

國立交通大學

電信工程研究所

碩士論文



基於額外資訊相關性的合併訊源通道編碼機制
A Joint Source-Channel Coder Design
Based on The Correlation of Extrinsic Information

研究生：葉遠呈
指導教授：張文輝 博士

中華民國九十九年六月

基於額外資訊相關性的合併訊源通道編碼機制
A Joint Source-Channel Coder Design
Based on The Correlation of Extrinsic Information

研究生：葉遠呈

Student：Yuan-Cheng Yeh

指導教授：張文輝

Advisor：Wen-Whei Chang



Computer and Information Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

基於額外資訊相關性的合併訊源通道編碼機制

學生：葉遠呈

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

在數位多媒體通訊系統裡，合併訊源通道編碼演算法可有效提升系統效能的強健性。本論文將針對位元及索引層級的疊代訊源通道解碼機制，探討系統傳送端兩項關鍵元件的設計。索引指定，先觀察疊代解碼過程中額外資訊相關性的實質變化，再配合二位元置換演算法以設計最佳的索引指定。至於交錯器的設計，我們從接收端的角度，提出適用於疊代訊源通道解碼的交錯器。最後經由附加性白高斯雜訊通道的系統模擬，驗證我們提出的索引指定與交錯器，有效提升了疊代解碼的效益與整體系統的強健性。

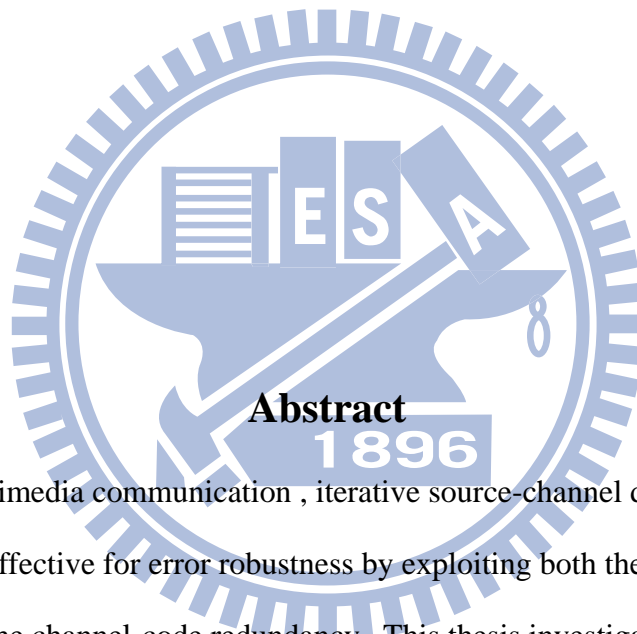
A Joint Source-Channel Coder Design

Based on The Correlation of Extrinsic Information

Student : Yuan-Cheng Yeh Advisor : Dr. Wen-Whei Chang

Institute of Communications Engineering

National Chiao Tung University



Abstract

In digital multimedia communication , iterative source-channel decoding(ISCDC) has been shown effective for error robustness by exploiting both the source residual redundancy and the channel-code redundancy . This thesis investigates the encode design for use in conjunction with bit-level as well as index-level ISCDC . Firstly , we compare different index assignment schemes in terms of the correlation of extrinsic information resulting form the ISCDC process .Also proposed is a correlation-based index assignment scheme . Secondly , we focus on the interleaver design based on the performance of iterative decoding . Simulation results indicate that the combined use of correlation-based index assignment and interleaver can improve the error robustness for AWGN channel.

誌謝

一篇論文的完成，所代表的不只是個人努力的成果，它同時蘊含許多人的心力與支持。首先要特別感謝我的指導教授張文輝老師無私的教導，讓我在兩年的研究生涯裡培養研究應有的精神與態度。同時也要感謝實驗室朋友們在課業與研究上的扶持，尤其是博班學長吳俊鋒與李承龍所提供的建議使我獲益良多。最後，要感謝我的家人在這兩年來的支持，讓我能心無旁騖完成學業與研究。



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	viii
第一章 緒論	1
1.1 研究動機與方向.....	1
1.2 章節概要.....	3
第二章 疊代訊源通道解碼機制	5
2.1 位元層級編碼架構.....	6
2.2 位元層級疊代解碼演算法	7
2.2.1 通道解碼演算法.....	9
2.2.2 軟性位元訊源解碼器.....	13
2.3 索引層級編碼架構.....	16
2.4 索引層級疊代解碼演算法	17
2.4.1 索引層級 BCJR 演算法.....	18
2.4.2 索引層級訊源解碼器.....	21

第三章 索引指定的最佳化設計	22
3.1 基於額外資訊轉換圖的索引指定設計	23
3.1.1 額外資訊轉換圖	23
3.1.2 基於 EXIT Chart 的索引指定設計	29
3.1.3 基於 EXIT Chart 索引指定設計之缺陷	32
3.2 基於資訊相關性的索引指定設計	33
第四章 交錯器的最佳化設計	40
4.1 疊代解碼效益	41
4.2 基於資訊關聯性的交錯器設計	43
4.2.1 通道解碼器輸入資訊關聯性的分析	43
4.2.2 基於疊代解碼效益的交錯器設計	50
4.3 訊源解碼器資訊關聯性的分析	51
4.4 交錯器的最佳化設計	54
第五章 實驗模擬與結果分析	59
5.1 實驗環境設定	59
5.2 索引指定設計在 ISCD 效能之比較	60
5.3 交錯器設計在 ISCD 之效能比較	70
第六章 結論與未來展望	76
參考文獻	78

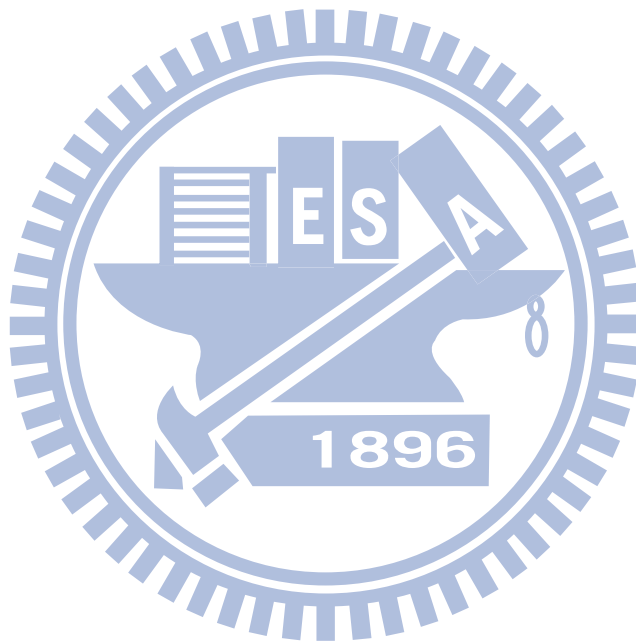
圖目錄

圖 2.1 位元層級疊代編碼架構圖	6
圖 2.2 位元層級疊代訊源通道解碼架構圖	8
圖 2.3 索引層級疊代編碼架構圖	16
圖 2.4 索引層級疊代訊源解碼架構圖	17
圖 2.5 區段性籬柵圖 ($M = 2$)	18
圖 3.1 軟性輸入輸出解碼器	23
圖 3.2 額外資訊轉換圖(NBC) (a)通道解碼器 (b)訊源解碼器	26
圖 3.3 軌跡圖與額外資訊轉換圖(NBC)	28
圖 3.4 訊源解碼器額外資訊轉換圖 (a)NBC (b) Gray code	30
圖 3.5 額外資訊轉換圖與軌跡圖(NBC, 搭配區塊與隨機交錯器)	33
圖 3.6 額外資訊相關特性圖 $\rho'(\tau)$ (a) NBC (b) Gray Code	37
圖 3.7 二位元置換演算法流程圖	39
圖 4.1 交錯器 Φ 與解交錯器 Φ^{-1}	41
圖 4.2 疊代解碼中輸入資訊之間關聯性流程圖	43
圖 4.3 $\rho_{SBSD}^{(n)}(L_{SBSD}^{(ext),(n)}(u_i), \tilde{u}_t(m))$ (NBC)	46
圖 4.4 $\rho_{SBSD}^{(n)}(L_{SBSD}^{(ext),(n)}(u_i), \tilde{u}_t(m))$ (表 3.1 中的 BIA_E)	49
圖 4.5 $\rho^{(n^+)}(L_{CD}^{(ext),(n^+)}(y_{\Phi(i_m)}^s), L_{CD}^{(ext),(n^+)}(y_i^s))$	52
圖 4.6 交錯器設計流程圖	58

圖 5.1 位元層級不同索引指定的效能比較(10 次疊代)	61
圖 5.2 BIA_C 的額外資訊相關特性圖($CSNR = -3 \text{ dB}$)	63
圖 5.3 不同索引指定之額外資訊相關特性比較 ($CSNR = -3 \text{ dB}$) ..	63
圖 5.4 額外資訊轉換圖比較	64
圖 5.5 索引層級兩種索引指定的效能比較(10 次疊代)	66
圖 5.6 相異索引指定的額外資訊轉換圖、軌跡圖之比較	67
圖 5.7 不同索引指定的資訊相關性變化圖之比較	68
圖 5.8 不同交錯器的效能比較 (第 10 次疊代)	70
圖 5.9 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_{148}), \tilde{y}_l^s)$	72
圖 5.10 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_{149}), \tilde{y}_l^s)$	72
圖 5.11 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(1)}(u_{150}), \tilde{y}_l^s)$	73
圖 5.12 不同交錯器效能之比較 (第 5 次疊代)	74
圖 5.13 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_{50}), \tilde{y}_l^s)$	75

表目錄

表 3.1 自然索引指定和基於 EXIT 設計的最佳化索引指定	31
表 5.1 位元層級不同索引指定的比較	61
表 5.2 索引層級不同索引指定的比較	66



第一章 緒論

1.1 研究動機與方向

在數位多媒體傳輸系統中，訊源編碼器與通道編碼器為系統中重要的組成元件。前者主要負責將來源訊號量化壓縮以減少傳送位元量，後者則加入人工冗息以保護傳送位元免於通道雜訊的干擾。傳統的通訊系統器設計是根據 C. E. Shannon 提出的分離定律(Shannon's separation theorem)[1]，分別針對訊源編碼器與通道編碼進行設計。該理論是基於兩項假設：設計訊源編碼器時假設通道編碼器可完全修正傳輸錯誤，而在設計通道編碼器時則假設訊源編碼器能完全壓縮移除輸入訊號間的冗息(redundancy)。然而因運算複雜度和整體延遲的諸多限制，使得系統的具體實現無法達到理論上的要求。有鑒於此，前人研究提出了合併訊源通道編碼(joint source-channel coding)[2]機制，利用訊源編碼器輸出資料間的殘餘冗息(residual redundancy)提升系統的強健性。相關研究包括訊源控制通道解碼(source-controlled channel decoding, SCCD)[3]以及軟性位元訊源解碼(softbit source decoding, SBSDD)[4]。而近幾年來研究更致力於 SCCD 與 SBSDD 的整合，並且進一步根據渦輪準則(turbo-principle)而發展出更先進的疊代訊源通道解碼機制(iterative

source-channel decoding, ISCD)[5][6][7][8][9]。主要構想是於通道與訊源兩解碼器間建立疊代架構，交換彼此輸出的額外資訊 (extrinsic information)，以提升系統的強健效能。

在疊代訊源通道解碼的相關研究中，額外資訊轉換圖 (extrinsic information transfer chart, EXIT chart) [10][11][12] 已成為理論分析疊代解碼效能的重要工具。其主要原理為尋找通道與訊源兩解碼器的額外資訊轉換曲線之交點，停止點，當停止點越接近理論的上限值時代表該系統的解碼效能越好。隨後 Adrat 觀察到索引指定 (index assignment) 的改變會使停止點的位置產生明顯變化，據以發展出基於額外資訊轉換圖的索引指定機制 [13]，並成功的讓系統效能獲得明顯的提升。其缺點是受限於額外資訊轉換圖基本假設的要求，無法考量整個系統來進行最佳的索引指定設計。因此本文稍後將針對此一缺點進行推論，同時從實際疊代解碼效能的角度切入，探討額外資訊相關性與疊代解碼效能之間的關係。最後我們將提出一個基於額外資訊相關性的索引指定設計方式，期望能找出最匹配於 ISCD 系統的最佳索引指定。

另一方面，我們也對交錯器 (interleaver) 的設計進行探討。在渦輪碼的通道編碼研究領域裡，交錯器主要是配置於兩個並聯的通道編碼器之間，負責打散輸入資料序列，避免兩通道編碼器同時產生低

權重(low weight)的碼字(codeword)。因此渦輪碼交錯器設計的目的主要是避免低權重碼字的產生，改善距離譜(distance spectrum)，例如[14][15][16][17][18]。而在 ISCD 中，常使用的交錯器有區塊交錯器(block interleaver)及隨機交錯器(S-random interleaver)。在系統傳輸端，它們處於訊源和通道兩編碼器中間，置換訊源編碼器輸出資料的位置以削減資料中殘餘的冗息，再送到後級的通道編碼器，其目的是讓訊源與通道兩編碼器的輸入資料儘可能夠滿足不相關的假設。然而區塊與隨機交錯器只考慮到系統傳送端，並未將接收端的疊代解碼架構納入設計考量因素。因此有別於傳統觀點，本研究將引入[19]的概念，從接收端系統出發，分析解碼演算法與疊代解碼時資訊關聯性的變化，並據以設計適用於疊代訊源通道解碼系統的交錯器。

1.2 章節概要

第二章將介紹位元及索引層級的疊代訊源通道解碼機制，並針對通道與訊源解碼演算法進行探討。第三章則先介紹額外資訊轉換圖和其在索引指定上的應用，接下來再點出此設計方式的缺點並加以改良。在第四章裡將介紹[19]的交錯器設計，並衍生此概念而發展出適用於疊代訊源通道解碼系統的交錯器設計。第五章進行實驗模擬與分

析。第六章則為結論與未來展望。



第二章 疊代訊源通道解碼機制

20 世紀末，由 Berrou 等人所提出的渦輪碼(Turbo code)[20]不但具有高效能的更錯能力外，其疊代解碼概念更是深深影響往後通訊的發展，疊代訊源通道解碼(ISCD)便是其中之一。ISCD 依據渦輪準則(turbo principle)在通道解碼器與訊源解碼器兩者之間建立起可疊代解碼架構，讓兩解碼器可交換彼此的額外資訊(extrinsic information)，有效的提升重建信號的品質。然而在傳統的 ISCD 中，解碼器之間的額外資訊交換仍屬於位元層級(bit-level)，其通道解碼器輸出的額外資訊必須先從位元層級轉到索引層級(index-level)後，才能作為訊源解碼器的事前額外資訊；反之，訊源解碼器輸出的額外資訊則要從索引層級轉到位元層級後，才能送到通道解碼器的輸入端。問題是在這轉換的過程中，往往會有許多資訊從中流失，造成系統解碼效能未能達到最佳的狀態。為此，[9]提出基於索引層級的 ISCD (Index-Level ISCD)架構，來避免位元與索引層級轉換過程中資訊的流失，並且已被證實這樣的系統設計能擁有比傳統疊代解碼更好的解碼效能。接下來，本章的前兩節將介紹位元層級疊代解碼傳送端與接收端的架構，後兩小節則介紹索引層級疊代解碼傳送端與接收端的架構，並且對每個解碼器所使用的演算法進行描述。

2.1 位元層級編碼架構

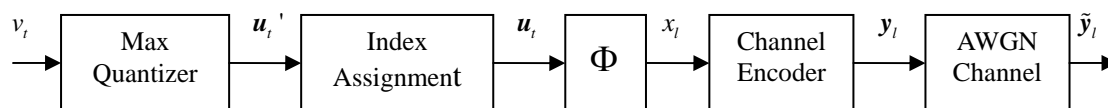


圖 2.1 位元層級疊代編碼架構圖

考慮現有一傳輸架構如圖 2.1 所示， v_t 為一個連續信號在時間點 t 被取樣的輸入信號。經由 M 位元的純量量化器處理後， v_t 被對應到最接近碼字的索引值 $u_t' \in \{0, 1, \dots, 2^M - 1\}$ ，接著再根據索引指定 (index assignment) 機制將此索引值對映至一組長度為 M 位元向量 $u_t = \{u_t(1), \dots, u_t(m), \dots, u_t(M)\}$ 來表示。而交錯器 Φ 則是針對一個時間間隔 T 內的索引集合 $U_1^T = \{u_1, u_2, \dots, u_T\}$ ，將其位元組合的位置重新排列成 $x_1^L = \{x_1, x_2, \dots, x_L\}$ 以削去相鄰位元之間的相關性。因此交錯器的長度 $L = T * M$ 且 $x_l = u_t(\lambda), l = 1, 2, \dots, L$ 。接下來為了提升傳送資料的正確性， x_1^L 被送入通道編碼器進行編碼。在本文中，通道編碼器採用由 J 個暫存器組成且碼率 (code rate) 為 $1/2$ 的遞迴系統迴旋碼 (recursive systematic convolutional code) 來實現。通道編碼器以即時的方式連續對輸入位元 x_l ，產生輸出向量 $y_l = \{y_l^s, y_l^p\}$ ， $l = 1, 2, \dots, L$ 。其中 $y_l^s = x_l$ 為系統位元可提供給解碼端做系統效能的評估；同位位元 y_l^p 則扮演保護系統位元的角色。特別要注意的是，我們假設了交錯器長度

L 遠大於編碼器暫存器數目 J ，所以編碼暫存器歸零所需的終止位元不被計入 y_l 中。最後， y_l 被二進制相移鍵控(binary phase shift keying)調變後，位元"0" $\rightarrow +1$ 、位元"1" $\rightarrow -1$ ，送入無記憶可加性白色高斯雜訊通道後獲得信號 \tilde{y}_l ，其通道轉換機率(channel transition probability)可表示成

$$p(\tilde{y}_l | y_l) = \left(\sqrt{\frac{E_s}{\pi N_0}} \right)^2 e^{-\frac{E_s}{N_0} \|\tilde{y}_l - y_l\|^2} \quad (2.1)$$

其中 E_s 為傳送信號能量，通道雜訊變異數 $\sigma^2 = \frac{N_0}{2E_s}$ ， $\|\cdot\|$ 為範數(norm)。

2.2 位元層級疊代解碼演算法

根據渦輪準則，系統進行疊代程序(iterative processing)必須滿足下列要求：

1. 組成區塊的連結
2. 交錯器
3. 軟性輸入/輸出程序(Soft-In-Soft-Out process)
4. 額外資訊(extrinsic information)的交換

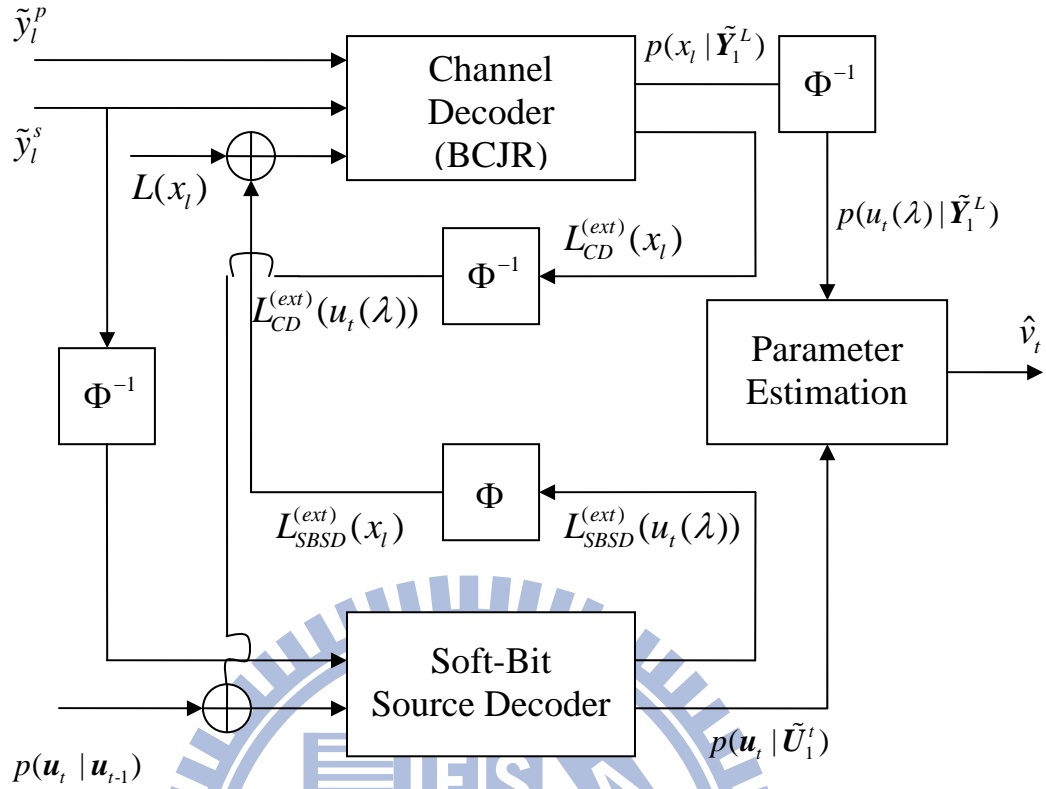


圖 2.2 位元層級疊代訊源通道解碼架構圖

因此接收端的通道解碼器與訊源解碼器，分別使用 BCJR 演算法[21]和軟性位元訊源解碼器(SBSD)[4]來實現，並且在兩者之間以交錯器/解交錯器連結，如圖 2.2 所示。當接收端收到 \tilde{y}_i 後先由通道編碼器進行解碼動作，而解碼後所產生的額外資訊 $L_{CD}^{(ext)}(x_i)$ 則透過交錯器形成 $L_{CD}^{(ext)}(u_i(\lambda))$ 傳到訊源解碼器的輸入端，形成第 0^+ 次疊代。接下來進行第 1 次疊代時，訊源解碼器再整合 $L_{CD}^{(ext)}(u_i(\lambda))$ 與其他輸入資訊計算出訊源額外資訊 $L_{SBSD}^{(ext)}(u_i(\lambda))$ ，隨後並由解交錯器作用得到 $L_{SBSD}^{(ext)}(x_i)$ ，做為第 1^+ 次疊代解碼時通道解碼器輸入資訊的一部分。如此反覆上述的

動作，兩解碼器不斷地交換彼此的額外資訊來執行解碼，系統將隨著疊代次數的上升得到更好的效能。而在後續的小節中，將分別介紹通道解碼器與訊源解碼器所使用的演算法。

2.2.1 通道解碼演算法

在通道解碼器中，最大後驗機率(maximum a posterior probability, MAP) 則採用 BCJR 演算法[21]來計算。它利用接收訊號 $\tilde{y}_l = \{\tilde{y}_l^s, \tilde{y}_l^p\}$ 、訊源額外資訊 $L_{SBSD}^{(ext)}(x_l)$ 、事前統計機率 $L(x_l)$ 與事先建立好的籬柵圖(trellis diagram)，來求得每個時間點的分枝計量(branch metric)、順向計量(forward metric)和反向計量(backward metric)，並據此計算出碼字位元 x_l 的後驗機率，如下

$$\begin{aligned}
 p(x_l = i | \tilde{Y}_1^L) &= \sum_{(s_{l-1}, s_l) \in \Omega_l^i} p(s_{l-1}, s_l | \tilde{Y}_1^L) \\
 &= \sum_{(s_{l-1}, s_l) \in \Omega_l^i} \frac{p(s_{l-1}, s_l, \tilde{Y}_1^L)}{p(\tilde{Y}_1^L)} = \sum_{(s_{l-1}, s_l) \in \Omega_l^i} C \cdot p(s_{l-1}, s_l, \tilde{Y}_1^L), i = 0, 1
 \end{aligned} \tag{2.2}$$

當中 Ω_l^i 代表籬柵圖上符合 $x_l = i$ 的所有狀態轉換 (s_{l-1}, s_l) 集合。接下來假設， s_{l-1} 已知的情況下 \tilde{Y}_1^{l-1} 與 s_l 、 \tilde{y}_l 互為獨立，且 s_l 已知的情況下 \tilde{Y}_{l+1}^L 與 s_{l-1} 、 \tilde{Y}_1^l 獨立。因此， $p(s_{l-1}, s_l, \tilde{Y}_1^L)$ 可簡化為：

$$\begin{aligned}
 p(s_{l-1}, s_l, \tilde{Y}_1^L) &= p(s_{l-1}, s_l, \tilde{Y}_1^l) \cdot p(\tilde{Y}_{l+1}^L | s_{l-1}, s_l, \tilde{Y}_1^l) \\
 &= p(s_{l-1}, \tilde{Y}_1^{l-1}) \cdot p(s_l, \tilde{y}_l | s_{l-1}, \tilde{Y}_1^{l-1}) \cdot p(\tilde{Y}_{l+1}^L | s_{l-1}, s_l, \tilde{Y}_1^l) \\
 &= p(s_{l-1}, \tilde{Y}_1^{l-1}) \cdot p(s_l, \tilde{y}_l | s_{l-1}) \cdot p(\tilde{Y}_{l+1}^L | s_l) \\
 &= \alpha_{l-1}(s_{l-1}) \cdot \gamma_l(s_{l-1}, s_l) \cdot \beta_l(s_l)
 \end{aligned} \tag{2.3}$$

其中順向計量 $\alpha_l(s_l)$ 、反向計量 $\beta_l(s_l)$ 可分別由以下遞迴公式計算

$$\begin{aligned}
\alpha_l(s_l) &= p(s_l, \tilde{\mathbf{Y}}_1^l) = \sum_{s_{l-1}} p(s_{l-1}, s_l, \tilde{\mathbf{Y}}_1^l) \\
&= \sum_{s_{l-1}} p(s_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \cdot p(s_l, \tilde{\mathbf{y}}_l | s_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \\
&= \sum_{s_{l-1}} p(s_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \cdot p(s_l, \tilde{\mathbf{y}}_l | s_{l-1}) \\
&= \sum_{s_{l-1}} \alpha_{l-1}(s_{l-1}) \cdot \gamma_l(s_{l-1}, s_l)
\end{aligned} \tag{2.4}$$

$$\begin{aligned}
\beta_l(s_l) &= p(\tilde{\mathbf{Y}}_{l+1}^L | s_l) = \sum_{s_{l+1}} p(s_{l+1}, \tilde{\mathbf{Y}}_{l+1}^L | s_l) \\
&= \sum_{s_{l+1}} p(\tilde{\mathbf{Y}}_{l+2}^L | \tilde{\mathbf{y}}_{l+1}, s_l, s_{l+1}) \cdot p(\tilde{\mathbf{y}}_{l+1}, s_l, s_{l+1}) \cdot \frac{1}{p(s_l)} \\
&= \sum_{s_{l+1}} p(\tilde{\mathbf{Y}}_{l+2}^L | s_{l+1}) \cdot p(\tilde{\mathbf{y}}_{l+1}, s_{l+1} | s_l) \\
&= \sum_{s_{l+1}} \beta_{l+1}(s_{l+1}) \cdot \gamma_{l+1}(s_l, s_{l+1})
\end{aligned} \tag{2.5}$$

