

國立交通大學

電信工程學系

碩士論文



通道匹配疊代解碼演算法之研究
A study of channel-matched iterative decoding
algorithm

研究生：許忠安

指導教授：張文輝博士

中華民國九十六年六月

通道匹配疊代解碼演算法之研究

學生：許忠安

指導教授：張文輝 博士

國立交通大學電信工程學系碩士班

中文摘要



在數位通訊系統，訊源編碼器的設計受限於複雜度與延遲，其編碼輸出序列仍有殘餘冗息存在。為了加強系統的強健性，我們在接收端以渦輪碼原理設計疊代解碼演算法。其關鍵在於利用訊源殘餘冗息和通道編碼提供的保護位元，在通道解碼器與訊源解碼器之間交叉利用解碼輸出的額外訊息。藉由疊代次數增加，額外訊息的準確性也跟著增加，而解碼效能也隨之提升。此外，在無線通道環境中常出現叢集性錯誤，疊代解碼設計應考量錯誤之間的相關性。模擬結果顯示，基於迴旋碼的疊代解碼效能較好，特別是與通道匹配的解碼機制可進一步改善系統強健性。

A study of channel-matched iterative decoding algorithm

Student: Zo-Ein Shi Advisor: Dr. Wen-Whei Chang

Department of Communication Engineering

National Chiao Tung University

The logo of National Chiao Tung University is a circular seal. It features a gear-like outer border. Inside the circle, there is a stylized building or structure. Below the building, the year '1896' is inscribed. The word 'Abstract' is overlaid on the logo.

Abstract

Due to complexity and delay limit of source encoder, there exists source residual redundancy in digital communication. In order to enhance the system robustness, we proceed with iterative source-channel decoding based on a turbo principle at the receiver. The basic strategy is to exchange extrinsic information between the channel decoder and source decoder. Real channels are characterized by error bursts due to the combined effects of intersymbol interference and multipath fading. It is therefore believed that further enhancement can be realized through a more precise characterization of the channel on which the iterative decoder design is based. Simulation results indicate that channel-matched iterative decoding based on convolutional code achieves the highest robustness against channel errors.

致謝

本篇論文的完成，首先感謝指導教授張文輝老師這兩年來的悉心指導，讓我瞭解應該要以怎樣的態度作研究。尤其本來不是在通訊領域上有所接觸的我，使老師要加倍教導。衷心感激老師不嫌棄我的資質孱鈍，春風化雨。同時，也要謝謝實驗室的學長學弟們，平常在功課生活及研究上有所照顧與討論，讓我能順利作研究。還要感謝家人們及朋友們的鼓勵。最後再次感激實驗室 812A 的學長學弟及老師。



目錄

中文摘要	i
英文摘要	ii
致謝	iii
目錄	iv
圖目錄	vii
表目錄	ix
第一章 緒論	1
1.1 研究動機與方向	1
1.2 章節概要	2
第二章 疊代編解碼演算法	3
2.1 疊代編解碼架構	3
2.2 軟性位元訊源解碼	6
2.3 額外訊息推導	9
第三章 基於單一同位檢查碼的疊代解碼機制	12
3.1 通道編碼模型	12
3.2 位元後驗機率推導	13

3.3	通道解碼額外訊息的推導.....	17
3.4	訊源解碼額外訊息的推導.....	20
第四章	基於迴旋碼的疊代解碼機制	25
4.1	通道編碼模型.....	25
4.2	無記憶性通道的解碼額外訊息推導.....	27
4.3	記憶性通道的解碼額外訊息推導.....	34
4.3.1	通道解碼額外訊息的計算	35
4.3.2	訊源解碼額外訊息的計算.....	37
第五章	實驗模擬與結果分析	39
5.1	基於單一同位檢查碼之疊代解碼實驗.....	39
5.1.1	系模擬之步驟說明	39
5.1.2	結果分析.....	42
5.2	基於迴旋碼之疊代解碼實驗.....	43
5.2.1	系統模擬之步驟說明.....	43
5.2.2	結果分析.....	44
5.3	記憶性通道之疊代解碼實驗.....	45
5.3.1	系統模擬之步驟說明.....	45
5.3.2	結果分析.....	53

第六章 結論與未來展望	55
參考文獻	56
附錄 A 對數相似比率代數	57
附錄 B 對數相似比率應用	58



圖目錄

圖 2.1	疊代編解碼機制	4
圖 2.2	疊代解碼機制	5
圖 2.3	區塊交錯器	6
圖 2.4	隨機交錯器	6
圖 3.1	碼字的位元組合	13
圖 3.2	接收區塊的相對位置	14
圖 3.3	碼字位元的相對位置	14
圖 4.1	$R=1/2$ 的迴旋系統碼	26
圖 4.2	迴旋系統碼的狀態圖(左)和籬柵圖(右)	26
圖 4.3	籬柵終止	27
圖 4.4	前項遞迴機率籬柵圖	30
圖 4.5	後項遞迴機率籬柵圖	31
圖 4.6	Gilbert 模型	35
圖 5.1	基於單一同位檢查碼之疊代解碼結果	42
圖 5.2	基於迴旋碼之疊代解碼結果	44
圖 5.3	使用區塊交錯器($L=150$)但不考慮通道記憶性之結果	47
圖 5.4	利用通道記憶性與區塊交錯器($L=150$)之結果	48
圖 5.5	不考慮通道記憶性與交錯器之結果	49

圖 5.6	考慮通道記憶性但不使用交錯器之結果	50
圖 5.7	考慮通道記憶性與隨機交錯器(L=150)之結果	51
圖 5.8	考慮通道記憶性與隨機交錯器(L=1500)之結果	52
圖 5.9	考慮通道記憶性與區塊交錯器(L=1500)之結果	53



表目錄

表 5.1	使用區塊交錯器(L=150)但不考慮通道記憶性的結果	46
表 5.2	利用通道記憶性與區塊交錯器(L=150)之結果	47
表 5.3	不考慮通道記憶性與交錯器之結果	48
表 5.4	考慮通道記憶性但不使用交錯器之結果	49
表 5.5	考慮通道記憶性與隨機交錯器(L=150)之結果	50
表 5.6	考慮通道記憶性與隨機交錯器(L=1500)之結果	51
表 5.7	考慮通道記憶性與區塊交錯器(L=1500)之結果	52



