是由一個訊息位元及一個檢查位元(parity check bit)組成，即 $w'_n = \{w'_n{}^s, w'_n{}^p\}$ ，其中 $w'_n{}^s$ 為訊息位元， $w'_n{}^p$ 為檢查位元。由於通道編碼器我們採用系統碼(systematic code)，所以 $w'_n{}^s = b'_n$ 也就是 $w'_n = \{w'_n{}^s, w'_n{}^p\} = \{b'_n, w'_n{}^p\}$ 。注意在封包經過通道編碼後， \underline{w}' 的長度為 \tilde{N} 而非 N ，這是因為此長度除了訊息位元長度之外，還包括了終止位元(termination bit)的長度。所有的位元都經過二位元鍵移調變(Binary Phase Shift Keying, BPSK modulation)，再經由附加性白色高斯雜訊(AWGN)通道傳送。

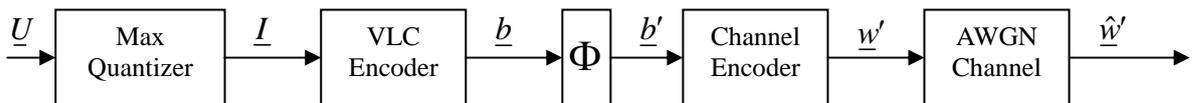


圖 3.1 系統傳輸端架構圖

定義經過通道雜訊干擾的封包為 $\hat{w}' = [\hat{w}'_1, \hat{w}'_2, \dots, \hat{w}'_N]$ ，其中第 n 位置的 $\hat{w}'_n = \{\hat{w}'_n{}^s, \hat{w}'_n{}^p\} = \{\hat{b}'_n, \hat{w}'_n{}^p\}$ ，它提供通道相關的資訊以協助解碼處理。假設通道為附加性白高斯(AWGN)的無記憶性通道，封包第 n 位置的接收資訊 $P(\hat{w}'_n | w'_n)$ 可以視為訊息位元通道資訊 $p(\hat{w}'_n{}^s | w'_n{}^s) = p(\hat{b}'_n | b'_n)$ 及檢查位元通道資訊 $p(\hat{w}'_n{}^p | w'_n{}^p)$ 機率的乘積。在通道訊雜比 $\frac{E_b}{N_0}$ 已知的情形下且雜訊之變異數為 $\sigma_{noise} = N_0/2$ ，我們可以分別計算：

$$p(\hat{w}'_n{}^s | w'_n{}^s) = p(\hat{b}'_n | b'_n) = \frac{1}{\sqrt{2\pi\sigma_{noise}}} \cdot \exp\left[-\frac{E_b}{N_0} (\hat{w}'_n{}^s - b'_n)^2\right] \quad (3.1)$$

$$p(\hat{w}'_n{}^p | w'_n{}^p) = \frac{1}{\sqrt{2\pi\sigma_{noise}}} \cdot \exp\left[-\frac{E_b}{N_0} (\hat{w}'_n{}^p - w'_n{}^p)^2\right] \quad (3.2)$$

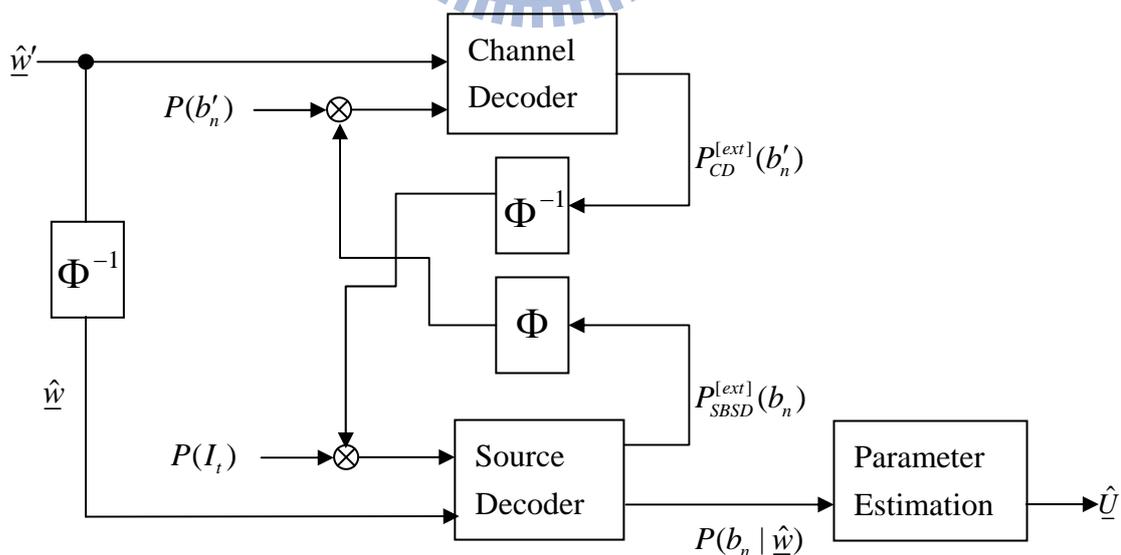


圖 3.2 系統接收端架構圖

接收端架構及其解碼流程如圖 3.2 所示。疊代解碼步驟如下所述：

1. 先設定疊代次數為 $iteration=0$ ，而所有位元的訊源解碼額外訊息初始機率值為 $P_{SBSD}^{[ext]}(b'_n)=1$ 。
2. 在通道解碼模組部分，受通道干擾的封包 $\hat{w}'=[\hat{w}'_1, \hat{w}'_2, \dots, \hat{w}'_N]$ 進入接收端，它提供了通道訊息 $p(\hat{b}'_n | b'_n)$ 與 $p(\hat{w}'_n^P | w_n^P)$ 。針對每一個位元，將事先統計所得的位元機率 $P(b'_n)$ 及訊源解碼的額外訊息 $P_{SBSD}^{[ext]}(b'_n)$ 視為通道解碼器之事前訊息。整合通道訊息與事前訊息之後，通道解碼演算法計算求出通道解碼額外訊息 $P_{CD}^{[ext]}(b'_n)$ 。
 $P_{CD}^{[ext]}(b'_n)$ 經過反交錯器 (de-interleaver) 後，得到重整順序的 $P_{CD}^{[ext]}(b_n)$ 再提供給訊源解碼器使用。
3. 在訊源解碼模組部分，結合索引機率 $P(I_i)$ 與位元層級的通道解碼額外訊息 $P_{CD}^{[ext]}(b_n)$ 求得訊源解碼所需之事前訊息，再考慮反交錯處理後封包 $\hat{w}=[\hat{w}_1, \hat{w}_2, \dots, \hat{w}_N]$ 提供的位元通道資訊，最後計算出每一個位元的訊源解碼額外訊息 $P_{SBSD}^{[ext]}(b_n)$ 及其後驗機率 $P(b'_n=i | \hat{b})$ 。
4. 將 $P_{SBSD}^{[ext]}(b_n)$ 回傳到通道解碼器再進行下一次的疊代計算，此時將疊代次數 $iteration \leftarrow iteration+1$ ，並將 $P_{SBSD}^{[ext]}(b_n)$ 經過交錯器成為 $P_{SBSD}^{[ext]}(b'_n)$ ，再進入步驟 2. 繼續執行即可。若已完成預設的疊代次數，即可利用位元後驗機率依最後後驗機率原則 (MAP) 估算解碼值。

3.2 通道解碼演算法

針對迴旋碼(convolutional code)，使用 BCJR 演算法可實現軟性輸入及軟性輸出的通道解碼模組。為了使訊息位元更具可靠性，它觀察整體接收的封包 \hat{w}' ，並利用迴旋碼的籬柵結構與遞迴計算公式以大幅降低其計算量及複雜度。在此，我們將先介紹如何運用 BCJR 演算法，計算接收封包經過通道解碼後的位元後驗機率，並於下個章節中說明額外訊息之計算。首先，考慮第 n 個位元的後驗機率：

$$\begin{aligned} P(b'_n = i | \hat{w}') &= \sum_{\sigma_1 \in S} \sum_{\sigma_2 \in S} P(S_{n-1} = \sigma_1, b'_n = i, S_n = \sigma_2 | \hat{w}') \\ &= C \cdot \sum_{\sigma_1 \in S} \sum_{\sigma_2 \in S} P(S_{n-1} = \sigma_1, b'_n = i, S_n = \sigma_2, \hat{w}') \end{aligned} \quad (3.3)$$

其中 $C = \frac{1}{p(\hat{w}')}$ 為與他項獨立之常數， S 為籬柵圖中所有狀態之集合，

$S_{n-1} = \sigma_1$ 及 $S_n = \sigma_2$ 分別為第 n 個位元的前一刻與後一刻的籬柵狀態。

假設通道無記憶性，且訊息位元 b'_n 互相獨立，則(3.3)式中的結合機率

(joint probability) $P(S_{n-1} = \sigma_1, b'_n = i, S_n = \sigma_2, \hat{w}')$ 可寫成：

$$\begin{aligned} P(S_{n-1} = \sigma_1, b'_n = i, S_n = \sigma_2, \hat{w}') &= \underbrace{p(\hat{w}'_{n+1} | S_n = \sigma_2)}_{\beta_n(\sigma_2)} \\ &\cdot \underbrace{p(\hat{w}'_n, b'_n = i, S_n = \sigma_2 | S_{n-1} = \sigma_1)}_{\gamma_n(i, \sigma_2, \sigma_1)} \\ &\cdot \underbrace{p(S_{n-1} = \sigma_1, \hat{w}'_1^{n-1})}_{\alpha_{n-1}(\sigma_1)} \end{aligned} \quad (3.4)$$

其中前向遞迴 α 及後向遞迴 β 分別為：

$$\alpha_n(\sigma_2) = \sum_{\sigma_1 \in S} \sum_{i=0}^1 \gamma_n(i, \sigma_2, \sigma_1) \alpha_{n-1}(\sigma_1) \quad (3.5)$$

$$\beta_{n-1}(\sigma_1) = \sum_{\sigma_2 \in S} \sum_{i=0}^1 \gamma_n(i, \sigma_2, \sigma_1) \beta_n(\sigma_2) \quad (3.6)$$

迴旋碼起始自全零的籬柵狀態 s_0 ，且強制在解碼結束時回到全零狀態，所以我們設定 α 及 β 的初始值：

$$\alpha_0(S_0) = \begin{cases} 1, & \text{if } S_0 = s_0 \\ 0, & \text{if } S_0 \neq s_0 \end{cases} \quad (3.7)$$

$$\beta_{\tilde{N}}(S_{\tilde{N}}) = \begin{cases} 1, & \text{if } S_{\tilde{N}} = s_0 \\ 0, & \text{if } S_{\tilde{N}} \neq s_0 \end{cases} \quad (3.8)$$

另外，利用機率之連鎖律及基於通道無記憶性及 $w_n'^s = b_n'$ 和 $\hat{w}_n'^s = \hat{b}_n'$ 之關係，(3.4)式的 $\gamma_n(i, \sigma_2, \sigma_1)$ 項可推導為：

$$\begin{aligned} \gamma_n(i, \sigma_2, \sigma_1) &= p(\hat{w}_n', b_n' = i, S_n = \sigma_2 | S_{n-1} = \sigma_1) \\ &= P(b_n' = i, S_n = \sigma_2 | S_{n-1} = \sigma_1) \cdot p(\hat{w}_n' | b_n' = i, S_n = \sigma_2, S_{n-1} = \sigma_1) \\ &= P(b_n' = i) \cdot p(\hat{w}_n' | w_n') \\ &= P(b_n' = i) \cdot p(\hat{b}_n' | b_n' = i) \cdot p(\hat{w}_n'^p | w_n'^p) \end{aligned} \quad (3.9)$$

其中 $P(b_n' = i)$ 為位元事前機率， $p(\hat{b}_n' | b_n' = i)$ 為訊息位元通道訊息， $p(\hat{w}_n'^p | w_n'^p)$ 為檢查位元通道訊息。我們現在完成所有計算需要的元素，將(3.5)、(3.6)及(3.9)式代回(3.4)式，即可得到(3.3)式的後驗機率。接下來，我們將利用位元後驗機率，推導出位元額外訊息的資訊。

3.3 解碼輸出之額外訊息

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以及訊源解碼器各別輸出的額外訊息。

(A) 通道解碼額外訊息

3.2 節所計算的位元後驗機率，可據以推導出通道解碼額外資訊 $P_{CD}^{[ext]}(b'_n)$ ，再回傳到訊源解碼器的事前訊息輸入端。首先考慮(3.9)式的 $\gamma_n(i, \sigma_2, \sigma_1)$ 項：

$$\begin{aligned}\gamma_n(i, \sigma_2, \sigma_1) &= P(b'_n = i) \cdot p(\hat{b}'_n | b'_n = i) \cdot p(\hat{w}'_n{}^p | w'_n{}^p) \\ &\triangleq P(b'_n = i) \cdot p(\hat{b}'_n | b'_n = i) \cdot \gamma_n^{[ext]}(i, \sigma_2, \sigma_1)\end{aligned}\quad (3.10)$$

在此，我們定義了 $\gamma_n^{[ext]}(i, \sigma_2, \sigma_1) = p(\hat{w}'_n{}^p | w'_n{}^p)$ 。將(3.5)、(3.6)及(3.10)式代入(3.4)式，可得(3.3)式的後驗機率為

$$\begin{aligned}P(b'_n = i | \hat{w}') &= C \cdot \sum_{\sigma_1 \in S} \sum_{\sigma_2 \in S} \alpha_{n-1}(\sigma_1) \cdot \gamma_n(i, \sigma_2, \sigma_1) \cdot \beta_n(\sigma_2) \\ &= C \cdot P(b'_n = i) \cdot p(\hat{b}'_n | b'_n = i) \cdot \sum_{\sigma_1 \in S} \sum_{\sigma_2 \in S} \alpha_{n-1}(\sigma_1) \cdot \gamma_n^{[ext]}(i, \sigma_2, \sigma_1) \cdot \beta_n(\sigma_2) \\ &= C \cdot P(b'_n = i) \cdot p(\hat{b}'_n | b'_n = i) \cdot P_{CD}^{[ext]}(b'_n = i)\end{aligned}\quad (3.11)$$

其中通道解碼額外資訊為

$$P_{CD}^{[ext]}(b'_n) = \sum_{\sigma_1 \in S} \sum_{\sigma_2 \in S} \alpha_{n-1}(\sigma_1) \cdot \gamma_n^{[ext]}(i, \sigma_2, \sigma_1) \cdot \beta_n(\sigma_2)\quad (3.12)$$

在第一次疊代之後，一般作法是將 $P(b'_n = i)$ 與來自訊源解碼器的額外資訊 $P_{SBS}^{[ext]}(b'_n)$ 整合而成為事前機率。因此若依下式作替代：

$$P(b'_n = i) \leftarrow \left[P_{SBS D}^{[ext]}(b'_n = i) \cdot P(b'_n = i) \right] \quad (3.13)$$

則(3.11)式的後驗機率 $P(b'_n = i | \hat{w}')$ 可改寫為：

$$P(b'_n = i | \hat{w}') = C \cdot P(b'_n = i) \cdot P_{SBS D}^{[ext]}(b'_n = i) \cdot p(\hat{b}'_n | b'_n = i) \cdot P_{CD}^{[ext]}(b'_n = i) \quad (3.14)$$

其中 $P(b'_n = i) \cdot P_{SBS D}^{[ext]}(b'_n = i)$ 為事前訊息， $p(\hat{b}'_n | b'_n = i)$ 為通道訊息，而 $P_{CD}^{[ext]}(b'_n = i)$ 則為通道解碼器輸出的額外資訊。

(B) 訊源解碼額外訊息

第二章所介紹的可變長度碼解碼演算法，將在此疊代解碼架構中扮演軟性位元訊源解碼的角色。由圖 3.2 可以知道，訊源解碼器輸出的額外訊息可視為通道解碼器的事前訊息。回顧第二章中，可變長度碼繪出籬柵圖，且利用籬柵狀態推導 BCJR 演算法計算後驗機率 $P(b_n = i | \hat{b})$ 。我們將利用該籬柵圖上的路徑轉移機率 $\gamma_n(i, \sigma_2, \sigma_1)$ 及後驗機率 $P(b_n = i | \hat{b})$ ，進而推導出於疊代解碼中所需的訊源解碼額外訊息。將(2.9)式代入(2.4)式得到 $\gamma_n(i, \sigma_2, \sigma_1)$ 為：

$$\gamma_n(i, \sigma_2, \sigma_1) = \frac{1}{C(\sigma_1)} \cdot p(\hat{b}_n | b_n = i) \cdot \sum_{\lambda \in \mathcal{P}_{\sigma_1, \sigma_2, i}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v) \quad (3.15)$$

由(2.1)、(2.2)與(3.15)式可知位元後驗機率：

$$\begin{aligned}
P(b_n = i | \hat{b}) &= \sum_{\sigma_1 \in \mathcal{S}} \sum_{\sigma_2 \in \mathcal{S}} P(S_{n-1} = \sigma_1, b_n = i, S_n = \sigma_2 | \hat{b}) \\
&= C' \cdot \sum_{\sigma_1 \in \mathcal{S}} \sum_{\sigma_2 \in \mathcal{S}} \alpha_{n-1}(\sigma_1) \cdot \beta_n(\sigma_2) \cdot \gamma(i, \sigma_2, \sigma_1) \\
&= C' \cdot p(\hat{b}_n | b_n = i) \\
&\quad \cdot \sum_{\sigma_1 \in \mathcal{S}} \sum_{\sigma_2 \in \mathcal{S}} \frac{1}{C(\sigma_1)} \cdot \alpha_{n-1}(\sigma_1) \cdot \beta_n(\sigma_2) \cdot \sum_{\lambda \in \mathcal{P}_{\sigma_1, \sigma_2, i}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v) \\
&\triangleq C' \cdot p(\hat{b}_n | b_n = i) \cdot P_{SBSD}^{[ext]}(b_n = i)
\end{aligned} \tag{3.16}$$