藉由假設 $\tilde{\mathbf{y}}_l^p$ 與 $\tilde{\mathbf{y}}_l^s$ 獨立，可算得分枝計量

$$\begin{aligned}
\gamma_l(s_{l-1}, s_l) &= p(s_l, \tilde{\mathbf{y}}_l | s_{l-1}) = p(x_l = i, s_l, \tilde{\mathbf{y}}_l | s_{l-1}) \\
&= p(\tilde{\mathbf{y}}_l | x_l = i, s_{l-1}, s_l) \cdot p(s_l | s_{l-1}, x_l = i) \cdot p(x_l = i) \\
&= p(\tilde{\mathbf{y}}_l^p | x_l = i, s_{l-1}, s_l) \cdot p(\tilde{\mathbf{y}}_l^s | x_l = i, s_{l-1}, s_l) \cdot \\
&\quad p(s_l | s_{l-1}, x_l = i) \cdot p(x_l = i) \\
&= p(\tilde{\mathbf{y}}_l^p | x_l = i, s_{l-1}) \cdot p(\tilde{\mathbf{y}}_l^s | x_l = i) \cdot p(s_l | s_{l-1}, x_l = i) \cdot p(x_l = i)
\end{aligned} \tag{2.6}$$

其中 $p(x_l = i)$ 為事前資訊； $p(s_l | s_{l-1}, x_l = i) = 1$ ，若 $(s_{l-1}, s_l) \in \Omega_4^i$ ，反之

則為 0。接下來將 (2.1) 代入 (2.6)，則

$$\gamma_l(s_{l-1}, s_l) = p(x_l = i) \cdot \frac{E_s}{\pi N_0} \cdot \exp\left\{-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_l - \mathbf{y}_l\|^2\right\} \tag{2.7}$$

最後，後驗機率可被寫成

$$\begin{aligned}
p(x_l = i | \tilde{\mathbf{Y}}_1^L) &= \sum_{(s_{l-1}, s_l) \in \Omega_l^i} C \cdot p(s_{l-1}, s_l, \tilde{\mathbf{Y}}_1^L) \\
&= \sum_{(s_{l-1}, s_l) \in \Omega_l^i} C \cdot \alpha_{l-1}(s_{l-1}) \cdot \gamma_l(s_{l-1}, s_l) \cdot \beta_l(s_l) \\
&= C \cdot p(x_l = i) \cdot p(\tilde{y}_l^s | x_l = i) \\
&\quad \cdot \sum_{(s_{l-1}, s_l) \in \Omega_l^i} \alpha_{l-1}(s_{l-1}) \cdot \beta_l(s_l) \cdot p(\tilde{y}_l^p | x_l = i, s_{l-1}) \\
&= C \cdot p(x_l = i) \cdot p_c(x_l = i) \cdot p_{CD}^{(ext)}(x_l = i)
\end{aligned} \tag{2.8}$$

其中 $p_c(x_l = i) = p(\tilde{y}_l^s | x_l = i)$ ，且通道額外資訊為

$$p_{CD}^{(ext)}(x_l = i) = \sum_{(s_{l-1}, s_l) \in \Omega_l^i} \alpha_{l-1}(s_{l-1}) \cdot \beta_l(s_l) \cdot p(\tilde{y}_l^p | x_l = i, s_{l-1}) \tag{2.10}$$

特別強調的是在尚未啟動疊代時，事前資訊 $p_a(x_l = i) = p(x_l = i)$ ；經過一次疊代解碼後 $p_a(x_l = i) = p(x_l = i) \cdot p_{SBS D}^{(ext)}(x_l = i)$ ，故疊代解碼過程中

$$p(x_l = i | \tilde{\mathbf{Y}}_1^L) = C \cdot p(x_l = i) \cdot p_{SBS D}^{(ext)}(x_l = i) \cdot p_c(x_l = i) \cdot p_{CD}^{(ext)}(x_l = i) \tag{2.11}$$

最後使用賈可賓對數函數(Jacobian logarithm function)來降低運算量且避免機率值過小的數值精準度問題。定義賈可賓對數函數：

$$\log(e^x + e^y) = \max\{x, y\} + \log(1 + e^{-|x-y|}) \tag{2.12}$$

對公式(2.8)使用 Jacobian 對數函數，於是

$$\begin{aligned}
L(x_l | \tilde{\mathbf{Y}}_1^L) &= \ln \frac{p(x_l = 0 | \tilde{\mathbf{Y}}_1^L)}{p(x_l = 1 | \tilde{\mathbf{Y}}_1^L)} \\
&= L_a^{(ext)}(x_l) + L_c(x_l) + L_{CD}^{(ext)}(x_l)
\end{aligned} \tag{2.13}$$

$$\text{其中事前資訊對數值 } L_a(x_l) = \ln \left[\frac{p(x_l = 0)}{p(x_l = 1)} \right], \tag{2.14}$$

$$\text{通道相關資訊對數值 } L_c(x_l) = \ln \left[\frac{p(\tilde{x}_l | x_l = 0)}{p(\tilde{x}_l | x_l = 1)} \right], \quad (2.15)$$

通道額外資訊對數值

$$\begin{aligned} L_{CD}^{(ext)}(x_l) &= \ln \left[\frac{p_{CD}^{(ext)}(x_l = 0)}{p_{CD}^{(ext)}(x_l = 1)} \right] \\ &= \max_{(s_{l-1}, s_s) \in \Omega_0^*} \{ \hat{\alpha}_{l-1}(s_{l-1}) + \hat{\beta}_l(s_l) + \ln p(\tilde{y}_l^p | x_l = 0, s_{l-1}) \} \\ &\quad - \max_{(s_{l-1}, s_s) \in \Omega_1^*} \{ \hat{\alpha}_{l-1}(s_{l-1}) + \hat{\beta}_l(s_l) + \ln p(\tilde{y}_l^p | x_l = 1, s_{l-1}) \} \end{aligned} \quad (2.16)$$

$$\hat{\alpha}_{l-1}(s_l) = \ln \alpha_{l-1}(s_l) = \max_{s_{l-2}}^* \left[\hat{\alpha}_{l-2}(s_{l-2}) + \hat{\gamma}_{l-1}(s_{l-2}, s_{l-1}) \right], \quad (2.17)$$

$$\hat{\beta}_l(s_l) = \ln \beta_l(s_l) = \max_{s_{l+1}}^* \left[\hat{\beta}_{l+1}(s_{l+1}) + \hat{\gamma}_{l+1}(s_l, s_{l+1}) \right], \quad (2.18)$$

$$\hat{\gamma}_l(s_{l-1}, s_l) = \ln \gamma_l(s_{l-1}, s_l) = \frac{(1 - 2 \cdot x_l) \ln p(x_l = i)}{2} + \frac{L_c}{2} \|\tilde{\mathbf{y}}_l - \mathbf{y}_l\|^2 \quad (2.19)$$

$$L_c = 4 \frac{E_s}{N_0}. \quad (2.20)$$

疊代過程中令 $L_a(x_l) = \ln \left[\frac{p(x_l = 0)}{p(x_l = 1)} \right] + L_{SBS D}^{(ext)}(x_l)$ ，並根據(2.13)可知通

道解碼器輸出的額外資訊對數值為

$$L_{CD}^{(ext)}(x_l) = L(x_l | \tilde{\mathbf{Y}}_1^L) - L_c(x_l) - L_{SBS D}^{(ext)}(x_l) - \ln \left[\frac{p(x_l = 0)}{p(x_l = 1)} \right] \quad (2.21)$$

2.2.2 軟性位元訊源解碼器

在傳送端原始輸入訊號取樣點之間是存有關聯性的，例如相鄰的聲音、影像訊號，並且即使經過量化與索引指定之後，相鄰兩個索引值仍具有關聯性。因此軟性位元訊源解碼器在解碼時，將使用事先訓練統計出的事前資訊來提高解碼的正確性。本節的軟性位元訊源解碼演算法[4]，利用索引 u_t 的一階事前訊息(1st-order a priori

knowledge)來計算其後驗機率

$$\begin{aligned} p(\mathbf{u}_t = n | \tilde{\mathbf{U}}_1^t) &= p(\mathbf{u}_t = n | \tilde{\mathbf{U}}_1^{t-1}, \tilde{\mathbf{u}}_t) \\ &= C \cdot p(\tilde{\mathbf{u}}_t / \mathbf{u}_t = n) \cdot \sum_{n'=0}^{2^M-1} [p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n')] \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1}) \end{aligned} \quad (2.22)$$

其中 $p(\tilde{\mathbf{u}}_t / \mathbf{u}_t = n)$ 是通道相關資訊，假設通道為無記憶性後可由下列公式計算

$$p(\tilde{\mathbf{u}}_t / \mathbf{u}_t = n) = \prod_{m=1}^M p(\tilde{u}_t(m) | u_t(m)) \quad (2.23)$$

而 $p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n')$ 則是一階事前索引統計機率。在疊代解碼過程中，額外考慮通道解碼器輸出的額外資訊，事前機率可由

$$p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot \prod_{m=1}^M p_{CD}^{(ext)}(u_t(m) | u_t = n) \text{ 來取代。}$$

訊源解碼器輸出額外資訊是屬於索引層級，若要將其提供給通道解碼器做為位元層級的事前資訊，必須先將訊源解碼演算法所算出的索引後驗機率轉成位元層級，其位元層級的後驗機率

$$p(u_t(\lambda) = i | \tilde{\mathbf{U}}_1^T) = \sum_{(\mathbf{u}_t | u_t(\lambda) = i)} p(\mathbf{u}_t | \tilde{\mathbf{U}}_1^T) \quad (2.24)$$

$$= C \cdot p(u_t(\lambda) = i) \cdot p(\tilde{u}_t(\lambda) | u_t(\lambda) = i) \cdot p_{CD}^{(ext)}(u_t(\lambda) = i) \cdot p_{SBS D}^{(ext)}(u_t(\lambda) = i)$$

其中 $i=0,1$ ， $(\mathbf{u}_t | u_t(\lambda) = i)$ 表示所有符合 $u_t(\lambda) = i$ 的索引值 \mathbf{u}_t ，

訊源額外資訊

$$p_{SBS D}^{(ext)}(u_t(\lambda) = i) = \sum_{(\mathbf{u}_t | u_t(\lambda) = i)} \left[\prod_{\substack{m=1 \\ m \neq \lambda}}^M p(\tilde{u}_t(m) | \mathbf{u}_t) \cdot p_{CD}^{(ext)}(u_t(m) | \mathbf{u}_t) \right] \cdot \left[\sum_{n'=0}^{2^M-1} p(\mathbf{u}_t^{(ext)}(\lambda) | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1}) \right] \quad (2.25)$$

$$= \sum_{(\mathbf{u}_t | u_t(\lambda) = i)} \Theta(\tilde{\mathbf{u}}_t^{(ext)} | \mathbf{u}_t^{(ext)}) \cdot \sum_{n'=0}^{2^M-1} p(\mathbf{u}_t^{(ext)}(\lambda) | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1})$$

其中

$$p(\mathbf{u}_t^{(ext)}(\lambda) | \mathbf{u}_{t-1} = n') = \frac{p(\mathbf{u}_t | \mathbf{u}_{t-1} = n')}{p(u_t(\lambda) = i)} \quad (2.26)$$

$$\Theta(\tilde{\mathbf{u}}_t^{(ext)} | \mathbf{u}_t^{(ext)}) = \exp \left\{ \sum_{\substack{m=1 \\ m \neq \lambda}}^M \frac{1 - 2u_t(\lambda)}{2} \left[L(\tilde{u}_t(\lambda) | u_t(\lambda)) + L_{CD}^{(ext)}(u_t(\lambda)) \right] \right\} \quad (2.27)$$

接下來如同上節，為了降低運算量且避免機率值過小的數值精準度問題，使用 Jacobian 對數函數計算對數相似值。由(2.25)，訊源額外資訊的對數相似值

$$L_{SBS D}^{(ext)}(u_t(\lambda)) = \ln \frac{p_{SBS D}^{(ext)}(u_t(\lambda) = 0)}{p_{SBS D}^{(ext)}(u_t(\lambda) = 1)} \quad (2.28)$$

$$= \ln \frac{\sum_{(\mathbf{u}_t | u_t(\lambda)=0)} \Theta(\tilde{\mathbf{u}}_t^{(ext)} | \mathbf{u}_t^{(ext)}) \cdot \sum_{n=0}^{2^M-1} p(\mathbf{u}_t^{(ext)}(\lambda) | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1})}{\sum_{(\mathbf{u}_t | u_t(\lambda)=1)} \Theta(\tilde{\mathbf{u}}_t^{(ext)} | \mathbf{u}_t^{(ext)}) \cdot \sum_{n=0}^{2^M-1} p(\mathbf{u}_t^{(ext)}(\lambda) | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1})}$$

最後進行參數重建時，可選擇對通道解碼器輸出或者訊源解碼器來進行重建與評估。若要對通道解碼器做評估，必須先將後驗對數相似值轉回後驗機率值，即

$$p(x_t = 0 | \tilde{\mathbf{Y}}_1^L) = \frac{\exp\{L(x_t | \tilde{\mathbf{Y}}_1^L)\}}{1 + \exp\{L(x_t | \tilde{\mathbf{Y}}_1^L)\}}, \quad (2.29)$$

$$p(x_t = 1 | \tilde{\mathbf{Y}}_1^L) = \frac{1}{1 + \exp\{L(x_t | \tilde{\mathbf{Y}}_1^L)\}} \quad (2.30)$$

由通道解碼器輸出的位元層級後驗機率 $p(x_t | \tilde{\mathbf{Y}}_1^L)$ 經解交錯器作用後可得 $p(u_t(\lambda) | \tilde{\mathbf{Y}}_1^L)$ ，所以通道解碼器輸出的索引層級後驗機率可從下式計算

$$p(\mathbf{u}_t | \tilde{\mathbf{U}}_1^t) = \prod_{\lambda=1}^M p(u_t(\lambda) | \tilde{\mathbf{U}}_1^t) \quad (2.31)$$

利用最小化均方差(minimum mean-squared error, MMSE)重建參數，

即

$$\tilde{v}_t = \sum_{n=0}^{2^M-1} p(\mathbf{u}_t = n | \tilde{\mathbf{U}}_1^t) \cdot v_t(n) \quad (2.32)$$

當中 $v_t(n)$ 為量化器輸出的量化值。

2.3 索引層級編碼架構

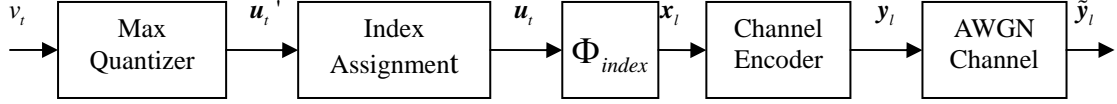


圖 2.3 索引層級疊代編碼架構圖

有別於位元層級的疊代解碼架構，索引層級的疊代編碼是採用一個索引層級的交錯器，它每次置換的單位為一個索引向量，並且不會改變索引向量裡組成位元的先後順序。故定義經交錯器作用後的資料 $\mathbf{X}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ ， $\mathbf{x}_l = \mathbf{u}_l$ ， $l = 1, 2, \dots, T$ 。而接下來的通道編碼器也如同 2.1 節所述，為一 (2,1) 的遞迴系統迴旋碼，然而在編碼上是必須以下列方式進行：假設現在一次送入一個索引向量進入通道編碼器進行編碼，則可得到 $2M$ 個碼字輸出序列， $\mathbf{Y}_1^T = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T\}$ ， $\mathbf{y}_l = \{\mathbf{y}_l^s, \mathbf{y}_l^p\} = \{\mathbf{x}_l, \mathbf{y}_l^p\}$ ，並且其整體碼率不會被改變，仍與位元層級序列的傳輸架構一樣。而經過無記憶可加性白色高斯雜訊通道作用後，其索引層級的通道轉換機率為

$$p(\tilde{\mathbf{y}}_l^s | \mathbf{x}_l) = \left(\sqrt{\frac{E_s}{\pi N_0}} \right)^M e^{-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_l^s - \mathbf{x}_l\|^2} \quad (2.33)$$

$$p(\tilde{\mathbf{y}}_l^p | \mathbf{x}_l) = \left(\sqrt{\frac{E_s}{\pi N_0}} \right)^M e^{-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_l^p - \mathbf{y}_l^p\|^2} \quad (2.34)$$

2.4 索引層級疊代解碼演算法

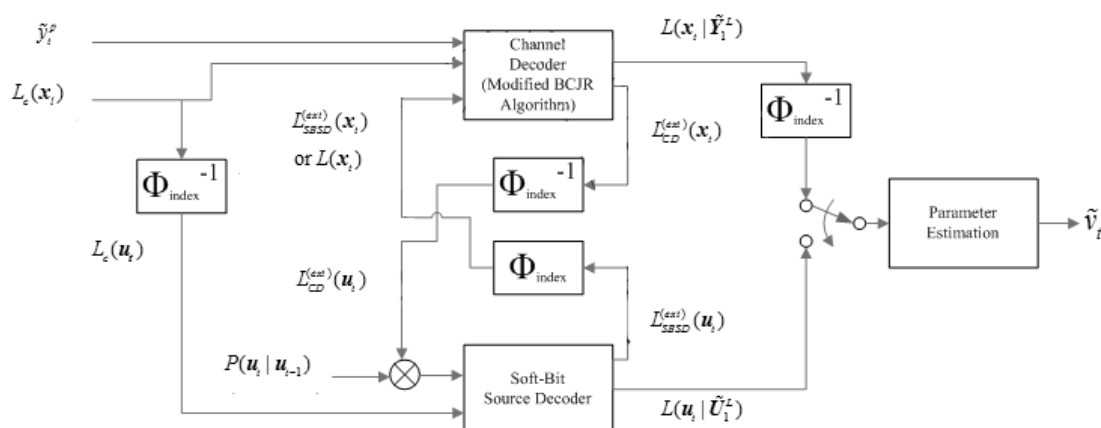


圖 2.4 索引層級疊代訊源解碼架構圖

額外資訊的交換是疊代訊源通道解碼當中重要的議題之一，然而輸入至兩解碼器的事前訊息層級卻是相異的：通道解碼器需要位元層級的事前資訊；相反的，訊源解碼器需要的則是索引層級的事前資訊。可是實際在疊代解碼的過程中，流動於兩解碼器之間的額外資訊卻又屬於位元層級，使得部分資訊在層級轉換時流失。以 2.2.2 節的為例，位元層級的額外資訊 $p_{CD}^{(ext)}(u_i(\lambda))$ 直接假設相互獨立，並相乘轉換至索引層級額外資訊 $p_{CD}^{(ext)}(\mathbf{u}_i)$ 後送入訊源解碼器當成事前訊息。因此為了改善這種現象，[9] 提出基於索引層級的疊代訊源通道解碼，利用改良性 BCJR 演算法(modified BCJR algorithm)讓通道解碼器所使用的演算法處於索引階層，並可直接輸出索引層級的額外資訊，避免層級轉換的問題發生。所以接下來我們將分別介紹索引層級的通道

解碼器與訊源解碼器。

2.4.1 索引層級 BCJR 演算法

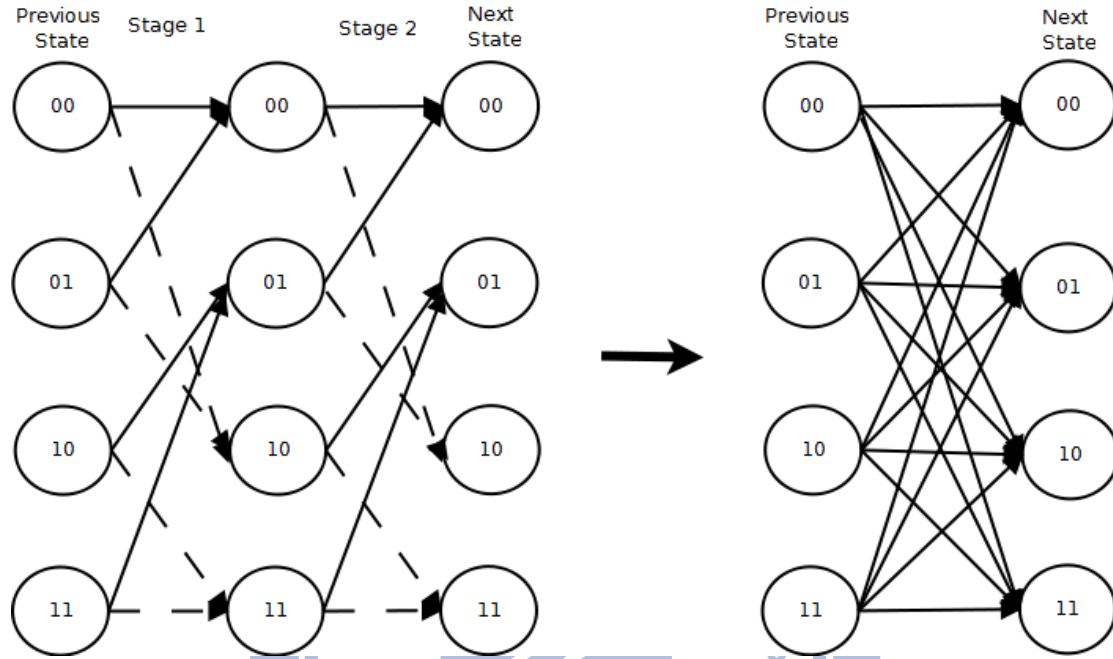


圖 2.5 區段性籬柵圖 ($M=2$)

改良性的 BCJR 演算法的最大特色在於每個時間點內可計算出索引層級信號 \tilde{y}_l 的後驗機率。而為了執行索引層級的運算，必須先改良位元層級籬柵圖使其成為區段性籬柵圖(sectionalized trellis diagram)，如圖 2.5 所示。接下來再利用區段性籬柵圖來計算後驗機率：

$$p(\mathbf{x}_l = n | \tilde{\mathbf{Y}}_1^L) = C \sum_{s_l} p(\mathbf{x}_l = n, s_l, \tilde{\mathbf{Y}}_1^L) = C \sum_{s_l} \alpha_l^x(n, s_l) \cdot \beta_l^x(n, s_l) \quad (2.35)$$

此處 $C = 1/p(\tilde{\mathbf{Y}}_1^L)$ ，順向計量 $\alpha_i^x(n, s_l) = p(\mathbf{x}_l = n, s_l, \tilde{\mathbf{Y}}_1^l)$ ，反向計量

$\beta_i^x(n, s_l) = p(\tilde{\mathbf{Y}}_{l+1}^L | \mathbf{x}_l = n, s_l, \tilde{\mathbf{Y}}_1^l)$ 。並且 $\alpha_i^x(n, s_l)$ 、 $\beta_i^x(i, s_l)$ 分別可由下列

式子遞迴公式求得

$$\begin{aligned}\alpha_i^x(i, s_l) &= \sum_{s_{l-1}} \sum_{n'} p(\mathbf{x}_l = n, s_l, \mathbf{x}_{l-1} = n', s_{l-1}, \tilde{\mathbf{y}}_l, \tilde{\mathbf{Y}}_1^{l-1}) \\ &= \sum_{s_{l-1}} \sum_{n'} \alpha_{l-1}^x(n', s_{l-1}) \gamma_{n,n'}(\tilde{\mathbf{y}}_l, s_l, s_{l-1})\end{aligned}\quad (2.36)$$

$$\begin{aligned}\beta_i^x(i, s_l) &= \sum_{s_{l+1}} \sum_{n'} p(\mathbf{x}_{l+1} = n', s_{l+1}, \tilde{\mathbf{Y}}_{l+1}^L | \mathbf{x}_l = n, s_l, \tilde{\mathbf{Y}}_1^l) \\ &= \sum_{s_{l+1}} \sum_{n'} \beta_{l+1}^x(n', s_{l+1}) \gamma_{n',n}(\tilde{\mathbf{y}}_{l+1}, s_{l+1}, s_l)\end{aligned}\quad (2.37)$$

$$\begin{aligned}\gamma_{n,n'}(\tilde{\mathbf{y}}_l, s_l, s_{l-1}) &= p(\mathbf{x}_l = n, s_l, \tilde{\mathbf{y}}_l | \mathbf{x}_{l-1} = n', s_{l-1}, \tilde{\mathbf{Y}}_1^{l-1}) \\ &= p(s_l | \mathbf{x}_{l-1} = n', s_{l-1}) \cdot p(\mathbf{x}_l = n | \mathbf{x}_{l-1} = n') \cdot p(\tilde{\mathbf{y}}_l | \mathbf{x}_l = n, s_l)\end{aligned}\quad (2.38)$$

接下來同樣利用 Jacobian 對數函數來降低運算量與提升數值精準

度，因此可定義順、反向計量的對數相似值

$$\begin{aligned}\hat{\alpha}_i^x(n, s_l) &= \log \alpha_i^x(n, s_l) \\ &= \max_{s_{l-1}}^* \{ \max_{n'}^* \{ \hat{\alpha}_{l-1}^x(n', s_{l-1}) + \hat{\gamma}_{n,n'}(\tilde{\mathbf{y}}_l, s_l, s_{l-1}) \} \}\end{aligned}\quad (2.39)$$

$$\begin{aligned}\hat{\beta}_i^x(n, s_l) &= \log \beta_i^x(n, s_l) \\ &= \max_{n'}^* \{ \max_{s_{l+1}}^* \{ \hat{\beta}_{l+1}^x(n', s_{l+1}) + \hat{\gamma}_{n',n}(\tilde{\mathbf{y}}_{l+1}, s_{l+1}, s_l) \} \}\end{aligned}\quad (2.40)$$

$$\begin{aligned}\hat{\gamma}_{n,n'}(\tilde{\mathbf{y}}_l, s_l, s_{l-1}) &= \log \gamma_{n,n'}(\tilde{\mathbf{y}}_l, s_l, s_{l-1}) \\ &= \log p(x_l = n) + \log p(s_l | \mathbf{x}_{l-1} = n', s_{l-1}) + \\ &\quad \log p(\tilde{\mathbf{x}}_l | \mathbf{x}_l = n) + \log p(\tilde{\mathbf{y}}_l^p | \mathbf{x}_l = n, s_l)\end{aligned}\quad (2.41)$$

$p(x_l = n)$ 為事前索引層級機率，在尚未執行疊代解碼前 $p(x_l = n)$ ，一

次疊代解碼後 $p(x_l = n) \leftarrow p_{SBS D}^{(ext)}(x_l = n)$ 。

而通道解碼器輸出的後驗對數相似值

$$L(\mathbf{x}_l = n | \tilde{\mathbf{Y}}_1^L) = \log \frac{p(\mathbf{x}_l = n | \tilde{\mathbf{Y}}_1^L)}{p(\mathbf{x}_l = 0 | \tilde{\mathbf{Y}}_1^L)} \quad (2.42)$$

$$= \max_{s_l}^* \{ \hat{\alpha}_l^x(n, s_l) + \hat{\beta}_l^x(n, s_l) \} - \max_{s_l}^* \{ \hat{\alpha}_l^x(0, s_l) + \hat{\beta}_l^x(0, s_l) \}$$