第一章 緒論

1.1 研究動機與方向

在 1948 年，C. E. Shannon[1]發表了兩篇重要的理論：訊源編碼理論(source coding theorem)和通道編碼理論(channel coding theorem)。訊源編碼理論指出，假設在通道不會發生錯誤情形下，當資料壓縮速率至少為訊源的熵(entropy)時，則在訊源解碼時不會有任何的失真發生。通道編碼理論指出，在訊源受到錯誤更正碼保護後，假如經由無記憶通道傳輸時，資料傳輸速率小於或等於通道容量(channel capacity)時，在解碼時不會有任何錯誤發生。依據上述理論，訊源編碼和通道編碼雖然可以分別獨立處理，即在設計訊源編碼時假設通道為零錯誤，在設計通道編碼時假設訊源完全被壓縮，即訊源編碼輸出位元視為獨立同等分佈(i. i. d)，可達到傳輸最佳化。但仍有諸多條件限制，以致於在實際通訊系統裡無法實現。舉例而言，對訊源和通道編碼器的複雜度和延遲沒有限制，且訊源和通道的編碼區塊長度要趨於無限長度才可符合理論需求。但在真實通訊系統裡，複雜度和延遲必須有所限制，所以當系統限制條件變多時，訊源和通道編碼分開設計的模式變成次佳。因此實際情形是，訊源編碼後仍有殘餘冗息存在，且經過通道解碼後仍有殘餘的位元錯誤存在。為了對抗通道雜訊所衍生的錯誤，通道解碼器必須提供軟性輸出，而不是作

硬性判斷，且訊源解碼應利用到訊源編碼殘餘冗息。若通道環境會發生叢發性錯誤，即通道有記憶特性，則在解碼時必須考量到記憶特性以對抗雜訊干擾。

1.2 章節概要

第二章介紹疊代解碼概念及軟性位元訊源解碼架構。第三章則推導疊代解碼演算法。第四章介紹基於迴旋碼的疊代解碼演算。第五章為系統模擬。第六章為結論與未來展望。



第二章 疊代編解碼演算法

依據 C. E. Shannon 的編碼理論，只要資料傳輸速率小於或等於通道容量(channel capacity)時，則一定存在著一種通道編碼技術，使其通道錯誤修正為一任意小的數值。然而並未指出如何去建構此最佳理論系統，常見的線性區塊碼與迴旋碼都與 Shannon 的理論值有一段差距。直到 1993 年，C. Berrou, A. Glavieux, 和 P. Thitimajshima 共同提出渦輪碼(turbo codes)[2]才有突破。其編碼器由兩個遞迴系統迴旋碼(recursive systematic convolutional codes)平行連結(parallel concatenation)組成，並且以交錯器(interleaver)分開。解碼時則以疊代方式在兩通道解碼器之間，交叉利用輸出的額外訊息(extrinsic information)，此額外訊息來自於兩個通道編碼器提供的冗息。渦輪碼原理可應用在本論文的疊代架構[3]裡，不同的是我們在通道解碼器與訊源解碼器之間進行，而額外訊息則來自於通道編碼的冗息與訊源編碼的殘餘冗息。在本章中我們首先介紹採用渦輪碼原理的疊代解碼架構，接著介紹疊代解碼的組成元件。

2.1 疊代編解碼架構

考慮在時間 τ 內傳送一組由 M 個參數組成的參數向量

$\underline{u}_\tau = \{u_{\tau,1}, u_{\tau,2}, \dots, u_{\tau,M}\}$ 至一雜訊通道，如圖 2.1 所示。針對一特定參數 $u_{\tau,k}$ ，進行 m 個位元的量化器處理後，所得的索引值經由位元映射(bit mapping)後，產生一組由 m 個位元所構成的位元組合

$\underline{x}_{\tau,k} \triangleq \{x_{\tau,k}(0), \dots, x_{\tau,k}(m-1)\}$ 。交錯器 Φ 以明確的映射方式插置整組位元組合 $\underline{x}_\tau = \{\underline{x}_{\tau,1}, \dots, \underline{x}_{\tau,M}\}$ 而得 $\tilde{\underline{x}}_\tau = \{\tilde{\underline{x}}_{\tau,1}, \dots, \tilde{\underline{x}}_{\tau,M}\}$ 。因量化後殘餘冗息存在於位元組合 $\underline{x}_{\tau,k}$ 之間，若取代之以彼此沒有相關性的位元組合 $\tilde{\underline{x}}_{\tau,k}$ 進行通道編碼，則產生的同位檢查位元就會與 $\underline{x}_{\tau,k}$ 沒有相關性，這樣一

來從通道解碼和訊源解碼攫取的額外訊息互相獨立，即可避免重複利用到相同訊息的可能。通道編碼的碼字 $\underline{y}_\tau$ 採用系統碼(systematic codes)形式，則在接收端即可直接取得訊息位元(information bits)及其通道的相關訊息。

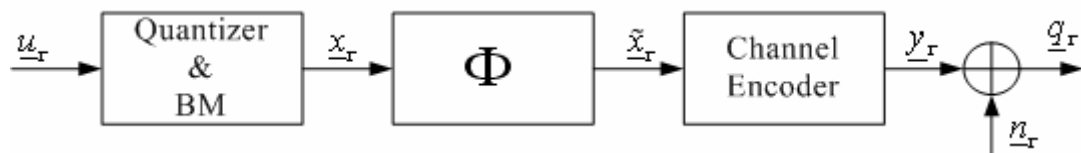


圖 2.1 疊代編碼機制

在接收端為了能進行疊代解碼，通道解碼器與訊源解碼器必須能提供軟性輸入/輸出的功能，且軟性輸出值須能拆解成 3 項：事前訊息

$p(x_{\tau,k}(\lambda))$ ，通道相關訊息 $p(q_{\tau,k}(\lambda)|x_{\tau,k}(\lambda))$ ，與額外訊息

$\{p_{cd}^{(ext)}(x_{\tau,k}(\lambda)), p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))\}$ 。如圖 2.2 所示，為了簡化，標記符號不

考慮交錯器影響。在第一次疊代解碼過程裡，沒有訊源解碼輸出的額外訊息 $p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ 當作通道解碼器的附加事前訊息。反之，通道解碼輸出的額外訊息 $p_{cd}^{(ext)}(x_{\tau,k}(\lambda))$ 則可傳遞給訊源解碼，視為其附加的事前訊息而提升解碼輸出額外訊息 $p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ 的準確性。第二次疊代時再把 $p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ 傳遞給通道解碼當作附加的事前訊息，之後疊代依此類推。藉由額外訊息在兩解碼器之間的交叉傳遞，隨著疊代次數增加，額外訊息的準確性也跟著提升。