其中 $C' = \frac{1}{p(\hat{b})}$ 為一常數。並定義訊源解碼額外訊息為：

$$P_{SBSD}^{[ext]}(b_n = i) = \sum_{\sigma_1 \in \mathcal{S}} \sum_{\sigma_2 \in \mathcal{S}} \frac{1}{C(\sigma_1)} \cdot \alpha_{n-1}(\sigma_1) \cdot \beta_n(\sigma_2) \cdot \sum_{\lambda \in \mathcal{P}_{\sigma_1, \sigma_2, i}} P(I_0 = \lambda | S_v = s_0, \hat{b}_1^v) \tag{3.17}$$

一般在進行疊代時，我們會將 $p(\hat{b}_n | b_n = i)$ 與來自通道解碼器的額外資訊 $P_{CD}^{[ext]}(b_n)$ 整合成為事前機率，因此可得位元後驗機率如下：

$$P(b_n = i | \hat{b}) = C' \cdot P_{CD}^{[ext]}(b_n = i) \cdot p(\hat{b}_n | b_n = i) \cdot P_{SBSD}^{[ext]}(b_n = i) \tag{3.18}$$

其中 $P_{CD}^{[ext]}(b_n = i)$ 為事前訊息、 $p(\hat{b}_n | b_n = i)$ 為通道訊息，而 $P_{SBSD}^{[ext]}(b_n = i)$ 則為訊源解碼器輸出的額外資訊。

第四章 雙向性可變長度碼之最佳化設計

前面的章節討論了針對可變長度碼的軟性解碼演算法及其在疊代解碼上的應用。了解到針對具有特定機率分佈的訊號源使用的可變長度碼，其編碼碼字之間存在的殘餘冗息特性提昇了疊代解碼強健性。然而可變長度碼的編碼方式眾多，以霍夫曼碼為例，它符合了前置條件(Prefix condition)，這使得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。另一種被廣範使用的可變長度編碼方式為雙向性可變長度碼(Reversible variable length codes, RVLCs)，它不但符合前置條件，也符合後置條件(Suffix condition)。因此支援前後二方向的解碼，其效果加強了解碼的錯誤更正能力，使我們能盡可能的從收到的資料中解出與原始訊息位元相符的資訊，實驗也證實雙向性可變長度碼在疊代解碼上有很好的效能。

因此本章首先將介紹雙向性可變長度碼及其編碼方式。接著說明在軟性解碼架構中額外資訊轉換圖的分析方法，並用以探討疊代系統之效能。最後，將運用額外資訊轉換分析結果針對雙向性可變長度碼進行索引指定的調整。

4.1 雙向性可變長度碼

可變長度碼的每個碼字長度不同，基於不同的訴求，其編碼方式也不一樣，如何設計最佳的可變長度碼至今仍是開放的議題。如前所述，雙向性可變長度碼符合前置條件以及後置條件，也因此支援雙向性的解碼。所謂的前置與後置條件定義如下：

(1) 前置條件(Prefix Condition)：

任何一碼字不可以是其他較長碼字之字首。

(2) 後置條件(Suffix Condition)：

任何一碼字不可以是其他較長碼字之字尾。

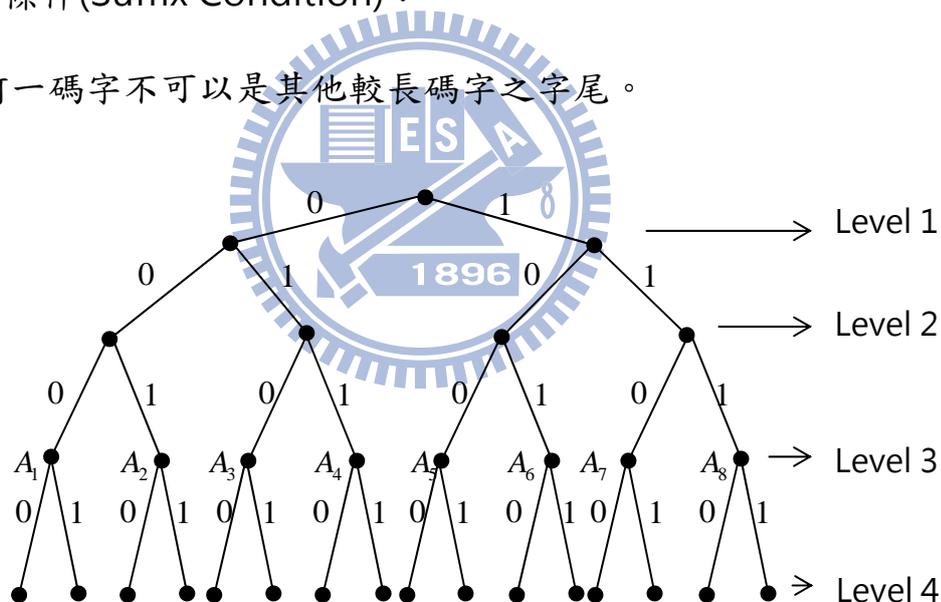


圖 4.1 四位元碼字樹狀圖

在可變長度編碼設計中，一般做法是利用位元樹狀圖來表示選取碼字的分佈情形。四位元樹狀圖如圖 4.1 所示，依位元長度分為第一層至第四層。其中以 A_1, A_2, \dots, A_8 代表第三層(即位元長度為 3)中八個可被選取的碼字，我們定義其為「候選碼字」，而一般碼字的選擇流

程都會是由上至下層依序進行選取。假設一個已知雙向性可變長度碼的碼書內，有 $[0,0]$ 這個第二層的碼字，為了符合前置條件，則候選碼字中的 A_1 及 A_2 則不可能出現在此碼書中；相同的，為了符合後置條件， A_5 亦不可能出現在此碼書中。也就是說，若我們挑選了 $[0,0]$ 作為編碼碼書裡的碼字，定義為「目標碼字」，也就同時使 A_1 、 A_2 、 A_5 三個碼字無效。此時第三層剩下的有效碼字為五個。而這樣經過前置與後置條件判斷後確定可以使用的碼字，我們定義為「可用候選碼字」。反之，若挑選了 $[1,0]$ 作為我們的目標碼字，為了符合前置條件，則候選碼字中的 A_5 及 A_6 則不可能出現在此碼書中，且為了符合後置條件， A_3 及 A_7 亦不可能出現。即 A_3 、 A_5 、 A_6 、 A_7 四個碼字將被排除出可用候選碼字的集合，則第三層剩下的可用候選碼字為 A_1 、 A_2 、 A_4 、 A_8 。由這兩個例子可以了解，在挑選較短的碼字時，將會受到前置條件與後置條件的影響，而使得下一層可用候選碼字數量不同。

如前述，基於不同的訴求，可變長度碼的編碼方式也不一樣。以下我們將基於最短平均長度以及錯誤更正能力兩個方向，分別介紹兩種不同的雙向性可變長度碼的編碼方式。

4.1.1 基於最小平均長度的設計

一個雙向性可變長度碼的設計為了使其平均長度最小，其編碼的碼字長度將會參考平均傳輸位元量可以近似理論熵值的霍夫曼碼。然而，為了同時符合前置與後置條件，雙向性可變長度碼勢必在碼字長度要求上需要適度的放寬。一般編碼方式將由最短長度的碼字開始挑選，而為了滿足平均長度最小，在編碼的演算法中，其關鍵是讓每一層都有最多的可用候選碼字能被選擇。因此當我們挑選某層的碼字時，將使該層候選碼字有其挑選的優先順序。首先將候選碼字中互相為位元互補的兩個碼字成為一組。依前置與後置條件計算每一個候選碼字的下一層可用候選碼字數量，並由大至小排列。若有相同的可用候選碼字數量，則以十進制值較小的碼字為優先。表 4.1 列出第三層的所有碼字，顯示每一個碼字若被挑選後，第四、五、六層所剩的可用候選碼字數量，並以該數量由大至小排列。

第三層碼字	各層可用候選碼字之數量		
候選碼字	第四層	第五層	第六層
000	13	24	44
111	13	24	44
010	12	21	37
101	12	21	37
001	12	20	33
110	12	20	33
011	12	20	33
100	12	20	33

表 4.1 第三層碼字之可用候選碼字表

在表 4.1 中，若我們編碼時在第三層需要的目標碼字為 3 個時，則前 2 個碼字首先選「000」及「111」，因為它們擁有第四層最多的可用候選碼字。在選擇「000」及「111」之後，剩下的所有第三層候選碼字擁有相同的第四層可用候選碼字數量。所以我們再參考第五層的可用候選碼字，而發現「010」在這些剩下的第三層候選碼字中擁有最多的可用後選碼字，所以我們挑選「010」作為我們的第三個編碼碼字。利用這樣的挑選方式將碼字一一編入雙向性可變長度碼的碼書。以下介紹編碼演算法：

1. 首先定義 $n(i)$ 為一組作為參考的可變長度碼(例如霍夫曼碼)之第 i 層的碼字數量。並定義 $n_{rev}(i)$ 為目標雙向性可變長度碼之第 i 層的碼字數量，且設定每一層的初始值 $n_{rev}(i) = n(i)$ 。
2. 針對第 i 層的每一個候選碼字，依前置與後置條件計算下一層可用候選碼字的數量，定義為 $avail(i)$ 。並依前頁所敘述的方式作排列。

此時會發生兩種可能：

(1) 若 $n_{rev}(i) \leq avail(i)$ ，即可用候選碼字足夠提供目標碼字的數量，則 $n_{rev}(i)$ 不變。

(2) 若 $n_{rev}(i) > avail(i)$ ，即可用候選碼字不足以提供目標碼字的數量時，則將不足的 $(n_{rev}(i) - avail(i))$ 數目碼字移至下一層挑選，即 $n_{rev}(i+1) = n_{rev}(i+1) + (n_{rev}(i) - avail(i))$ ，並將本層的碼字數量更

改為 $n_{rev}(i) = avail(i)$ 。

3. 依照排序將可用候選碼字編入目標碼書，接著再回到步驟 2 進行下一層的編碼。

這樣的演算法因為盡可能使下一層的可用候選碼字越多越好，也就是盡可能減少第二步驟中可用候選碼字不足以提供目標碼字的數量，而使得目標碼字必需由下一層的候選碼字挑選的情形發生。所以上述方式將能找到一組符合前置條件與後置條件且具最小平均長度的雙向性可變長度碼。

4.1.2 基於錯誤更正能力的設計

除了傳輸位元量為最小，我們也期望能使雙向性可變長度碼具有更好的錯誤更正能力。根據前人的研究[12]，一組可變長度碼的錯誤率與自由距離(free distance)有強烈的關聯性，即自由距離值越大，錯誤更正能力也就越強。

首先定義自由距離 d_f 以及區塊距離 d_b (block distance)。考慮 $\mathcal{I} = \{0, 1, \dots, 2^{M-1}\}$ 為量化的索引集合，並令 $\mathcal{C}(\mathcal{I}) = \{C(0), C(1), \dots, C(2^{M-1})\}$ 為索引集合 \mathcal{I} 所對應之編碼碼書，且其碼長分佈為 $l(\mathcal{I}) = \{l(0), l(1), \dots, l(2^{M-1})\}$ 。假設一個索引序列 $\underline{I} = \{I_1, I_2, \dots\}$ 在可變長度碼 $\mathcal{C}(\mathcal{I})$ 編碼後得到總長度為 N 的二位元字串

$\underline{b}=[c(I_1),c(I_2),\dots]=[b_1,b_2,\dots,b_N]$ ，並進行傳送。此外，定義集合 \mathcal{B} 為所有可能由可變長度碼 $\mathcal{C}(I)$ 作編碼後總長度為 N 的碼字組合。而 d_H 表示為漢明距離(Hamming distance)。則自由距離 d_f 及區塊距離 d_b 之定義為：

$$d_f = \min \left\{ d_H(\underline{b}^{(m)}, \underline{b}^{(n)}) \mid \underline{b}^{(m)}, \underline{b}^{(n)} \in \mathcal{B}; \underline{b}^{(m)} \neq \underline{b}^{(n)} \right\} \quad (4.1)$$

$$d_b = \min \left\{ d_H(\mathcal{C}(i), \mathcal{C}(j)) \mid \mathcal{C}(i), \mathcal{C}(j) \in \mathcal{C}; l(i) = l(j); i \neq j \right\} \quad (4.2)$$

針對霍夫曼碼而言，可由觀察發現有兩個碼字只不同於最後一個位元的情形，故其自由距離及區塊距離為 $d_f = d_b = 1$ 。

根據[17]證明，雙向性可變長度碼具有 $d_f \geq \min(2, d_b)$ 的特性。其中對於不同長度的碼字，基於前置與後置條件而使 $d_f \geq 2$ 恆成立。然而針對相同長度的碼字，經證明得到 $d_f \geq d_b$ 的結論。故只要能使區塊距離 d_b 至少為 2，則該雙向性可變長度碼就具有 $d_f \geq 2$ 的特性。我們將介紹以下的演算法，使目標雙向性可變長度碼書具有 $d_f \geq 2$ 的條件，讓錯誤更正能力提升：

1. 首先定義 $n(i)$ 為一組作為參考的可變長度碼(例如霍夫曼碼)之第 i 層的碼字數量。並定義 $n_{rev}(i)$ 為目標雙向性可變長度碼之第 i 層的碼字數量，且設定每一層的初始值 $n_{rev}(i) = n(i)$ 。
2. 針對第 i 層的每一個候選碼字，依前置與後置條件計算下一層可用候選碼字的數量，定義為 $avail(i)$ ，並依 4.1.1 節所介紹的方式作

排列。接下來依照順序將候選碼字一個個與已挑選完成並列入碼書的所有相同長度之碼字計算漢明距離 d_H ，若 $d_H \geq 2$ 則該候選碼字可被選為目標碼字而列入碼書。若 $d_H < 2$ 則不被列入，再取下一個候選碼字計算漢明距離 d_H 。而如此經過比對後，候選碼字中可被列入碼書的碼字數目定義為 $avail_{dis}(i)$ 。

3. 此時會發生兩種可能：

(1) 若 $n_{rev}(i) \leq avail_{dis}(i)$ ，即候選碼字中，符合漢明距離 $d_H \geq 2$ 的數量足以提供目標碼字作選擇，則 $n_{rev}(i)$ 不變。

(2) 若 $n_{rev}(i) < avail_{dis}(i)$ ，即符合漢明距離 $d_H \geq 2$ 的數量不足提供目標碼字作選擇，則將不足的碼字數量移至下一層挑選時，則將不足的 $(n_{rev}(i) - avail_{dis}(i))$ 數量碼字移至下一層挑選，即 $n_{rev}(i+1) = n_{rev}(i+1) + (n_{rev}(i) - avail_{dis}(i))$ ，並將本層的碼字數量更改為 $n_{rev}(i) = avail_{dis}(i)$ 。

4. 依照排序將符合漢明距離 $d_H \geq 2$ 的可用候選碼字列入目標碼書，接著再回到步驟 2 進行下一層的編碼。

利用上述的編碼方式，可以找出一組符合前置與後置條件，且自由距離 $d_f \geq 2$ 的雙向性可變長度碼，其錯誤更正能力將有顯著的功效。將其運用在疊代解碼系統中，預期亦可使疊代能發揮良好的解碼效能。

4.2 額外資訊轉換特性分析

在第三章所述的疊代解碼演算法，其原理是運用通道解碼器與訊源解碼器之間互相交換額外訊息來提升錯誤更正的能力。為了能藉由理論分析而事前了解系統疊代的效能，我們使用前人基於消息理論所推導的額外資訊轉換圖(extrinsic information transfer chart, EXIT chart)來進行疊代效益的分析。額外資訊轉換的模擬，是在計算單一解碼器的輸入事前訊息及輸出額外訊息兩者間消息量的變化關係。而透過額外資訊轉換圖，我們將不需要執行大量的系統模擬，即可分析此解碼器在疊代解碼時所提供的效能。而基於額外資訊轉換的分析，將提供我們在系統設計上效能的觀察，進一步促使我們發展系統關鍵元件的最佳化設計。在本節將針對額外資訊轉換圖做推導及介紹。

在先前章節所介紹的解碼演算法中，所有資訊皆以機率值表示。然而在此為了計算訊號消息量，所有的資訊將改以對數相似比率(Log-Likelihood Ratio)表示。首先考慮一個軟性輸入軟性輸出(soft-input/soft-output, SISO)解碼器，如圖 4.2 所示。令 y 為經二位元相位鍵移(BPSK)調變後所傳送的訊息， \hat{y} 為 y 通過高斯雜訊通道在接收端收到的實數值，且訊號 y 的事前機率為 $P(y)$ ，則位元事前訊息的對數相似比率表示為 $L(y) \triangleq \ln \frac{P(y=+1)}{P(y=-1)}$ 。而 $L(\hat{y}|y)$ 表示為通道訊息的對數相似比率， $L_{in}^{[ext]}(y)$ 為由另一個解碼器傳回的額外訊息，而