假設通道具無記憶性，並將(2.39)與(2.40)式帶入(2.42)中可計算出

$$L(\mathbf{x}_l = n | \tilde{\mathbf{Y}}_1^L) = L_a(\mathbf{x}_l = n) + L_c(\mathbf{x}_l = n) + L_{CD}^{(ext)}(\mathbf{x}_l = n) \quad (2.43)$$

其中

$$\text{事前資訊的對數相似值 } L_a(\mathbf{x}_l = n) = \log \left[\frac{p(\mathbf{x}_l = n)}{p(\mathbf{x}_l = 0)} \right], \quad (2.44)$$

$$\text{通道相關的對數相似值 } L_c(\mathbf{x}_l = n) = \log \left[\frac{p(\tilde{\mathbf{x}}_l | \mathbf{x}_l = n)}{p(\tilde{\mathbf{x}}_l | \mathbf{x}_l = 0)} \right], \quad (2.45)$$

通道編碼器輸出的額外資訊

$$L_{CD}^{(ext)}(\mathbf{x}_l = n) = \max_{s_l}^* \{ \hat{\beta}_l^x(n, s_l) + \log p(\tilde{\mathbf{y}}_l^p | \mathbf{x}_l = n, s_l) + \quad (2.46)$$

$$\max_{s_{l-1}}^* \{ \log p(s_l | \mathbf{x}_l = n, s_{l-1}) + \max_{n'}^* \{ \hat{\alpha}_{l-1}^x(n', s_{l-1}) \} \}$$

$$- \max_{s_l}^* \{ \hat{\beta}_l^x(0, s_l) + \log p(\tilde{\mathbf{y}}_l^p | \mathbf{x}_l = 0, s_l) \}$$

$$+ \max_{s_{l-1}}^* \{ \log p(s_l | \mathbf{x}_l = 0, s_{l-1}) \} + \max_{n'}^* \{ \hat{\alpha}_{l-1}^x(n', s_{l-1}) \} \}$$

最後在疊代解碼中，由(2.43)，通道額外資訊對數相似值

$$L_{CD}^{(ext)}(\mathbf{x}_l = n) = L(\mathbf{x}_l = n | \tilde{\mathbf{Y}}_1^L) - L_{SBS}^{(ext)}(\mathbf{x}_l = n) - L_c(\mathbf{x}_l = n) \quad (2.47)$$

2.4.2 索引層級訊源解碼器

由於通道解碼器使用索引層級 BCJR 演算法，因此訊源解碼器輸入端的額外資訊不用再經過不同層級間轉換動作，直接將此額外資訊取代事前資訊來計算後驗機率。因此在疊代解碼當中，一次事前統計資訊 $p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n')$ 直接以 $p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot p_{CD}^{(ext)}(\mathbf{u}_t = n)$ 取代，故 (2.22) 式可改寫成

$$\begin{aligned}
 p(\mathbf{u}_t = n | \tilde{\mathbf{U}}_1^t) &= C \cdot p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = n) \cdot \sum_{n'=0}^{2^M-1} p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot \\
 &\quad p_{CD}^{(ext)}(\mathbf{u}_t = n) \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1}) \\
 &= C \cdot p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = n) \cdot p_{CD}^{(ext)}(\mathbf{u}_t = n) \cdot \\
 &\quad \sum_{n'=0}^{2^M-1} p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1}) \\
 &= C \cdot p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = n) \cdot p_{CD}^{(ext)}(\mathbf{u}_t = n) \cdot p_{SBSD}^{(ext)}(\mathbf{u}_t = n)
 \end{aligned} \tag{2.48}$$

其中訊源解碼器輸出的額外資訊

$$p_{SBSD}^{(ext)}(\mathbf{u}_t = n) = \sum_{n'=0}^{2^M-1} p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot p(\mathbf{u}_{t-1} = n' | \tilde{\mathbf{U}}_1^{t-1}) \tag{2.49}$$

且其對數相似值為

$$L_{SBSD}^{(ext)}(\mathbf{u}_t = n) = \ln \left[\frac{p_{SBSD}^{(ext)}(\mathbf{u}_t = n)}{p_{SBSD}^{(ext)}(\mathbf{u}_t = 0)} \right] \tag{2.50}$$

第三章 索引指定的最佳化設計

索引指定(Index Assignment)機制是疊代訊源通道解碼中一項重要的研究課題，並且已有數種索引指定的設計方式被提出，例如[13]、[22]。[22]的設計方式主要是根據傳送端資料壓縮後殘餘的關聯性以及解碼端訊源解碼器(SBSD)的特性。而[13]的設計模式主要是在訊源通道解碼的額外資訊轉換圖(EXIT Chart)中，不同索引指定會影響訊源解碼器的額外資訊轉換圖，使之產生相異的曲度與收斂值大小之現象來選取最佳的索引指定，並由實驗證明[13]的索引指定設計會得到較好的解碼效能。但是依據使用額外資訊轉換圖交點來設計索引指定的方式，卻沒有考慮到兩項議題：第一，未考慮整個疊代訊源通道解碼系統的架構來進行設計，例如交錯器所帶來的影響，因此使得在某些情況下會得不到好的解碼效能；第二，這種設計方式並不適用於基於索引層級的疊代訊源通道解碼機制。故本章節將分成兩部分，首先介紹「基於額外資訊轉換圖索引指定的設計 (EXIT-based Index Assignment Design)」，接著我們將針對上述設計中未考慮的部分提出一個最佳化的索引指定設計方式，以期可依據系統的設定找出最佳的索引指定機制，達到更好的解碼效能。

3.1 基於額外資訊轉換圖的索引指定設計

本節先介紹額外資訊轉換圖，並分析通道/訊源解碼器的額外資訊轉換圖特性，接著介紹基於額外資訊轉換圖的索引指定設計。而此方法的缺失將在 3.2 節中做完整的說明。

3.1.1 額外資訊轉換圖

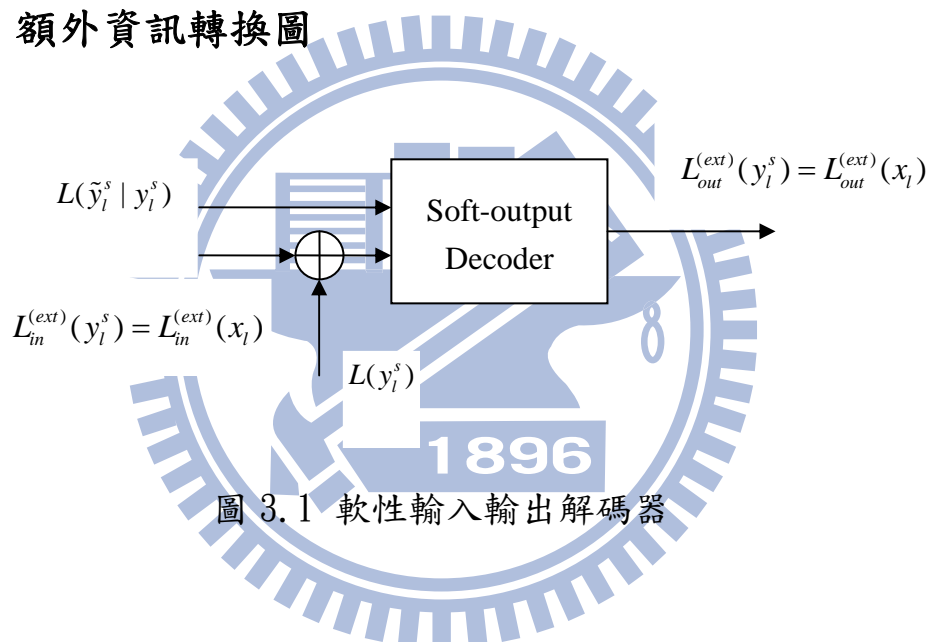


圖 3.1 軟性輸入輸出解碼器

在[10][11]提到額外資訊轉換圖的主要功能，是可用來評估疊代訊源通道解碼的效能並預測疊代收斂的次數，其原理主要是觀察解碼過程中事前及事後互消息(mutual information)的變化。首先，介紹互消息的定義，假設 p 、 q 是兩隨機變數，其熵值(entropy)分別為 $H(p)$ 、 $H(q)$ ，則 p 對 q 的互消息為 $I(p; q) = H(q) - H(q|p)$ ，代表觀察 p 的資訊之後使得 q 的不確定性所下降的程度。圖 3.1 代表疊代

解碼系統中的任一軟性輸入輸出的解碼器，其中 $L(\tilde{y}_i^s | y_i^s)$ 為通道相關的對數相似值(channel related LLR)， $L_{in}^{(ext)}(y_i^s) = L_{in}^{(ext)}(x)$ 是另一個解碼器輸出的額外資訊，並定義事前資訊 $L^{(apri)}(y_i^s) = L_{in}^{(ext)}(y_i^s) + L(y_i^s)$ ， $L(y_i^s)$ 代表事前消息的對數相似值，而 $L_{out}^{(ext)}(y_i^s) = L_{out}^{(ext)}(x)$ 則是此解碼器輸出的額外資訊。由於疊代解碼的過程中，任一解碼器都會利用另一個解碼器輸出的額外資訊來計算傳送資料 x_i 的事後機率，因此我們必須計算每個解碼器的事前互消息 $I^{[apri]}$ 與事後互消息 $I^{[ext]}$ 來評估該解碼器的效能。前者定義為解碼器輸入端的事前資訊 $L^{(apri)}(y_i^s)$ 對傳送端資料 y_i^s 的互消息，後者則是解碼器輸出端的額外資訊 $L_{out}^{(ext)}(y_i^s)$ 對傳送端資料 y_i^s 的互消息。

在計算 $I^{[apri]}$ 、 $I^{[ext]}$ 時，為了簡化整體的計算量與系統分析的複雜度，根據[10]做以下假設：

(a) 在交錯器長度夠大的情況下，經過幾次疊代後，事前資訊

$L^{(apri)}(y_i^s)$ 與通道相關資訊 $L(\tilde{y}_i^s | y_i^s)$ 不相關。

(b) 疊代次數越多，軟性解碼器輸出的額外資訊(未經交錯器)會越趨

近於高斯分佈(Gaussian Distribution)。

因此任一解碼器輸入端的事前訊息 $L^{(apri)}(y_i^s)$ ，可被模擬成平均值

$\mu_A = \sigma_A^2 / 2$ ，變異數 σ_A^2 的高斯隨機變數，並用此模擬值來計算該解

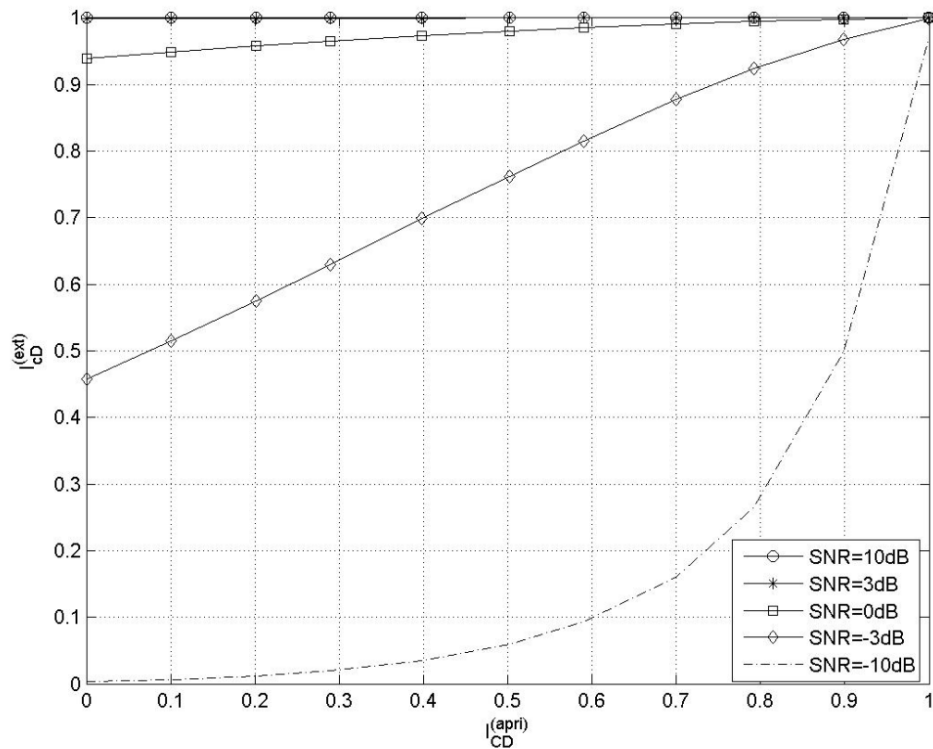
碼器的事前互消息 $I^{[apri]}$ 。之後再將通道相關資訊 $L(\tilde{y}_i^s | y_i^s)$ 與模擬產生

的 $L^{(apri)}(y_i^s)$ 引入解碼器的輸入端進行解碼，隨之得到的輸出結果即可用來求得事後互消息 $I^{[ext]}$ 。由第二章公式(2.1)可知通道相關資訊 $L(\tilde{y}_i^s | y_i^s)$ 為 E_s/N_0 的函數，故 $I^{[ext]}$ 可表示為 $I^{[apri]}$ 與 E_s/N_0 的函數， $I^{[ext]} = f(I^{[apri]}, E_s/N_0)$ ， $f(\cdot)$ 為通道解碼或訊源解碼演算法。換句話說，只要傳送信號能量與通道雜訊能量確定，可調整 σ_A^2 以得到我們想要觀察的 $I^{[apri]}$ 與 $I^{[ext]}$ ，進而產生訊源/通道解碼器的額外資訊轉換圖，並藉由觀察額外資訊轉換圖中兩項互消息的變化來評估解碼效能的好壞。例如在相同的實驗環境設定下，輸入等量的事前互消息至解碼器輸入端，則得到高輸出互消息的解碼器性能會較突出，因為由互消息的意義可知它比較能夠準確地估測傳送資料。

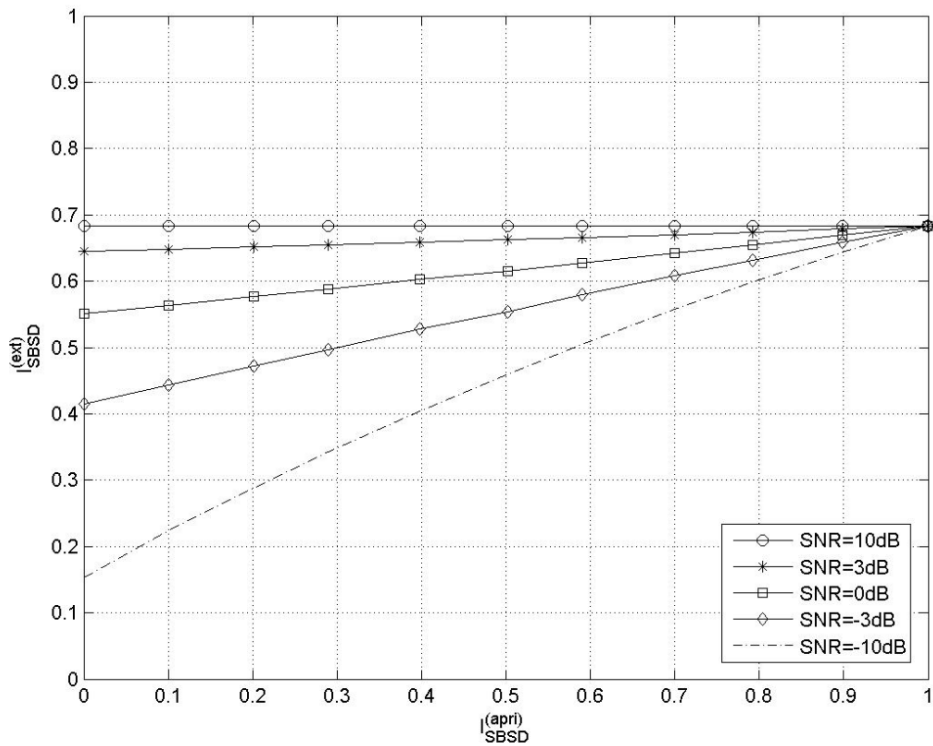
最後，我們將呈現不同通道環境下的訊源通道解碼之額外資訊轉換圖。實驗環境設定方面：原始傳送訊號 v_i 被模擬成相關係數 $r = 0.95$ 的一階高斯馬可夫程序(1-st order Gauss-Markov process)，並採用量化位元數 $M = 3$ 的 Lloyd-Max 量化器；而量化器的輸出則使用自然二位元碼(natural binary code, NBC)的位元對應(bit mapping)方式來產生二位元序列 x_i 。之後， x_i 經過一個生成矩陣

$$G(D) = \begin{bmatrix} 1 & \frac{g^{(1)}(D)}{g^{(0)}(D)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1+D^2}{1+D+D^2} \end{bmatrix} \text{ 的通道編碼處理形成 } y_i \text{ 再傳送至}$$

通道中；通道訊雜比 E_s/N_0 則設定為 10, 3, 0, -3, -10 dB 五種情況。



(a)



(b)

圖 3.2 額外資訊轉換圖(NBC) (a)通道解碼器 (b)訊源解碼器

圖 3.2(a) 中橫軸及縱軸分別為為通道解碼器的事前互消息 $I_{CD}^{(apri)}$ 及事後互消息 $I_{CD}^{(ext)}$ ，並由圖中可看到隨著通道訊雜比的增加，解碼效能也跟著往上提升。例如在高訊雜比的通道 ($E_s/N_0 = \{10, 3, 0\}dB$) 時，通道解碼器可以在低 $I_{CD}^{(apri)}$ 的輸入時就可得到較高的 $I_{CD}^{(ext)}$ ；但是在 $E_s/N_0 = \{-3, -10\}dB$ 的通道下則需要較高的 $I_{CD}^{(apri)}$ 輸入方能產生較高的 $I_{CD}^{(ext)}$ 。另一方面，無論在何種通道環境下都可發現 $I_{CD}^{(apri)}$ 與 $I_{CD}^{(ext)}$ 之間恆存在著正相關的關係。在圖 3.2(b) 中，訊源解碼器的額外資訊轉換圖亦有相似的現象產生。

為了要由理論上來預估疊代訊源通道解碼的收斂狀況時，則先將訊源解碼器額外資訊轉換圖的橫軸、縱軸交換，再與通道解碼器額外資訊轉換圖結合，尋找兩曲線的交點，稱之為停止交點(stopping intersection)。要把訊源解碼器額外資訊轉換圖的橫軸、縱軸置換的原因，是在疊代解碼的過程中，每個解碼器都會以另一個解碼器輸出的額外資訊做為輸入來進行解碼，因此必須置換其中一個解碼器額外資訊轉換圖的橫軸、縱軸來完成此項動作。而通常我們會選擇訊源解碼器額外資訊轉換圖作為置換的目標，讓 $I_{CD}^{(apri)} = I_{SBSD}^{(ext)}$ 且 $I_{SBSD}^{(apri)} = I_{CD}^{(ext)}$ 。在觀察實際解碼收斂狀況方面，則是把每次實際疊代解碼產生的額外訊息 $I_{CD}^{(ext)}$ 、 $I_{SBSD}^{(ext)}$ 直接用來計算 $I_{CD,n}^{(ext)}$ 、 $I_{SBSD,n}^{(ext)}$ ，n 代表第幾次疊代，並且分別將 $I_{SBSD,n}^{(ext)}$ 、 $I_{CD,n}^{(ext)}$ 視為的 x、y 座標，直接在額外資

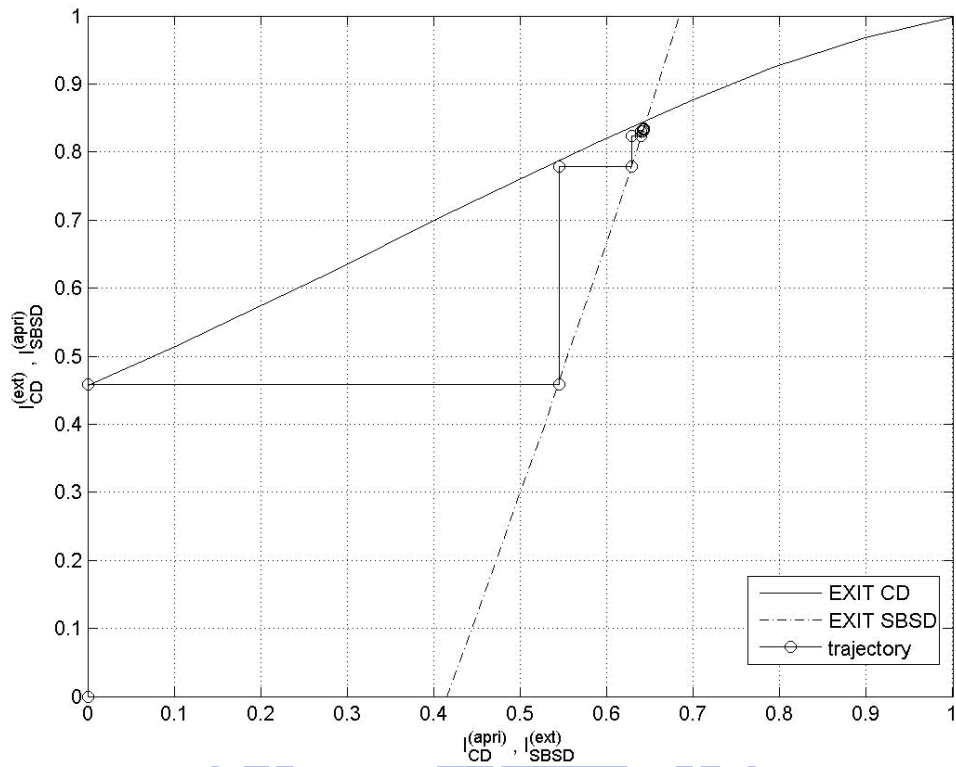


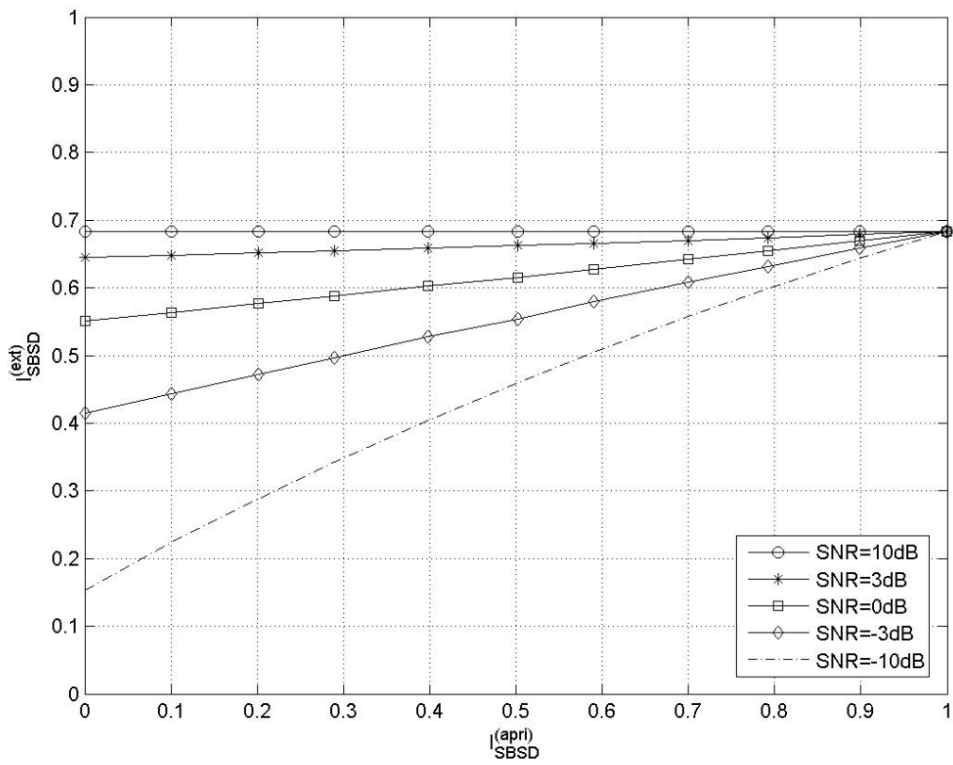
圖 3.3 軌跡圖與額外資訊轉換圖(NBC)

訊轉換圖上進行標示，形成所謂的軌跡圖(trajjectory)。圖 3.3 則顯示此系統在訊雜比 $E_s/N_0 = -3dB$ 時的軌跡圖與額外資訊轉換圖。

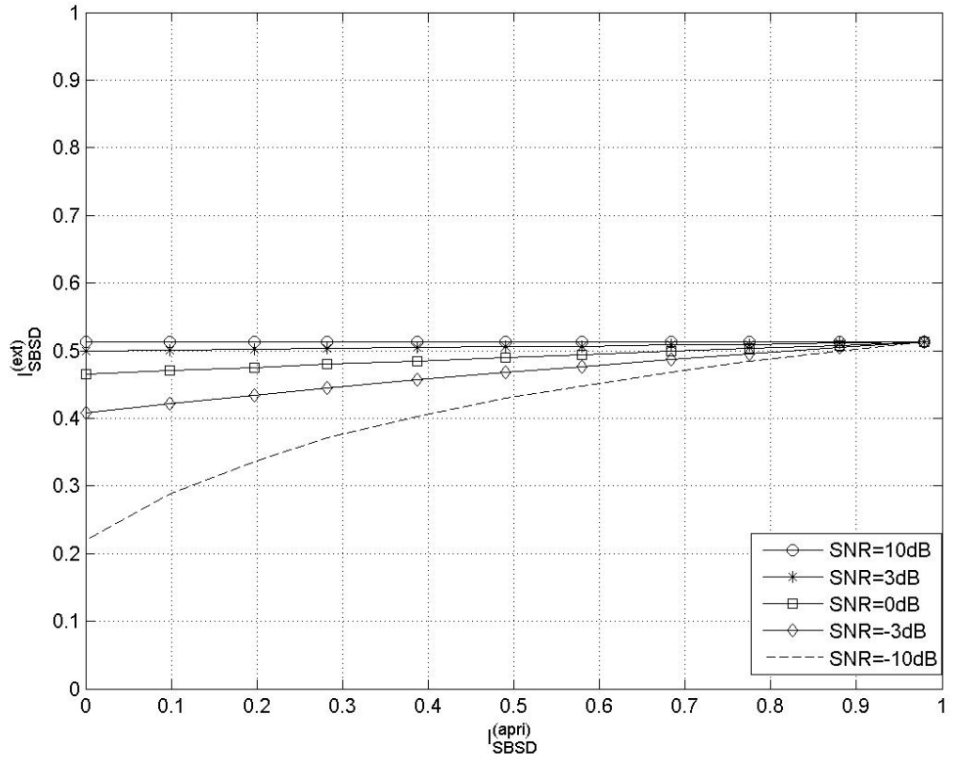
3.1.2 基於 EXIT Chart 的索引指定設計

在上節中的圖 3.2 裡，不但可看到通道解碼器與訊源解碼器的額外資訊轉換特性外，若再仔細比對分析則可發現到有兩個現象存在於它們之間。第一，通道解碼器與訊源解碼器所適用的通道環境不同。在通道環境較差的情況下，例如 $E_s/N_0 = -3 \text{ dB}$ ，即使輸入相同的事前互消息 $I^{(apri)}$ ，訊源解碼器擁有較高的事後互消息 $I^{(ext)}$ 輸出，即訊源解碼器在雜訊干擾較強的環境下能有較佳的效能；但是在通道環境好的時候， $E_s/N_0 > -3 \text{ dB}$ ，反而是通道解碼器對於受雜訊干擾小的輸入資料有不錯的更正能力，輸出的事後互消息 $I^{(ext)}$ 較高。所以疊代訊源通道解碼利用此項特點，藉著建立可疊代解碼架構讓兩解碼器相互交換額外資訊，以補足個自己不擅長的通道環境，使系統整體的效能獲得提升。第二，在訊源解碼器裡，不同的索引指定機制將產生不同 $I_{SBSD}^{(ext)}$ 的趨勢與大小，進而改變額外資訊轉換圖中停止交點的位置。如圖 3.4 所示，無論在任何通道環境下自然二位元碼的 $I_{SBSD}^{(ext)}$ 表現皆優於格雷碼(Gray Code)的 $I_{SBSD}^{(ext)}$ ，這就代表了索引指定的選擇將直接影響疊代訊源通道解碼效能。