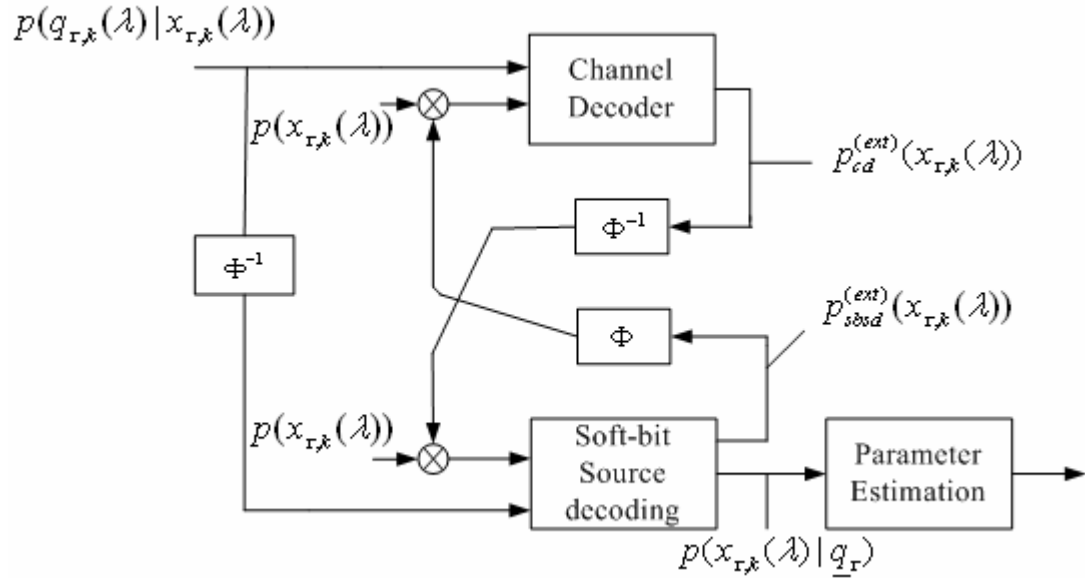


圖 2.2 疊代解碼機制

此外，交錯器的大小和結構也會影響系統效能。以下介紹在實驗中使用的交錯器。

[1] 區塊交錯器(block interleaver)

一個長度 $L \cdot K$ 區塊交錯器可以以一個 $L \times K$ 的矩陣表示。資料依序寫入矩陣的列向量而以行向量讀取。如圖 2.3 所示，假設有一長度

24 的輸入序列，以 3×8 矩陣插置。

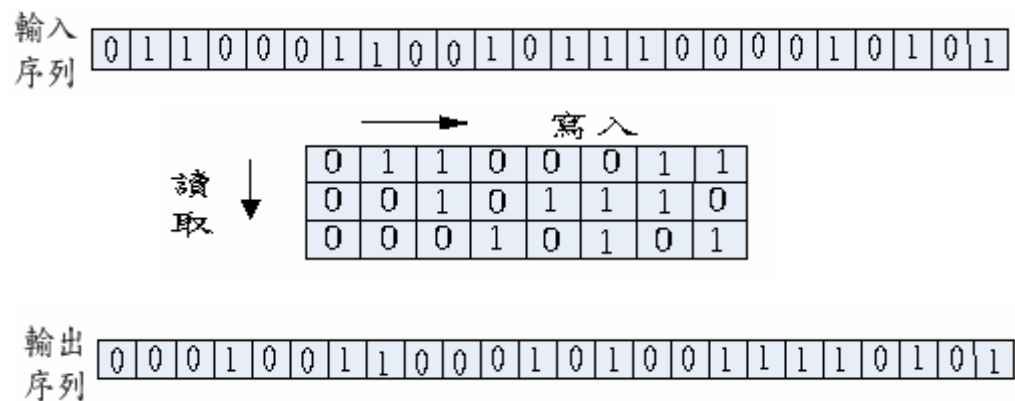


圖 2.3 區塊交錯器

[2] 隨機交錯器(random interleaver)

隨機交錯器使用一固定的隨機排列次序把輸入序列映射至輸出

序列。如圖 2.4 所示，假設有一長度 24 的輸入序列。

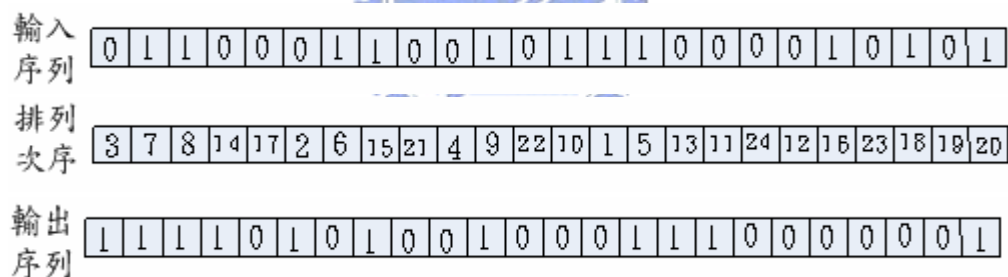


圖 2.4 隨機交錯器

2.2 軟性位元訊源解碼

在數位通訊系統裡，即使訊源受到通道編碼保護來對抗雜訊，但在通道解碼後仍不能避免殘餘錯誤的存在。不同於硬性判斷(hard decision)，軟性位元訊源解碼[4]利用訊源編碼後的殘餘冗息，並結合參數預估以隱匿殘餘的錯誤。傳輸架構如圖 2.5 所示。

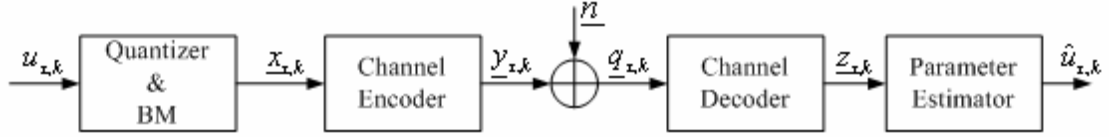


圖 2.5 軟性位元訊源解碼器

考慮在時間 τ 傳送一參數 $u_{\tau,k}$ ，進行量化處理得最近似碼字

$Q(u_{\tau,k}) = c_{x_{\tau,k}}$ ，其中 $x_{\tau,k}$ 為該碼字對應的索引值。經由位元映射(bit mapping)後，產生一組由 m 個位元所構成的位元組合

$\underline{x}_{\tau,k} \triangleq \{x_{\tau,k}(0), \dots, x_{\tau,k}(m-1)\}$ 。通道編碼以系統碼且碼率為 $1/2$ 為例，則輸出位元組 $\underline{y}_{\tau,k} \triangleq \{(x_{\tau,k}(\lambda), \pi_{\tau,k}(\lambda)), \lambda = 0, 1, \dots, m-1\}$ ，其中 $x_{\tau,k}(\lambda)$ 為訊息位元而 $\pi_{\tau,k}(\lambda)$ 為同位檢查位元。在接收端，分別令訊息位元與同位檢查位元的接收實數序列為 $\underline{q}_{\tau,k} \triangleq \{q_{\tau,k}(\lambda), \lambda = 0, 1, \dots, m-1\}$ ，以及 $\underline{z}_{\tau,k} \triangleq \{z_{\tau,k}(\lambda), \lambda = 0, 1, \dots, m-1\}$ 。至於訊源解碼過程中的參數預估，則根據下列步驟依序完成。

[1] 事前訊息的計算

為了適當利用訊源編碼的殘餘冗息，量化索引可模擬成 n 階馬可夫程序

$$p(x_{\tau,k} | x_{1,k}^{\tau-1}) = p(x_{\tau,k} | x_{\tau-n,k}^{\tau-1}) \quad (2.1)$$