$L_{out}^{[ext]}(y)$ 則為此解碼器輸出的額外訊息。

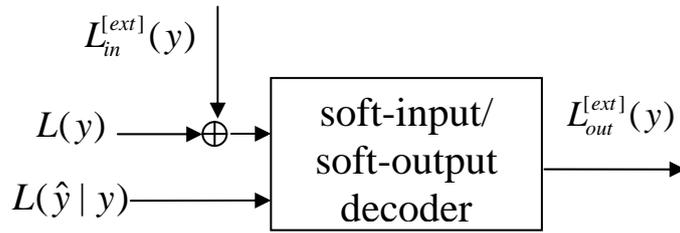


圖 4.2 軟性輸入軟性輸出解碼器

額外訊息轉換圖是利用兩個互消息所繪出的分析圖，分別為輸入事前訊息 $L^{[apri]}(y) = L(y) + L_{in}^{[ext]}(y)$ 與傳送訊息 y 之間的互消息，表示為事前互消息 $I^{[apri]}$ ，以及輸出額外訊息 $L_{out}^{[ext]}(y)$ 與傳送訊息 y 之間的互消息，記之為額外互消息 $I^{[ext]}$ 。這裡所謂任兩個隨機變數 X 和 Y 之間的互消息，定義為熵值(entropy) $H(X)$ 和 $H(X|Y)$ 的差值，即

$$I(X;Y) = H(X) - H(X|Y), 0 \leq I(X;Y) \leq H(X) \quad (4.3)$$

其意義為在事件 Y 已知的情形下， X 所減少的不確定性，即象徵當互消息 $I(X;Y)$ 值越大，兩個隨機變數 X 和 Y 的相關性就越大。以下將逐一說明如何計算事前互消息與額外互消息。

首先，單一解碼器的疊代過程可得到以下兩項假設：

- (1) 如果交錯器夠長，則在經過多次疊代後，輸入事前訊息 $L^{[apri]}(y)$ 與通道訊息 $L(\hat{y}|y)$ 在統計上可視為不相關。
- (2) 經過多次疊代後，單一解碼器的輸入端和輸出端所對應的額外對數相似比率 $L_{in}^{[ext]}(y)$ 和 $L_{out}^{[ext]}(y)$ 皆近似於高斯分佈。

基於上述假設可知，輸入事前訊息 $L_m^{[ext]}(y) = \xi$ 為一個獨立的高斯隨機變數，若令 n_A 為平均值 0，且變異數為 σ_A^2 之高斯隨機變數，則

$L_m^{[ext]}(y) = \mu_A \cdot y + n_A$ ，其平均值為隨機變數 $\mu_A \cdot y$ ， $y \in \{\pm 1\}$ ，且 $\mu_A = \frac{\sigma_A^2}{2}$ 。

其機率密度函數為 $p_A(\xi|y) = \frac{1}{\sqrt{2\pi\sigma_A^2}} e^{-\frac{(\xi - \mu_A \cdot y)^2}{2\sigma_A^2}}$ 。經由推導可知，高斯

分佈具有以下兩項特性： $p_A(\xi|y=-1) = p_A(-\xi|y=+1)$ 與 $p_A(\xi|y) = p_A(-\xi|y)e^{y \cdot \xi}$ ，則可推得

$$e^{-y \cdot \xi} = \frac{p_A(\xi|-y)}{p_A(\xi|y)} \quad (4.4)$$

藉由上述說明，在額外資訊轉移的模擬過程，將使用高斯隨機變數來表現不同程度的事前訊號，其控制變數為 σ_A 。而用二元隨機變數模擬訊號 y ，其中位元機率 $p(y=+1) = p(y=-1) = \frac{1}{2}$ ，則傳送訊號為 $y=+1$ 時，其事前互消息為

$$\begin{aligned} I(\xi; y=+1) &= E_\xi \left\{ \log_2 \frac{p_A(\xi|y=+1)}{p_A(\xi)} \right\} \\ &= E_\xi \left\{ \log_2 \frac{p_A(\xi|y=+1)}{\sum_{y=-1,+1} p_A(\xi|y) \cdot p(y)} \right\} \\ &= E_\xi \left\{ \log_2 \frac{p_A(\xi|y=+1)}{p_A(\xi|y=+1) + p_A(\xi|y=-1)} \right\} \\ &= 1 - E_\xi \left\{ \log_2 \left(1 + \frac{p_A(\xi|y=-1)}{p_A(\xi|y=+1)} \right) \right\} \\ &= 1 - E_\xi \left\{ \log_2 (1 + e^{-\xi}) \right\} \end{aligned} \quad (4.5)$$

而當訊息位元 $y = -1$ 時，其互消息為

$$I(\xi; y = +1) = 1 - E_{\xi} \left\{ \log_2(1 + e^{\xi}) \right\} \quad (4.6)$$

統合(4.5)與(4.6)式，可得事前互消息在模擬過程中的近似式

$$\begin{aligned} I^{[apri]} = I(\xi; y) &= 1 - E_{\xi} \left\{ \log_2(1 + e^{-y \cdot \xi}) \right\} \\ &\approx 1 - \frac{1}{N} \sum_{n=1}^N \log_2(1 + e^{-y_n \cdot \xi_n}) \quad , 0 \leq I(\xi; y) \leq 1 \end{aligned} \quad (4.7)$$

其中 N 為封包位元長度， y_n 與 ξ_n 分別代表在封包中第 n 位置之傳送訊息及該位置所對應之輸入事前訊息。當封包長度 N 足夠長時，則(4.7)式之近似式即成立。同理可推，輸出額外訊息也滿足高斯分佈的事實。

故設 $L_{out}^{[ext]}(y) = \psi$ ，其與傳送訊息 y 之間的互消息可表示成

$$\begin{aligned} I^{[ext]} = I(\psi; y) &= 1 - E_{\psi} \left\{ \log_2(1 + e^{-y \cdot \psi}) \right\} \\ &\approx 1 - \frac{1}{N} \sum_{n=1}^N \log_2(1 + e^{-y_n \cdot \psi_n}) \quad , 0 \leq I(\psi; y) \leq 1 \end{aligned} \quad (4.8)$$

其中額外訊息 $L_{out}^{[ext]}(y) = \ln \frac{P_{out}^{[ext]}(y = +1)}{P_{out}^{[ext]}(y = -1)}$ ， $P_{out}^{[ext]}(y)$ 為經過軟性解碼器計算

後輸出之額外訊息機率值。依據上述的推導，使用(4.7)與(4.8)可計算

互消息 $I^{[apri]}$ 與 $I^{[ext]}$ ，而額外資訊轉換圖則紀錄此兩者的關係，進而提

供作為效能的分析。如圖 4.2 所示， $L_{out}^{[ext]}(y)$ 是解碼器利用輸入訊號

$L(\hat{y}|y)$ 及 $L^{[apri]}(y)$ 所產生的輸出訊號，而 $L(\hat{y}|y)$ 又為 $\frac{E_b}{N_0}$ 的函數，因此

$I^{[ext]}$ 可表示為 $I^{[apri]}$ 和 $\frac{E_b}{N_0}$ 兩個變數的函數。所以在繪製額外資訊轉換圖

時，將固定訊雜比 $\frac{E_b}{N_0}$ ，以 $I^{[apri]}$ 為橫軸， $I^{[ext]}$ 為縱軸繪出一條曲線。每一條曲線代表某一特定通道品質 $\frac{E_b}{N_0}$ 反應出來的額外資訊轉換結果。

如前述，在模擬額外資訊轉移的過程中，我們將控制變數 σ_A 並使用高斯隨機變數來表現不同程度的事前訊號。為了找尋所需的 σ_A ，進一步將 $I(\xi; y)$ 按照定義展開，由[19]可知， $I(\xi; y)$ 為單一變數 σ_A 的函數，即 $I(\xi; y) = g(\sigma_A)$ ，且之間呈現遞增關係，

$$\lim_{\sigma_A \rightarrow 0} g(\sigma_A) = 0, \quad \lim_{\sigma_A \rightarrow \infty} g(\sigma_A) = 1 \quad (4.9)$$

故可以利用 σ_A 的設定值控制輸入事前互消息 $I^{[apri]}$ 的值，而 σ_A 的求取可利用 $\sigma_A = g^{-1}(I(\xi; y))$ ，並使用嘗試錯誤法，嘗試各種不同的 σ_A 直至出現需使用的 $I(\xi; y)$ 值。有了需要的 $I(\xi; y) = I^{[apri]}$ 值，即可在一已知通道訊雜比 $\frac{E_b}{N_0}$ 的通道品質之下，計算各個不同的 $I^{[apri]}$ 所輸出的 $I^{[ext]}$ 值，並得到額外資訊轉換特性圖。以下分別述說可變長度碼的疊代解碼中其軟性解碼的額外資訊轉換分析。

(A) 通道解碼器之額外資訊轉換特性

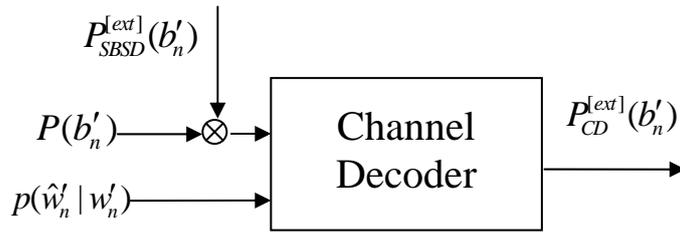


圖 4.3 通道解碼器架構圖

考慮第三章所說明之疊代解碼系統中的通道解碼器部分，如圖 4.3 所示。其中 $P(b'_n)$ 為位元事前機率， $p(\hat{w}'_n | w'_n)$ 為位元通道訊息， $P_{SBSD}^{[ext]}(b'_n)$ 為訊源解碼器輸出之額外訊息，而 $P_{CD}^{[ext]}(b'_n)$ 為通道解碼器輸出之額外訊息，可由(3.12)式求得。所有訊息之對數相似比率可由機率值求出，例如通道解碼輸出額外訊息之對數相似比率為 $I_{CD}^{[ext]}(b'_n) = \ln \frac{P_{CD}^{[ext]}(b'_n=0)}{P_{CD}^{[ext]}(b'_n=1)}$ 。模擬系統之額外資訊轉換特性時，利用設定 σ_A 方式控制輸入事前訊息機率，並結合通道訊息 $p(\hat{w}'_n | w'_n)$ 經過通道解碼運算求出額外訊息 $P_{CD}^{[ext]}(b'_n)$ ，則可由(4.8)式求出輸出之額外互消息 $I_{CD}^{[ext]}$ 。

圖 4.4 顯示通道解碼器所產生的額外資訊轉換圖，以通道解碼器之輸入事前互消息 $I_{CD}^{[apri]}$ 為橫軸，其輸出額外互消息 $I_{CD}^{[ext]}$ 為縱軸，分別繪出了通道品質為-6、0、3、6、15 dB 下的額外資訊轉換特性。當 σ_A^2 增加，代表輸入事前互消息的 $I_{CD}^{[apri]}$ 值也就越大。由圖中發現隨著可靠的輸入事前互消息 $I_{CD}^{[apri]}$ 增加，經過通道解碼可以得到越多可靠的輸出

額外互消息 $I_{CD}^{[ext]}$ 。每一條曲線皆呈現向點 $(I_{CD}^{[apri]}, I_{CD}^{[ext]}) = (1, 1)$ 集中的趨勢，也就是當 $I_{CD}^{[apri]}$ 提供至最大值 1 時， $I_{CD}^{[ext]}$ 亦可得到最大值 1 的可靠訊息。然而 $I_{CD}^{[apri]}$ 和 $I_{CD}^{[ext]}$ 之間並非線性遞增。值得注意的是，當訊雜比大於 3 (dB) 時， $I_{CD}^{[ext]}$ 幾乎不受事前消息影響。若 $I_{CD}^{[apri]}$ 固定的情況下，當通道品質越好時， $I_{CD}^{[ext]}$ 反應的可靠性越大。

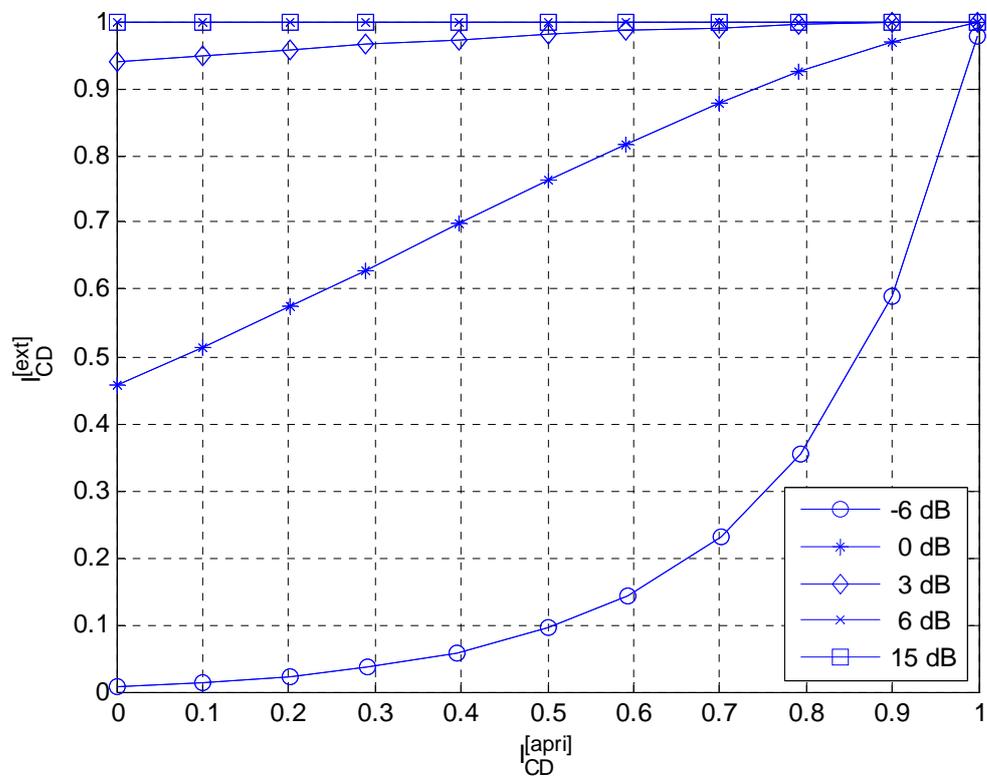


圖 4.4 通道解碼器之額外資訊轉換圖

(B) 訊源解碼器之額外資訊轉換特性

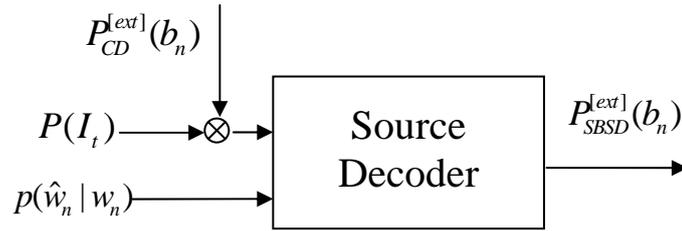


圖 4.5 訊源解碼器架構圖

考慮第三章說明之疊代解碼系統中的訊源解碼器部分，如圖 4.5 所示。其中 $P(I_t)$ 為索引事前機率， $p(\hat{w}_n | w_n)$ 為位元通道訊息， $P_{CD}^{[ext]}(b_n)$ 為通道解碼器輸出額外訊息，而 $P_{SBSD}^{[ext]}(b_n)$ 為訊源解碼器輸出之額外訊息，可由(3.17)式求得。且所有訊息之對數相似比率可由機率值求出。利用設定 σ_A 方式控制輸入事前訊息機率，模擬系統之額外資訊轉換特性。將事前訊息機率與通道訊息 $p(\hat{w}_n | w_n)$ 輸入訊源解碼器，並計算求出額外訊息 $P_{SBSD}^{[ext]}(b_n)$ 後，則可由(4.8)式求出輸出之額外互消息 $I_{CD}^{[ext]}$ 。

圖 4.6 為考慮傳送端以霍夫曼碼對傳送封包加以編碼，並針對訊源解碼器模擬所繪出的額外資訊轉換圖。其中橫軸 $I_{SBSD}^{[apri]}$ 為訊源解碼器之輸入事前互消息，縱軸 $I_{SBSD}^{[ext]}$ 為訊源解碼器之輸出額外互消息，且分別繪出了通道訊雜比為 0、3、6、15 dB 下的曲線。觀察此圖可以明顯發現在任何通道品質之下，曲線都非常平且低。當事前互消息 $I_{SBSD}^{[apri]}$ 為 0 時，額外互消息 $I_{SBSD}^{[ext]}$ 非常小，隨著 $I_{SBSD}^{[apri]}$ 值越來越大甚至提供至最大值 1 時， $I_{SBSD}^{[ext]}$ 仍無法有大幅度的增加。即使當通道品質相當好的

15 dB 時， $I_{SBSD}^{[ext]}$ 最大值仍在 0.1 以下。這也就表示即使提供解碼器再多的輸入事前互消息 $I_{SBSD}^{[apri]}$ ，也無法得到更可靠的輸出額外互消息

$I_{SBSD}^{[ext]}$ 。

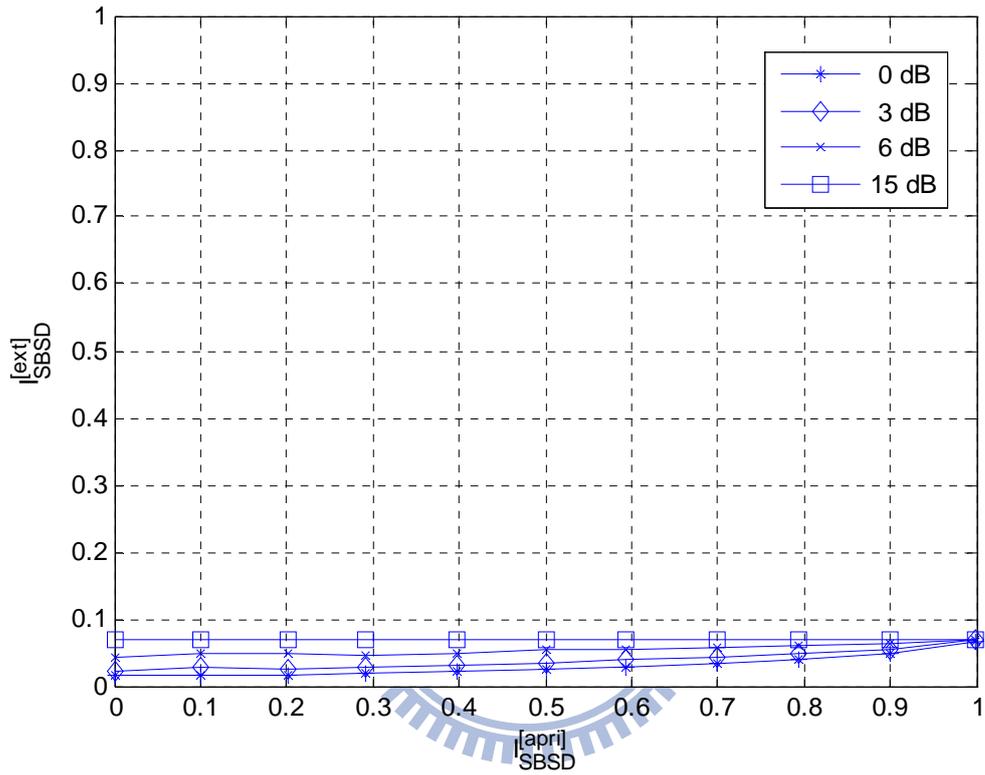


圖 4.6 訊源解碼器之額外資訊轉換圖(霍夫曼碼)