因此，由上述兩點，基於額外資訊轉換圖的索引指定設計將以 $I_{SBSD}^{(ext)} \big|_{I_{SBSD}^{(apri)}=1}$ 做為最佳化的準則，且採用二位元置換演算法[13][23] (binary switching algorithm, BSA)的方式來搜尋最佳索引指定。



(a)



(b)

圖 3.4 訊源解碼器額外資訊轉換圖 (a)NBC (b) Gray code

一開始選定 NBC 當做 BSA 的初始值，並計算對應的 $I_{SBSD}^{(ext)} |_{I_{SBSD}^{(apri)}=1}$ 。接著

將此索引指定做為置換的基準，固定該索引指定的第 $i=0$ 位置後，

[步驟一]：將其它位置 $j=i+1, \dots, 2^M - 1$ 的索引值依序與位置 i 的索引

值作置換，同時計算對應的 $I_{SBSD}^{(ext)} |_{I_{SBSD}^{(apri)}=1}$ 。

[步驟二]：過程中若找到比基準索引值還高且最大的 $I_{SBSD}^{(ext)} |_{I_{SBSD}^{(apri)}=1}$ 時，

則將此索引指定與 $I_{SBSD}^{(ext)} |_{I_{SBSD}^{(apri)}=1}$ 記錄下來，並以此索引指定

做為基準，令 $i=0$ 後跳回步驟一；若無，則進入步驟三。

[步驟三]：若 $i < 2^M - 2$ 則將 $i=i+1$ 並進行步驟一；反之代表 BSA 收

斂，基準索引指定為最佳。表 3.1 為利用此方法在

$E_s/N_0 = -3 \text{ dB}$ 時所找出的最佳索引指定。

表 3.1 自然索引指定和基於 EXIT 設計的最佳化索引指定 ($M=3$)

自然二位元碼 (NBC)		基於 EXIT 設計的最佳索引指定 (BIA_E)	
0	000	0	000
1	001	5	101
2	010	6	110
3	011	3	011
4	100	4	100
5	101	1	001
6	110	2	010
7	111	7	111

3.1.3 基於 EXIT Chart 索引指定設計之缺陷

3.1.2 節中的索引指定必須滿足下列兩項假設：

(a) 在交錯器長度夠大的情況下，則經過幾次疊代後，事前資訊

$L^{(apri)}(y_i^s)$ 與通道相關資訊 $L(\tilde{y}_i^s | y_i^s)$ 不相關。

(b) 疊代次數越多，軟性解碼器輸出的額外資訊(未經交錯器)會越趨

近於高斯分佈。

換句話說，當這兩項假設成立後我們才能簡化系統分析的複雜度；將每個解碼器輸入端的事前資訊模擬成高斯分佈；事前互消息 $I^{[apri]}$ 與事後互消息 $I^{[ext]}$ 的公式推導；單獨計算疊代解碼架構中任一解碼器的 $I^{[apri]}$ 和 $I^{[ext]}$ 。但是這也使得索引指定的設計過程中，無法考量到交錯器的配置，並造成設計後的索引指定與選擇交錯器搭配上的問題發生。

以圖 3.5 來說明交錯器對系統的影響，它是 NBC 分別搭配區塊交錯器與隨機交錯器(大小為 300 bit, $S=4$)後，在通道訊雜比等於 -3dB 下疊代 10 次後所產生的軌跡圖。從圖上可發現兩者軌跡圖最終都停留在停止交點附近，但是搭配隨機交錯器的系統無論是在疊代過程中或是停止交點都有優於搭配區塊交錯器系統的表現。大約在第二次疊代後，採隨機交錯器系統中的解碼器可獲得較高的事前互消息 $I^{[apri]}$ 與事後互消息 $I^{[ext]}$ ，並且最終停止交點為(0.648, 0.857)，

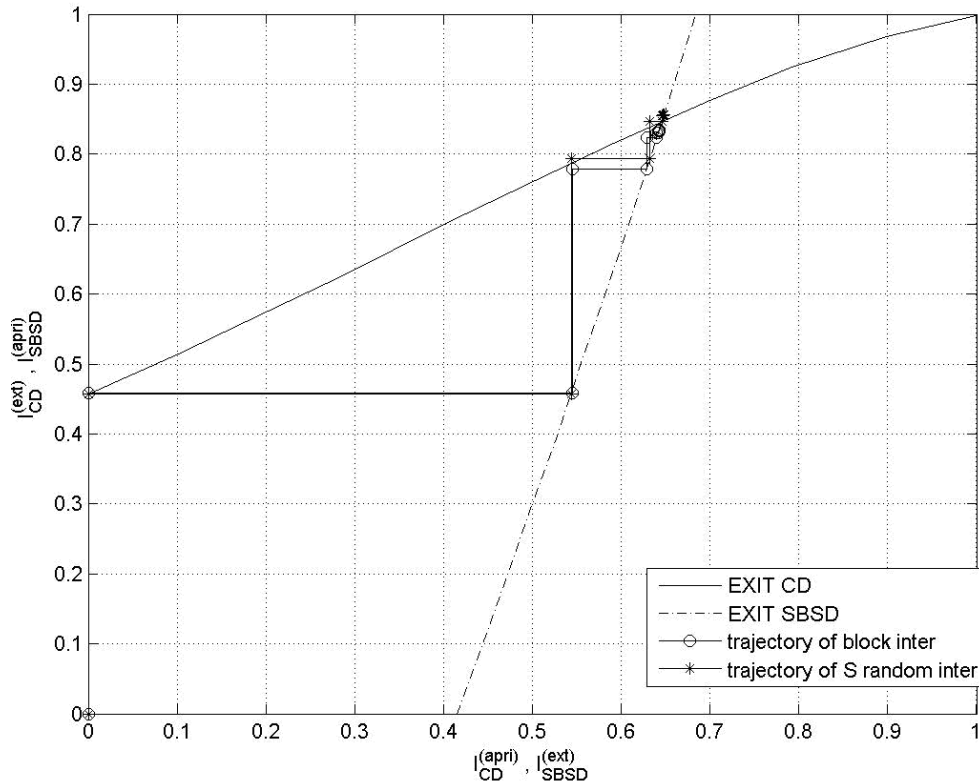


圖 3.5 額外資訊轉換圖與軌跡圖

(NBC，搭配區塊交錯器與隨機交錯器)

優於使用區塊交錯器系統的停止交點(0.642, 0.834)。因此不同的交錯器確實會影響系統的表現，但是 3.1 節的設計方式卻因 a、b 的假設而忽略交錯器對系統的影響。

3.2 基於資訊相關性的索引指定設計

本節將針對額外資訊轉換圖的索引指定設計方式之缺失，提出一個基於資訊相關性的成本函數，利用其設計出一個合乎整體系統設定的最佳化索引指定設計。有別於上節的設計方式，這一個新的索引指

定設計可以更廣泛用於位元及索引層級的疊代訊源通道解碼過程。

要將交錯器納入考量並找出符合整體系統設定的索引指定，必須回歸到實際的系統架構上，找尋可具體反應疊代解碼效能的指標。然而，[13]已說明了不同索引指定有不同的額外資訊轉換圖，象徵它們的額外資訊會有不同的變化，同時額外資訊轉換圖的差異又將導致解碼效能好壞，所以額外資訊的變化可具體反應該系統的解碼效能。更進一步分析，通道解碼器額外資訊 $p_{CD}^{(ext)}(x_l)$ 源自於後驗機率 $p(x_l | \tilde{Y}_l^L)$ ，故每次疊代解碼時結果的好壞都會影響 $p_{CD}^{(ext)}(x_l)$ 產生的最小化均方差預估值之特性，例如其相關性的變化。因此我們依據額外資訊之間的相關性建立了一個成本函數，鏈結索引指定與其對應的成本函數可評估當下疊代解碼好壞的想法，並透過二位元置換演算法將不同的索引指定引進系統中，希望能找到符合整個系統架構的最佳索引指定。首先定義由 $p_{CD}^{(ext)}(x_l)$ 產生的最小化均方差預估值

$$\tilde{v}_{t,CD} = \sum_{n=0}^{2^M-1} p_{CD}^{(ext)}(\mathbf{u}_t = n) \cdot v_t(n) \quad (3.1)$$

當中 $p_{CD}^{(ext)}(\mathbf{u}_t)$ 為通道解碼器額外資訊所形成的索引層級事後機率。

假設原始傳送信號 v_t 為一階高斯馬可夫程序，其相關係數為 r ，即

$$v_t = r \cdot v_{t-1} + n_t \quad (3.2)$$

其中 n 為平均值 $\mu = 0$ 、變異數 $\sigma^2 = \sigma_n^2$ 的白色雜訊程序(white-noise process)。其自相關函數(autocorrelation function)

$$\rho(\tau) = E[v_t \cdot v_{t+\tau}], \tau = 0, 1, \dots, T \quad (3.3)$$

同理， $\tilde{v}_{t,CD}$ 與 $\tilde{v}_{t+\tau,CD}$ 之間的自相關函數為

$$\rho'(\tau) = E[\tilde{v}_{t,CD} \cdot \tilde{v}_{t+\tau,CD}], \tau = 0, 1, \dots, T \quad (3.4)$$

而 $\rho'_n(\tau)$ 則是第 n 次疊代後，用 $p_{CD}^{(ext)}(u_t(\lambda))$ 產生的最小化均方差預估
值所計算的自相關函數。

在整個疊代訊源通道解碼架構中，無論是通道解碼器或者訊源解碼器，它們的事前資訊輸入端包含事前統計資訊與額外資訊這兩部分。事前統計資訊在系統設計初期被決定後便固定不變，而額外資訊則是隨著解碼器效能而變化，並且影響下一級解碼器的事前資訊。因此對訊源解碼器而言，當通道解碼器可更正大部分接收資料中的錯誤時，不但可以提供良好的額外資訊給訊源解碼器外，同時也會讓基於額外資訊所產生的自相關函數 $\rho'(\tau)$ 的統計特性更接近原始傳送資料的 $\rho(\tau)$ 統計特性，進而幫助訊源解碼器輸出更正確的結果。另一方面，對通道解碼器而言，假若由其輸出的額外資訊所產生的統計特性 $\rho'(\tau)$ (公式(3.4)) 能夠在疊代解碼的過程中越來越接近 $\rho(\tau)$ ，就象徵了訊源解碼器解碼後輸出資料的正確性高，並且它能提供較好的額外資訊當作通道解碼器的事前資訊，幫助通道解碼器更正更多因傳輸造成的錯誤。最後，在疊代解碼過程中，上述兩個現象會不斷的分別在通道解碼器與訊源解碼器中發生，而最終系統效能達到收斂時，

$\rho_n'(\tau)$ 將不會再有明顯的變化。

以使用自然二位元碼(NBC)與格雷碼(Gray Code)為索引指定的系統為例，在與 3.1.2 節相同實驗環境設定下，觀察於通道訊雜比為 -3 dB 中兩者 $\rho'(\tau)$ 的變化，結果顯示於圖 3.6。如同稍早描述，隨著疊代次數的增加與系統效能上升，兩系統的 $\rho'(\tau)$ 也越來越接近 $\rho(\tau)$ 。但是經過仔細觀察後，仍可察覺到有段明顯差距存在於兩者的 $\rho'(\tau)$ 之間，例如在同樣經過五次疊代解碼後，採自然二位元碼系統的 $\rho'(\tau)$ 會較採格雷碼系統的 $\rho'(\tau)$ 接近 $\rho(\tau)$ 。這個現象反應了使用自然二位元碼的系統在第五次疊代後，它的效能會優於使用格雷碼的系統。因此，可由以上的推論與實際觀測，證實了 $\rho'(\tau)$ 與系統效能之間確實存有正相關的特性。有鑑於此，我們將成本函數定義為 $|\rho(\tau) - \rho_n'(\tau)|$ ，而一個好的索引指定必須要使 $|\rho(\tau) - \rho_n'(\tau)|$ 達到最小值。因此基於資訊相關性的最佳化索引指定設計可轉換成一個最佳化的問題：

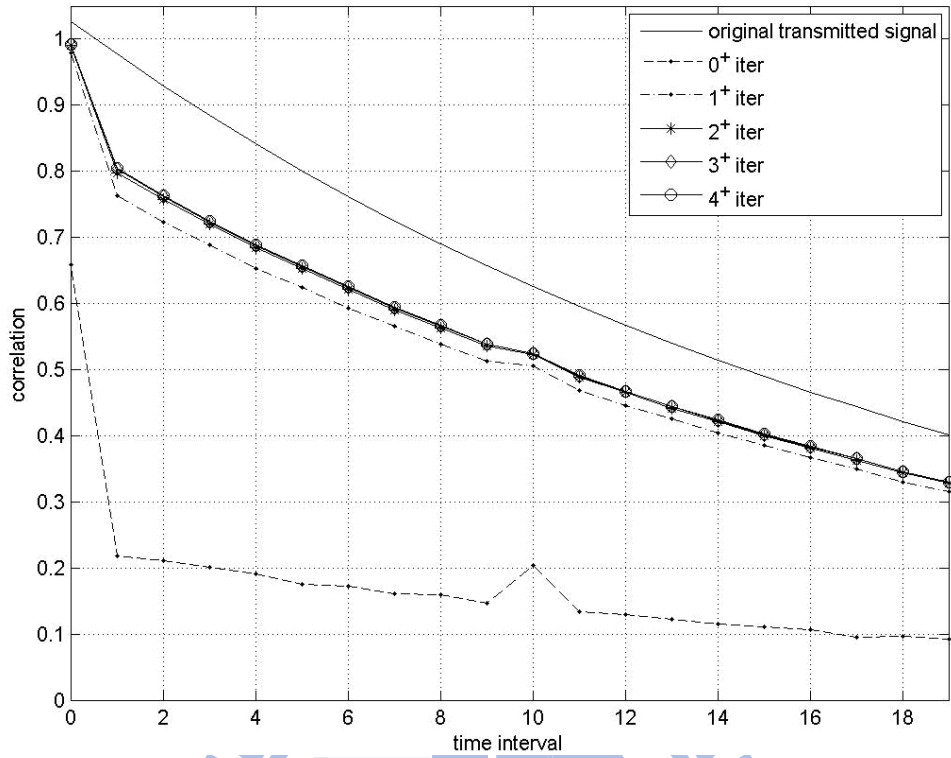
已知： v_i 是一階高斯馬可夫程序模擬其相關因子 $r=0.95$ ，

量化器為量化位元數 $M=3$ 的 Lloyd-Max，

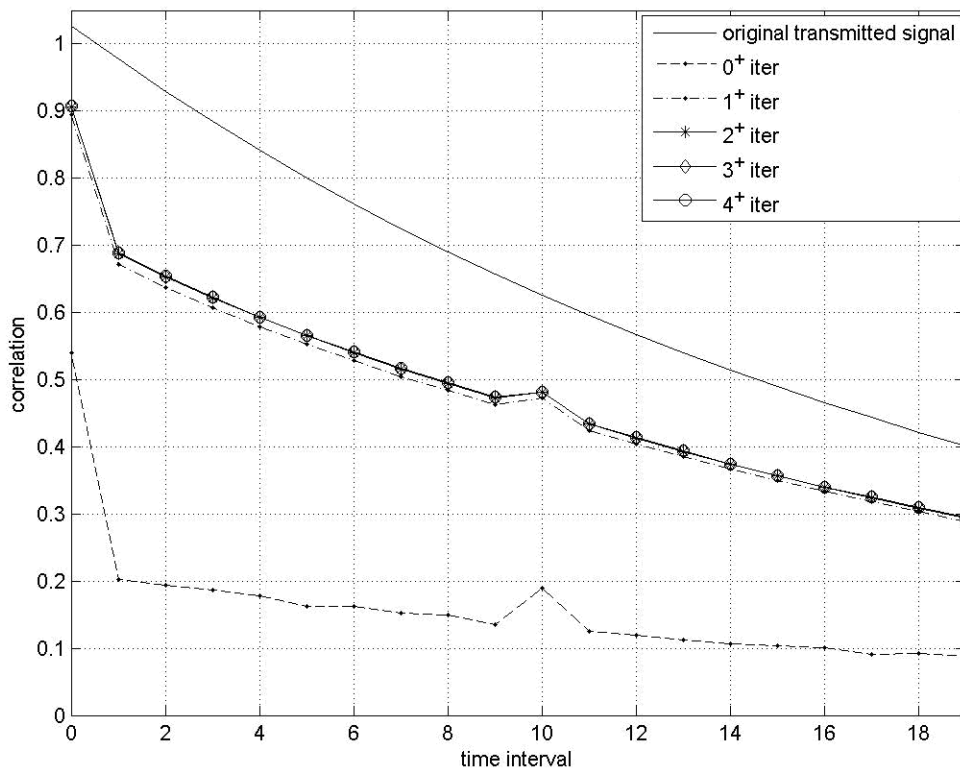
區塊交錯器的長度等於 300 位元，

$$\text{通道編碼矩陣為 } G(D) = \begin{bmatrix} g^{(1)}(D) \\ g^{(0)}(D) \end{bmatrix} = \begin{bmatrix} 1 + D^2 \\ 1 + D + D^2 \end{bmatrix},$$

訊雜比為 -3dB 的通道雜訊以高斯程序產生。



(a)



(b)

圖 3.6 額外資訊相關特性圖 $\rho'(\tau)$ (a) NBC (b) Gray Code

令集合 Λ 是 $\{0, 1, \dots, 2^M - 1\}$ 所有排列組合的聯集， ϕ 與 ϕ_{opt} 分別為索引指定、最佳索引指定，則

$$\phi_{opt} = \arg \min_{\phi \in \Lambda} \left[\left| \rho(1) - \rho_{k^*}'(1) \right| \right] \quad (3.5)$$

其中 $\rho_{k^*}'(1)$ 為疊代解碼收斂時的一階自相關值； k^* 代表疊代解碼收斂次數。

最後，針對上述最佳化設計的完整流程顯示於圖 3.7，過程中除了同樣使用 BSA 來降低運算量外，還有以下兩點需特別注意：

1. 為了降低原始傳送訊號以及通道雜訊對索引指定最佳化的影響，設

計索引指定時，需使用同一組原始傳送訊號與通道雜訊。

2. 本方法可針對索引或位元層級 ISCD，分別進行其索引指定設計。

在索引層級的解碼過程，只需直接以索引層級的通道解碼器額外

資訊 $p_{CD}^{(ext)}(\mathbf{u}_t)$ 計算 $\rho_n'(1)$ 即可。另一方面，當要設計位元層級系統

的索引指定時，則必須先把通道解碼器輸出的位元層級額外資訊

$p_{CD}^{(ext)}(x_t)$ 轉至索引層級額外資訊，如下：

$$p_{CD}^{(ext)}(\mathbf{u}_t) = \prod_{\lambda=1}^M p_{CD}^{(ext)}(u_t(\lambda)) \quad (3.6)$$

接著才能計算 $\rho_n'(1)$ 。(註： $p_{CD}^{(ext)}(u_t(\lambda))$ 表示經由解交錯器作用後

的通道解碼器額外資訊 $p_{CD}^{(ext)}(x_t)$)

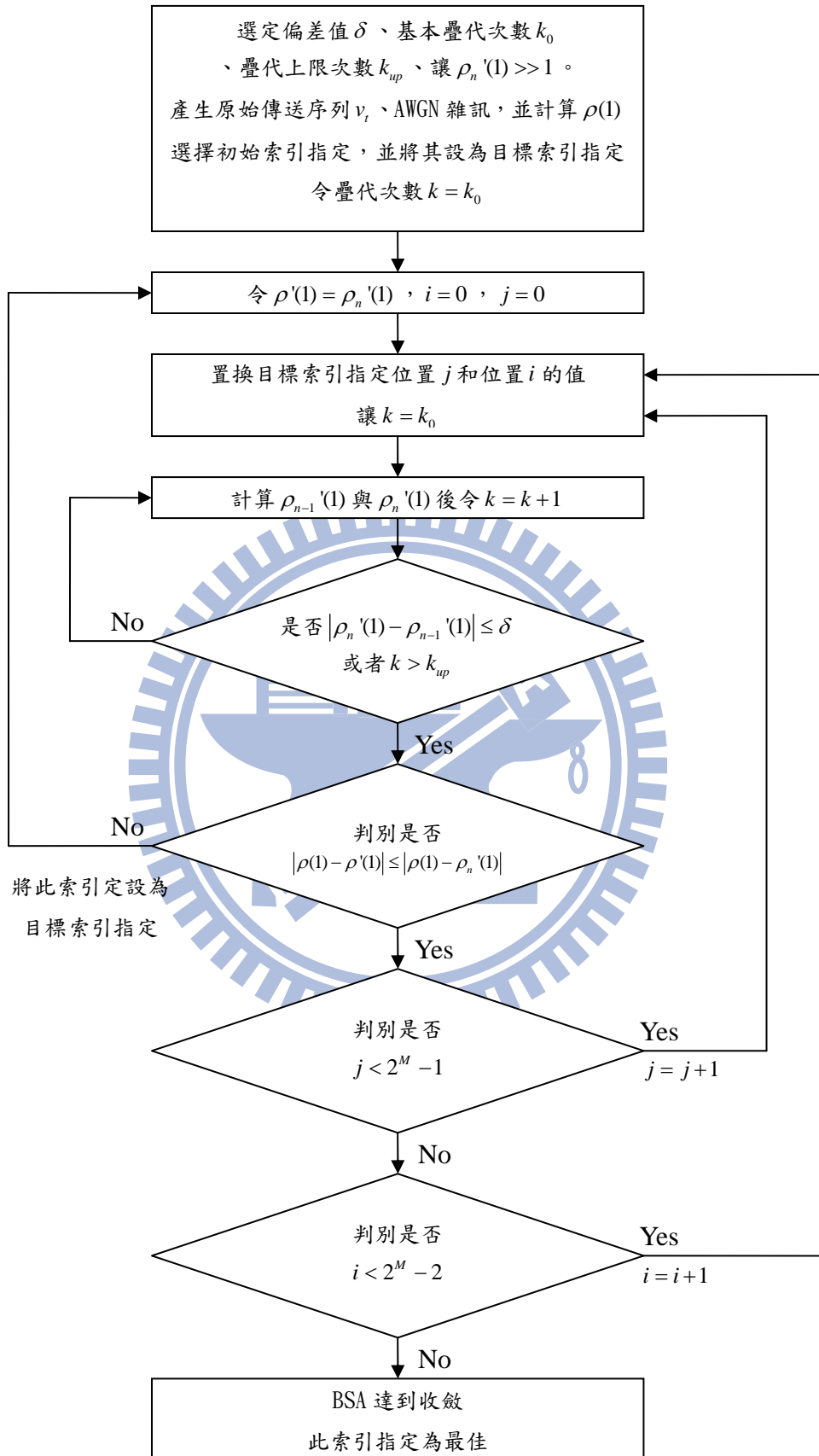


圖 3.7 二位元置換演算法流程圖

第四章 交錯器的最佳化設計

就通道編碼研究而言，自從渦輪碼被提出後，交錯器的設計成為疊代通道解碼架構中一項重要的研究課題，並且已有數種成熟的設計方式被提出[14][15][16][17][18]。然而，對於疊代訊源通道解碼而言，一般還是採用傳統的區塊交錯器與隨機交錯器兩大類，而且缺乏相關研究來說明何者能提供較好的疊代訊源通道解碼效能。系統傳送端的原始信號，如影像及聲音，經訊源編碼器(Source Encoder)量化壓縮後所產生的輸出資料，將會被送至通道編碼器進行編碼。根據Shannon's separation theorem[1]，若訊源編碼器輸出的資料間不存在任何冗息，則訊源編碼器與通道編碼器可考慮個別設計。然而實際上訊源編碼器輸出的資料仍然存在殘餘冗息，這些冗息能在接收端幫助解碼器進行通道錯誤的檢測與補償處理，以加強系統的強健性。所以在傳送端中，交錯器設計的目的是降低訊源編碼器輸出資料間的關聯性，使得通道編碼器可忽略該資料的殘餘冗息(例如 AK0、AK1)；而在接收端，通道解碼器和訊源解碼器也需透過交錯器、解交錯器的連結才能行使疊代解碼。因此，有別於以往只專注傳送端交錯器所扮演的角色，本章節將延伸渦輪碼交錯器的設計理念[19]，從疊代解碼效益的角度切入，賦予交錯器在疊代訊源通道解碼中新的定位。最後

更進一步的提出設計交錯器的成本函數，將其轉換為一個最佳化問題，並延伸此概念至索引層級疊代訊源通道解碼的交錯器設計。

4.1 疊代解碼效益

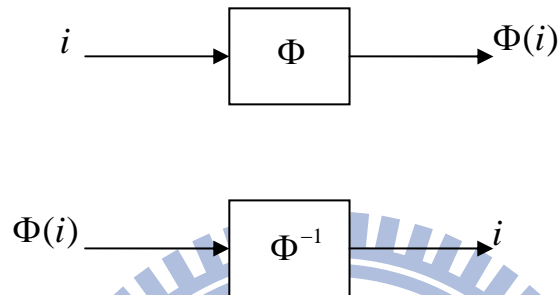


圖 4.1 交錯器 Φ 與解交錯器 Φ^{-1}

圖 4.1 分別為長度 L 的交錯器與解交錯器。交錯器可將輸入序列中第 i 個位置 ($i \in \{1, 2, \dots, L\}$) 的資料對應到輸出序列裡第 $\Phi(i)$ 個位置；第 $\Phi(i)$ 位置的資料則可被解交錯器對應回第 i 個位置。在疊代解碼架構中，訊源解碼器的額外資訊經交錯器置換後，做為下一疊代通道解碼器的事前資訊使用；另一方面，通道解碼器的額外資訊經解交錯器置換後，被當成下一級訊源解碼器的事前資訊。所以對整個疊代訊源通道解碼而言，交錯器的特性可讓通道解碼器第 $\Phi(i)$ 個位置的輸入資料，使用訊源解碼器第 i 個位置的額外資訊來進行解碼。而解交錯器則使訊源解碼器第 i 個位置的額外資訊做為通道解碼器第 $\Phi(i)$ 個位置的事前資訊。因此要如何設計交錯器，使得疊代架構中通道與訊源解