其中 $x_{1,k}^{\tau-1} \triangleq \{x_{1,k}, x_{2,k}, \dots, x_{\tau-1,k}\}$ ，而 $x_{\tau-n,k}^{\tau-1} \triangleq \{x_{\tau-n,k}, \dots, x_{\tau-1,k}\}$ 。若 $n=0$ ，我們稱 $p(x_{\tau,k})$ 為零階事前訊息(0^{th} order a priori knowledge, AK0)，因

其在統計上的描述為零階的馬可夫程序，即為一沒有記憶性的程序。

若 $n=1$ ， $p(x_{\tau,k} | x_{\tau-1,k})$ 稱為一階事前訊息(first-order a priori knowledge, AK1)，因其代表了一階的馬可夫程序。這些機率值皆以訓練大量語料求得。至於階數設定則需考慮三項要點：量化索引的殘餘冗息，運算複雜度，及效能和運算複雜度之間的取捨。

[2] 量化索引事後機率的計算

(A) AK0 模型：

如果量化索引序列提供零階事前訊息，利用貝氏定理則事後機率為

$$p(x_{\tau,k} | z_{\tau,k}) = C \cdot p(z_{\tau,k} | x_{\tau,k}) \cdot p(x_{\tau,k}) \quad (2.2)$$

其中正規化常數 $C = \frac{1}{\sum_{x_{\tau,k}} p(z_{\tau,k} | x_{\tau,k}) \cdot p(x_{\tau,k})}$ 。假設通道為 AWGN，則

$$\begin{aligned} p(z_{\tau,k} | x_{\tau,k}) &= \prod_{\lambda=0}^{m-1} p(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \\ &= \prod_{\lambda=0}^{m-1} \frac{1}{\sqrt{\pi N_0}} \cdot e^{-\frac{E_s}{N_0} (z_{\tau,k}(\lambda) - x_{\tau,k}(\lambda))^2} \end{aligned} \quad (2.3)$$

(B) AK1 模型

如果量化索引序列提供一階事前訊息，則可利用事前機率

$p(x_{\tau,k} | x_{\tau-1,k})$ 和接收序列 $z_{1,\tau}^r \triangleq \{z_{1,\tau}, z_{2,\tau}, \dots, z_{\tau,k}\}$ ，求得個別索引的事後機

率。以遞迴形式呈現如下

$$p(x_{\tau,k} | z_{1,k}^{\tau}) = C \cdot p(z_{\tau,k} | x_{\tau,k}) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1}) \quad (2.4)$$

$$\text{其中正規化常數 } C = \frac{1}{\sum_{x_{\tau,k}} p(z_{\tau,k} | x_{\tau,k}) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}$$

[3] 參數估測

使用最小均方誤差準則(minimum mean square error criterion)，

重建後的參數為

$$\hat{u}_{\tau,k} = \sum_{x_{\tau,k}} c_{x_{\tau,k}} \cdot p(x_{\tau,k} | z_{1,k}^{\tau}) \quad (2.5)$$

其中 $c_{x_{\tau,k}}$ 為索引 $x_{\tau,k}$ 對應的量化參數碼字



2.3 額外訊息推導

為了方便說明，我們以 AK0 模型來推導訊源解碼輸出的額外訊息

$p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ ，亦即考量存在於索引 $x_{\tau,k}$ 所屬的個別位元之間的殘餘冗

息。令 $x_{\tau,k}^{(ext)} = \{x_{\tau,k}(j), j \neq \lambda\}$ ，則 $x_{\tau,k} = \{x_{\tau,k}^{(ext)}, x_{\tau,k}(\lambda)\}$ ，而其接收序列為

$z_{\tau,k} = \{z_{\tau,k}^{(ext)}, z_{\tau,k}(\lambda)\}$ 。該位元後驗機率可表示為

$$p(x_{\tau,k}(\lambda) | z_{\tau,k}) = \sum_{\{x_{\tau,k} : x_{\tau,k}(\lambda)\}} p(x_{\tau,k} | z_{\tau,k}) \quad (2.6)$$

其中 $\{x_{\tau,k} : x_{\tau,k}(\lambda)\}$ 代表具有相同 $x_{\tau,k}(\lambda)$ 的索引集合。利用貝氏定理與

連鎖法則，並假設通道獨立可展開得

$$\begin{aligned} p(x_{\tau,k}(\lambda) | z_{\tau,k}) &= \sum_{\{x_{\tau,k} : x_{\tau,k}(\lambda)\}} C \cdot p(z_{\tau,k} | x_{\tau,k}) \cdot p(x_{\tau,k}) \\ &= C \cdot p(x_{\tau,k}(\lambda)) \cdot p(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \end{aligned}$$

$$\cdot \sum_{x_{\tau,k}^{(ext)}} p(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) \cdot p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)) \quad (2.7)$$

基於通道無記憶性假設，(2.7)式最後一項可簡化為

$$\begin{aligned} & \sum_{x_{\tau,k}^{(ext)}} p(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) \cdot p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)) \\ &= \sum_{x_{\tau,k}^{(ext)}} p(z_{\tau,k}^{(ext)}, x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)) = p(z_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)) \end{aligned} \quad (2.8)$$

綜合(2.7)式和(2.8)式，訊源解碼的額外訊息定義為

$$\begin{aligned} p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) &= p(z_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)) \\ &= \frac{p(x_{\tau,k}(\lambda) | z_{\tau,k})}{C \cdot p(x_{\tau,k}(\lambda)) \cdot p(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda))} \end{aligned} \quad (2.9)$$

其中正規化常數 $C = \frac{1}{\sum_{x_{\tau,k}} p(z_{\tau,k} | x_{\tau,k}) \cdot p(x_{\tau,k})}$ 。

此外，我們也以單一同位檢查碼來說明通道解碼輸出的額外訊息

$$p_{cd}^{(ext)}(x_{\tau,k}(\lambda))。$$

假設有一訊息位元序列 $x_{\tau,k}(\lambda)$ ， $\lambda = 0, 1, \dots, m-1$ ，其單一同位檢查位元為

$$\pi_{\tau} = x_{\tau,k}(0) \oplus x_{\tau,k}(1) \oplus \dots \oplus x_{\tau,k}(m-1)。 \quad (2.10)$$

任一訊息位元 $x_{\tau,k}(\lambda)$ 可由同位檢查位元 π_{τ} 與其剩餘訊息位元

$\{x_{\tau,k}(j), j \neq \lambda\}$ 共同決定如下

$$x_{\tau,k}(\lambda) = x_{\tau,k}(0) \oplus \dots \oplus x_{\tau,k}(\lambda-1) \oplus x_{\tau,k}(\lambda+1) \oplus \dots \oplus x_{\tau,k}(m-1) \oplus \pi_{\tau}。 \quad (2.11)$$