4.3 索引指定最佳化設計

由上一節的介紹我們知道，額外資訊轉換圖是有效分析疊代解碼效能的一項工具，透過該分析可以了解編解碼設計對於疊代系統作用的評估，進而估測每一次資訊交換的平均消息增益。而在本章第一節中，我們談到不同的可變長度碼設計，其結果顯示不同的設計理念透過犧牲些許的壓縮效益，將可提供充份的雜訊對抗能力。故本節將針對具有疊代解碼的通訊系統，運用額外資訊轉換分析，進行可變長度編碼索引指定的最佳化設計。

雙向性可變長度碼的訊源額外資訊轉換特性，如圖 4.7 所示。有別於霍夫曼碼，只要能提供解碼器越多的輸入事前互消息 $I_{SBSD}^{[apri]}$ ，就可以得到更可靠的輸出額外互消息 $I_{SBSD}^{[ext]}$ 。不但如此，當 $I_{SBSD}^{[apri]}$ 提供至最大值 1 時， $I_{SBSD}^{[ext]}$ 也可以得到近似最大值 1 的輸出額外訊息。所以針對疊代解碼的系統，將選擇雙向可變長度碼作為最佳化的初始樣本，再運用二元置換演算法進行碼書索引指定的調整，以期能找到具有最佳額外資訊轉換特性的編碼組合。

然而，計算精確且完整的額外資訊轉換特性需要大量的模擬計算，因此我們將嘗試找尋一組較精簡的參考值以取代繁復的模擬。在實驗結果觀察中發現，針對不同的可變長度編碼所得的互消息曲線上， $(I_{0.8}^{[apri]}=0.8, I_{0.8}^{[ext]})$ 與 $(I_1^{[apri]}=1, I_1^{[ext]})$ 兩點的位置可以有效區分出不同

編碼間的優劣差異。其中 $I_{0.8}^{[ext]}$ 及 $I_1^{[ext]}$ 分別代表 $I^{[apri]}$ 為 0.8 及 1 時所輸出的額外互消息。而一個好的編碼可在事前互消息 $I^{[apri]}=1$ 的情形下使疊代系統輸出額外互消息 $I^{[ext]}$ 趨近於 1，因此若兩組編碼均具有 $(I^{[apri]}, I^{[ext]})=(1, 1)$ 之特性時，代表兩者在疊代效果上均有可能良好的表現，然而，兩者間些許的差異將可由點 $(I^{[apri]}=0.8, I_{0.8}^{[ext]})$ 的不同分辨出來。據此，在後續設計中當我們考慮以一組較佳的雙向性可變長度碼作為初始碼書時，將僅使用 $(I^{[apri]}=0.8, I_{0.8}^{[ext]})$ 作為調整之重要參考值。

圖 4.7 繪出了兩種不同的雙向性可變長度碼在通道品質為 3dB 下所繪出的額外資訊轉換曲線，其中虛線代表的是 4.1.1 節所介紹具最短平均長度的雙向性可變長度碼(RVLC₁)，而實線代表的是在 4.1.2 節中所介紹具自由距離 $d_f \geq 2$ 的雙向性可變長度碼(RVLC₂)。發現此兩條曲線在 $I_{SBSD}^{[apri]}=0.8$ 時的距離為最大，代表觀察事前互消息為 $I_{SBSD}^{[apri]}=0.8$ 得到的額外互消息 $I_{SBSD}^{[ext]}$ 具有相當的鑑別度。且注意 RVLC₂ 為具有達到 $(I_{SBSD}^{[apri]}, I_{SBSD}^{[ext]})=(1, 1)$ 能力的編碼，在已知這樣的條件下，即可直接以 $I_{SBSD}^{[apri]}=0.8$ 得到的 $I_{SBSD}^{[ext]}$ 作為參考值。故在接下來的索引指定設計部分，將以一組具自由距離 $d_f \geq 2$ 的雙向性可變長度碼為初始樣本，且判別額外資訊轉換特性好壞的參考值 I^{ref} 指定為事前互消息 $I_{SBSD}^{[apri]}=0.8$ 時所計算出的額外互消息 $I_{SBSD}^{[ext]}$ 。

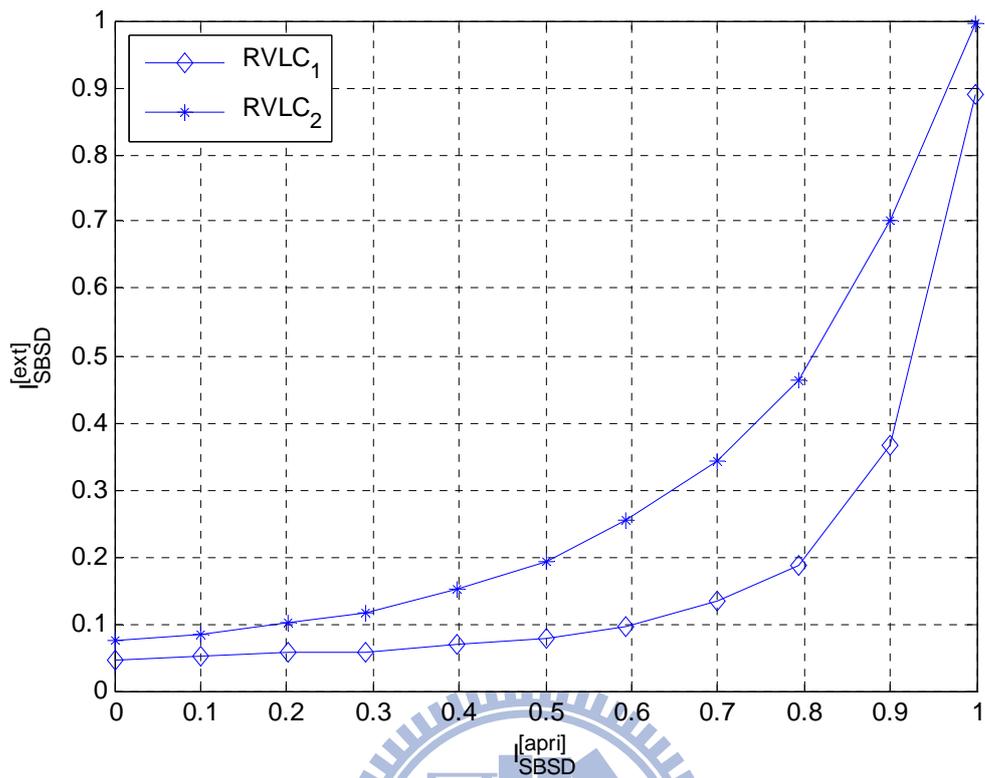


圖 4.7 RVLC 的 EXIT Chart

我們所提出的最佳化設計，將基於一組可變長度碼書中所有碼字的索引值互相做調整，希望能找到調整索引值後輸出額外互消息為最多的額外資訊轉換特性，並選擇該編碼組合為最佳選擇。針對這樣的議題，最直觀的做法就是考慮完全搜尋法(full search algorithm)，也就是嘗試所有可能的索引值排列組合，計算出全部組合的額外資訊轉換特性，再從中比較最好的輸出額外訊息，找到最好的組合作為我們的最佳索引指定。這樣的方式雖然可以找到全域性最大值(global maximum)，但運算量非常龐大。為了遵守可變長度編碼的原則以及減少計算量，我們將針對每一層分別各自採用計算量較低的二位元置

換演算法(binary switching algorithm)[13][14]，圖 4.8 為二位元置

換演算法之流程圖：

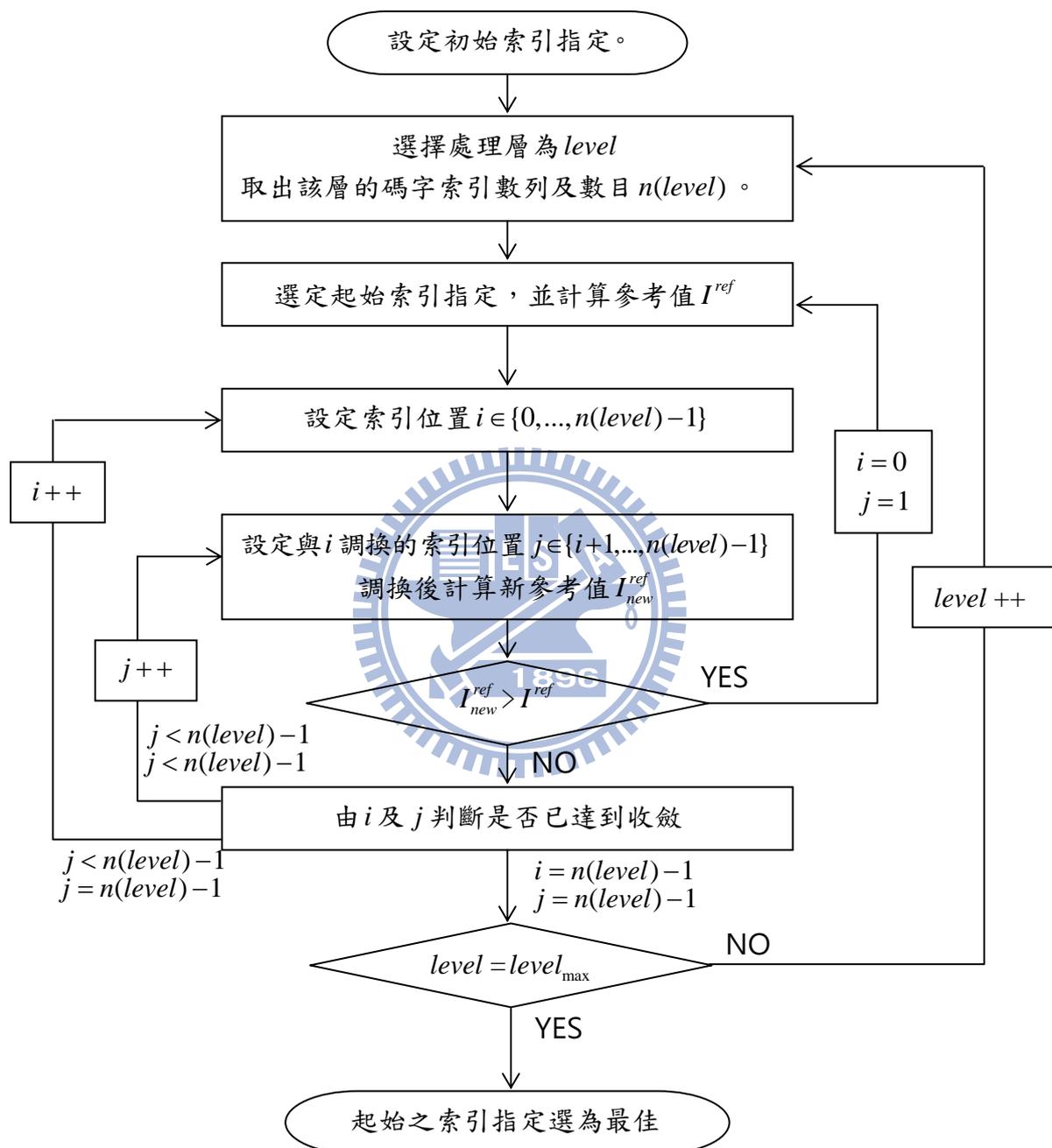


圖 4.8 可變長度碼之二位元置換演算法

索引機率	碼字長度	初始索引指定	碼字	最佳索引指定	碼字
0.1017	3	1	[0, 0, 0]	2	[1, 1, 1]
0.10106	3	2	[1, 1, 1]	1	[0, 0, 0]
0.10044	4	3	[0, 1, 0, 1]	7	[0, 0, 1, 1]
0.0976	4	4	[1, 0, 1, 0]	8	[1, 1, 0, 0]
0.08969	4	5	[0, 1, 1, 0]	3	[0, 1, 0, 1]
0.08928	4	6	[1, 0, 0, 1]	4	[1, 0, 1, 0]
0.07649	4	7	[0, 0, 1, 1]	5	[0, 1, 1, 0]
0.07514	4	8	[1, 1, 0, 0]	6	[1, 0, 0, 1]
0.06241	5	9	[0, 0, 1, 0, 0]	12	[1, 0, 0, 0, 1]
0.05777	5	10	[1, 1, 0, 1, 1]	10	[1, 1, 0, 1, 1]
0.04287	5	11	[0, 1, 1, 1, 0]	9	[0, 0, 1, 0, 0]
0.04113	5	12	[1, 0, 0, 0, 1]	11	[0, 1, 1, 1, 0]
0.02515	6	13	[0, 1, 0, 0, 1, 0]	13	[0, 1, 0, 0, 1, 0]
0.02314	6	14	[1, 0, 1, 1, 0, 1]	14	[1, 0, 1, 1, 0, 1]
0.00865	7	15	[0, 1, 0, 0, 0, 1, 0]	15	[0, 1, 0, 0, 0, 1, 0]
0.00748	7	16	[1, 0, 1, 1, 1, 0, 1]	16	[1, 0, 1, 1, 1, 0, 1]
平均長度			4.14639		4.14639

表 4.2 索引指定對照表

由表 4.2 看到在開始運作二位元置換演算法之前，所設定的初始索引指定為索引機率大至小排列。在進行索引指定的二位元置換演算法後，將會更改索引機率所對應的編碼碼字。然而置換方式為針對每一層分別各自置換，故其平均長度不變。以下說明其置換步驟：

首先，如表 4.2 依索引機率設定一組作為初始值的索引指定後，計算這組碼書的額外互消息參考值 I^{ref} ，且該組碼書中碼字最短長度記為 $level_{min}$ 和最長長度記為 $level_{max}$ 。接著將考慮的碼層變數 $level$ 由

碼字長度最短的 $level_{\min}$ 層開始，取出該層的所有索引值，且令該層索引值數量為 $n(level)$ ，則該層具 $n(level)$ 個索引位置 $\{0, \dots, n(level)-1\}$ 。令 i 為該層之觀察索引位置，且設定 $i=0$ ，再分別與下面的索引位置 $j=i+1, \dots, n(level)-1$ 逐步置換，每置換一次就計算新的參考值 I_{new}^{ref} ，若新的參考值 I_{new}^{ref} 比原本的參考值 I^{ref} 大則進行置換，否則索引位置維持不變，再選擇下一個 j 來與 i 置換。若發生置換則對調索引位置 i 與 j 的索引值，並重新設定 $i=0$ 與 $j=i+1, \dots, n(level)-1$ 兩兩置換。若觀察索引位置 i 與全數的 j 都已試著換過，但沒有應置換的情形 ($I_{new}^{ref} > I^{ref}$) 發生的話，則將觀察索引位置 i 由 i 移至 $i+1$ ，再重覆與其他可以置換的情況相比較，直到觀察索引位置 i 移至 $n(level)-1$ ，且所有觀察的索引不再具有 I^{ref} 增加的置換發生時，則該索引指定即為該層的最佳索引指定。這樣的步驟由 $level_{\min}$ 一直進行到 $level_{\max}$ ，每一層都完整地執行一次，就能找到每一層的最佳索引指定。

最後針對初始指定是根據索引機率由大至小排列之原因說明如下。我們注意到二位元置換演算法的特性為觀察索引位置 i 總是由第一位置 $i=0$ 開始與其他索引位置 j 逐一置換的方式做搜尋。所以若我們設法讓機率較大的索引值有最多的置換可能，就能做最有效率的置換，故一開始設定的初始索引指定為同長度的碼字之間，以機率由大至小排列的索引值，即可有效的找到一組最佳索引指定。

第五章 實驗模擬與結果分析

在前面的章節已經詳細的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，我們將模擬系統的傳送端與接收端，計算其位元及索引錯誤率。除此之外，我們亦計算系統的額外資訊轉換圖並用以分析各種可變長度碼的效能，再深入討論分析我們的推論。

本章分為三項實驗，首先我們將利用第二章的軟性訊源解碼機制探討各種可變長度碼在傳輸上的效益。實驗二則將各種可變長度碼代入第三章之疊代訊源通道解碼系統，個別模擬其額外資訊轉換圖，並以位元錯誤率(bit error rate, BER)及索引錯誤率(symbol error rate, SER)加以討論。最後實驗三則利用 4.3 節所介紹的二位元置換演算法所設計的碼書，模擬其額外資訊轉換特性及錯誤率，並加以比較討論。

[實驗一]