碼器可交換到滿足其需求的事前資訊就成為設計的關鍵，同時這也是 [19] 所提出的基於疊代效益的交錯器設計概念。

由於渦輪碼中使用兩個通道解碼器，所以 [19] 交錯器設計的理念就是分析通道解碼演算法，從中了解通道解碼器對於事前資訊特性的要求。並且因為實際疊代解碼過程中，交錯器的選擇將影響演算法的特性，所以可藉由觀察實際解碼時資訊關聯性的統計特性，並依此設計交錯器，讓交錯器能提供解碼器理論上需求的事前資訊，以達到疊代解碼效能的提升。

然而，基於疊代解碼效益的交錯器設計確實能提升系統效能，但是上述設計模式卻只能滿足單一解碼器對事前資訊需求。事實上，無論疊代訊源通道解碼或渦輪碼都是由兩個解碼器組合而成，必須要同時考量交錯器與解交錯器可否提供良好的事前資訊給下一級解碼器。所以接下來的 4.2、4.3 節中，分別觀察疊代過程中交錯器設計對於通道與訊源解碼器裡資訊關聯性的影響。4.4 節則反應兩解碼器對於事前資訊的需求，來改良原始交錯器設計演算法。新的演算法除了考量交錯器的設計是否能符合通道解碼器對事前資訊的要求外，我們更進一步的考量了解交錯器對於訊源解碼器的影響，同時進行交錯器與解交錯器設計，期望此交錯器的設計可為疊代訊源通道解碼效能帶來提升。

4.2 基於資訊關聯性的交錯器設計

本節將探討 BCJR 演算法在理論上對事前資訊所作的假設，並且分析實際疊代解碼時通道解碼器中資訊關聯性的變化，最後將總結理論與實際的差異來設計交錯器。

4.2.1 通道解碼器輸入資訊關聯性的分析

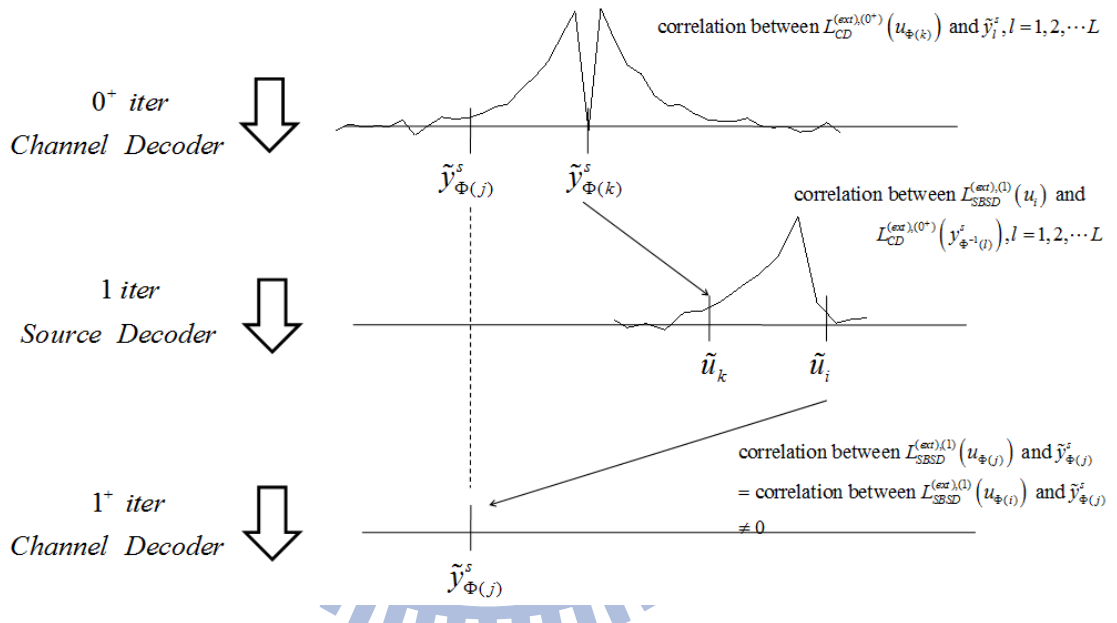
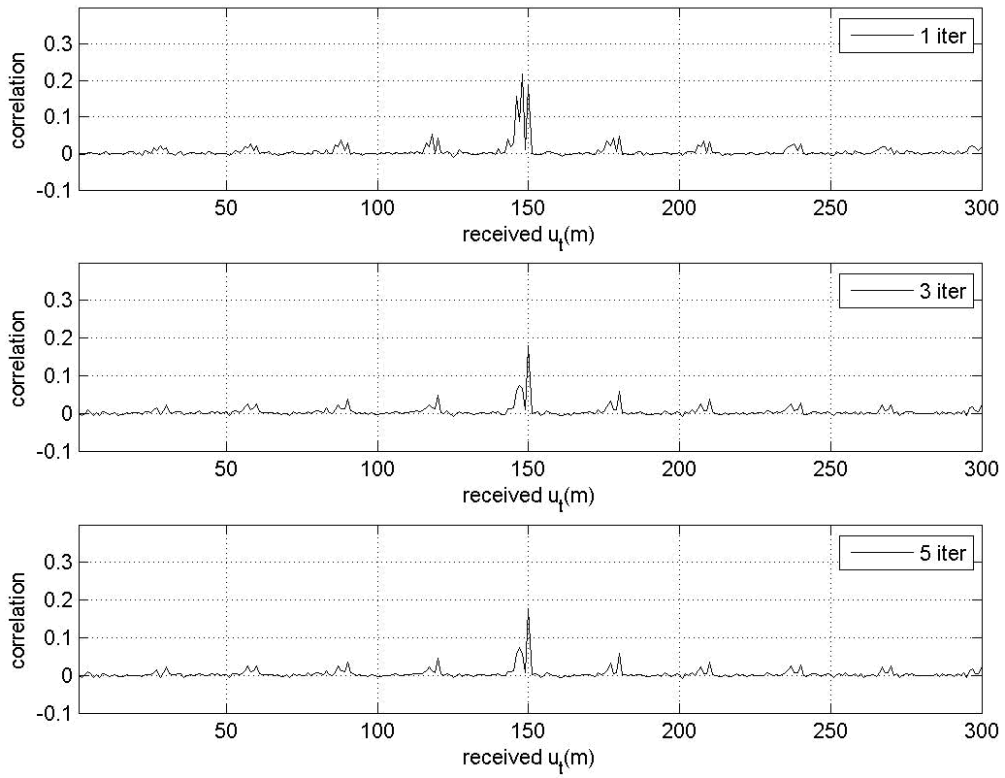
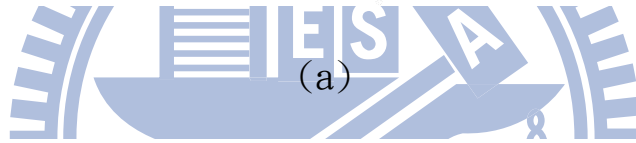
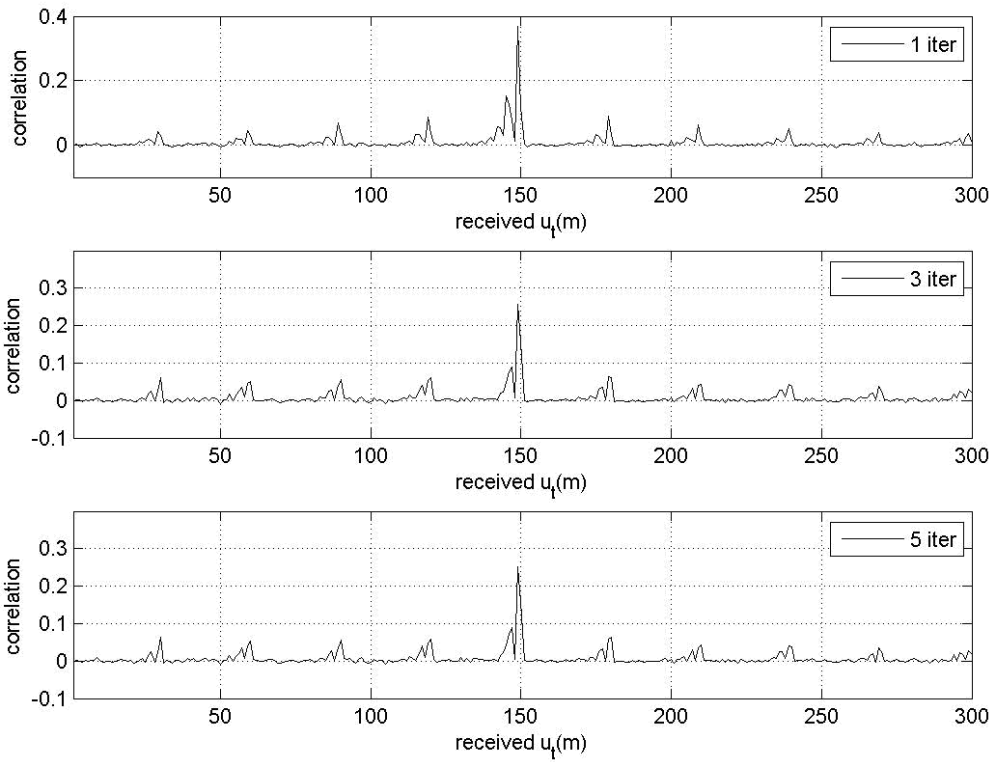


圖 4.2 疊代解碼中輸入資訊之間關聯性流程圖

第二章中我們介紹了通道解碼器所使用的 BCJR 演算法，其中為了簡化公式推導而做了若干假設，例如式子(2.3)中假設 y_l 與 \tilde{Y}_1^{l-1} 、 \tilde{Y}_{l+1}^L 相互獨立，且 y_i^s 與 y_i^p 之間也相互獨立；而(2.6)則意味著疊代解碼過程中回傳的第 l 位置訊源額外資訊 $L_{SBS D}^{(ext)}(x_l)$ 要與 y_i^s 與 y_i^p 獨立。上述前項假設可在 AWGN 通道環境下成立，但是後項假設在實際解碼過程中

卻不一定成立。圖 4.2 顯示 0^+ 至 1^+ 次疊代時，輸入資訊間關聯性形成的流程圖。為了稍後說明進行以下定義：令一個交錯器長度內的原始序列 $U_1^T = \{u_1(1), u_1(2), \dots, u_i(m) \dots u_T(M)\} = \{u_1, u_2, \dots, u_i, \dots, u_L\}$ ， $m=1, 2, \dots, M$ 、 $L=M \cdot T$ ，且經交錯器作用後的 u_i 等於 $y_{\Phi(i)}^s$ ； K 代表 u_i 所屬的索引及之前索引組成位元的位置之集合。現今 u_i 、 $u_j \in U_1^T$ ， $k \in K$ 。

在第 0^+ 次疊代解碼時由於尚未有額外資訊輸入，通道解碼器只能根據事前統計機率與通道相關機率進行 BCJR 演算法，又因為通道編碼器暫存器的影響，所以解碼後輸出的 $L_{CD}^{(ext),(0^+)}(y_{\Phi(k)}^s)$ 將只會與鄰近部分的 \tilde{y}_i^s 產生關聯性，例如 $\tilde{y}_{\Phi(j)}^s$ 。接下來 $L_{CD}^{(ext),(0^+)}(y_{\Phi(k)}^s)$ 經解交錯器作用後送回訊源解碼器，因訊源編碼輸出位元之間殘餘冗息，故被當成訊源解碼器第 i 個位置的事前資訊使用。所以 1 次疊代解碼後， $L_{SBSD}^{(ext),(1)}(u_i)$ 與 $L_{CD}^{(ext),(0^+)}(y_{\Phi(k)}^s)$ 之間便存在關聯性，並且 $L_{SBSD}^{(ext),(1)}(u_i)$ 會更進一步透過 $L_{CD}^{(ext),(0^+)}(y_{\Phi(k)}^s)$ 間接地與 $\tilde{y}_{\Phi(j)}^s$ 建立關聯性。進入 1^+ 次疊代解碼時， $L_{SBSD}^{(ext),(1)}(u_j)$ 做為通道解碼器中第 $\Phi(j)$ 位置的事前資訊，幫助進行解碼。若此時交錯器發生 $\Phi(i) = \Phi(j)$ 的狀況，將會造成實際疊代解碼狀況與 BCJR 假設不符的情形。



(b)

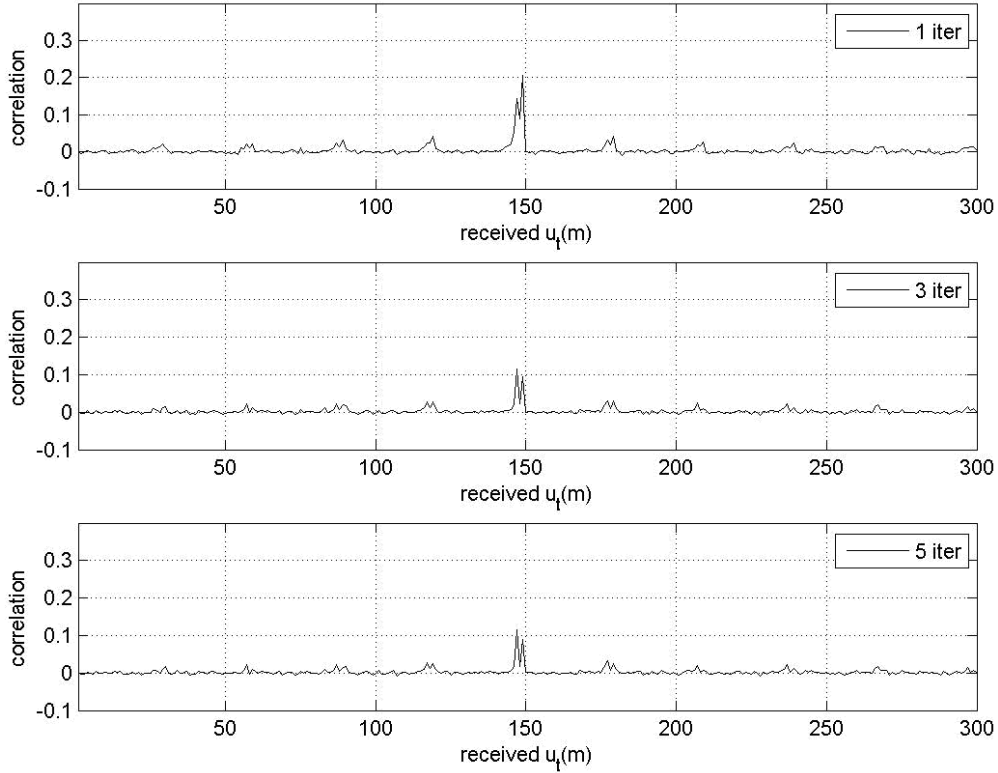


圖 4.3 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(n)}(u_i), \tilde{u}_t(m))$ (NBC) (a) $i=148$
 (b) $i=149$ (c) $i=150$

所以接下來我們將參照 3.1.1 節實驗環境的設定，並採用 NBC 搭配 10×30 的區塊交錯器。讓 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(n)}(i), \tilde{u}_t(m))$ 代表 $L_{SBS D}^{(ext),(1)}(u_i)$ 與接收值 $\tilde{u}_t(m)$ 之間的相關係數，其中 $t=1, 2, \dots, T'$ 、 $m=1, 2, 3$ ，兩隨機變數 p 、 q 之間的相關係數

$$\rho(p, q) = \frac{Cov[pq]}{\sqrt{Var[p] \cdot Var[q]}} \quad (4.1)$$

圖 4.3(a)(b)(c) 為第 n 次疊代中 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{u}_t(m))$ 的變化，

$i=148,149,150$ ， $n=1,3,5$ 。值得注意的是，在一個長度 $L=300$ 的交錯器區塊裡，位元 $\{u_{148},u_{149},u_{150}\}$ 是第50個索引的組成位元，因此令 $\{u_{148},u_{149},u_{150}\}$ 分別為 $\mathbf{u}_{50}=\{u_{50}(1),u_{50}(2),u_{50}(3)\}$ 。從圖中，我們明顯地看到三個現象：

[1] 三者都顯示同一索引的組成位元彼此間有最高的關聯性，而對前幾個時間點的索引位元則有次高的相關性，並有較小關聯性分佈

於部份的區域內。這些現象的發生是因為位元層級額外資訊

$L_{SBSD}^{(ext),(n)}(u_{50}(m))$ 的源於索引層級後驗機率 $p(\mathbf{u}_{50}=n|\tilde{\mathbf{U}}_1^T)$ ，所以

$L_{SBSD}^{(ext),(n)}(u_{50}(m))$ 會對同一索引內其他組成位元 $\{\tilde{u}_{50}(1),\tilde{u}_{50}(2),\dots$

$\tilde{u}_{50}(m-1),\tilde{u}_{50}(m+1)\dots\tilde{u}_{50}(M)\}$ 有最高的相關性。其次，訊源解碼

器在解碼時，利用一階事前訊息來計算後驗機率，因此後驗機率

$p(\mathbf{u}_{50}=n|\tilde{\mathbf{U}}_1^T)$ 與前面幾個索引 \tilde{u}_{49} 、 $\tilde{u}_{48}\dots$ 有次大的關聯性，隨後

並影響由其生成的 $L_{SBSD}^{(ext),(n)}(u_{50}(m))$ ，使之與 \tilde{u}_{49} 、 $\tilde{u}_{48}\dots$ 關聯性的產生。

至於分佈在部分區域的關聯性則是上一級的通道解碼器輸出的

額外資訊所造成。因進行通道解碼時也會利用到系統位元接收

值的資訊，使得 $L_{CD}^{(ext),(n-1)^+}(\tilde{y}_{\Phi(k)}^s)$ 與部分系統位元接收值產生關聯。

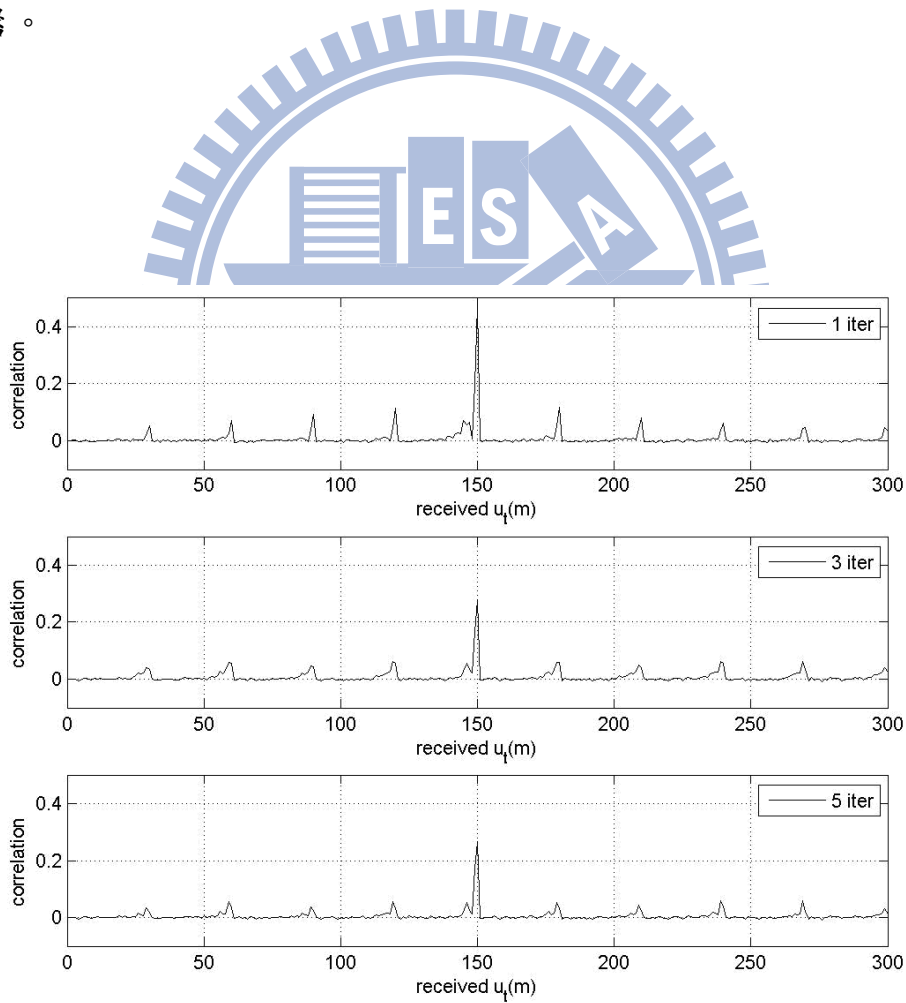
接下來 $L_{CD}^{(ext),(n-1)^+}(\tilde{y}_{\Phi(k)}^s)$ 經解交錯器送至訊源解碼器的事前資訊輸

入端，讓 $L_{SBSD}^{(ext),(n)}(u_i)$ 與其他不相鄰的系統位元接收值具有關聯性。

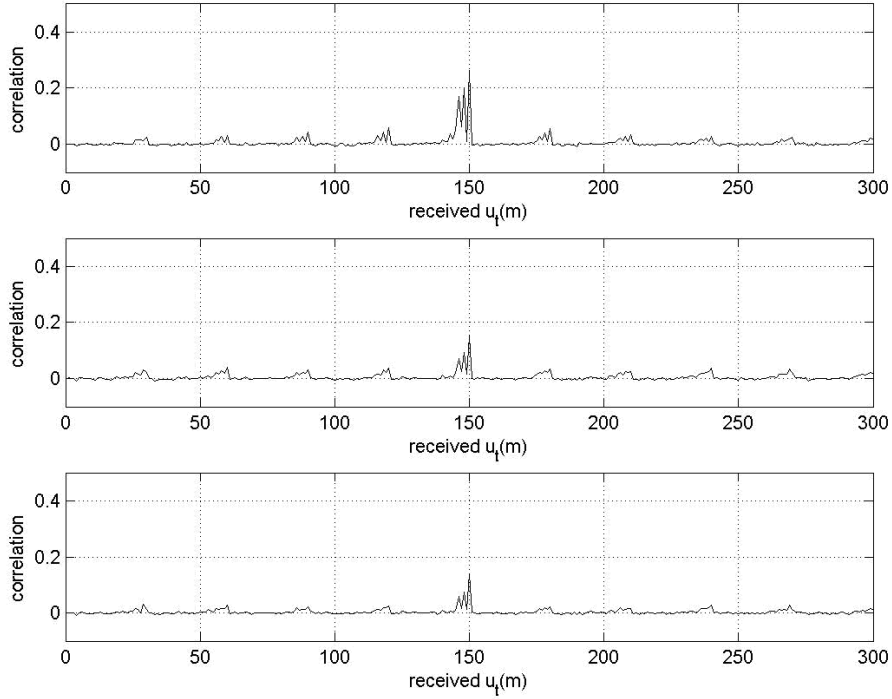
[2] 隨著疊代次數的增加，額外資訊 $L_{SBSD}^{(ext),(n)}(u_i)$ 與 $\tilde{u}_i(m)$ 的關聯性逐次

降低，這現象代表疊代解碼正在逐漸收斂。

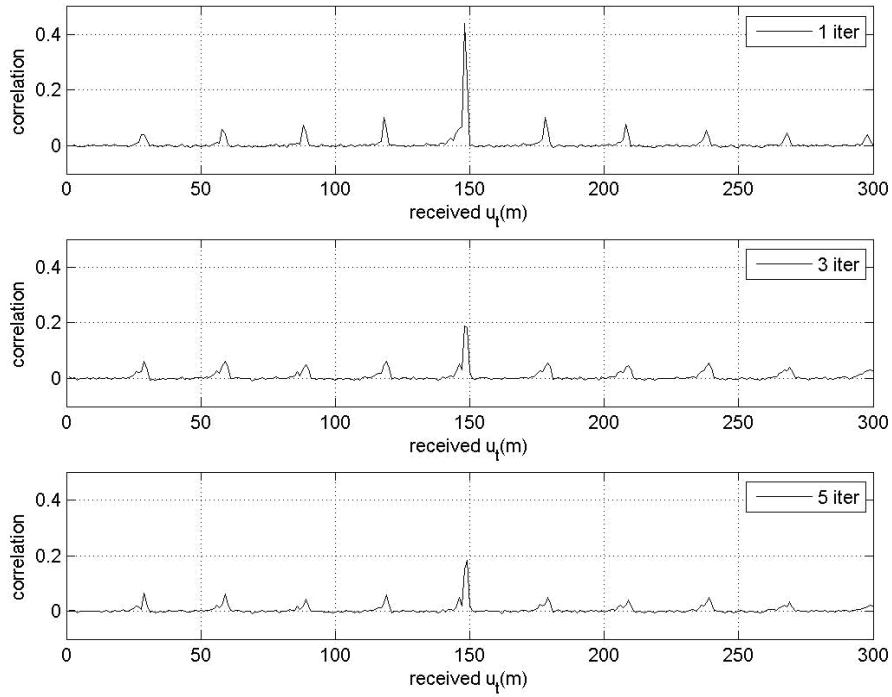
- [3] $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(n)}(u_i), \tilde{u}_i(m))$, $i=148,149,150$ ，三者的大小會依系統設定的索引指定而變化。圖 4.4 顯示的是表 3.1 中 BIA_E 之 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(n)}(u_i), \tilde{u}_i(m))$, $i=148,149,150$ 。與圖 4.3 的 NBC 結果比較，雖然高相關性的出現位置不會改變，但是索引指定的設定卻影響著關聯性的強度，這意味著特定的索引指定有其最佳的交錯器。



(a)



(b)



(c)

圖 4.4 $\rho_{SBSD}^{(n)}(L_{SBSD}^{(ext),(n)}(u_i), \tilde{u}_t(m))$ (表 3.1 中的 BIA_E)

(a) $i=148$ (b) $i=149$ (c) $i=150$

4.2.2 基於疊代解碼效益的交錯器設計

上節中，我們已觀察了疊代解碼時資訊關聯性的實際變化，接下來將根據第一次疊代資訊關聯性 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{u}_i(m))$ 來進行交錯器的設計。首先針對此關聯性建立一個統計模型：設定 U_1^T 經通道後形成 $\tilde{U}_1^T = \{\tilde{u}_1(1), \tilde{u}_1(2), \dots, \tilde{u}_T(M)\} = \{\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_L\}$ ， $\tilde{u}_i(m) \in \tilde{U}_1^T$ 。因此 $\tilde{u}_i(m)$ 可寫為 \tilde{u}_j ， $j = \{1, 2, \dots, L\}$ ，且 $\tilde{u}_i(m)$ 經交錯器作用後等於 $\tilde{y}_{\Phi(j)}^s$ 。將實測值