定義 $\underline{x}_{\tau,k} = (x_{\tau,k}(\lambda), x_{cd}^{(ext)})$ ，其中

$$x_{cd}^{(ext)} \triangleq (x_{\tau,k}(1), \dots, x_{\tau,k}(\lambda-1), x_{\tau,k}(\lambda+1), \dots, x_{\tau,k}(m), \pi_{\tau}) \quad (2.12)$$

為同位檢查位元與其剩餘訊息位元 $x_{\tau,k}(j)$ 組成。若經由 AWGN 通道雜訊干擾，接收一實數序列 $\underline{q}_{\tau} = \{q_{\tau,k}(\lambda), q_{cd}^{(ext)}\}$ 。藉由貝氏定理且假設通道無記憶性，則該位元的後驗機率可表示成

$$\begin{aligned} p(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) &= C \cdot p(\underline{q}_{\tau} | x_{\tau,k}(\lambda)) \cdot p(x_{\tau,k}(\lambda)) \\ &= C \cdot p(x_{\tau,k}(\lambda)) \cdot p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \cdot p(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \end{aligned} \quad (2.13)$$

其中正規化常數 $C = \frac{1}{\sum_{x_{\tau,k}(\lambda)} p(\underline{q}_{\tau} | x_{\tau,k}(\lambda)) \cdot p(x_{\tau,k}(\lambda))}$

定義通道解碼輸出的額外訊息為

$$\begin{aligned} p_{cd}^{(ext)}(x_{\tau,k}(\lambda)) &= p(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \\ &= \frac{p(x_{\tau,k}(\lambda) | \underline{q}_{\tau})}{C \cdot p(x_{\tau,k}(\lambda)) \cdot p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda))} \end{aligned} \quad (2.14)$$

第三章 基於單一同位檢查碼的疊代解碼機制

疊代解碼演算首要工作為決定個別位元的後驗機率(a posteriori probability)，為了攫取最大的可靠增益以提升其準確性，接收位元區塊必須完整地利用。我們嘗試以簡單的單一同位檢查碼(single parity check code)執行通道編碼，同時整合訊源編碼所殘餘的冗息，完成位元後驗機率的相關推導。結果發現其可拆解成四項，其中兩項為額外消息(extrinsic information)，分別由同位檢查位元與訊源編碼殘餘冗息所提供。類似於渦輪碼原理(turbo principle)，此兩項額外消息將是疊代解碼製作的關鍵元件，我們將在以下章節中詳述其推導過程。



3.1 通道編碼模型

所謂的 (n, k) 二位元線性區塊碼(binary linear block codes)，其編碼機制為由一組 k 個位元組成的向量 $\underline{x}_r$ 映射至 n 個位元組成的向量 $\underline{y}_r$ ，此為一對一關係。此編碼過程可用 $\underline{y}_r = \underline{x}_r \cdot G$ 來表示，其中 G 為一 $k \times n$ 維度的矩陣。假設碼字(codeword) $\underline{y}_r$ 之間作位元與位元的 modulo-2sum 的運算後仍為一合理碼字，則稱之為線性區塊碼。一個區塊碼稱之為系統碼(systematic codes)，表示碼字 $\underline{y}_r$ 所屬的 n 個位

元之中有 k 個位元來自 $\underline{x}_\tau$ 本身，剩下的 $r=n-k$ 個位元則稱之為同位檢查位元 (parity check bits)。單一同位檢查碼 (single parity check codes) 就是一系統碼，以 $(4, 3)$ 碼且 $r=1$ 為例，

$$\underline{y}_\tau = \underline{x}_\tau \cdot \begin{pmatrix} 1001 \\ 0101 \\ 0011 \end{pmatrix}$$

則同位檢查碼

$$\pi_\tau = x_\tau(1) \oplus x_\tau(2) \oplus x_\tau(3) = \sum_{\lambda=1}^3 \oplus x_\tau(\lambda)$$

若考慮時間 τ 內有一組參數向量 $\underline{u}_\tau$ ，由三個參數 $\{u_{\tau,k}, k=1, 2, 3\}$ 組成。

個別參數 $u_{\tau,k}$ 經量化器處理後，所得的索引值經由位元映射 (bit mapping) 後，產生一組 3 個位元的組合 $\underline{x}_{\tau,k} \triangleq \{x_{\tau,k}(1), x_{\tau,k}(2), x_{\tau,k}(3)\}$ 。

同位檢查碼附加在每一行上， $\pi_\tau(\lambda) = x_{\tau,1}(\lambda) \oplus x_{\tau,2}(\lambda) \oplus x_{\tau,3}(\lambda)$ ，

$\lambda=1, 2, 3$ ，則碼字 $\underline{y}_\tau \triangleq \{x_{\tau,1}(\lambda), x_{\tau,2}(\lambda), x_{\tau,3}(\lambda), \pi_\tau(\lambda)\}$ ， $\lambda=1, 2, 3$ ，如圖

3.1 所示。

$x_{\tau,1}$			
$x_{\tau,2}$			
$x_{\tau,3}$			
π_τ			

圖 3.1 碼字的位元組合

3.2 位元後驗機率推導

假設通道編碼如上一節所提，考慮經由雜訊干擾的通道接收一組

12 個實數組成的序列 $\underline{q}_\tau \triangleq \{q_{\tau,k}(\lambda), q_{cd}^{(ext)}, q_{sbsd}^{(ext)}, q_{RES}\}$ 。相對於特定的接收位置 $q_{\tau,k}(\lambda)$ ， $q_{cd}^{(ext)}$ 及 $q_{sbsd}^{(ext)}$ 分別應用於通道解碼軟性位元訊源解碼，而 q_{RES} 則包括其餘 6 個位置的接收值。接收區塊相對位置如圖 3.2 所示。

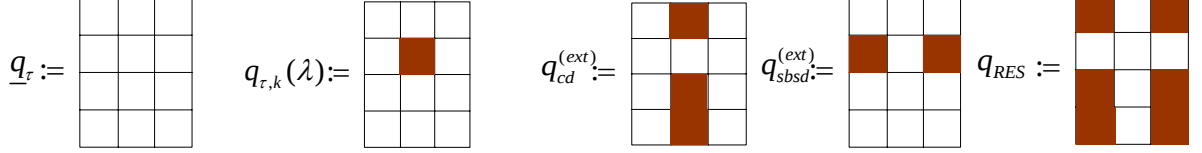


圖 3.2 接收區塊的相對位置

依此區塊劃分原則，碼字 $\underline{y}_\tau$ 所屬的 12 個位元也可區分為

$\underline{y}_\tau \triangleq \{x_{\tau,k}(\lambda), x_{cd}^{(ext)}, x_{sbsd}^{(ext)}, x_{RES}\}$ ，其相對位置如圖 3.3 所示。