目的：探討不同可變長度碼之解碼效能

模擬環境：考慮如圖 2.1 之傳輸架構，符號 U_t 採用一階高斯-馬可夫

訊號源，其相關因子設定為 $\rho=0.9$ ，且其變異數為

$\sigma_u^2=1$ 。訊號源關係式表示為 $U_t = \rho \cdot U_{t-1} + n_t$ 。其中 n_t 為平

均值為 0 的附加性白色高斯雜訊。

每一次模擬以封包 $\underline{U}=[U_1, U_2, \dots, U_K]$ 為單位傳送。每一個封包含有 $K=100$ 個符號。而每個符號都獨立經由 $M=4$ 位元的純量量化器處理後得到 16 階索引序列 $\underline{I}=[I_1, I_2, \dots, I_K]$ ，其中 $I_k \in \mathcal{I}=\{0, 1, \dots, 2^{M-1}\}$ 。接著根據可變長度碼將封包內不同的索引值個別作編碼，得到位元總長度為 N 的二位元字串 $\underline{b}=[c(I_1), c(I_2), \dots, c(I_K)]=[b_1, b_2, \dots, b_N]$ 。由於碼書為可變長度碼，故每個封包所輸出的位元總長度 N 不盡相同。以二位元相位鍵移調變後，經附加性高斯白雜訊通道傳送。接收端將受到雜訊干擾的資料以第二章的方式解碼，並計算其索引錯誤率。

結果：

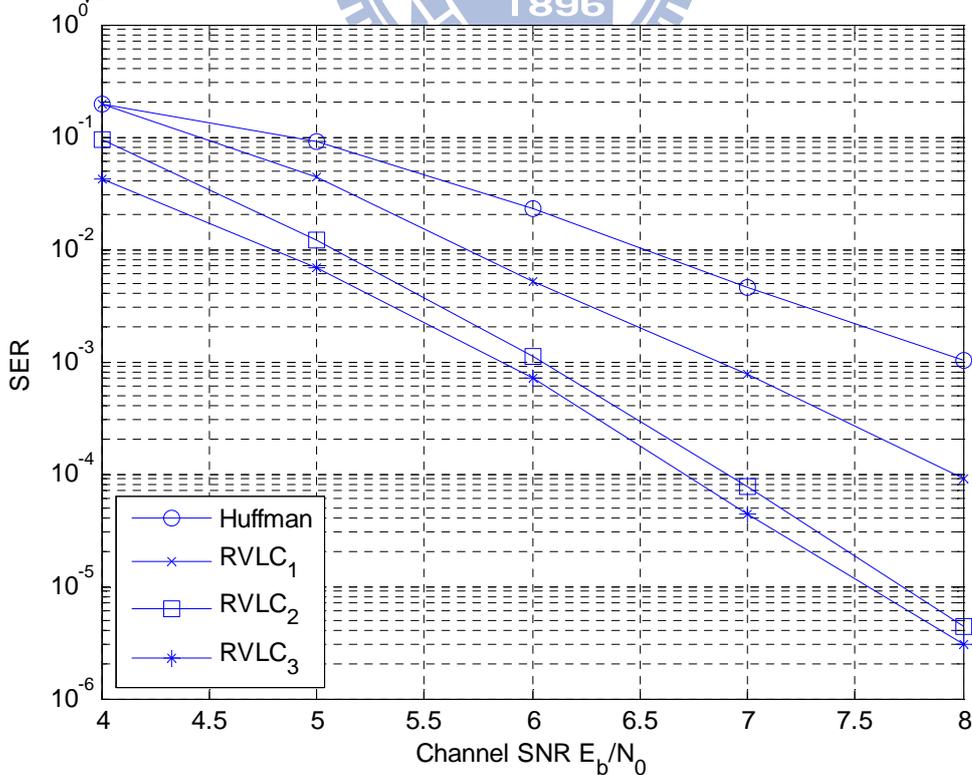


圖 5.1 不同可變長度碼之解碼效能比較

討論：本實驗以第二章之架構探討不同可變長度碼的解碼效能，其結果如圖 5.1 所示，橫軸為通道訊雜比(Channel SNR, CSNR)，縱軸為索引錯誤率。其中 Huffman 為霍夫曼碼，RVLC₁ 為 4.1.1 節所介紹具最短平均長度的雙向性可變長度碼，而 RVLC₂ 代表的是在 4.1.2 節中所介紹具自由距離 $d_f \geq 2$ 的雙向性可變長度碼，最後 RVLC₃ 為 4.3 節所說明經過索引指定設計之雙向性可變長度碼。觀察圖可以發現，在通道品質較好時，霍夫曼碼的索引錯誤率中高於其他雙向性可變長度碼，其原因是霍夫曼碼只符合前置條件，而雙向性可變長度碼則同時符合前置與後置條件。這些差別使得解碼器執行 BCJR 演算法將有明顯的差異。因此解碼執行完成所輸出的後驗機率進行估算時，位元錯誤進而使得索引錯誤變得嚴重。另外我們發現 RVLC₁ 與 RVLC₂ 亦有明顯的差距，如同第四章所述，這是因為 RVLC₂ 具有較好的錯誤更正能力($d_f \geq 2$)。也由此驗證了自由距離與錯誤更正能力的關係。而錯誤率最低的 RVLC₃ 則表示了經過索引指定設計可使可變長度碼具有更好的解碼效果。

[實驗二]

目的：針對不同的可變長度碼，以額外資訊轉換特性分析疊代解碼效能，並模擬其疊代系統錯誤率加以討論。

模擬環境：考慮如圖 3.1 之傳輸架構，符號 U_t 採用一階高斯-馬可夫訊號源，其相關因子設定為 $\rho=0.9$ ，且其變異數為 $\sigma_U^2=1$ 。訊號源關係式表示為 $U_t = \rho \cdot U_{t-1} + n_t$ 。其中 n_t 為平均值為 0 的附加性白色高斯雜訊。

每一次模擬以封包 $\underline{U}=[U_1, U_2, \dots, U_K]$ 為單位傳送。每一個

封包含有 $K=100$ 個符號。而每個符號都獨立經由 $M=4$

位元純量量化器處理後得到 16 階索引序列

$\underline{I}=[I_1, I_2, \dots, I_K]$ ，其中 $I_k \in \mathcal{I}=\{0, 1, \dots, 2^{M-1}\}$ 。接著依可變

長度碼將封包內不同的索引值個別作編碼，得到位元總長

度為 N 的二位元字串 $\underline{b}=[c(I_1), c(I_2), \dots, c(I_K)]=[b_1, b_2, \dots, b_N]$ 。接

著將編碼完成的封包傳入 S-random 交錯器，且設定

$S=\sqrt{\frac{N}{2}}$ ，交錯器的詳細流程於附錄 A 詳述。接著將交錯

器輸出之碼串 $\underline{b}'=[b'_1, b'_2, \dots, b'_N]$ 輸入通道編碼器，在此我們

採用編碼率 $R_c=\frac{1}{2}$ 且暫存器長度為 2 的迴旋碼

(Convolutional code)，編碼生成矩陣為

$G(D)=\begin{bmatrix} 1 & \frac{(1+D^2)}{(1+D+D^2)} \end{bmatrix}$ 。將通道編碼器輸出之封包

$\underline{w}' = [w'_1, w'_2, \dots, w'_N]$ 以二位元相位鍵移調變後，送入已知訊

雜比 $\frac{E_b}{N_0}$ (dB) 的附加性白高斯雜訊通道傳送。注意在封包

經過通道編碼後， \underline{w}' 的長度為 \tilde{N} 而非 N ，這是因為此長度除了訊息位元長度之外，還包括了終止位元的關係。

接收端考慮如圖 3.2 之架構圖，實驗中設定疊代次數為

10 次。其中訊源解碼器之演算法如第二章所述，通道解

碼器之演算法如 3.2 節所推導，其解碼步驟則如 3.1 節所

說明。在多次疊代完成後，分別計算出位元錯誤率，以及

索引錯誤率。

結 果：

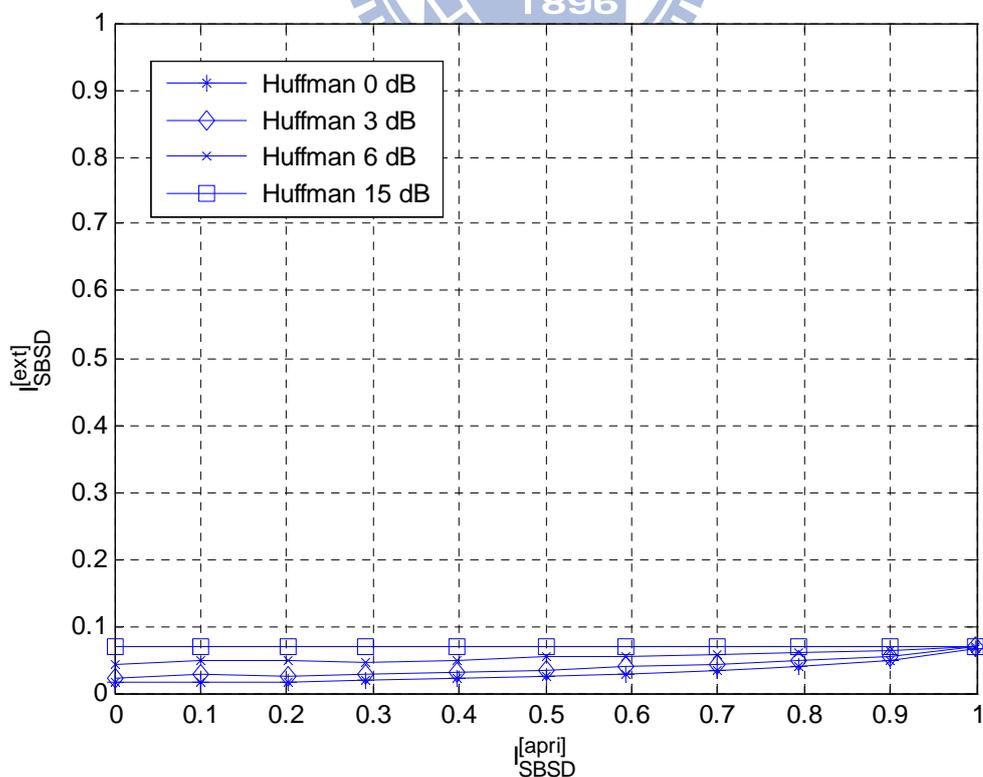


圖 5.2 Huffman Code 之 EXIT Chart

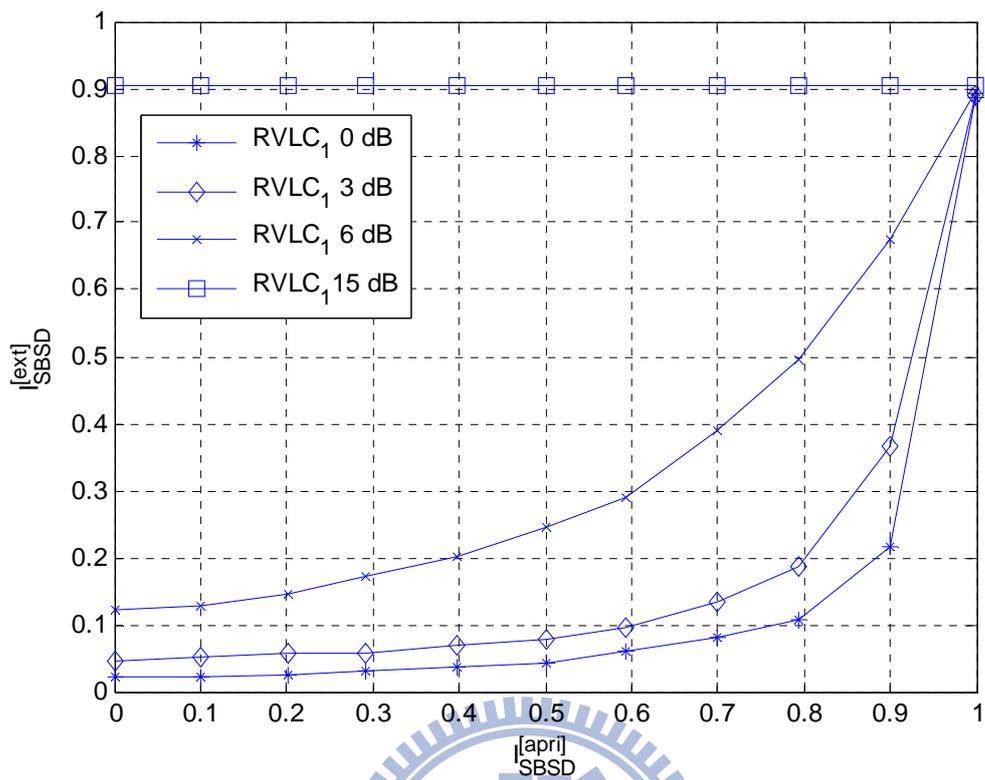


圖 5.3 RVLC₁ 之 EXIT Chart

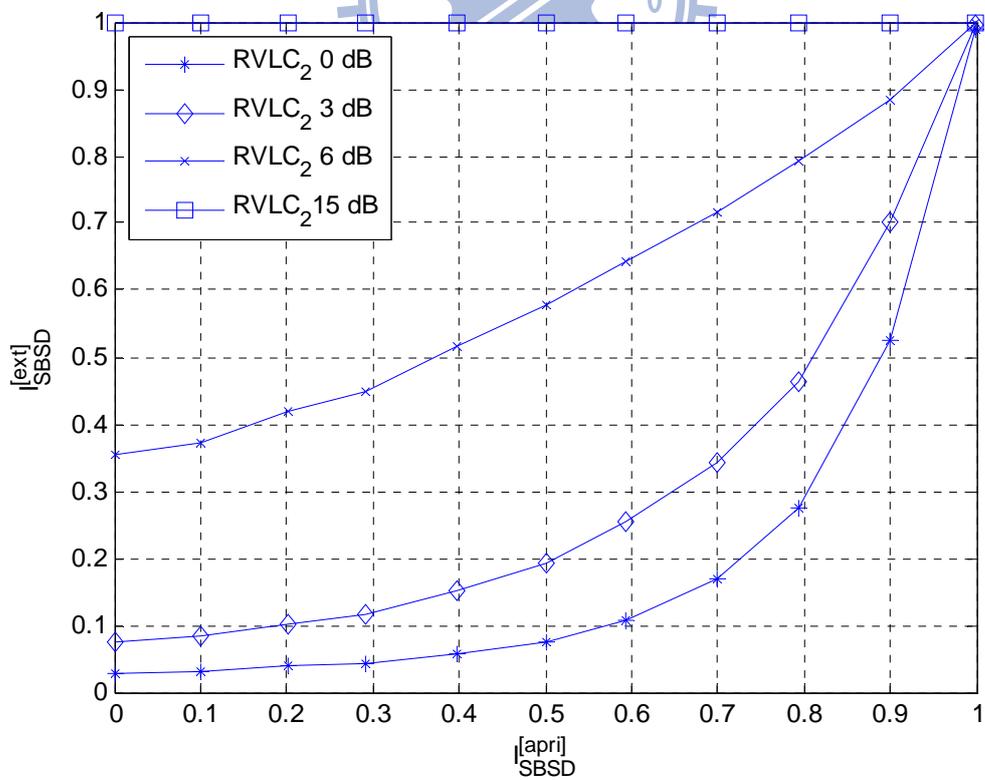


圖 5.4 RVLC₂ 之 EXIT Chart

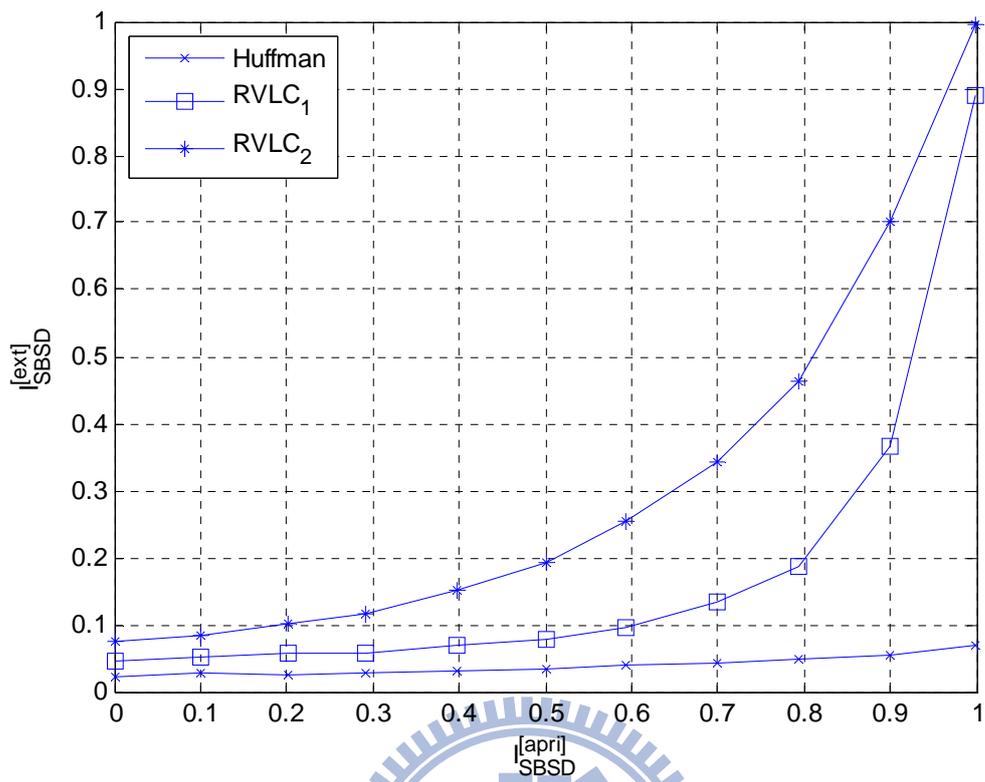


圖 5.5 可變長度碼之 EXIT Chart 比較 (CSNR=3dB)