$\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{u}_i(m))$ 送入交錯器轉換形成 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{y}_{\Phi(j)}^s)$

後，根據[19]中使用的曲線配適法(curve fitting)，產生一個以 exponential 為底的指數函數 $\rho_{SBS D}^{(1)}(i, \Phi(j))$ ，

$$\rho_{SBS D}^{(1)}(i, \Phi(j)) = \rho_1(i, \Phi(j)) + \sum_{\Phi(g)=1}^L \rho_1(i, g) \rho_2(\Phi(j), \Phi(g)) \quad (4.2)$$

其中 $\rho_1(i, \Phi(j))$ 為 $L_{CD}^{(ext),0^+}(x_l) = 0$ ， $l = 1, 2, \dots, L$ 時， $L_{SBS D}^{(ext),(1)}(u_i)$ 與 $\tilde{y}_{\Phi(j)}^s$ 之間的關聯性； $\rho_2(\Phi(j), \Phi(g))$ 則代表 $L_{CD}^{(ext),0^+}(y_{\Phi(j)}^s)$ 與 $\tilde{y}_{\Phi(g)}^s$ 之間的關聯性。

以上兩者都是藉由曲線配適法所得的。由於 $\rho_{SBS D}^{(1)}(i, \Phi(j))$ 為位置 i 、

$\Phi(j)$ 的函數且能代表 $L_{SBS D}^{(ext),(1)}(u_i)$ 和 $\tilde{y}_{\Phi(j)}^s$ 之間的關聯性，因此可將

$\rho_{SBS D}^{(1)}(i, \Phi(j))$ 當作成本函數，並根據 BCJR 演算法中對於事前資訊獨立假設的要求，找出可讓函數 $\rho_{SBS D}^{(1)}(i, \Phi(j))$ 為最小值的 $\Phi(j)$ 即為 $\Phi(i)$ 。

換句話說，此統計模型輸出的值若越小，代表 $L_{SBS D}^{(ext),(n)}(u_i)$ 與 $\tilde{y}_{\Phi(i)}^s$ 越無

關聯，越能符合 BCJR 演算法的要求。

4.3 訊源解碼器資訊關聯性的分析

為了設計適用於 ISCD 的交錯器，我們將依 4.2 節所述方式，針對訊源解碼器的資訊關聯性進行分析。

在接收端，訊源解碼器可利用量化索引序列的殘餘冗息來隱藏傳輸錯誤，因此在疊代解碼中訊源解碼器的事前資訊是屬於索引層級。但是在傳統的疊代訊源通道解碼系統裡，通道解碼輸出的額外資訊卻屬於位元層級，因此額外資訊的交換必須先從位元層級轉至索引層級。然而層級轉換的過程中，通道解碼額外資訊一般是假設互為獨立的。舉例而言，1 次疊代後索引後驗機率(式 2.22)裡的事前機率

$p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n')$ 直接由 $p(\mathbf{u}_t = n | \mathbf{u}_{t-1} = n') \cdot \prod_{m=1}^M p_{CD}^{(ext)}(u_t(m) | \mathbf{u}_t = n)$ 來取代，就是基於索引 \mathbf{u}_t 組成位元 $\{u_t(1), u_t(2), \dots, u_t(M)\}$ 的額外資訊彼此獨立的假設。所以接下來，我們將實際觀察通道解碼額外資訊間的關聯性是否與上述假設相符合。

令交錯器長度內的原始序列 $\mathbf{U}_1^T = \{u_1(1), u_1(2), \dots, u_1(m), \dots, u_T(M)\} = \{u_1, u_2, \dots, u_M, \dots, u_L\}$ ， $L = M \cdot T$ 。設索引 \mathbf{u}_i 的組成位元 $\{u_i(1), u_i(2), \dots, u_i(M)\} \in \mathbf{U}_1^T$ ，則 $\{u_i(1), u_i(2), \dots, u_i(M)\} = \{u_{i_1}, u_{i_2}, \dots, u_{i_M}\}$ ，其中 $i_1, i_2, \dots, i_M = 1, 2, \dots, L$ 、 $i_1 \neq i_2 \neq \dots \neq i_M$ 。而經交錯器作用後的 u_{i_m} 等於 $y_{\Phi(i_m)}^s$ 。利用(4.1)式，定義 $\rho^{(n)}(L_{CD}^{(ext),(n)}(y_{\Phi(i_m)}^s), L_{CD}^{(ext),(n)}(y_i^s))$ 為第 n 次

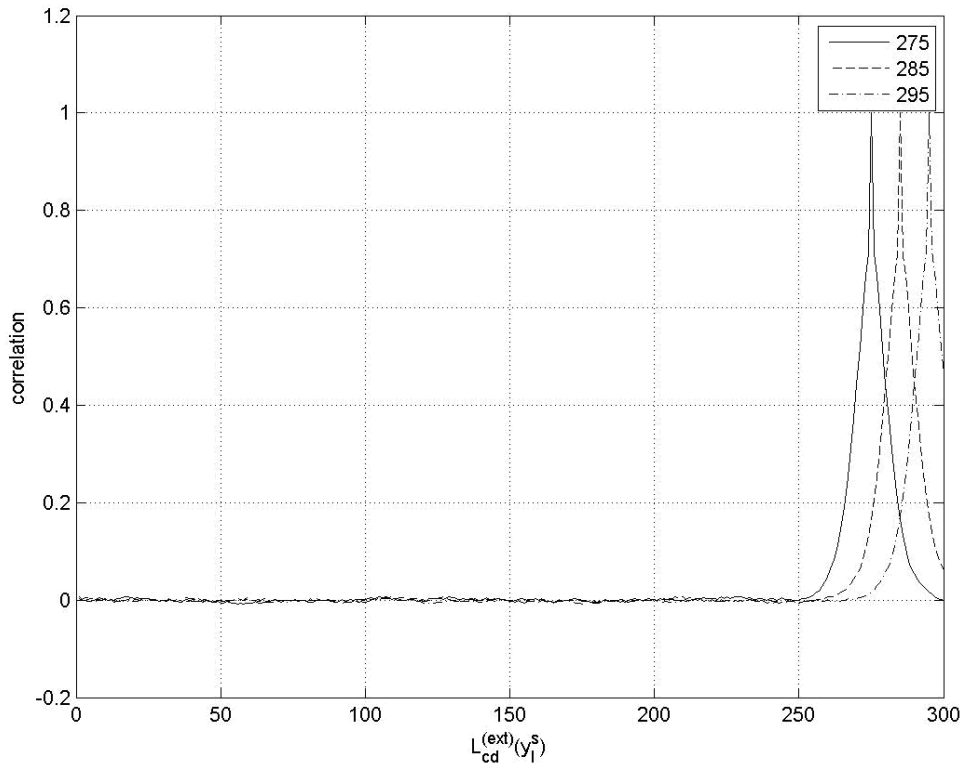


圖 4.5(a) $\rho^{(0^+)}(L_{CD}^{(ext),(0^+)}(y_{\Phi(i_m)}^s), L_{CD}^{(ext),(0^+)}(y_l^s))$, $i_m = \{148, 149, 150\}$

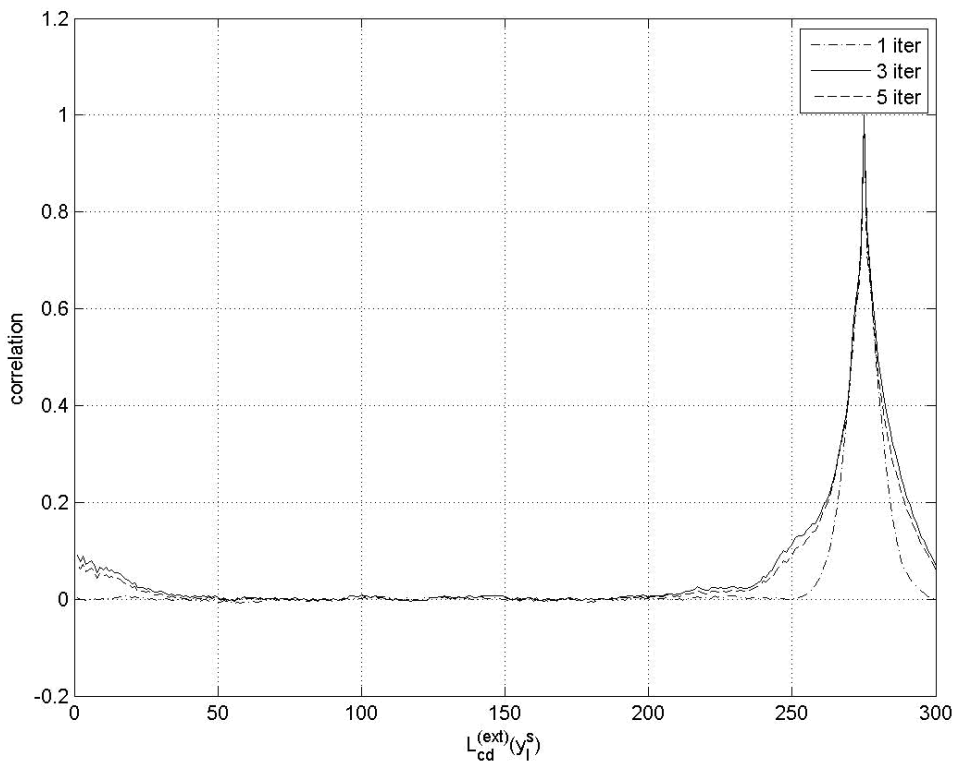


圖 4.5(b) $\rho^{(n^+)}(L_{CD}^{(ext),(n^+)}(y_{\Phi(148)}^s), L_{CD}^{(ext),(n^+)}(y_l^s))$, $n = 1, 3, 5$

疊代解碼時， $L_{CD}^{(ext),(n)}(y_{\Phi(i_m)}^s)$ 與 $L_{CD}^{(ext),(n)}(y_l^s)$ 之間的關聯性， $l=1,2,\dots,L$ 。

在與 3.1.1 節實驗環境的設定且搭配 $10*30$ 的區塊交錯器下，圖 4.5

(a) 所顯示的是 $\rho^{(0^+)}(L_{CD}^{(ext),(0^+)}(y_{\Phi(i_m)}^s), L_{CD}^{(ext),(0^+)}(y_l^s))$ ， $i_m = \{148, 149, 150\}$ 。

如同 4.2.1 節所述， u_{i_m} 恰好為 u_{50} 的組成位元。在本圖中因區塊交錯

器的影響，使得 $\Phi(i_m) = \{275, 285, 295\}$ ，並且清楚的觀察到

$L_{CD}^{(ext),(0^+)}(y_{\Phi(i_m)}^s)$ 不但與鄰近的 $L_{CD}^{(ext),(0^+)}(y_l^s)$ 有較大的關聯性外，彼此之間

亦存在不小的關聯性，例如 $\rho^{(0^+)}(L_{CD}^{(ext),(0^+)}(y_{285}^s), L_{CD}^{(ext),(0^+)}(y_{295}^s)) \approx 0.5$ 。圖

4.6 則為 $\rho^{(n^+)}(L_{CD}^{(ext),(n^+)}(y_{\Phi(148)}^s), L_{CD}^{(ext),(n^+)}(y_l^s))$ ， $n=1,3,5$ 。明顯的，隨著

疊代次數 n 增加，曲線涵蓋的範圍也隨之擴大，代表 $L_{CD}^{(ext),(n^+)}(y_{\Phi(148)}^s)$ 與

更多的 $L_{CD}^{(ext),(n^+)}(y_l^s)$ 產生關聯。綜合上述現象可歸納出 $L_{CD}^{(ext),(n^+)}(y_{\Phi(i_m)}^s)$ 彼

此間不但存有關聯性的現象外，且關聯性又將隨疊代次數增加而上升，這違背 u_{50}

組成位元 $\{u_{50}(1), u_{50}(2), u_{50}(3)\}$ 的通道解碼額外資訊彼此

獨立之假設。

由以上範例，我們了解到解交錯器的設計確實會影響訊源解碼輸入

端資訊的關聯性。在本文的疊代訊源通道解碼系統中，對於索引 u_t

組成位元 $\{u_t(1), u_t(2), u_t(3)\} = \{u_{i_1}, u_{i_2}, u_{i_3}\}$ ，解交錯器的設計應盡量避免

$\{\Phi(i_1), \Phi(i_2), \Phi(i_3)\}$ 彼此過於靠近，如此方能讓 $\{u_t(1), u_t(2), u_t(3)\}$ 的事

前通道額外資訊可互相保持不相關。並根據 [10][25]，將輸入至訊

源解碼器的事前資訊視為高斯程序，因此 $L_{CD}^{(ext),(0^+)}(\Phi(\tilde{x}_t(1)))$ 、

$L_{CD}^{(ext),(0^+)}(\Phi(\tilde{x}_t(2)))$ 、 $L_{CD}^{(ext),(0^+)}(\Phi(\tilde{x}_t(3)))$ 可相互獨立，滿足訊源解碼演算法理論上的假設。

4.4 交錯器的最佳化設計

本節我們將以[19]裡設計交錯器的概念為基礎，融合 4.3 節中對於解交錯器的要求，發展出適合整個系統的信號交錯器與解交錯器設計。然而[19]所提出的交錯器設計，卻只使用單一成本函數來對系統設計(4.2 式)，無法滿足疊代訊源通道解碼的需求。以下將對此現象進行說明。假設 $\mathbf{u}_i, \mathbf{u}_j \in U_1^{T'}$ ，且其組成位元分別為 $\{u_{i_1}, u_{i_2}, \dots, u_{i_M}\}$ 與 $\{u_{j_1}, u_{j_2}, \dots, u_{j_M}\}$ 。由實驗我們發現 u_{i_m} 和 u_{j_m} 有相同的 ρ_1 ， $m = 1, 2, \dots, M$ ，並與其他組成位元的 ρ_1 相異。而相異的 ρ_1 將導致不同 $\rho_{SBSD}^{(1)}(i, \Phi(j))$ 的產生。所以我們提出下列方式進行改良：

1. 以 exponential 為底的指數函數在疊代訊源通道解碼的信號交錯器設計已不敷使用，因此我們經由模擬經驗，決定採用高斯函數(Gauss function)的線性組合來建立 $\rho_1(i, j)$ 的統計模型。
2. 於(4.2)式中使用 $\rho_{1,i'}(i, j)$ 替代 $\rho_1(i, j)$ ，其中 $i' = \text{mod}(i, M)$ ，以解決不同 ρ_1 的問題。

因此(4.2)式可改寫為

$$\rho_{SBSD}^{(1)}(i, \Phi(j)) = \rho_{1,i'}(i, \Phi(j)) + \sum_{\Phi(g)=1}^L \rho_{1,i'}(i, g) \rho_2(\Phi(j), \Phi(g)) ,$$

$$i' = \text{mod}(i, M) \quad (4.3)$$

其中

$$\rho_{1,i'}(i, \Phi(j)) = \sum_{\sigma} a_{\sigma,i'}^{(1)} \exp \left[\frac{-((i + M - i' - \Phi(j)) - b_{\sigma,i'}^{(1)})^2}{(c_{\sigma,i'}^{(1)})^2} \right]$$

$$, \Phi(j) \neq i, \Phi(j) \leq (i + M - i')$$
(4.4)

$$\rho_2(\Phi(j), \Phi(g)) = a^{(2)} \cdot \exp[b^{(2)} \cdot |j - g|], j \neq g \quad (4.5)$$

在(4.3)中令 $\Phi(i) = \Phi(j)$ ，則 $L_{SBS D}^{(ext),(1)}(u_i)$ 與 $\tilde{y}_{\Phi(i)}^s$ 相關係數的統計模型

$$\rho_{SBS D}^{(1)}(i, \Phi(i)) = \sum_{g=1}^L \rho_{1,i'}(i, g) \rho_2(\Phi(i), \Phi(g)) \quad (4.6)$$

接下來為了滿足4.3節中提出的解交錯器設計，我們將藉由以下交錯器的設計來間接滿足對解交錯器的要求。令

$$u_i = \{u_{i_1}, u_{i_2}, \dots, u_{i_M}\} \in U_1^{T'}$$

，則其組成位元的位置分別為 $\{i_1, i_2, \dots, i_M\}$ 。觀察4.6式中的 $\rho_{1,i'}$ 可發現其與差值 $(i - g)$ 大小呈負相關的特性，且 ρ_2 也有相同之情況。因此一次對 $\{i_1, i_2, \dots, i_M\}$ 進行交錯器的設計，可讓 $\{\Phi(i_1), \Phi(i_2), \dots, \Phi(i_M)\}$ 彼此保持一段距離。總結以上推論，我們已建立了資訊關聯性與交錯器設計兩者之間的關聯，所以可使用(4.6)式來組成適用於疊代訊源通道解碼中交錯器設計的成本函數。其目的是依通道及訊源解碼演算法的要求，找出最佳交錯器使得成本函數達到最小值。因此基於資訊關聯性的交錯器設計可轉換成一個最佳化的問題：

已知： v_t 是一階高斯馬可夫程序模擬其相關因子 $r=0.95$ ，位元數

$M=3$ 的 Lloyd-Max 純量量化器，交錯器的長度等於 300 位

元，訊雜比為 -3dB 的通道雜訊以高斯程序產生，

$$\text{通道編碼矩陣為 } G(D) = \begin{bmatrix} g^{(1)}(D) \\ g^{(0)}(D) \end{bmatrix} = \begin{bmatrix} 1 + D^2 \\ 1 + D + D^2 \end{bmatrix}。$$

令索引 u_t 的組成位元為 $\{u_t(1), u_t(2), u_t(3)\}$ ， $\{i_1, i_2, i_3\}$ 依序代表該組成位

元的位置，而 $\{\Phi(i_1), \Phi(i_2), \Phi(i_3)\}$ 為 $\{i_1, i_2, i_3\}$ 經交錯器置換後的位置，

則

$$\{\Phi(i_1), \Phi(i_2), \Phi(i_3)\}_{opt} = \arg \min_{\Phi(i_1), \Phi(i_2), \Phi(i_3) \in \Psi} \sum_{m=1}^3 \rho_{SBS D}^{(1)}(i_m, \Phi(i_m)) \quad (4.6)$$

其中集合 $\Psi = \{1, 2, \dots, 300\}$ 。

最後，圖 4.6 為最佳化交錯器設計的流程，然而其具體實現必須留意

下列重點：

1. 經交錯器置換後的位置 ψ 為 M 筆相異數字的排列，而對任意 M 筆

相異數字的組合總共有 $M!$ 種排列的可能性。假設一個交錯器長度

為 L ，則要設計第一筆索引的組成位元之 ψ_{opt} 需要嘗試 $C_M^L (M!)$ 種

組合，這將造成龐大的運算量與延遲。因此實際設計時，對任意

一組 ψ 只考慮一種組合 $\psi = \{\Phi(i_1), \Phi(i_2) \dots \Phi(i_M)\}$ ，並且

$\Phi(i_1) < \Phi(i_2) < \dots < \Phi(i_M)$ ，以降低運算量。

2. 為了更進一步降低運算量，我們可以從 $\rho_{SBS D, i'}^{(1)}(i, \Phi(i))$ 的函數特性

進行觀察，並且發現其內部包含高斯函數的運算。每次進行交錯器設計時就必須重複把 $(i-g) \cdot |\Phi(i) - \Phi(g)|$ 可能發生的組合帶入

$\rho_{SBSD,i}^{(1)}(i, \Phi(i))$ 中，執行 exponential 指數運算後找出最小

$\rho_{SBSD,i}^{(1)}(i, \Phi(i))$ ，如此將使得整個設計時間被拉長。可是若仔細探究

$(i-g) \cdot |\Phi(i) - \Phi(g)|$ 可能發生的組合即可發現 $(i-g) \cdot$

$|\Phi(i) - \Phi(g)| = \{1, 2, \dots, L\}$ ，這是因為 $\rho_{1,i}(i, g)$ 、 $\rho_2(\Phi(j), \Phi(g))$ 性質

的緣故。所以我們在進行交錯器設計演算法前，可事先求得所有

$\rho_{1,i}(i, g)$ 、 $\rho_2(\Phi(j), \Phi(g))$ 並且將之儲存，提供演算法執行過程中

計算 $\rho_{SBSD}^{(1)}(i, \Phi(i)) = \sum_{\substack{g=1 \\ g \neq i}}^L \rho_{1,i}(i, g) \rho_2(\Phi(i), \Phi(g))$ (註：4.6 式)，如此可

避免重複進行多次方指數運算，縮短設計所花費的時間。

3. 本方法亦可適用於索引層級 ISCD 的交錯器設計，只需將(4.1)以索引層級的互消息 $I(\tilde{\mathbf{u}}_t; \mathbf{u}_t | 0)$ (symbol-level mutual information)[11]來替代，

$$\begin{aligned}
 I(\tilde{\mathbf{u}}_t; \mathbf{u}_t | 0) &= E_{\tilde{\mathbf{u}}_t} \left\{ \log_2 \frac{p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = 0)}{p(\tilde{\mathbf{u}}_t)} \right\} \\
 &= M - E_{\tilde{\mathbf{u}}_t} \left\{ \log_2 \left(1 + \sum_n \frac{p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = n)}{p(\tilde{\mathbf{u}}_t | \mathbf{u}_t = 0)} \right) \right\}
 \end{aligned} \tag{4.7}$$

其中 M 為量化器位元數目。

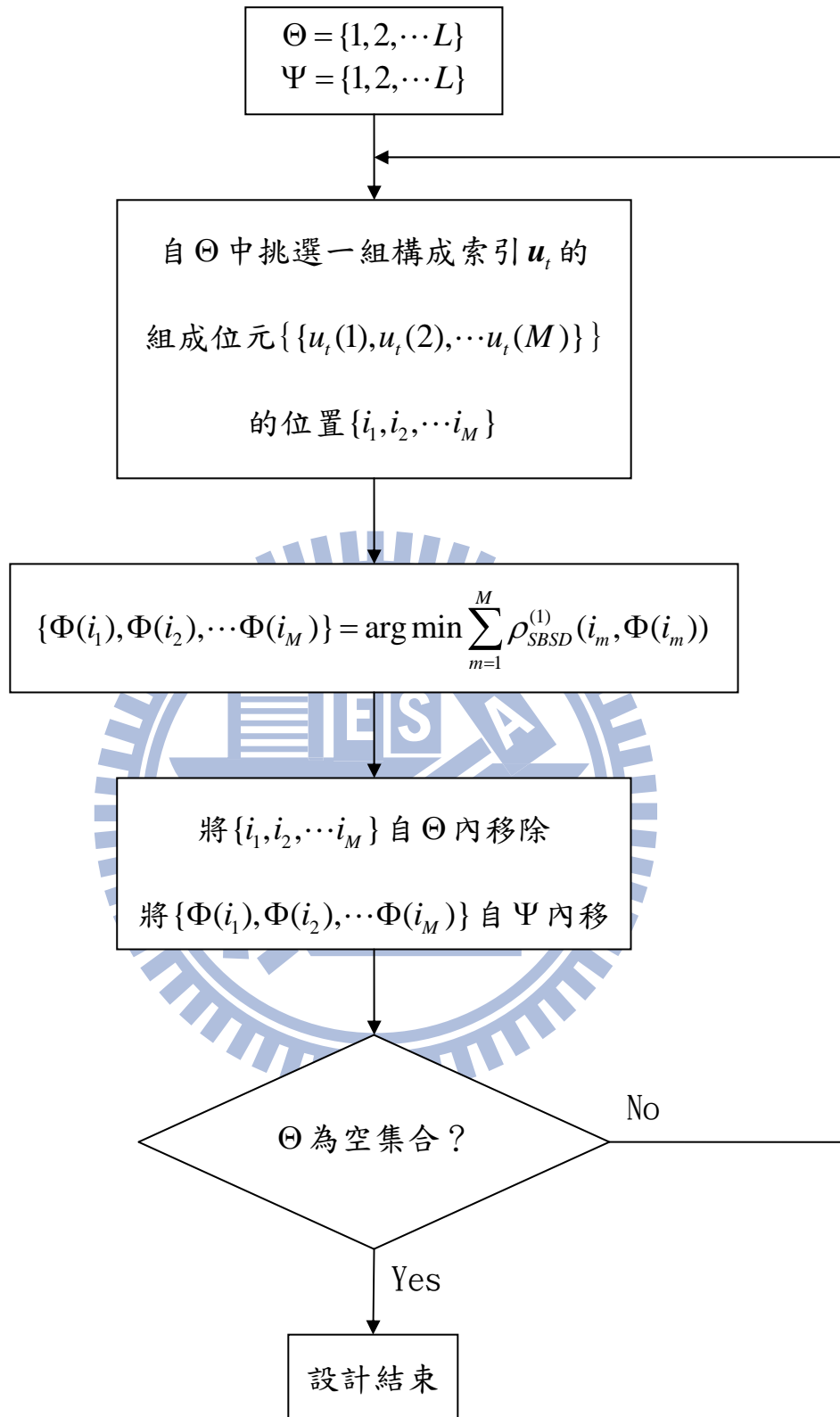


圖 4.6 交錯器設計流程圖

第五章 實驗模擬與結果分析

在前面的章節已介紹索引指定與交錯器的原理與設計，接下來將根據我們所提出的最佳化設計來進行實驗和驗證。首先 5.1 節先說明基本實驗環境的設定；5.2 節進行基於資訊相關性的索引指定設計，並且將與目前最佳的基於額外資訊轉換圖的索引指定設計來比較；5.3 節則是實現基於疊代解碼效益的交錯器設計，且將設計出來的解碼器與數種常用的解碼器進行系統效能的比較。而 5.2、5.3 節中，我們將同時呈現位元與索引不同層級的疊代訊源通道解碼系統的設計結果，並加以比較。

5.1 實驗環境設定

本節主要敘述後續實驗的基本實驗設定，而對於每個實驗所需的設定細節將在往後的實驗過程中逐次進行說明。

整體系統架構如圖 2.1。其中訊號源 v_t 以變異數 $\sigma_v^2 = 1$ 且相關係數 $r = 0.95$ 的一階高斯馬可夫程序產生，如下式

$$v_t = r \cdot v_{t-1} + n_t \quad (5.1)$$

當中 n_t 是平均值等於 0 的白色高斯雜訊。

量化器採取量化位元數 $M = 3$ 的 Lloyd-Max 純量量化器；交錯器長度

$L = 300$ 位元；通道編碼器選擇遞迴系統迴旋碼且其生成矩陣

$$G(D) = \begin{bmatrix} 1 & \frac{g^{(1)}(D)}{g^{(0)}(D)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1+D^2}{1+D+D^2} \end{bmatrix} \quad (5.2)$$

而傳送前的調變為二進制相移鍵控，可將位元"0" $\rightarrow +1$ 、位元"1" $\rightarrow -1$ 。
最後傳送通道雜訊以無記憶可加性白色高斯雜訊模擬，其平均值與變異數分別為0及 $N_0/2$ 。