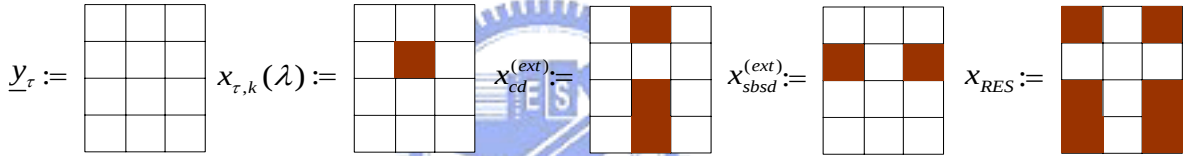


圖 3.3 碼字位元的相對位置

為了得到特定位元 $x_{\tau,k}(\lambda)$ 的最大準確性預估值，其位元後驗機率可由整組碼字 $\underline{y}_\tau$ 的邊際分佈機率(marginal distribution)求得

$$p(x_{\tau,k}(\lambda) | \underline{q}_\tau) = \sum_{\{\underline{y}_\tau: x_{\tau,k}(\lambda)\}} p(\underline{y}_\tau | \underline{q}_\tau) \quad (3.1)$$

其中 $\{\underline{y}_\tau: x_{\tau,k}(\lambda)\}$ 代表具有相同 $x_{\tau,k}(\lambda)$ 位元值的碼字集合。利用貝氏定理

(Bayes' theorem)及連鎖法則(chain rule)，可展開得

$$p(\underline{y}_\tau | \underline{q}_\tau) = C \cdot p(\underline{q}_\tau | \underline{y}_\tau) \cdot p(\underline{y}_\tau) = C \cdot p(\underline{q}_\tau | \underline{y}_\tau) \cdot p(x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}, x_{\tau,k}(\lambda))$$

$$p(x_{cd}^{(ext)}, x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \cdot p(x_{\tau,k}(\lambda)) \quad (3.2)$$

其中 C 為正規化常數 $\frac{1}{\sum_{\underline{y}_\tau} p(\underline{q}_\tau | \underline{y}_\tau) \cdot p(\underline{y}_\tau)}$ 。

進一步的簡化則須考慮三個假設，首先假設訊源解碼與通道解碼過程互相獨立，則

$$\begin{aligned} p(x_{cd}^{(ext)}, x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) &= p(x_{cd}^{(ext)} | x_{sbsd}^{(ext)}, x_{\tau,k}(\lambda)) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \\ &= p(x_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \end{aligned} \quad (3.3)$$

接著假設 x_{RES} 與 $x_{\tau,k}(\lambda)$ 沒有相關性，而且假設通道無記憶性，則

$$p(x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}, x_{\tau,k}(\lambda)) = p(x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) \quad (3.4)$$

$$p(\underline{q}_\tau | \underline{y}_\tau) = p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \cdot p(q_{cd}^{(ext)} | x_{cd}^{(ext)}) \cdot p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)}) \cdot p(q_{RES} | x_{RES}) \quad (3.5)$$

整合(3.1)至(3.5)式， $x_{\tau,k}(\lambda)$ 的位元後驗機率為

$$\begin{aligned} p(x_{\tau,k}(\lambda) | \underline{q}_\tau) &= C \cdot p(x_{\tau,k}(\lambda)) \cdot p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \\ &\quad \cdot \sum_{x_{cd}^{(ext)}} p(q_{cd}^{(ext)} | x_{cd}^{(ext)}) \cdot p(x_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \\ &\quad \cdot \sum_{x_{sbsd}^{(ext)}} p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)}) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \\ &\quad \cdot \sum_{x_{RES}} p(q_{RES} | x_{RES}) \cdot p(x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) \end{aligned} \quad (3.6)$$

基於通道無記憶性假設，(3.6)式中最後一項可簡化成

$$\begin{aligned} \sum_{x_{RES}} p(q_{RES} | x_{RES}) \cdot p(x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) &= \sum_{x_{RES}} p(q_{RES}, x_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) \\ &= p(q_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) \end{aligned} \quad (3.7)$$

再利用通道解碼與訊源解碼互相獨立作用的假設，

$$p(q_{RES} | x_{cd}^{(ext)}, x_{sbsd}^{(ext)}) = p(x_{cd}^{(ext)}, x_{sbsd}^{(ext)} | q_{RES}) \cdot p(q_{RES}) / p(x_{cd}^{(ext)}, x_{sbsd}^{(ext)})$$

$$\begin{aligned}
&= p(x_{cd}^{(ext)} | q_{RES}) \cdot p(x_{sbsd}^{(ext)} | q_{RES}) \cdot p(q_{RES}) / (p(x_{cd}^{(ext)}) \cdot p(x_{sbsd}^{(ext)})) \\
&= p(q_{RES} | x_{cd}^{(ext)}) \cdot p(q_{RES} | x_{sbsd}^{(ext)}) / p(q_{RES}) \circ
\end{aligned} \tag{3.8}$$

將(3.7)和(3.8)式帶回(3.6)式可得

$$\begin{aligned}
p(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) &= C' \cdot p(x_{\tau,k}(\lambda)) \cdot p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \\
&\quad \cdot \sum_{x_{cd}^{(ext)}} p(q_{cd}^{(ext)} | x_{cd}^{(ext)}) \cdot p(x_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \cdot p(q_{RES} | x_{cd}^{(ext)}) \\
&\quad \cdot \sum_{x_{sbsd}^{(ext)}} p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)}) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \cdot p(q_{RES} | x_{sbsd}^{(ext)}) \tag{3.9}
\end{aligned}$$

其中 $C' = C / p(q_{RES})$ ，而最後一項在無記憶性通道假設下又可化簡成

$$\begin{aligned}
&\sum_{x_{sbsd}^{(ext)}} p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)}) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \cdot p(q_{RES} | x_{sbsd}^{(ext)}) \\
&= \sum_{x_{sbsd}^{(ext)}} p(q_{sbsd}^{(ext)} | q_{RES}, x_{sbsd}^{(ext)}, x_{\tau,k}(\lambda)) \cdot p(q_{RES} | x_{sbsd}^{(ext)}, x_{\tau,k}(\lambda)) \cdot p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) \\
&= \sum_{x_{sbsd}^{(ext)}} p(q_{sbsd}^{(ext)}, q_{RES}, x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)) = p(q_{sbsd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda)) \tag{3.10}
\end{aligned}$$

同理第三項亦可作此化簡，所以(3.9)式

$$\begin{aligned}
p(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) &= C' \cdot p(x_{\tau,k}(\lambda)) \cdot p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) \\
&\quad \cdot p_{cd}^{(ext)}(x_{\tau,k}(\lambda)) \cdot p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) \tag{3.11}
\end{aligned}$$

其中 $p_{cd}^{(ext)}(x_{\tau,k}(\lambda)) = p(q_{cd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda))$ ，

$p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = p(q_{sbsd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda))$ 由(3.11)式可知特定位元的後驗