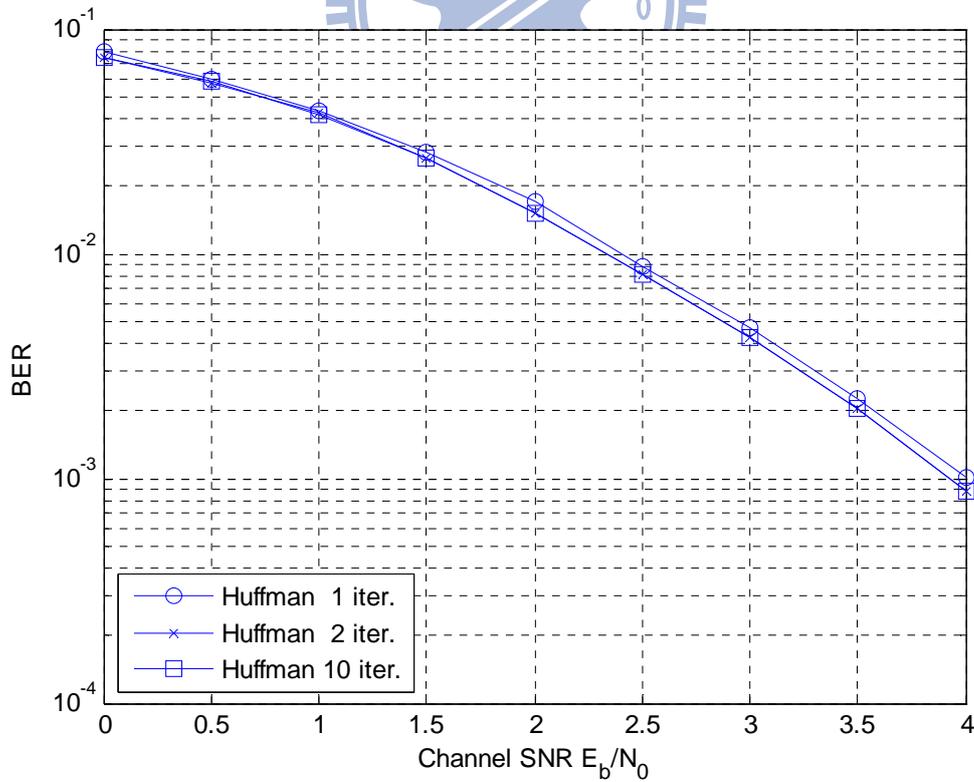


圖 5.6 Huffman Code 之 BER

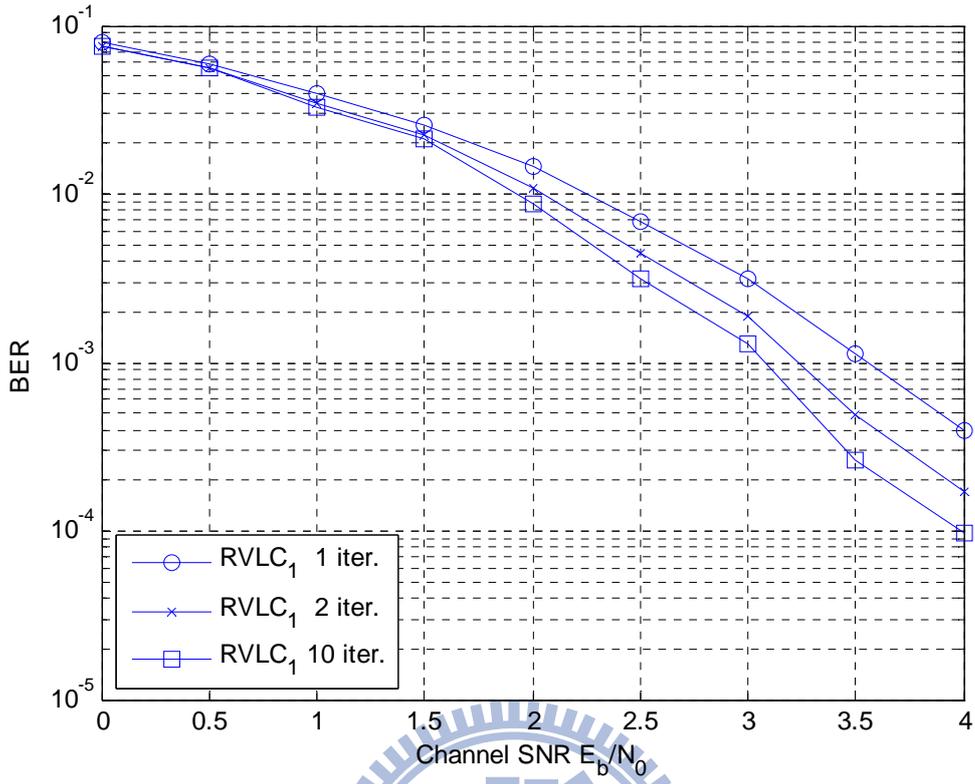


圖 5.7 RVLC₁ 之 BER

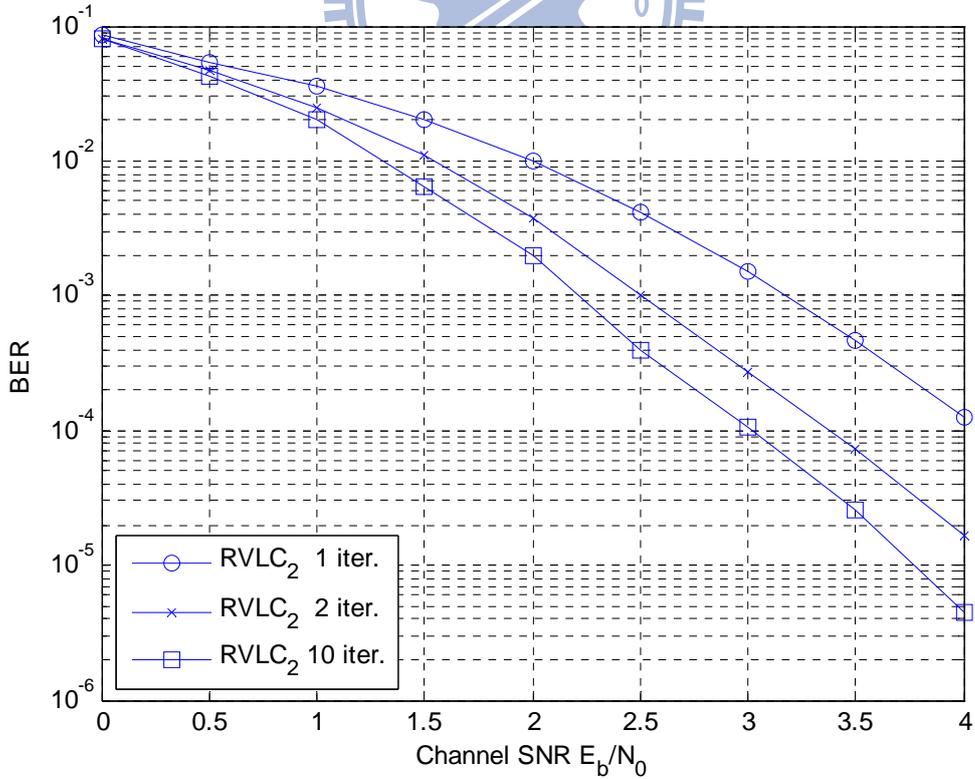
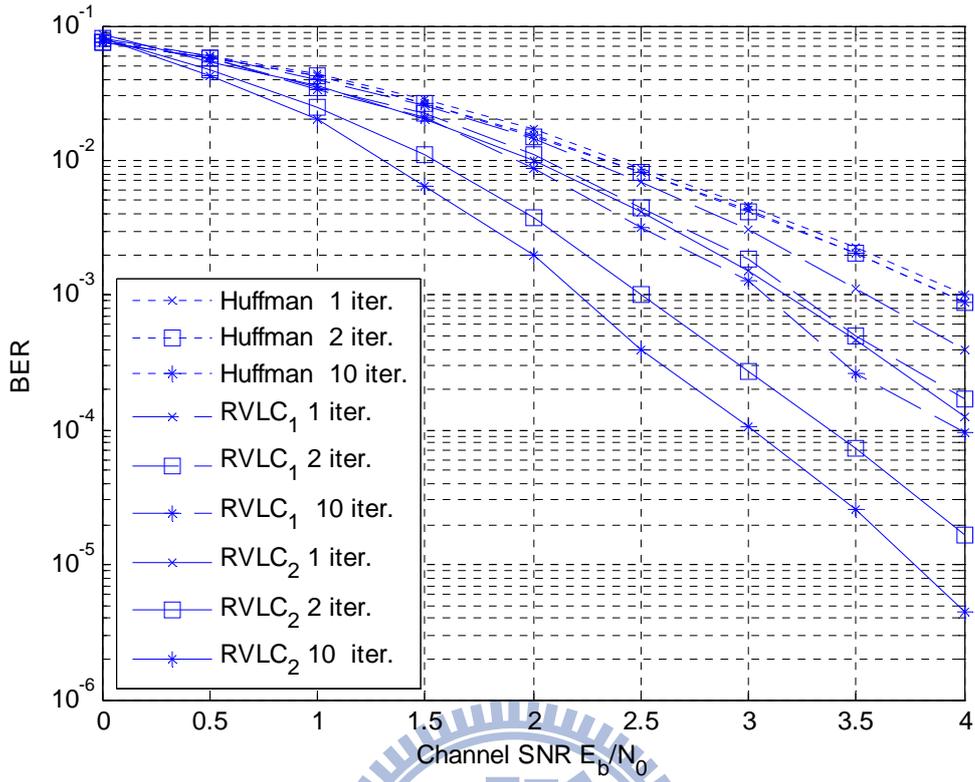
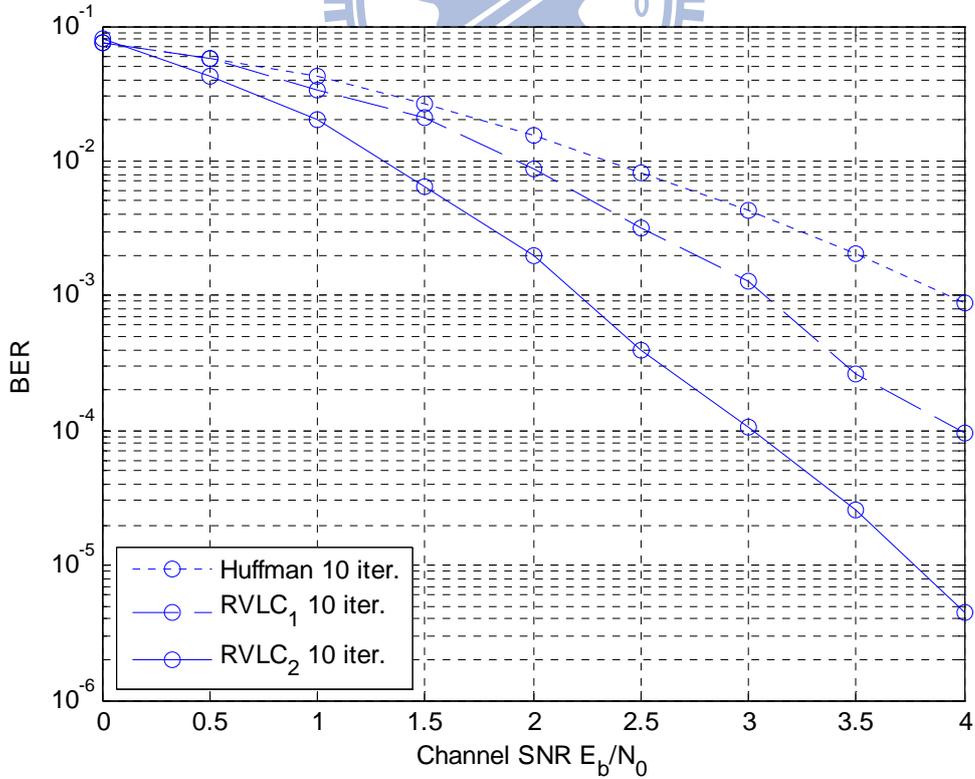