特別要注意的是以上設定同時適用於位元層級、索引層級的 ISCD，但是在索引層級系統中交錯器的部分要轉換成索引層級的交錯器，且長度 $L' = L/M = 100$ 個索引。

5.2 索引指定設計在 ISCD 效能之比較

實驗一 位元層級系統索引指定設計之比較

本實驗先把120000個信號源的取樣點送入系統中，再搭配隨機交錯器 ($S = 4$)，利用第三章基於資訊關聯性的索引指定設計流程來找尋最佳的索引指定，其結果如表 5.1。實驗一將比較在位元層級疊代訊源通道解碼上，自然二位元碼(NBC)、基於額外資訊相關性設計的索引指定(BIA_C)和基於額外資訊轉換圖設計的索引指定(BIA_E)三者的系統效能，結果如圖 5.1。除此亦呈現 BIA_C 與 BIA_E 兩種索引指定，分別於額外資訊轉換圖與額外資訊相關特性圖。

實驗結果

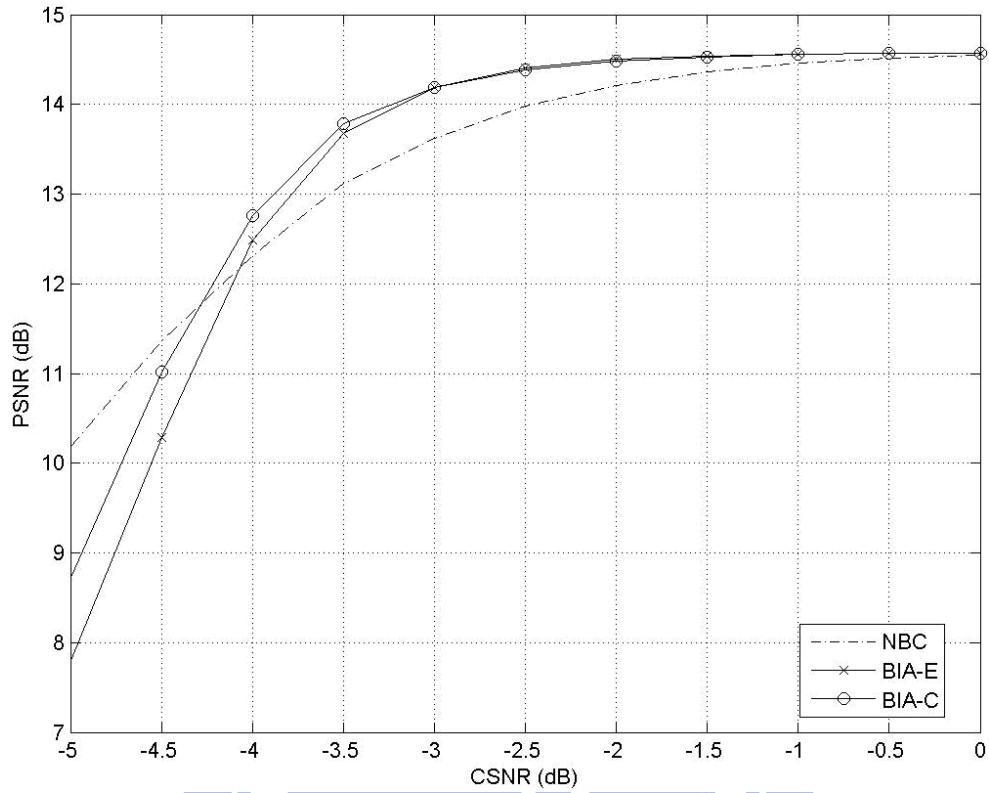


圖 5.1 位元層級不同索引指定的效能比較(10 次疊代)

表 5.1 位元層級不同索引指定的比較

NBC		BIA_C		BIA_E	
0	000	2	010	0	000
1	001	7	111	5	101
2	010	4	100	6	110
3	011	3	011	3	011
4	100	0	000	4	100
5	101	5	101	1	001
6	110	6	110	2	010
7	111	1	001	7	111

結果分析與討論

我們首先針對 $CSNR = -3 \text{ dB}$ 之情況將 BIA_C 在不同疊代過程中的額外資訊相關特性圖顯示於圖 5.2。可清楚的看到其相關係數 $\rho_n'(\tau)$ 隨著疊代次數增加而越接近原始傳送序列的自相關函數 $\rho(\tau)$ ，同時這也間接代表接收資料正確性的提升。而圖 5.3 則為不同索引指定在 10 次疊代後的額外資訊相關特性圖，從中發現 BIA_C 的 $\rho_{10}'(1)$ 最接近 $\rho(1)$ ，BIA_E 次之，NBC 差距最多。並且各索引指定的 $\rho_{10}'(\tau)$ 除了 $\tau=1$ 外，在其他 τ 值也有類似的優劣現象。因此我們預期 BIA_C 在 ISCD 系統有最佳的效能。

而圖 5.4(a)則是以額外資訊轉換圖和軌跡圖的角度觀察。首先可看到無論是 BIA_C 或 BIA_E，它們的停止點幾乎都在相同位置。代表在 $CSNR = -3 \text{ dB}$ 時額外資訊轉換圖認為理論上兩組索引指定在系統收斂時能夠達到相近的效能。前人提出的 BIA_E 根據較大的 $I_{SBSD}^{(ext)} \big|_{I_{SBSD}^{(apri)}=1}$ 來認定所對應到的索引指定有較佳的解碼效能。而 BIA_C 及 BIA_E 所對應的 $I_{SBSD}^{(ext)} \big|_{I_{SBSD}^{(apri)}=1}$ 分別為 0.89618 和 0.896789，似乎有違於圖 5.1 中 BIA_C 有較佳表現之事實。這將使得此 BIA_E 演算法無法找到最佳的索引指定。此外它們的軌跡圖也存在明顯的差異：BIA_C 所能交換到的互消息多於 BIA_E。主要原因來自於訊源解碼器的額外資訊轉換圖曲度的不同，如圖 5.4(b)放大圖所示。

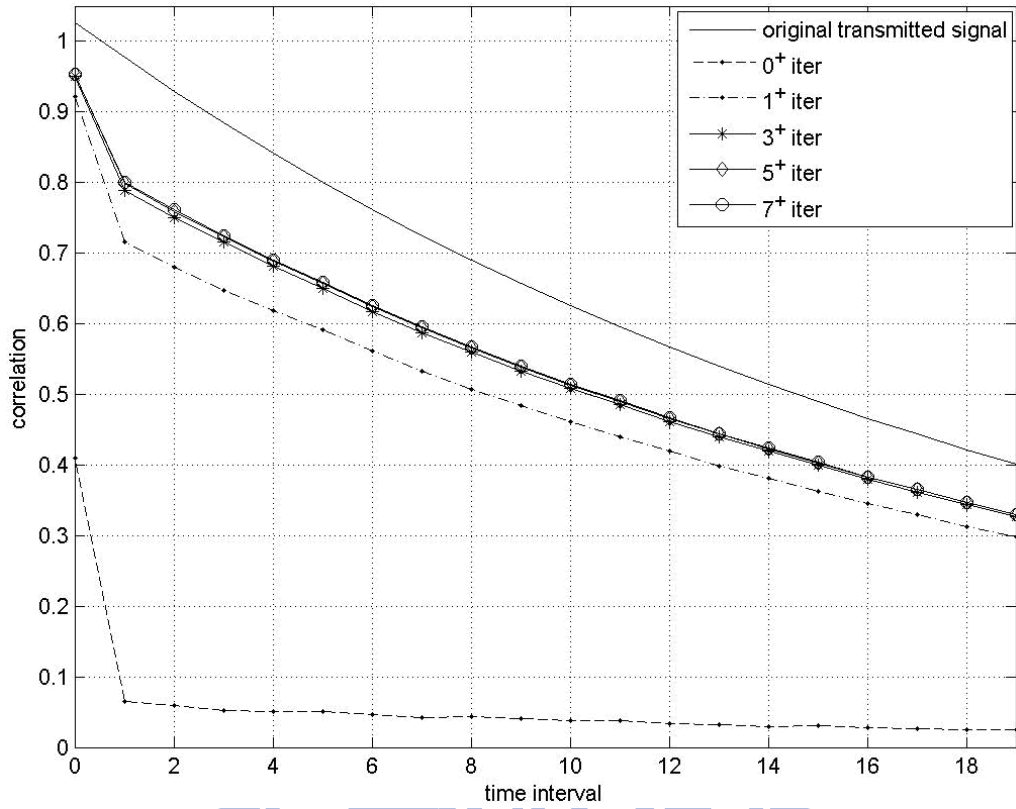


圖 5.2 BIA_C 的額外資訊相關特性圖 (CSNR = -3 dB)

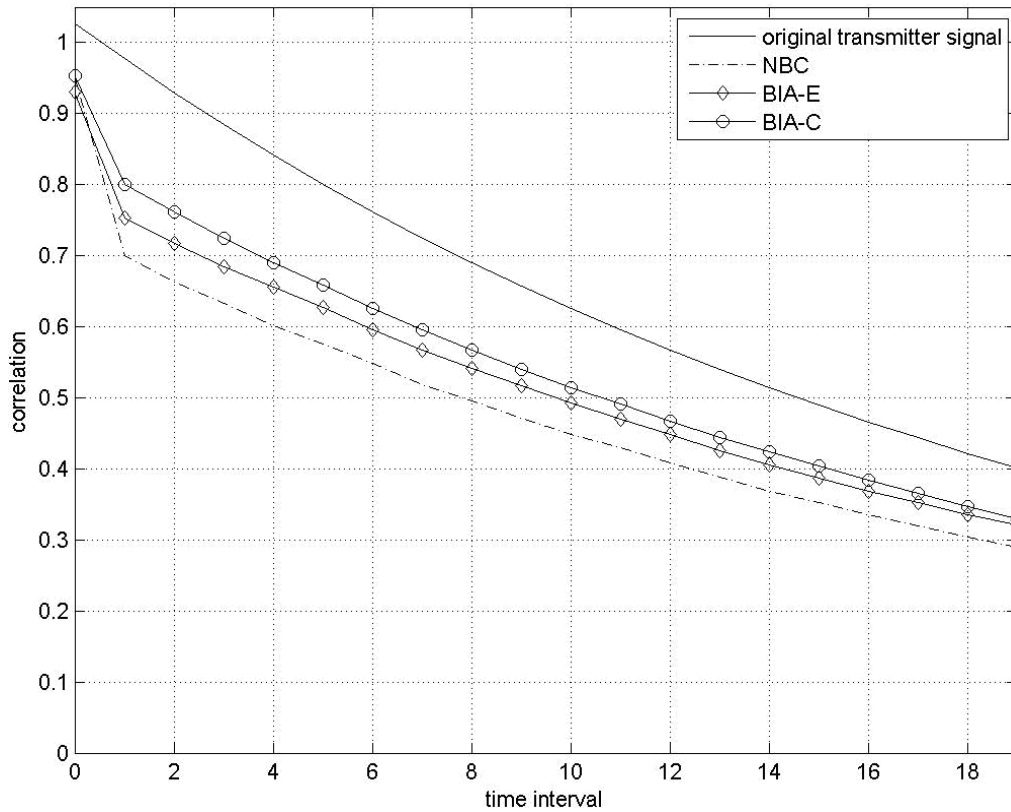


圖 5.3 不同索引指定之額外資訊相關特性比較 (CSNR = -3 dB)

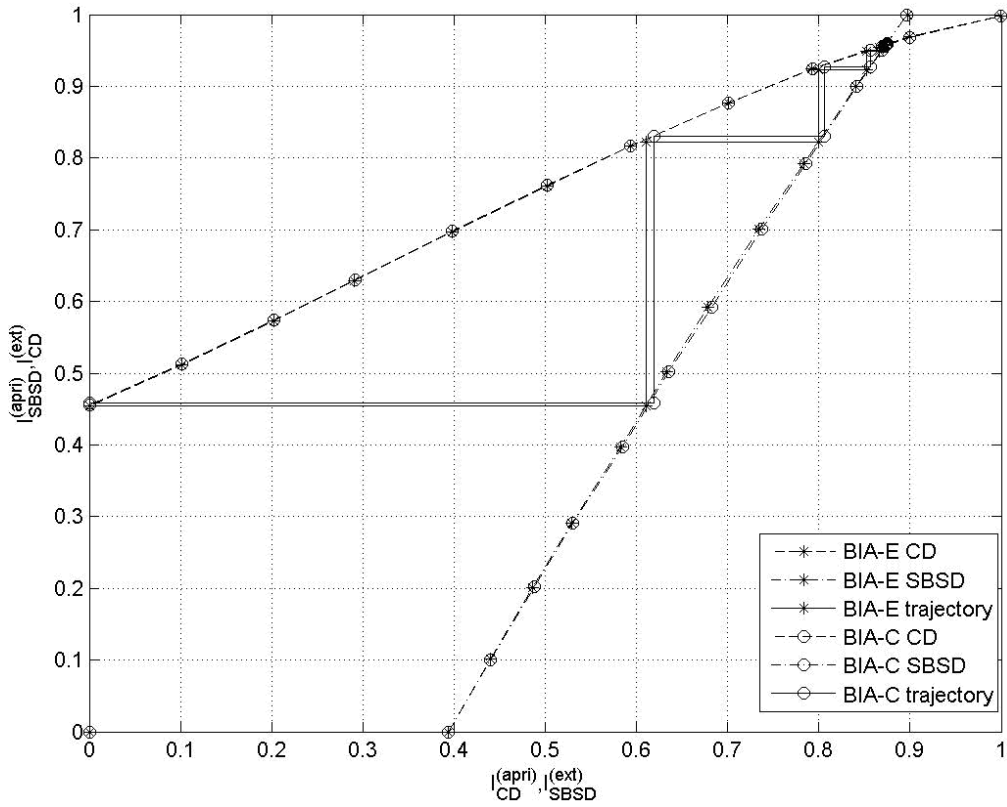


圖 5.4(a) 額外資訊轉換圖比較 ($CSNR = -3 \text{ dB}$)

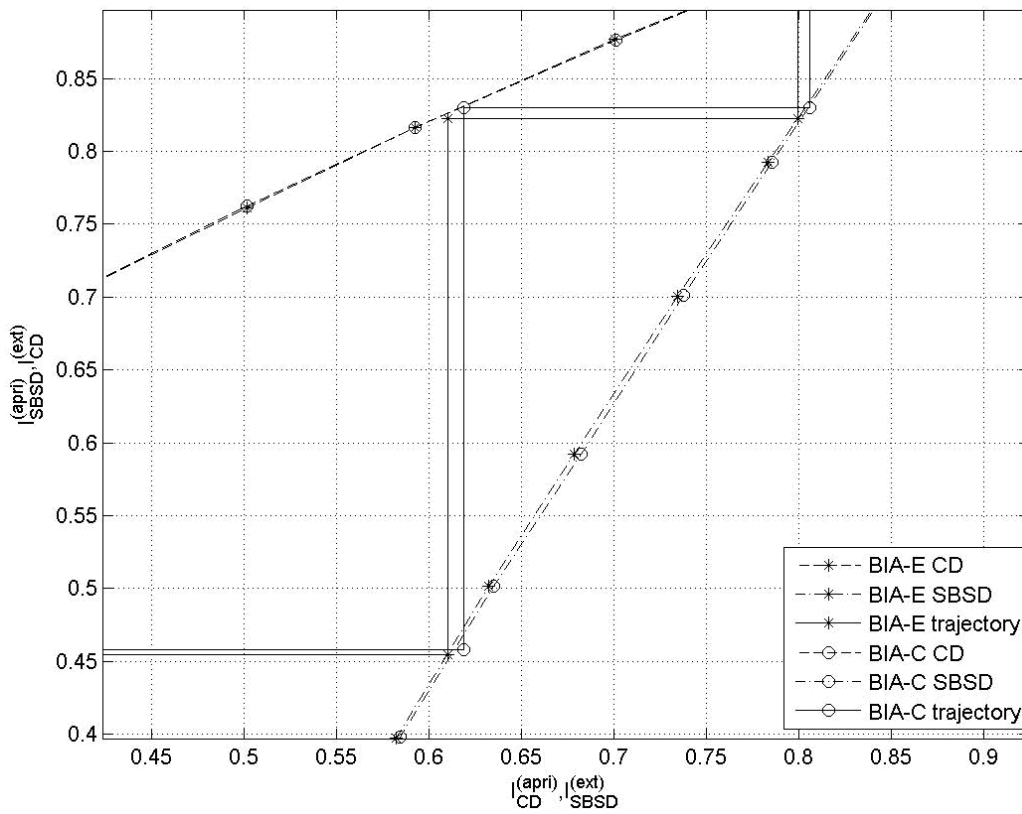


圖 5.4(b) 額外資訊轉換圖(放大)

最後，不同索引指定收斂時的系統效能顯示於圖 5.1。如之前所預測的，在 $SNR = -3 \text{ dB}$ 下 BIA_C 及 BIA_E 都有相近的效能且皆優於 NBC 的表現。而在 $SNR < -3 \text{ dB}$ 的通道環境下兩者效能逐漸變差，但是 BIA_C 的系統效能下降的幅度卻是較小的，這是因為它在疊代解碼過程中解碼器能交換較多互消息的緣故。相較之下，BIA_E 設計過程中只單純只考慮 $I_{SBS D}^{(ext)} |_{I_{SBS D}^{(pri)}=1}$ 的大小，更無法像 BIA_C 考量整個系統效能來進行設計。

實驗二 索引層級系統索引指定設計之比較

將 120000 個信號源的取樣點送入系統中，並搭配隨機交錯器 ($S=4$)，利用第三章基於資訊關聯性的索引指定設計流程來找尋最佳的索引指定，結果如表 5.2 所示。此實驗將在索引層級疊代訊源通道解碼上，比較自然二位元碼(NBC)與基於額外資訊相關特性設計的索引指定(SIA_C)的系統效能，其結果如圖 5.5。最後再使用額外資訊轉換圖和額外資訊相關特性圖分析此兩索引指定的差異，同時藉此說明在索引層級系統上基於額外資訊轉換圖設計索引指定的困難。

實驗結果

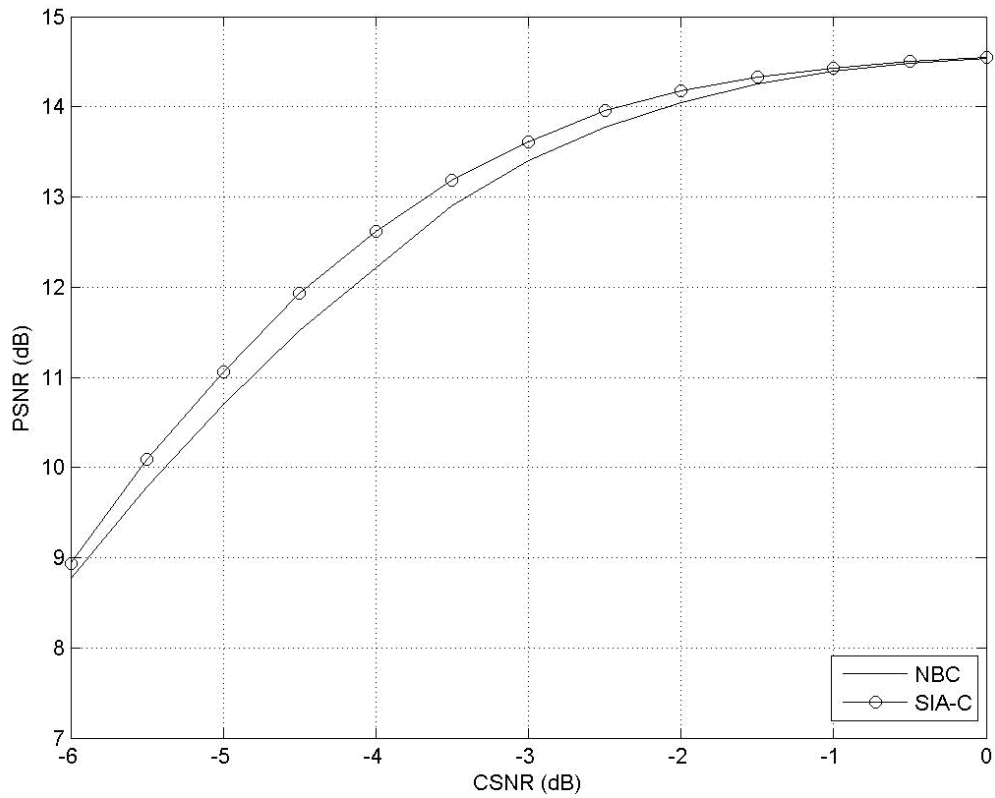


圖 5.5 索引層級兩種索引指定的效能比較(10 次疊代)

表 5.2 索引層級不同索引指定的比較

NBC		SIA_C	
0	000	0	000
1	001	5	101
2	010	2	010
3	011	4	100
4	100	3	011
5	101	1	001
6	110	6	110
7	111	7	111

結果分析與討論

在圖 5.5 中，可看到無論處於何種通道環境下，SIA_C 都有較好的表現。相較於 NBC，在 $CSNR = -6 \text{ dB}$ 時好約 0.17 dB ，而在 $SNR = -4.5 \text{ dB}$ 時也高出 0.42 dB 。

圖 5.6 是以額外資訊轉換圖對兩種索引指定的觀察。若以額外資訊轉換圖觀測停止點有其困難的，因為它們停止點的位置非常靠近。另一方面，NBC 與 SIA_C 在軌跡圖的收斂點卻又有明顯的不同，其座標分別為 $(1.503, 2.221)$ 、 $(1.513, 2.304)$ ，這意味著利用額外資訊轉換

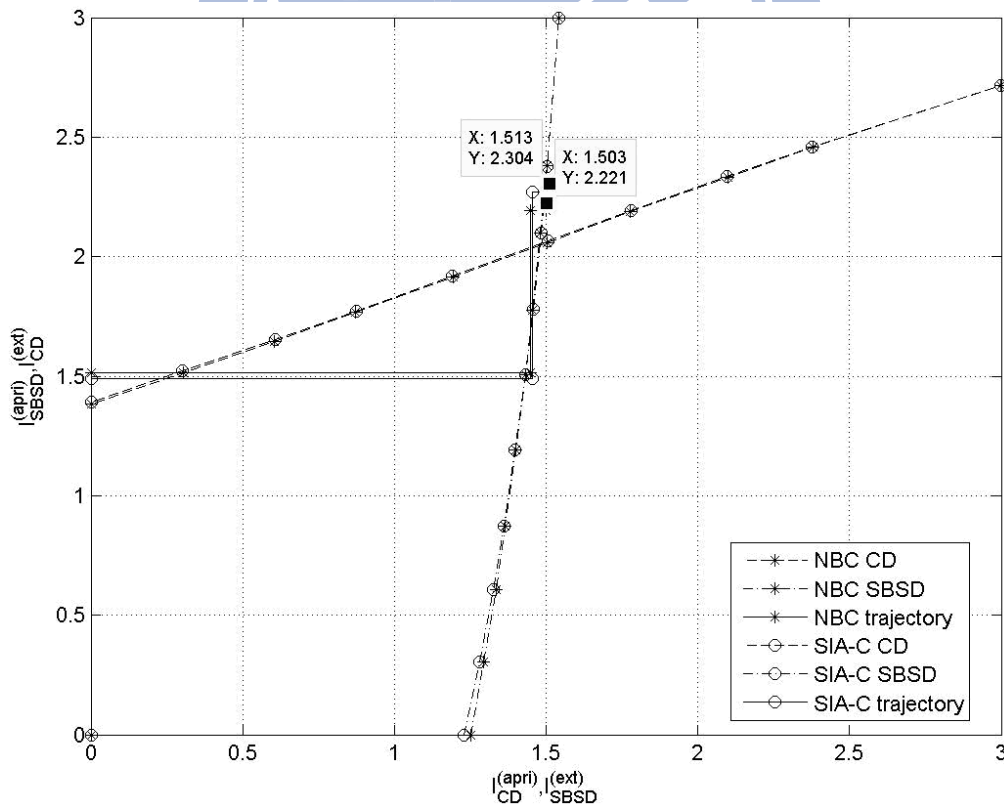


圖 5.6 相異索引指定的額外資訊轉換圖、軌跡圖之比較

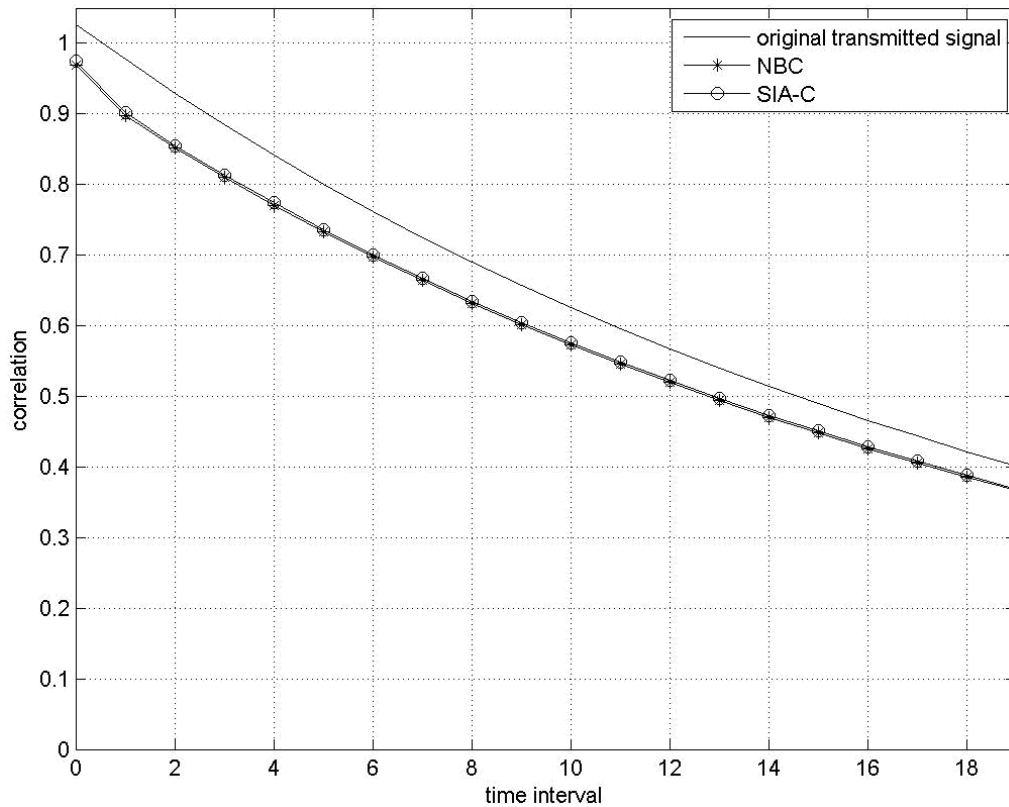


圖 5.7 不同索引指定的資訊相關性變化圖之比較

圖來設計索引指定是很困難的工作。在索引層級 ISCD 架構中，目前尚未有基於額外資訊轉換圖的設計索引指定方式被提出，且若要避免實驗一裡以 $I_{SBSD}^{(ext)} |_{I_{SBSD}^{(apri)}=1}$ 進行索引指定設計的缺陷，一個直覺的想法就是直接用該索引指定所對應的停止點位置來進行最佳化設計。問題是找尋停止點的過程所需運算量是龐大的，我們必須先求出通道解碼與訊源解碼額外資訊轉換曲線，再根據兩者的產生模型估算出停止點。而 NBC 與 SIA_C 的停止點又相當靠近，所以額外資訊轉換曲線的實測值與產生模型不允許有太大的誤差，這無形中又增加設計上的難度。相對而言，較可行的解決方案是基於額外資訊相關特性而設計的索引