機率可拆解成四個獨立成分，前兩項稱為固有訊息(intrinsic information)，第一項為事前訊息，第二項為與通道有關的訊息。後兩項稱為額外訊息(extrinsic information)，即它們是從其他接收值獲得訊息而不是從 $q_{\tau,k}(\lambda)$ 本身得知，其中 $p_{cd}^{(ext)}(x_{\tau,k}(\lambda))$ 來自於同位

檢查位元所提供以位元為基礎(bit level)的冗息以及 q_{RES} ,

$p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ 則來自於 q_{RES} 以及訊源編碼後以符號為基礎(parameter level)的殘餘冗息，相較於 $p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ ，(2.9)式中

$p(z_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda))$ $p(q_{sbsd}^{(ext)} | x_{\tau,k}(\lambda))$ 對於額外訊息的攫取只是從接收值 $q_{sbsd}^{(ext)}$ 獲得資訊，這也是造成軟性位元訊源解碼系統效能無法進一步地提升的主要原因。進一步分析(3.10)式可以得知額外訊息

$p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))$ 的攫取有 3 個來源。分別來自於通道相關訊息

$p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)})$ ，位元組合 $x_{\tau,k}$ 所屬位元之間的相關性 $p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda))$ ，

以及從其他位置接收值 q_{RES} 獲取 $x_{sbsd}^{(ext)}$ 的資訊 $p(q_{RES} | x_{\tau,k}(\lambda))$ 。假使

$x_{sbsd}^{(ext)}$ 個別位元分別經由通道編碼成為碼字內 x_{RES} 的一部份，則

$p(q_{RES} | x_{sbsd}^{(ext)})$ 可經由通道解碼攫取 $x_{sbsd}^{(ext)}$ 個別位元的額外訊息，這意味

著通道解碼輸出的額外訊息可以當作訊源解碼器輸入端的附加事前

訊息。同理可說明訊源解碼輸出的額外訊息可當作通道解碼器輸入端

的附加事前訊息。

3.3 通道解碼額外訊息的推導

為了得到通道額外訊息，通道解碼器必須可以提供軟性輸出值，此值定義為對數相似比率(log-likelihood ratio)。我們將以對數相似代數(log-likelihood algebra)[5]為基礎作演算。相較於在機率

領域計算，可以節省複雜的運算，同時避免了在計算機上以有限位數表示數值的問題。(3.11)式以對數相似比率表示

$$L(x_{\tau,k}(\lambda) | q_{\tau}) = \log \frac{p(x_{\tau,k}(\lambda)=1)}{p(x_{\tau,k}(\lambda)=-1)} + \log \frac{p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=1)}{p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=-1)} \\ + \log \frac{p_{cd}^{(ext)}(x_{\tau,k}(\lambda)=1)}{p_{cd}^{(ext)}(x_{\tau,k}(\lambda)=-1)} + \log \frac{p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)=1)}{p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)=-1)} \quad (3.12)$$

其中第二項在 AWGN 通道假設下可化簡為

$$L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) = \log \frac{p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=1)}{p(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=-1)} \\ = \log \frac{\exp \left\{ -\frac{E_s}{N_0} (q_{\tau,k}(\lambda) - 1)^2 \right\}}{\exp \left\{ -\frac{E_s}{N_0} (q_{\tau,k}(\lambda) + 1)^2 \right\}} = 4 \frac{E_s}{N_0} q_{\tau,k}(\lambda) \quad (3.13)$$

(3.13)式的化簡可視為以對數運算降低計算複雜度的具體實現。

(3.12)式中第3項代表傳送位元 $x_{\tau,k}(\lambda)$ 由通道解碼所得的額外訊息，

可表示為

$$L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{p_{cd}^{(ext)}(x_{\tau,k}(\lambda)=1)}{p_{cd}^{(ext)}(x_{\tau,k}(\lambda)=-1)} \\ = \log \frac{p(q_{cd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda)=1)}{p(q_{cd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda)=-1)} = L(q_{cd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda)) \quad (3.14)$$

我們暫時先不考慮 q_{RES} ，因其對 $L_{cd}^{(ext)}(x_{\tau,k}(\lambda))$ 的影響可藉由額外訊息

的迴授來達成。由 3.1 節可知 $x_{\tau,k}(\lambda)$ 與 $x_{cd}^{(ext)}$ 的關係以 modulo-2 sum

表示

$$x_{\tau,k}(\lambda) = \sum_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}}^{\oplus} x_{\tau,k}(l) \quad (3.15)$$

此累加必須作用於所有 $x_{cd}^{(ext)}$ 包括的 $x_{\tau,k}(l)$

$$L(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) = \log \frac{p(q_{cd}^{(ext)} | \left(\sum_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}}^{\oplus} x_{\tau,k}(l) \right) = 1)}{p(q_{cd}^{(ext)} | \left(\sum_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}}^{\oplus} x_{\tau,k}(l) \right) = -1)}$$

假設 $x_{\tau,k}(l)$ 互相獨立(詳細推導過程示於附錄 A)，則

$$L(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) = \log \frac{1 + \prod_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}} \tanh(L(q_{\tau,k}(\lambda) | x_{\tau,k}(l))/2)}{1 - \prod_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}} \tanh(L(q_{\tau,k}(\lambda) | x_{\tau,k}(l))/2)} \quad (3.17)$$

使用關係式 $\tanh(u/2) = (e^u - 1)/(e^u + 1)$ 則

$$L(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) = 2 \arctanh \prod_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}} \left\{ \tanh(L(q_{\tau,k}(l) | x_{\tau,k}(l))/2) \right\} \quad (3.18)$$

假設所有值 $|L(q_{\tau,k}(l) | x_{\tau,k}(l))/2|$ 都不一樣，且除了其中一項以外其他值都

很大，則 $\tanh$ 函數將趨進於飽和，

$\tanh(L(q_{\tau,k}(l) | x_{\tau,k}(l))/2) \approx \text{sign}\{L(q_{\tau,k}(l) | x_{\tau,k}(l))\}$ 。所以他們都可被提出，而

最小的一項留下。對這一項而言， $\tanh$ 和 artanh 函數作用抵銷，所

以

$$L(q_{cd}^{(ext)} | x_{\tau,k}(\lambda)) \approx \prod_{(x_{\tau,k}(\lambda) \text{ and } \pi_{\tau}) \text{ of } x_{cd}^{(ext)}} \text{sign}(L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda))) \cdot \min_{(x_{\tau,k}(\lambda) \text{ and } \pi_{\tau}) \text{ of } x_{cd}^{(ext)}} |L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda))| \quad (3.19)$$

我們以田符號簡略表達(3.19)式為

$$L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) = \sum_{x_{\tau,k}(l) \text{ of } x_{cd}^{(ext)}} L(q_{\tau,k}(l) | x_{\tau,k}(l))$$