圖 5.8 RVLC₂ 之 BER



(a) Huffman、RVLC₁、RVLC₂之1、2、10次疊代 BER 比較



(b) Huffman、RVLC₁、RVLC₂之第10次疊代 BER 比較

圖 5.9 可變長度碼之 BER 比較

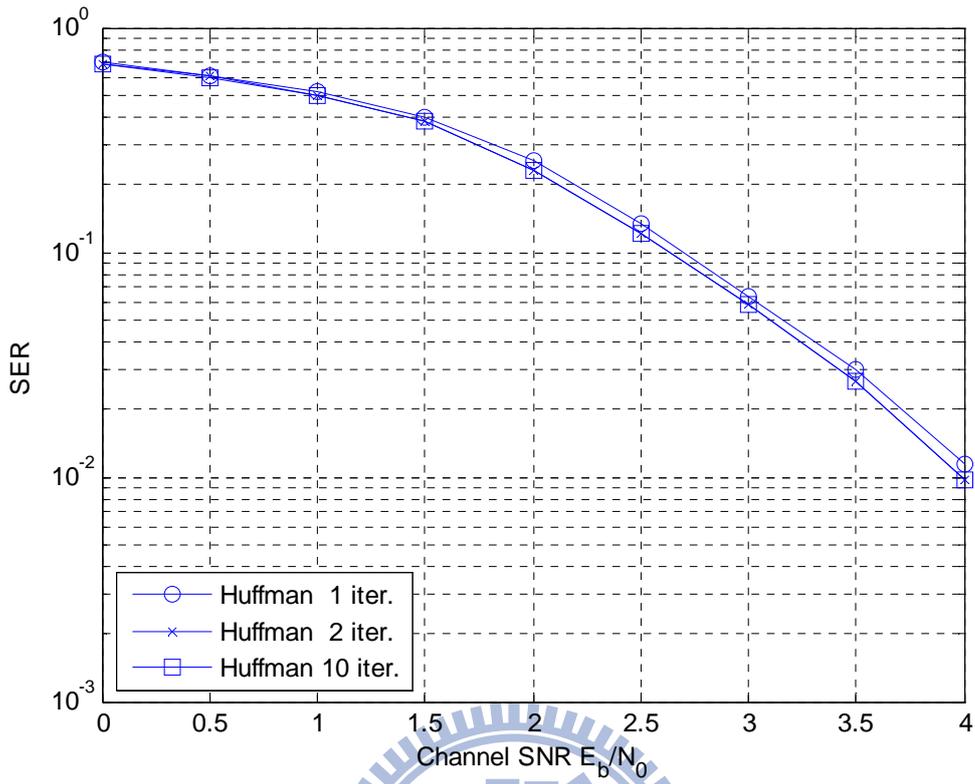


圖 5.10 Huffman Code 之 SER

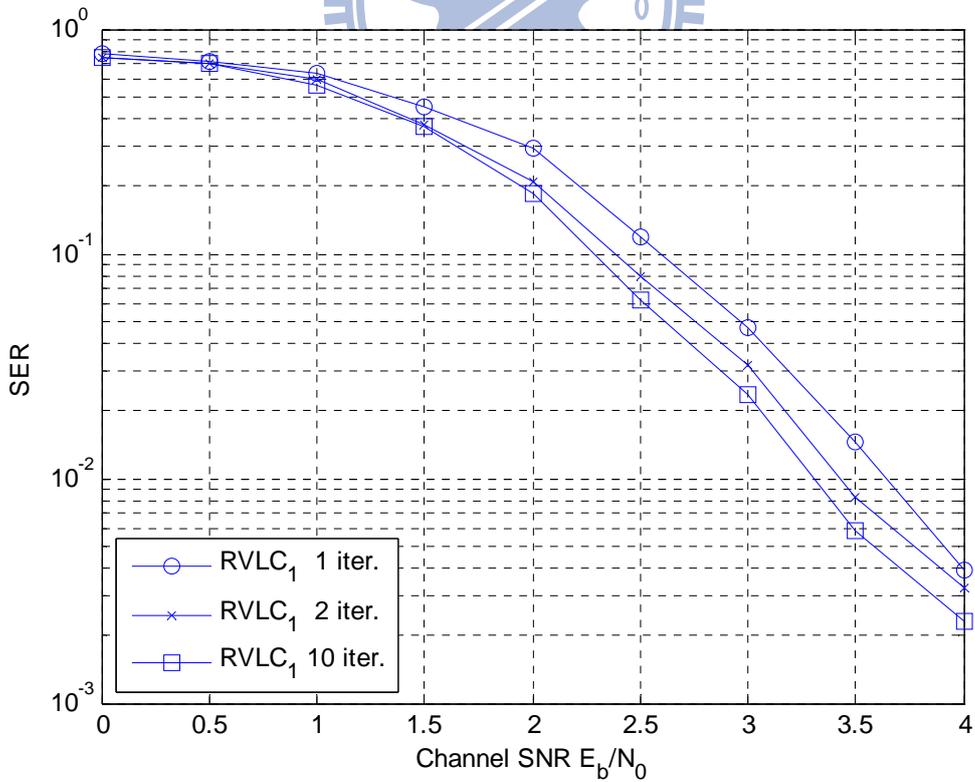


圖 5.11 RVLC₁ 之 SER

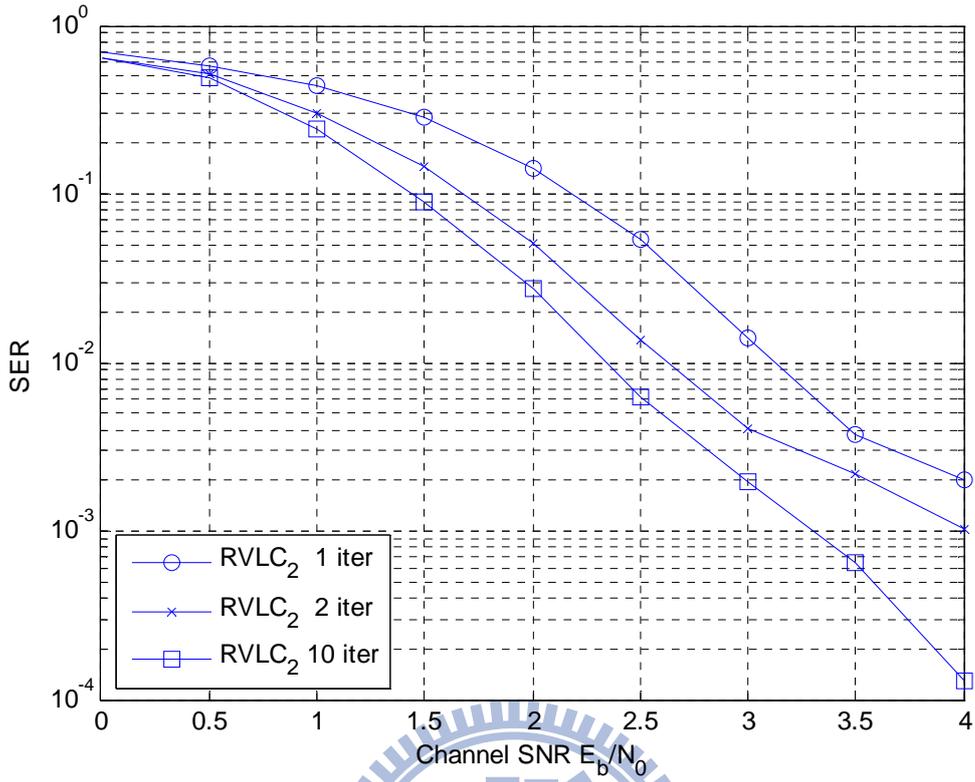
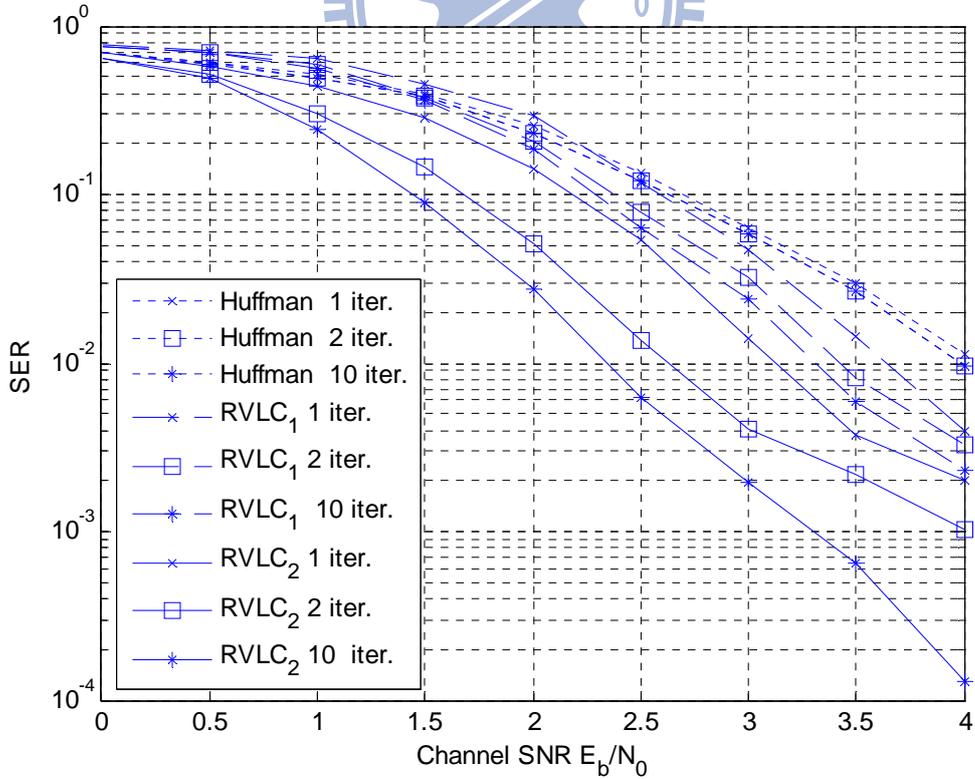
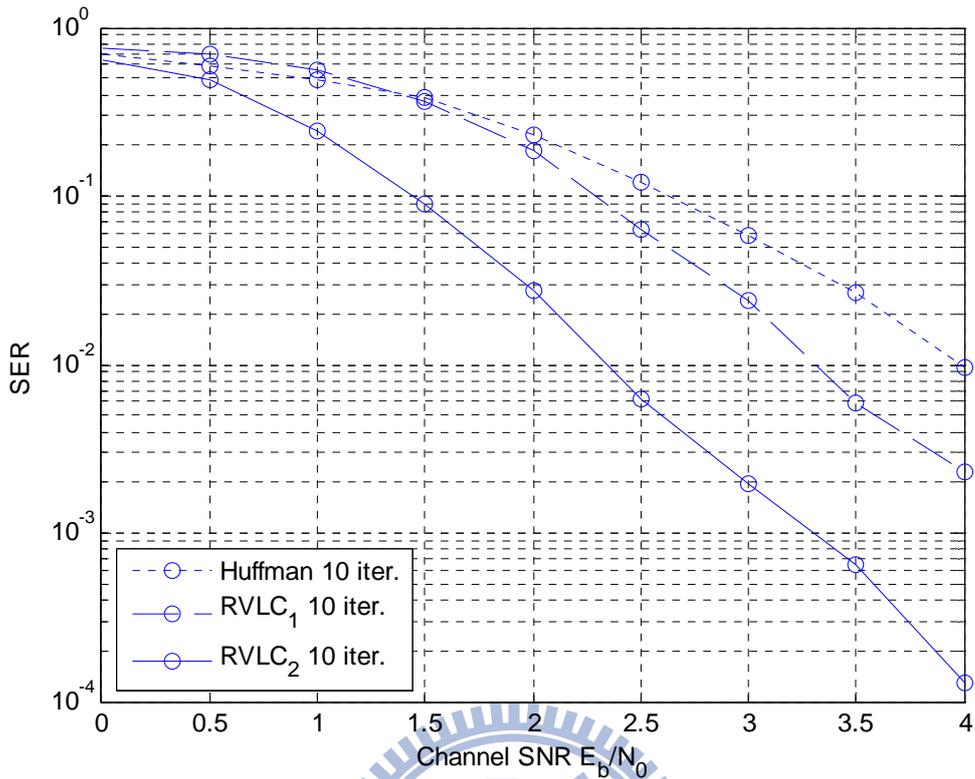


圖 5.12 RVLC₂ 之 SER



(a) Huffman、RVLC₁、RVLC₂ 之 1、2、10 次疊代 SER 比較



(b) Huffman、RVLC₁、RVLC₂之第10次疊代 BER 比較

圖 5.13 可變長度碼之 SER 比較

討論：圖 5.2 至圖 5.5 是不同可變長度碼之訊源解碼器額外資訊

轉換圖，通道訊雜比分別為 0、3、6、15dB，其中橫軸

為事前互消息 $I_{SBSD}^{[apri]}$ ，縱軸為額外互消息 $I_{SBSD}^{[ext]}$ 。如圖 5.2 所

示，可以明顯看出使用霍夫曼碼時，在任何通道品質之下

的曲線都非常平且低。這意味著即使提供再多的事前互消

息，系統也無法釋出更有力的額外互消息。再仔細比較圖

5.3 與圖 5.4，RVLC₁ 雖然相對於霍夫曼碼可以達到較高

的額外互消息量，但在事前互消息提供最大值 1 時僅達

到 $(I^{[apri]}, I^{[ext]})=(1, 0.9042)$ 。然而 RVLC₂ 卻在任何通道品質之下，只要事前互消息提供最大值 1 時，即可達到 $(I^{[apri]}, I^{[ext]})=(1, 1)$ 的特性。不但如此，我們再觀察圖 5.5，在固定訊雜比為 3dB 的情形下，比較此三種可變長度碼的額外資訊曲線。我們不但發現雙向性可變長度碼遠高於霍夫曼碼，亦發現 RVLC₂ 的整體曲線高於 RVLC₁。在此我們即判斷 RVLC₂ 在疊代效能上將優於 RVLC₁。

我們接下來利用實際模擬疊代訊源通道解碼機制，並以上述的可變長度碼分別計算出其錯誤率。圖 5.6 至圖 5.8 為三種可變長度碼在疊代系統所計算出來第 1、2、10 次疊代的位元錯誤率，其中橫軸為通道訊雜比，縱軸為位元錯誤率，而圖 5.9 為上述位元錯誤率之整體比較。圖 5.10 至圖 5.12 為三種可變長度碼的索引錯誤率，其中橫軸為通道訊雜比，縱軸為索引錯誤率，圖 5.13 為上述索引錯誤率之整體比較。觀察圖 5.6，霍夫曼碼在疊代第 1 次至第 2 次時位元錯誤率有些許下降，但可以發現第 2 次疊代至第 10 次疊代之間，錯誤率卻完全沒有改善的現象。這表示了霍夫曼碼無法發揮疊代系統的效能。當位元錯誤無法經疊代解碼改善的情形下，由圖 5.10 發現索引錯誤

率在第 2 次疊代至第 10 次疊代之間也完全沒有改善的趨勢。

接著我們再觀察圖 5.7 與圖 5.8，可以注意到圖 5.7 所表示 RVLC₁ 的位元錯誤率整體比霍夫曼碼低許多，並且在經過第 2 次疊代後錯誤率仍有下降的趨勢。這代表雙向性可變長度碼在疊代解碼系統上可以發揮效能。再與圖 5.8 相比較發現 RVLC₂ 的錯誤率又比 RVLC₁ 更低，並且注意到在每次疊代所減少的錯誤率比 RVLC₁ 明顯。對照 RVLC₁ 與 RVLC₂ 的額外資訊轉圖，每次疊代代表得到較好的事前機率，而 RVLC₂ 在額外互消息的表現上高於 RVLC₁，這使得 RVLC₂ 在每次疊代可以有更好的改善趨勢。同樣位元錯誤影響索引錯誤率，因此圖 5.11 與圖 5.12 也發現同樣的情形，因此我們驗證了在額外資訊轉換圖中的推測。這個觀察也提供了我們將以額外資訊轉換特性，設計雙向性可變長度碼之索引指定。

[實驗三]