指定。如圖 5.7 的額外資訊相關特性圖，雖然 NBC 和 SIA_C 的變化差距不大，但是由於不用模擬出每個索引指定的額外資訊相關特性圖的產生模型，只需比較一階自相關係數 $\rho_k^*(1)$ 差異，即可準確地設計出最佳化索引指定。



5.3 交錯器設計在 ISCD 之效能比較

實驗三 不同交錯器在位元層級系統上的實現

本章節中我們將輸入 $100 * L$ 個訊源取樣值，先採用 NBC 與區塊交錯器，並且重複 100 次來求得 $CSNR = -3 \text{ dB}$ 時 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{y}_i^s)$ ， $i=148,149,150$ 。接下來再根據 4.4 節分別算出三者的統計數學模型，以此作為交錯器設計的成本函數。最後並將分別比較區塊交錯器、隨機交錯器和基於疊代效益設計之交錯器（根據 [19] 所設計之交錯器 inter1、依本論文提出的設計方法而設計之交錯器 inter2）的系統效能，如圖 5.8。

實驗結果

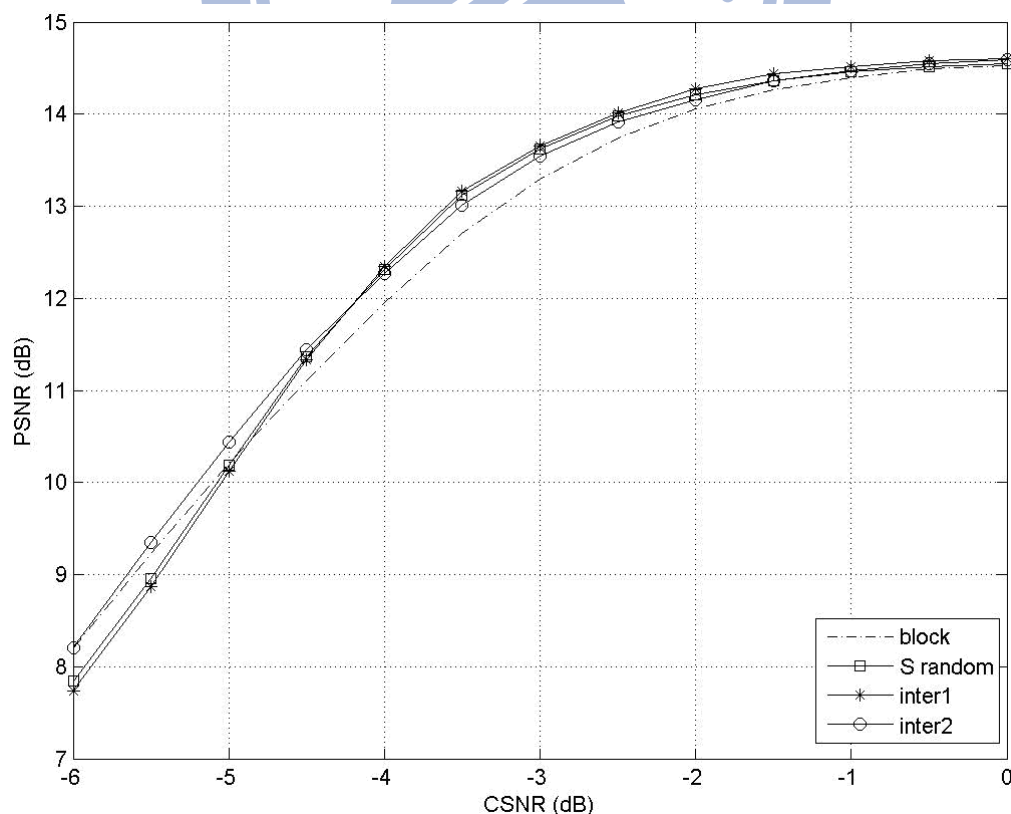


圖 5.8 不同交錯器的效能比較（第 10 次疊代）

結果分析與討論

首先，圖 5.9 至圖 5.11 同時顯示了 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(u_{148}), \tilde{y}_i^s)$ 、 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(1)}(u_{149}), \tilde{y}_i^s)$ 、 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(1)}(u_{150}), \tilde{y}_i^s)$ 與其各自對應的 $\rho_{SBS D}^{(1)}(i, \Phi(j))$ 。雖然每個 $\rho_{SBS D}^{(1)}(i, \Phi(j))$ 與 $\rho_{SBS D}^{(n)}(L_{SBS D}^{(ext),(1)}(u_i), \tilde{y}_i^s)$ 在某些地方有誤差，但是都在可接受的範圍之內。而圖 5.8 則是 NBC 搭配不同交錯器下系統效能的比較。相較於隨機交錯器，區塊交錯器在 $CSNR > -4.5 \text{ dB}$ 的通道環境下表現較差，但是於 $CSNR < -4.5 \text{ dB}$ 的情況下卻比較好，並且隨著 $CSNR$ 的下降，兩者之間的差距也跟著拉大。而 inter1 所表現出的趨勢則和隨機交錯器類似。inter2 的表現最為理想：在 $CSNR > -4.5 \text{ dB}$ 的通道環境下能夠優於區塊交錯器且接近隨機交錯器的效能；另一方面，在 $CSNR < -4.5 \text{ dB}$ 的通道環境下又能擁有最好的系統效能。所以印證了同時考量通道與訊源解碼器的交錯器設計，無論在何種通道環境下都能有不錯的疊代解碼效益，使得系統能最優。

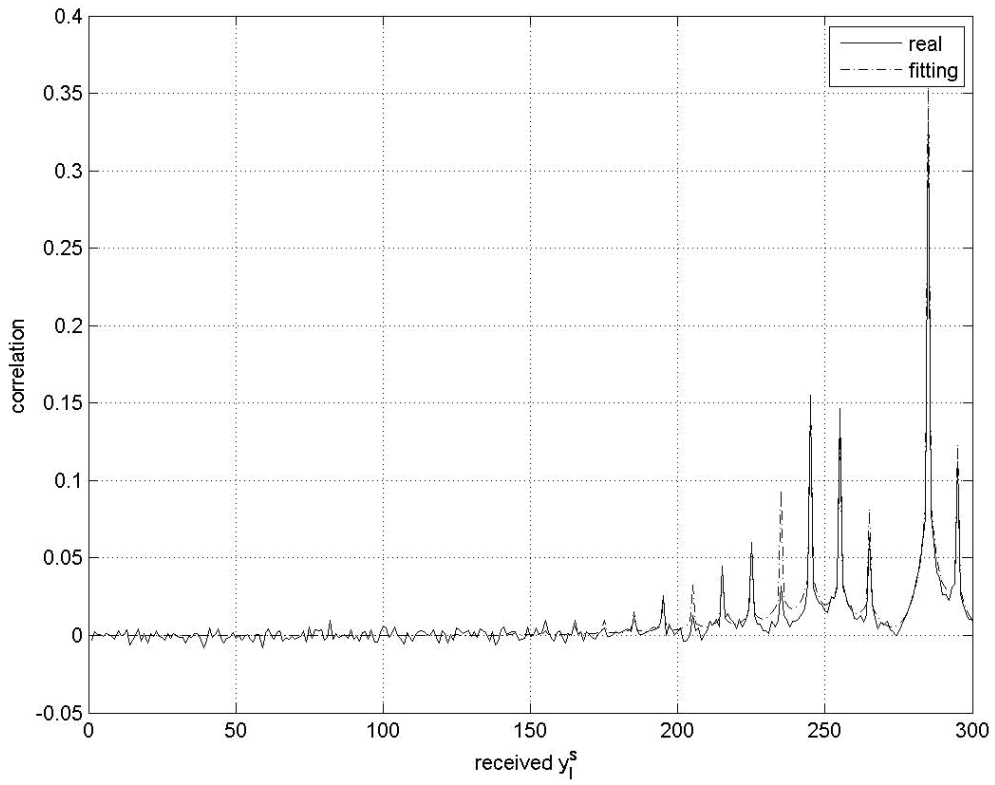


圖 5.9 $\rho_{SBSD}^{(1)}(L_{SBSD}^{(ext),(1)}(u_{148}), \tilde{y}_l^s)$

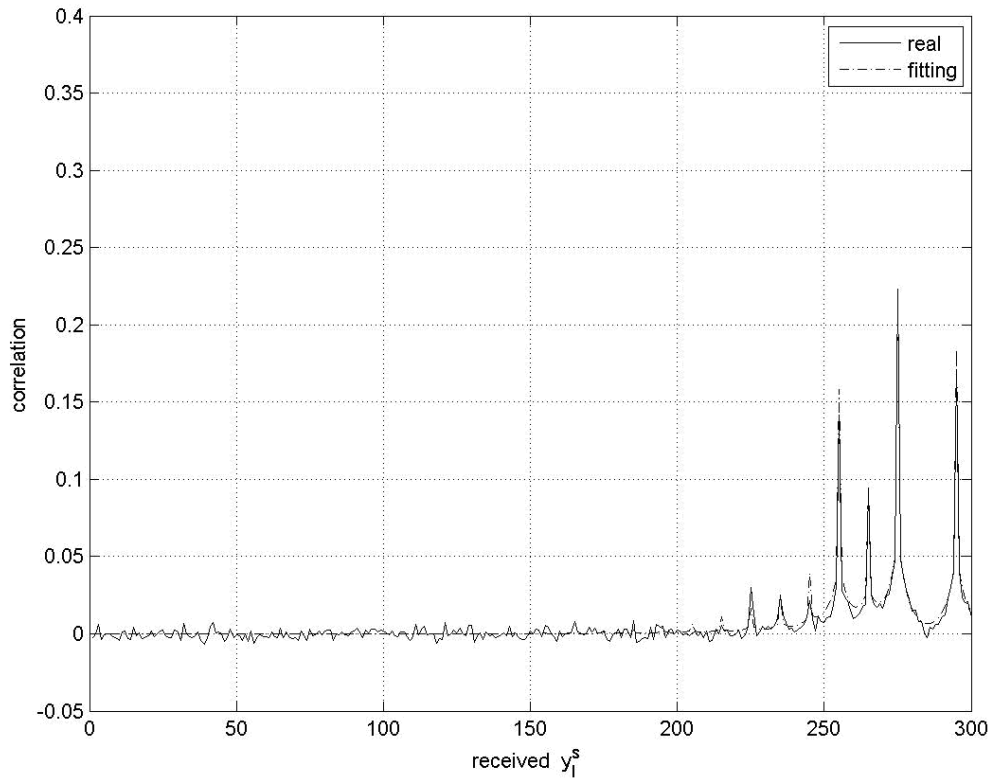


圖 5.10 $\rho_{SBSD}^{(1)}(L_{SBSD}^{(ext),(1)}(u_{149}), \tilde{y}_l^s)$

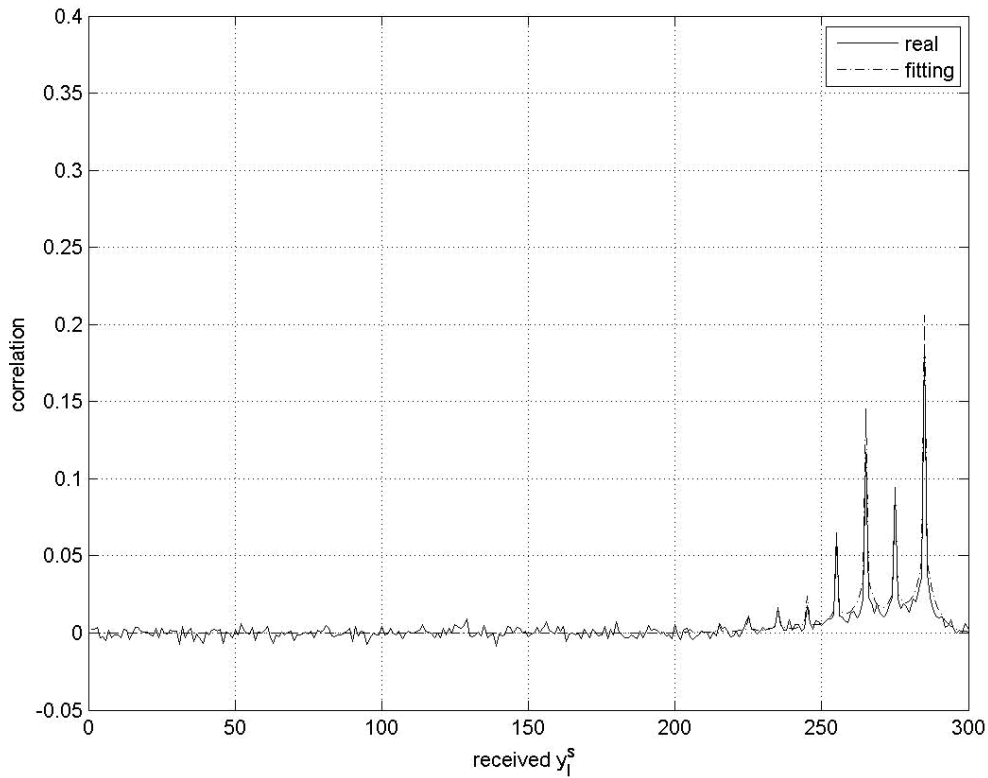


圖 5.11 $\rho_{SBSD}^{(n)}(L_{SBSD}^{(ext),(1)}(u_{150}), \tilde{y}_l^s)$

實驗四 不同交錯器索引層級系統上的實現

本實驗同樣以 $100 * L$ 個訊源取樣值，搭配 NBC 與區塊交錯器並且重複 100 次來求得 $SNR = -3dB$ 時的 $\rho_{SBSD}^{(1)}(L_{SBSD}^{(ext),(1)}(u_{50}), \tilde{u}_l)$ ，接下來再藉由 $\rho_{SBSD}^{(1)}(L_{SBSD}^{(ext),(1)}(u_{50}), \tilde{u}_l)$ 預估出產生模型並據以設計交錯器。最後我們將設計後的交錯器(`inter1(index)`)與區塊交錯器、隨機交錯器的系統效能進行比較，其結果如圖 5.12 所示。

實驗結果

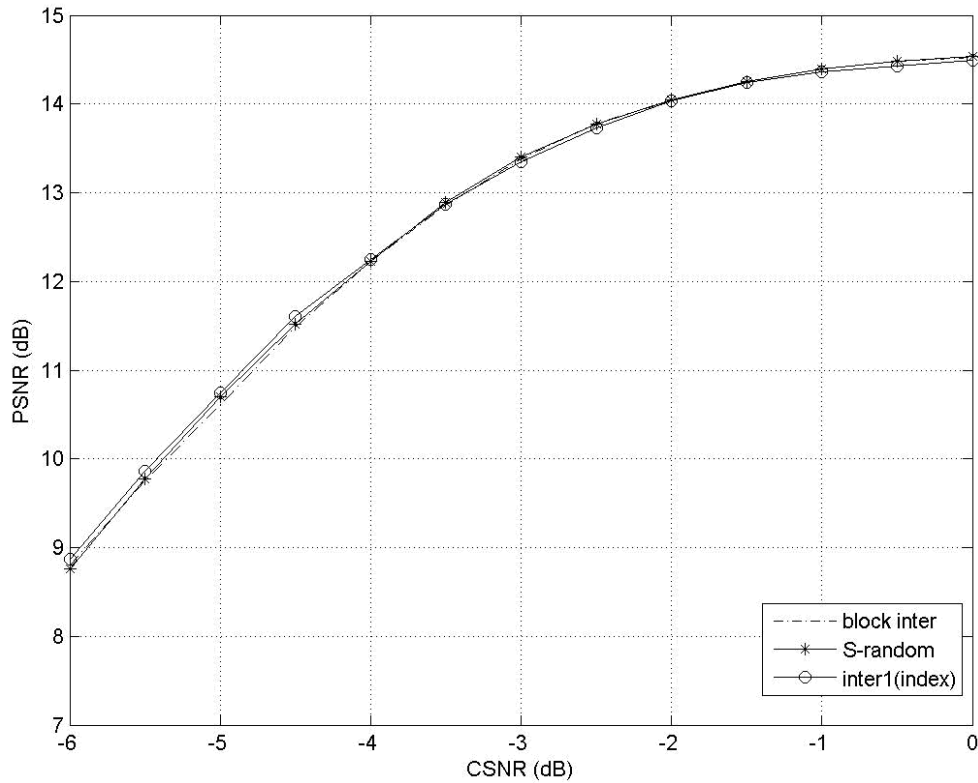


圖 5.12 不同交錯器效能之比較 (第 5 次疊代)

結果分析與討論

首先觀察圖 5.13，圖中清楚地表示 $\rho_{SBS D}^{(1)}(L_{SBS D}^{(ext),(1)}(\mathbf{u}_{50}), \tilde{\mathbf{y}}_l^s)$ 和 $\rho_{SBS D}^{(1)}(i, \Phi(i))$ 兩者的差異不大，這意味著我們可依此設計交錯器。而圖 5.12 則顯示了 3 種不同交錯器的系統效能，其中區塊交錯器和隨機交錯器的效能表現幾乎是完全一樣的，inter1(index) 則有別與前兩者的表現。首先在通道環境較差時 ($SNR < -4$ dB)，基於疊代解碼效益設計的交錯器表現較佳，並且在 $SNR = -6$ dB 下與隨機交錯器效能達 0.1 dB 的差距；而在 $SNR > -4$ dB 的通道環境中，三種交錯器的

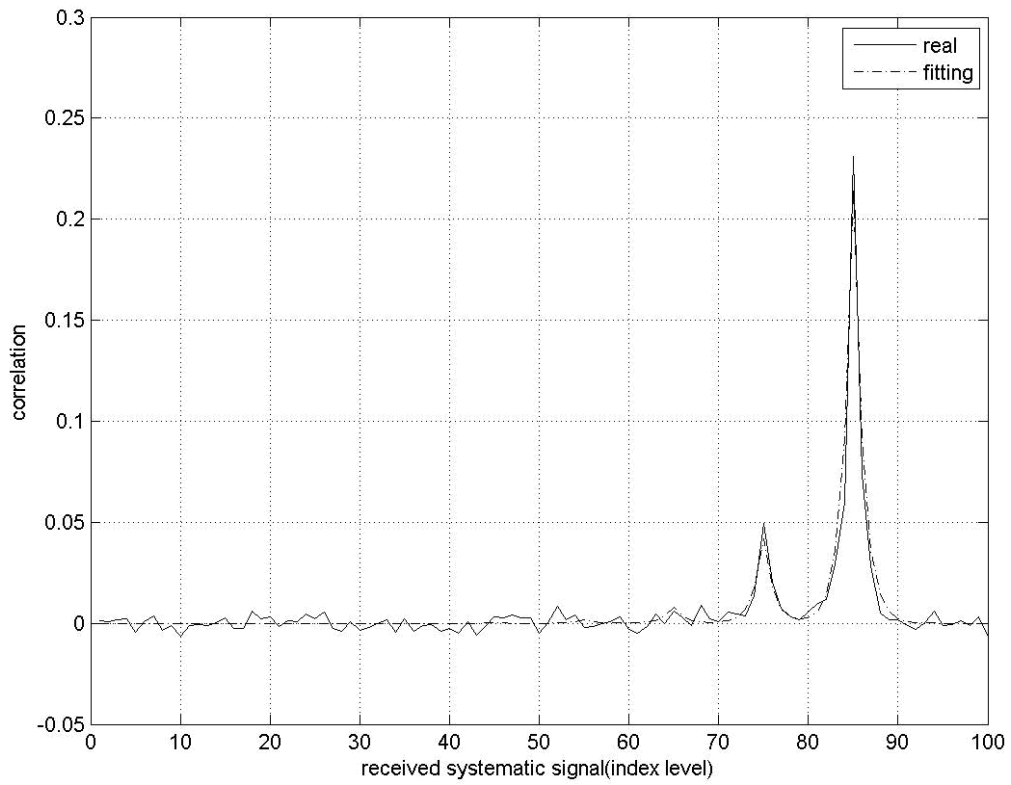


圖 5.13 $\rho_{SBSD}^{(1)}(L_{SBSD}^{(ext),(1)}(\mathbf{u}_{50}), \tilde{\mathbf{y}}_1)$

性能則很接近。



第六章 結論與未來展望

在本文中我們回歸到疊代訊源通道解碼演算法的本質，依渦輪準則分析疊代解碼效能的關鍵，並且分別探討渦輪準則三大要件，交錯器、軟性輸入輸出解碼器、額外資訊的交換，以期整體系統效能達到最佳化目的。

我們提出的索引指定設計主要是基於通道解碼額外資訊的自相關性變化，因為它不僅是反應系統效能的重要指標，同時也會隨著索引指定的改變而出現明顯差異。所以可將通道解碼額外資訊的一階自相關係數作為成本函數，來進行最佳化的索引指定設計。有別於傳統基於額外資訊轉換圖的索引指定設計，我們提出的設計方式是以整個系統架構為根本所衍伸的結果，因此索引指定更能與系統契合，並從實驗結果得到驗證。

而交錯器的設計則是根據[19]概念，以軟性輸入輸出解碼演算法的角度切入，探討解碼器在理論上對於事前資訊的要求，並藉由觀察實際事前資訊關聯性的變化，建立交錯器與系統效能的關聯。立足於此概念，我們融合通道與訊源解碼演算法在理論上的假設，推導出新的交錯器設計方式，讓每個解碼器在實際疊代解碼的過程中能交換到符合它們理論上需求的事前資訊。在第六章裡，我們比較區塊交錯

器、隨機交錯器、基於疊代效益設計的交錯器(inter1 和 inter2)，發覺依本文交錯器設計方法所產生的交錯器確實能提升 ISCD 效益，使系統有更好的表現。

雖然最佳化的索引指定和交錯器設計已被提出，當中卻存在龐大計算量的問題。起因是在最佳化的索引指定設計過程中，必須執行實際疊代解碼來統計通道解碼額外資訊相關性的變化，並且對於設定最佳化的系統參數需要實務經驗上的支持；而最佳化交錯器的設計則是需先觀察資訊關聯性實際的變化並求出產生模型，接下來以產生模型進行二位元置換演算法，但二位元置換演算法的運算量又跟交錯器的大小成正比。

因此展望未來的研究，索引指定的設計或許應考慮整合額外資訊轉換圖和額外資訊相關性的優點。先利用額外資訊轉換圖篩選出具有潛力的索引指定，再觀察這些索引指定的額外資訊相關性變化，從中找出最佳的索引指定。至於交錯器的設計，則仍有諸多改善空間，例如是否可避免對實際的資訊關聯性觀察並產生統計模型、切割交錯器長度進行交錯器設計等，這些都是能縮短開發時間的方法。

參考文獻

- [1] C.E. Shannon, "A Mathematical Theory of Communications," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, pp. 623-656, 1948.
- [2] L. Xiaobei, "Joint Source-channel Decoding and its Application to MELP Encoded Speech," Ph.D. thesis, *Nanyang Technological University*, Singapore, 2004.
- [3] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449-2457, Sep. 1995.
- [4] 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.
- [5] M. Adrat, "Iterative Source-Channel Decoding for Digital Mobile Communications," Ph.D. thesis, vol. 16 of ABDN, *Druck & Verlagshaus Mainz GMBH Aachen*, Aachen, Germany, 2003.
- [6] J. Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. on Information Theory*, vol. 42, no. 2, March 1996.
- [7] M. Adrat, P. Vary, and J. Spittka, "Iterative Source-Channel Decoder Using Extrinsic Information from Softbit-Source Decoding," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 4, pp. 2653-2656, Salt Lake City, Utah, USA, May 2001.
- [8] 許忠安，「通道匹配疊代解碼演算法之研究，國立交通大學碩士論文，民國九十六年。

- [9] 潘彥璋，「基於索引層級的疊代訊源通道解碼機制」，國立交通大學碩士論文，民國九十七年。
- [10] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [11] B. Scanavino, G. Montorsi, and S. Benedetto, "Convergence Properties of iterative decoders working at bit and symbol level," in *Proc. IEEE GLOBECOM*, San Antonio, TX, 2001, vol. 2, pp. 1037–1041.
- [12] S. Lin and D.J. Costello, "Error Control Coding," 2nd ed., *Pearson Prentice Hall*, ISBN 0-13-017973-6, 1983, 2004.
- [13] M. Adrat and P. Vary, "Iterative source-channel decoding: improved system design using EXIT charts," *EURASIP J. Appl. Signal Processing (Special Issue: Turbo Processing)*, pp. 928–941, May 2005.
- [14] J. Andersen and V. Zyablov, "Interleaver design for turbo coding," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sept. 1997, pp. 154–156.
- [15] S. Crozier, J. Lodge, P. Guinand, and A. Hunt, "Performance of Turbo-codes with relative prime and golden interleaving strategies," *International Mobile Satellite Conference, 6th*, Ottawa, Canada; CANADA; 16-18 June 1999. pp. 268-275. 1999.
- [16] F. Daneshgaran and M. Mondin, "Design of interleaver for turbo codes based on a cost function," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, Sept. 1997, pp. 255–258.
- [17] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations," *The Telecommunications and Data Acquisition Progress Report 42-122*, April–June 1995, Jet Propulsion Laboratory, Pasadena, California, pp. 56–65, August 15, 1995.

- [18] P. Robertson, "Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes," *Int. Conf. Universal Personal Commun.*, San Diego, CA, pp.183–187, Sep.–Oct. 1994.
- [19] J. Hokfelt, "On the Design of Turbo Codes", PhD Dissertation, Lund University, Lund, Sweden 2000
- [20] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo codes", *ICC '93, Conference Record*, Geneva, pp. 1064–1070, 1993
- [21] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform.Theory*, vol. IT-20, pp. 284-287, Mar. 1974.
- [22] N. Görtz, "Optimization of Bit Mappings for Iterative Source-Channel Coding," in *Proc. of International Symposium on Turbo Codes and Related Topics*, (Brest, France), pp. 255–258, Sept. 2003.
- [23] K. Zeger and A. Gersho, "Pseudo-Gray Coding," *IEEE Transactions on Communications*, vol. COM-38, pp. 2147–2158, Dec. 1990.
- [24] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. Elec. Eng., Linköping Univ., Sweden, 1996.
- [25] M. Adrat, U. von Agris, and P. Vary, "Convergence Behavior Of Iterative Source-Channel Decoding," in *Proc. of ICASSP 2003*, 2003, vol. 4, pp.269–272
- [26] 陳亞民，「渦輪碼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。