目的：探討可變長度碼之索引指定設計

模擬環境：以自由距離 $d_f \geq 2$ 的雙向性可變長度碼(RVLC₂)作為初始樣

本，並設定其初始索引是依索引機率由大至小排列。使用

二位元置換演算法進行索引指定的置換，搜尋到的最佳索

引指定其碼書命名為 RVLC₃，其中二位元置換演算法及其

參考值如 4.3 節所說明。

結果：

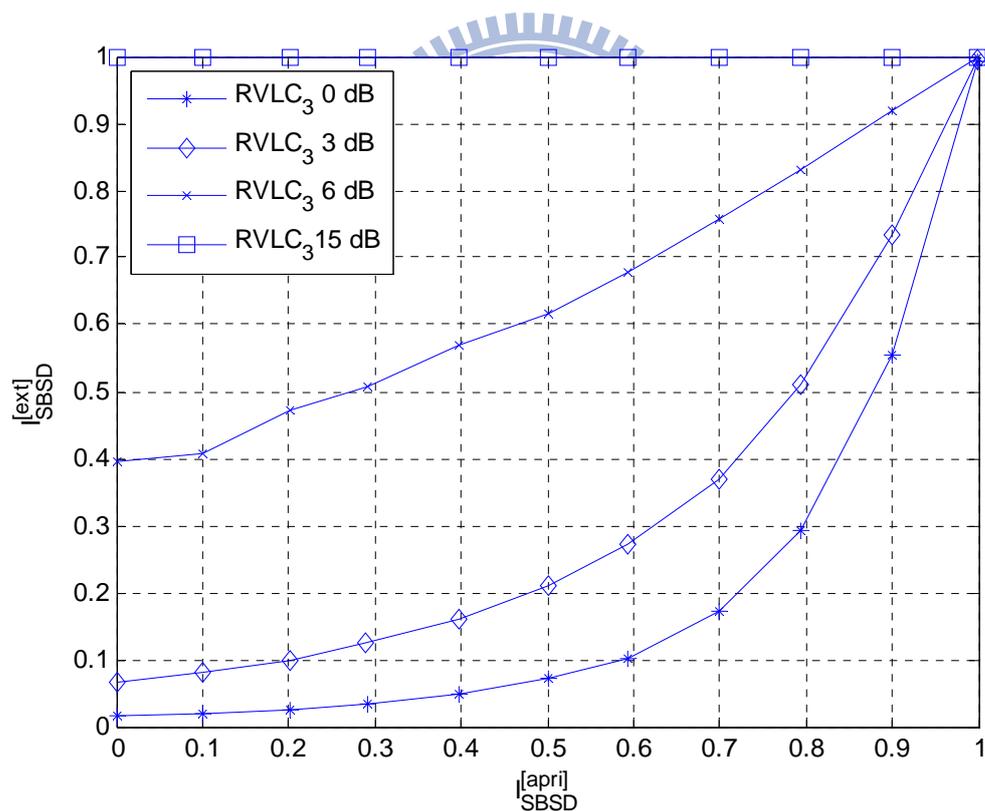


圖 5.14 RVLC₃ 之 EXIT Chart

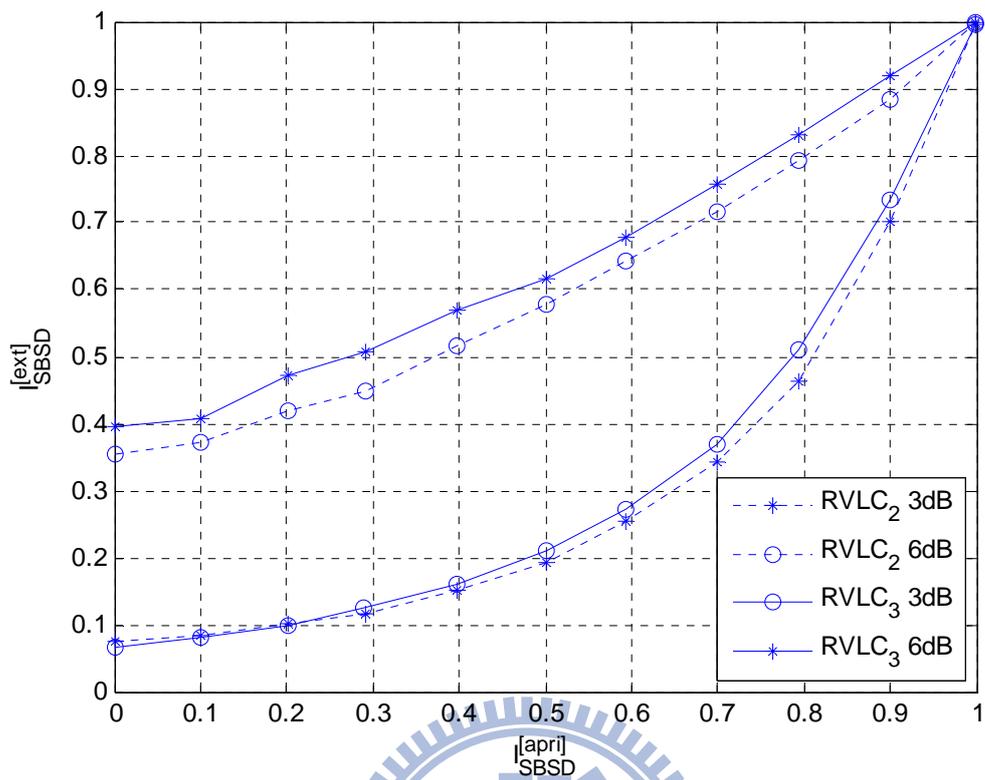


圖 5.15 RVLC₂ 與 RVLC₃ 之 EXIT Chart 比較

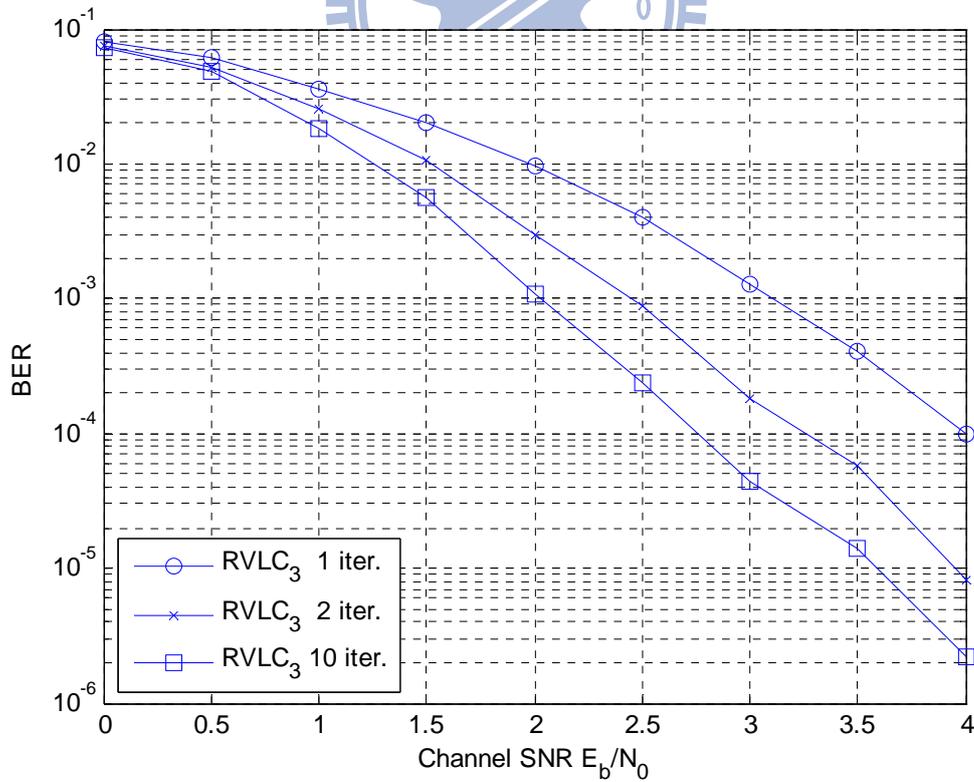


圖 5.16 RVLC₃ 之 BER

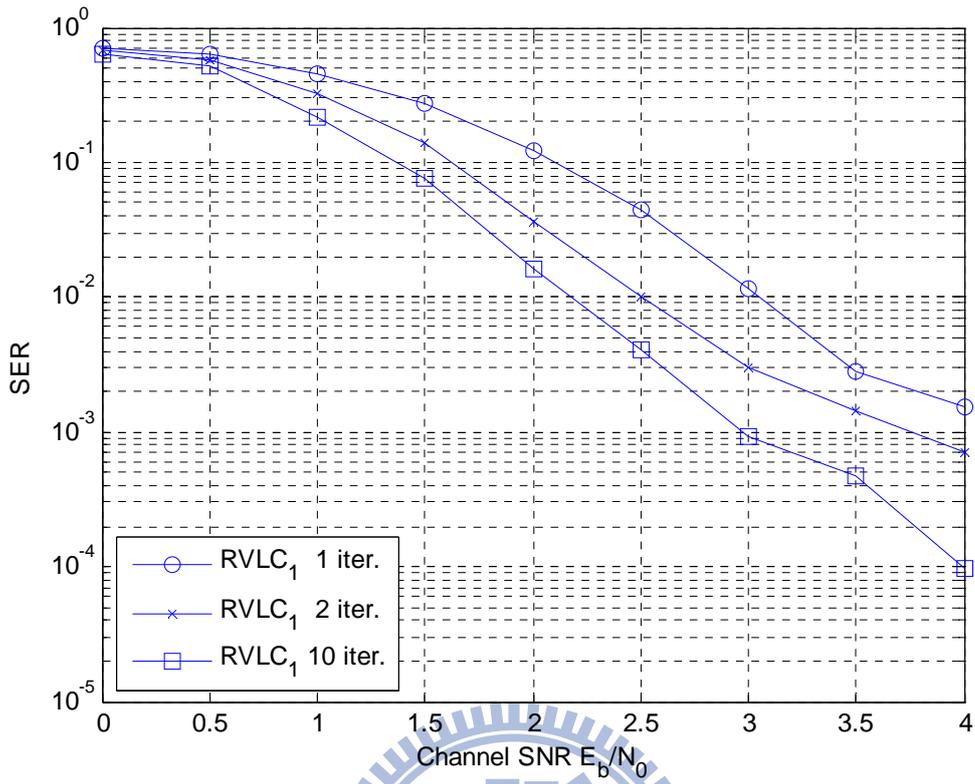


圖 5.17 RVLC₃ 之 SER

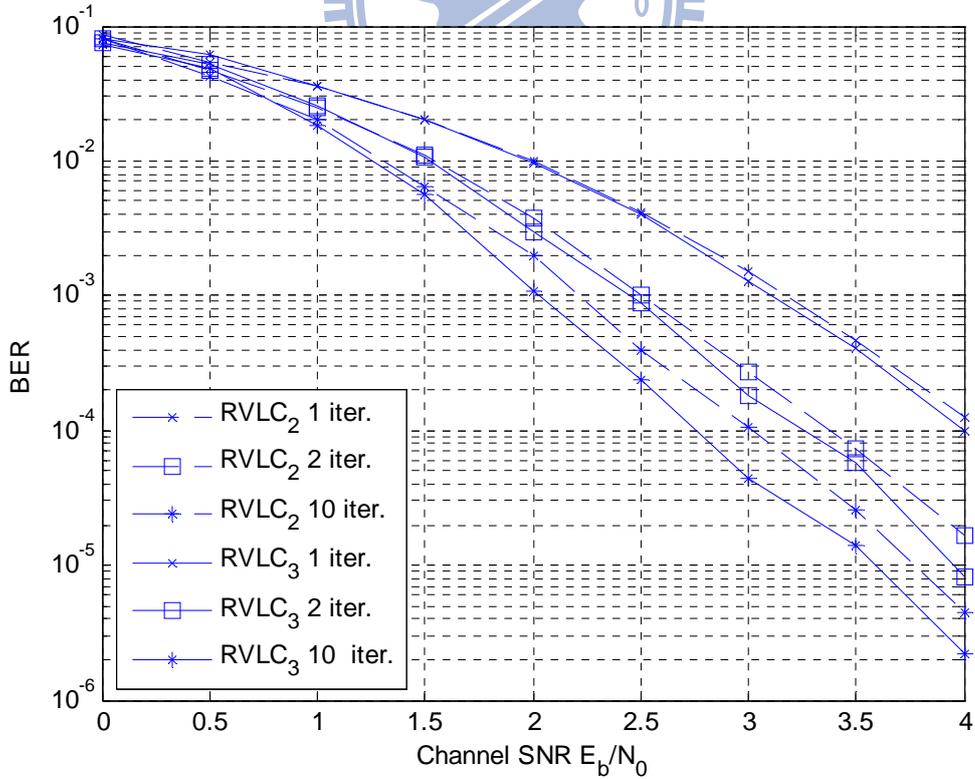


圖 5.18 RVLC₂ 與 RVLC₃ 之 BER 比較

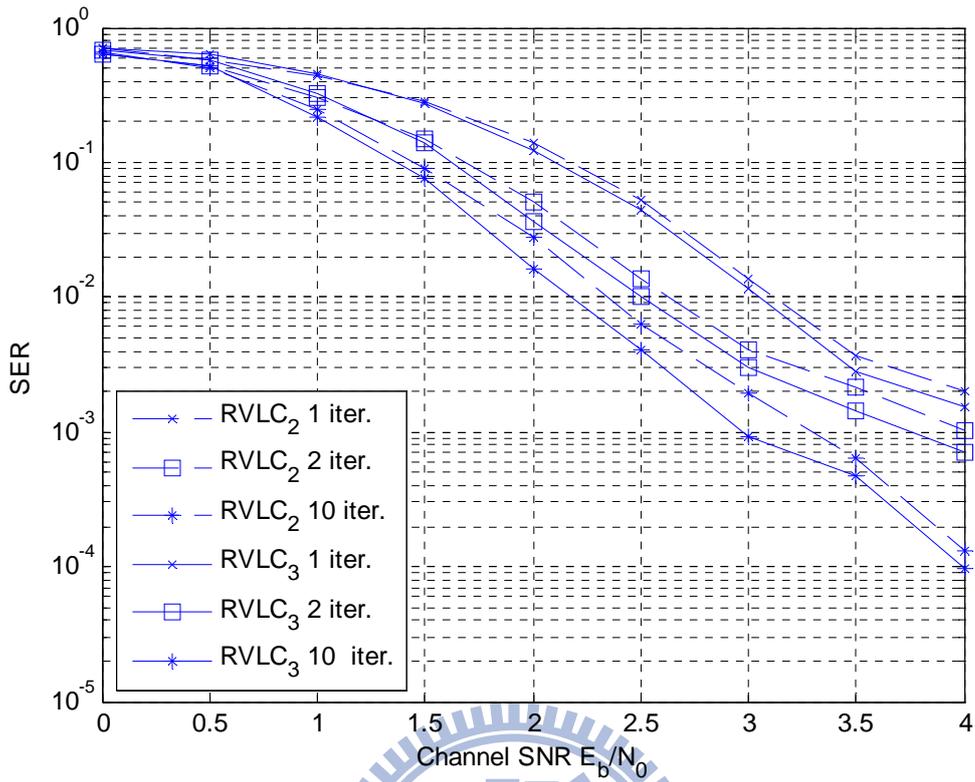


圖 5.19 RVLC₂ 與 RVLC₃ 之 SER 比較

討論：RVLC₃ 為基於額外資訊轉換特性，對索引指定做二位元置換演算法所設計出來的一組碼書。圖 5.14 為 RVLC₃ 模擬通道訊雜比分別為 0、3、6、15dB 之訊源解碼器額外資訊轉換圖，其中橫軸為事前互消息 $I_{SBSD}^{[apri]}$ ，縱軸為額外互消息 $I_{SBSD}^{[ext]}$ 。而圖 5.15 則將 RVLC₂ 與 RVLC₃ 在通道訊雜比在 3dB 及 6dB 下的額外資訊轉換特性作比較，可以發現 RVLC₃ 所輸出的額外資訊比 RVLC₂ 略高，且在 6dB 下有明顯的差距。我們再利用實際模擬疊代訊源解碼機制計算其位元錯誤率(圖 5.16)與索引錯誤率(圖 5.17)加以分析。

接著將上述之錯誤率與 RVLC₂ 做比較，並分別以圖 5.18 與圖 5.19 表示。我們可以發現即使不更動 RVLC₂ 碼書的內容，但經過索引指定的設計之後，RVLC₃ 在位元及索引層級的錯誤率都有明顯改善。此外，由於設計時在各層分開執行二位元置換演算法，由表 4.2 可以發現經過設計後的平均長度仍是不變的。因此我們可以在不更動編碼內容及其平均長度的情形下，找到一組索引指定使疊代解碼能更有效的運用訊源冗息，進一步降低位元及索引錯誤率。

透過以上的數據及討論，我們注意到實驗一考慮的是圖 2.1 之訊源編解碼架構，主要是探討不同的可變長度碼在無疊代且沒有通道編碼保護的情形下自身的傳輸效能。然而我們看到雙向性可變長度碼可以比霍夫曼碼取得更好的強健性能。然而由表 5.1 可以看出相較於霍夫曼碼，RVLC₁ 為了符合前置與後置條件其平均長度較長，相同的 RVLC₂ 為了滿足 $d_f \geq 2$ 而付出了更多的傳輸資源，顯示不同的設計理念透過犧牲些許的壓縮效益，將可提昇其雜訊對抗能力。再進一步考慮疊代系統，經過實驗了解到由於可變長度碼是基於特定機率分佈的訊源的編碼機制，故其殘餘冗息可用於在疊代系統中，使不同的編碼設計有相當差距的效能。因此若能善用這樣的冗息，即可設計更好的碼書。在實驗三我們討論了在不更動碼書的內容及平均長度下，經過

二位元置換演算法進行索引指定的設計之後，可以更進一步得到更好的編碼組合，這也表示了可以在不變動任何硬體設計及不增加資源耗費的情形下，利用可變長度碼及疊代解碼系統上特殊的額外資訊轉換特性，加以改良其效能並使傳輸更加強健。

索引機率	Huffman Code	RVLC ₁	RVLC ₂	RVLC ₃
0.1017	[1, 0, 0]	[0, 0, 0]	[0, 0, 0]	[1, 1, 1]
0.10106	[1, 0, 1]	[1, 1, 1]	[1, 1, 1]	[0, 0, 0]
0.10044	[1, 1, 0]	[0, 1, 0]	[0, 1, 0, 1]	[0, 0, 1, 1]
0.0976	[1, 1, 1]	[1, 0, 1]	[1, 0, 1, 0]	[1, 1, 0, 0]
0.08969	[0, 0, 0, 0]	[0, 1, 1, 0]	[0, 1, 1, 0]	[0, 1, 0, 1]
0.08928	[0, 0, 0, 1]	[1, 0, 0, 1]	[1, 0, 0, 1]	[1, 0, 1, 0]
0.07649	[0, 0, 1, 1]	[0, 0, 1, 1]	[0, 0, 1, 1]	[0, 1, 1, 0]
0.07514	[0, 1, 0, 0]	[1, 1, 0, 0]	[1, 1, 0, 0]	[1, 0, 0, 1]
0.06241	[0, 1, 1, 0]	[0, 0, 1, 0, 0]	[0, 0, 1, 0, 0]	[1, 0, 0, 0, 1]
0.05777	[0, 1, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]
0.04287	[0, 0, 1, 0, 0]	[0, 1, 1, 1, 0]	[0, 1, 1, 1, 0]	[0, 0, 1, 0, 0]
0.04113	[0, 0, 1, 0, 1]	[1, 0, 0, 0, 1]	[1, 0, 0, 0, 1]	[0, 1, 1, 1, 0]
0.02515	[0, 1, 0, 1, 1]	[0, 1, 1, 1, 1, 0]	[0, 1, 0, 0, 1, 0]	[0, 1, 0, 0, 1, 0]
0.02314	[0, 1, 0, 1, 0, 0]	[1, 0, 0, 0, 0, 1]	[1, 0, 1, 1, 0, 1]	[1, 0, 1, 1, 0, 1]
0.00865	[0, 1, 0, 1, 0, 1, 0]	[0, 0, 1, 0, 1, 0, 0]	[0, 1, 0, 0, 0, 1, 0]	[0, 1, 0, 0, 0, 1, 0]
0.00748	[0, 1, 0, 1, 0, 1, 1]	[1, 1, 0, 1, 0, 1, 1]	[1, 0, 1, 1, 1, 0, 1]	[1, 0, 1, 1, 1, 0, 1]
平均長度	3.82817	3.94835	4.14639	4.14639

表 5.1 不同可變長度碼之碼書

第六章 結論與未來展望

本論文主要探討可變長度碼在疊代訊源通道解碼機制上的設計與應用。首先探討可變長度碼在位元層級上的軟性訊源解碼機制，並利用其特殊的籬柵圖結構，運用 BCJR 演算法計算其後驗機率。此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的額外訊息，增加系統的強健性。可變長度碼因其設計目的不同而有各種編碼方式，相較於只滿足前置條件的霍夫曼碼，我們鎖定的雙向性可變長度碼具有較嚴謹的編碼條件，可以在平均長度與疊代解碼效能兩者之間取得較佳的平衡點。額外資訊轉換圖是模擬單一解碼器的輸入事前訊息及輸出額外訊息兩者間互消息量的變化關係，因此可視為有效分析疊代解碼效能的一項工具，利用這樣的分析，我們得以基於疊代系統的效能更進一步設計可變長度碼。我們提出的解決方案是利用疊代解碼輸出的額外資訊轉換特性，設計可變長度碼的索引指定。系統模擬證實，此方式可以在不變動硬體設計及不增加資源成本的情形下，加以改良並使傳輸效能更加強健。

現今行動通訊中，多媒體服務將成為趨勢，因此整合各種多媒體資料包括音樂、影像、語音…等並經由無線通訊傳輸將成為熱門議

題。而許多資料已有既定的壓縮編碼格式，例如 JPEG 為影像檔最為普遍的壓縮格式，一般使用霍夫曼碼作為其熵編碼(Entropy Coding)規格。然而經由我們的實驗證實，霍夫曼碼無法有效發揮疊代訊源通道解碼的效能。因此當疊代解碼系統運用在多媒體傳輸時，為了使訊源間的殘餘冗息得以充分利用，需要對可變長度碼做適當的調整。然而具有更正錯誤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再進一步探討的編碼設計目標。除此之外，我們在模擬中所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。在考慮更複雜的通道同時，將會使運算量大幅增加，除了通道考量，籬柵結構的 BCJR 演算法以及疊代解碼的計算量，將使解碼效率面臨相當大的考驗。因此如何克服龐大的運算，使得多媒體在傳輸上可以順利的解碼提供給使用者，將是重要議題。

附錄 A S-random 交錯器

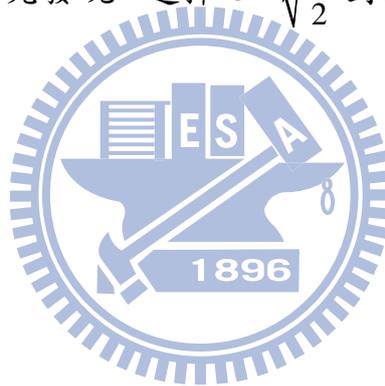
在系統中，可變長度碼以一個具有 K 個符號所組成的封包作為單位來傳輸，因此一個封包的位元總長度都不相同。而交錯器的使用目的希望能達到交錯後的資料相鄰之前的關聯性越小越好，因此我們選擇具良好交錯效果的 S-random 交錯器。其目標是要使交錯後的碼串像隨機選取的一樣，使相鄰的位元相關性最少。然而為了確保隨機選取的資料不會因巧合而發生高相關性的問題，因此設定一個數字 S ，使交錯後的碼串每個相鄰的位元都來自原始位置至少相差 S 以上的地方。

交錯器步驟：

1. 假設 S 為一常數。先空出與封包長度 N 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 $S = \sqrt{\frac{N}{2}}$ 對於資料的交錯可以達到很好的效果。



參考文獻

- [1] R. Thobaben and J. Kliewer, "Low-Complexity Iterative Joint Source-Channel Decoding for Variable-Length Encoded Markov Sources," *IEEE Trans. Commun.*, vol. 53, no. 12, pp. 2054–2064, Dec. 2005.
- [2] 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.
- [3] 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.
- [4] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449-2457, Sep. 1995.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] J. Kliewer and R. Thobaben, "Combining FEC and optimal soft-input source decoding for the reliable transmission of correlated variable-length encoded signals," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 83–91.

- [10] 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.
- [11] 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.
- [12] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel-decoding," in *Proc. IEEE Data Compression Conf.*, 2001, pp. 273–282.
- [13] M. Adrat and P. Vary, "Iterative source-channel decoding: improved system design using EXIT charts," *EURASIP J. Appl. Signal Processin*, pp. 928–941, May 2005.
- [14] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Trans. Commun.*, vol. 38, no. 12, pp. 2147–2158, 1990.
- [15] V. B. Balakirsky, "Joint source-channel coding with variable length codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Ulm, Germany, Jun. 1997, p.419
- [16] C. W. Tsai and J. L. Wu, "On constructing the Huffman-code based reversible variable length codes," *IEEE Trans. Commun.*, vol. 49, pp. 1506–1509, Sept. 2001.
- [17] K. Lakovic´ and J. Villasenor, "On design of error-correcting reversible variable length codes," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 337–339, Aug. 2002.
- [18] 潘彥璋,「基於索引層級的疊代訊源通解碼機制」,國立交通大學碩士論文,民國九十七年。
- [19] 陳亞民,「渦輪原則在合併訊源通道編碼之研究」,國立交通大學碩士論文,民國九十八年。