假使有額外訊息迴授，即把 q_{RES} 影響考慮進來且事前訊息不為零，則

$$\begin{aligned} L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) &= L(q_{cd}^{(ext)}, q_{RES} | x_{\tau,k}(\lambda)) = \\ &\sum_{x_{\tau,k}(\lambda) \text{ of } x_{cd}^{(ext)}} L(x_{\tau,k}(\lambda)) + L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) \end{aligned} \quad (3.20)$$

3.4 訊源解碼額外訊息的推導

依據不同程度的訊源冗息，我們將利用 0 階事前訊息與 1 階事前訊息來攫取額外訊息。同樣地 q_{RES} 暫時不予考慮，我們從訊源解碼器的軟性輸出值著手

$$\begin{aligned} L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) &= \log \frac{p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)=1)}{p_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)=-1)} \\ &= \log \frac{p(q_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)=1)}{p(q_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)=-1)} \\ &= \log \frac{\sum_{x_{sbsd}^{(ext)}} p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)=1) \cdot p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)})}{\sum_{x_{sbsd}^{(ext)}} p(x_{sbsd}^{(ext)} | x_{\tau,k}(\lambda)=-1) \cdot p(q_{sbsd}^{(ext)} | x_{sbsd}^{(ext)})} \end{aligned} \quad (3.20)$$

為了簡化標記符號，我們使用另一變數 z 來表示用於訊源解碼過程的接收序列 $z = q_{sbsd}$ 。

[1] 0 階事前訊息的應用

由於量化索引值的不均勻機率分佈，量化索引 $x_{\tau,k}$ 所屬的個別位元之

間存在著殘餘冗息。所以當只有考慮這層關係時 $x_{\tau,k} = (x_{\tau,k}^{(ext)}, x_{\tau,k}(\lambda))$,

其中 $x_{\tau,k}^{(ext)} = x_{sbsd}^{(ext)}$ 。(3.20)式可表示為

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = 1) \cdot p(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = -1) \cdot p(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})} \quad (3.21)$$

以 3 位元作量化為例，此累加 $x_{\tau,k}^{(ext)}$ 與有 $x_{\tau,k}(\lambda)$ 關連性共有 4 種組合。

假設通道無記憶性，

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = 1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(z_{\tau,k}(l) | x_{\tau,k}(l))}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = -1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(z_{\tau,k}(l) | x_{\tau,k}(l))} \quad (3.22)$$

此乘積必須作用於 $x_{\tau,k}^{(ext)}$ 所有位元。此外進一步分析， $p(z_{\tau,k}(l) | x_{\tau,k}(l))$ 可

表示成（詳細推導過程示於附錄 B）

$$p(z_{\tau,k}(l) | x_{\tau,k}(l)) = \Psi(l) \cdot \exp \left\{ \frac{x_{\tau,k}(l)}{2} \cdot L(z_{\tau,k}(l) | x_{\tau,k}(l)) \right\} \quad (3.23)$$

其中 $\Psi(l)$ 此項與其所處的位元位置有關，但對 $x_{\tau,k}(l) = 1$ 與 $x_{\tau,k}(l) = -1$

則分別為一常數。當對 $\Psi(l)$ 所有 $l, l \neq \lambda$ 作乘積時

$$\Psi_{x_{\tau,k}^{(ext)}} = \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} \Psi(l) = C \quad (3.24)$$

所以(3.22)式簡化成為

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)=1) \cdot \Theta_1(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)=-1) \cdot \Theta_1(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})} \quad (3.25)$$

其中 $\Theta_1(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) = \exp \left\{ \sum_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} \frac{x_{\tau,k}(l)}{2} \cdot L(z_{\tau,k}(l) | x_{\tau,k}(l)) \right\}$ ，而

$p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda))$ 可由參數的非均勻分佈求得，即 0 階馬可夫性質

$p(x_{\tau,k})$ 。此外 $\Theta_1(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})$ 包含著通道的相關訊息。若進一步考慮將

通道額外訊息當作附加的事前消息，即把 q_{RES} 的影響納入考量，則修

改(3.25)式如下

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)=1) \cdot \Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda)=-1) \cdot \Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})} \quad (3.26)$$

其中

$$\Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) = \exp \left\{ \sum_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} \frac{x_{\tau,k}(l)}{2} \cdot (L(z_{\tau,k}(l) | x_{\tau,k}(l)) + L_{cd}^{(ext)}(x_{\tau,k}(l))) \right\}$$

[2] 1 階事前訊息的應用

除了考量 $x_{\tau,k}(\lambda)$ 與 $x_{\tau,k}^{(ext)}$ 之間的相關性，同時考慮音框間

(inter-frame) 冗息將可大幅提升系統效能。我們從後驗機率著手，

令 $z_{1,k}^{\tau} = \{z_{1,k}, z_{2,k}, \dots, z_{\tau,k}\}$ ，

$$L(x_{\tau,k}(\lambda) | z_{1,k}^{\tau}) = \log \frac{p(x_{\tau,k}(\lambda)=1 | z_{1,k}^{\tau})}{p(x_{\tau,k}(\lambda)=-1 | z_{1,k}^{\tau})} \quad (3.27)$$

$$= \log \frac{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=1} p(x_{\tau,k} | z_{1,k}^{\tau})}{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=-1} p(x_{\tau,k} | z_{1,k}^{\tau})}$$

根據(2.4)式，利用個別索引事後機率替代 $p(x_{\tau,k} | z_{1,k}^{\tau})$ 可得

$$L(x_{\tau,k}(\lambda) | z_{1,k}^{\tau}) = \log \frac{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=1} p(z_{\tau,k} | x_{\tau,k}) \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=-1} p(z_{\tau,k} | x_{\tau,k}) \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \quad (3.28)$$

在通道無記憶性情況下， $p(z_{\tau,k} | x_{\tau,k})$ 可寫為個別位元的乘積，

$$L(x_{\tau,k}(\lambda) | z_{1,k}^{\tau}) = \log \frac{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=1} \prod_{m=1}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=-1} \prod_{m=1}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \quad (3.29)$$

$$\begin{aligned} &= \log \frac{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=1} p(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=1) \cdot \prod_{m=1, m \neq \lambda}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=-1} p(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)=-1) \cdot \prod_{m=1, m \neq \lambda}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \\ &= L(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + \log \frac{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=1} \prod_{m=1, m \neq \lambda}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}:x_{\tau,k}(\lambda)=-1} \prod_{m=1, m \neq \lambda}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k} | x_{\tau-1,k}) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \quad (3.30) \end{aligned}$$

此外，1 階事前消息 $p(x_{\tau,k} | x_{\tau-1,k})$ 可寫成

$$\begin{aligned} &p(x_{\tau,k}^{(\lambda-1)}, x_{\tau,k}^{(K)} | x_{\tau,k}(\lambda), x_{\tau-1,k}) \cdot p(x_{\tau,k}(\lambda) | x_{\tau-1,k}) \quad (3.31) \\ &= p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda), x_{\tau-1,k}) \cdot p(x_{\tau,k}(\lambda) | x_{\tau-1,k}) \end{aligned}$$

假設 $x_{\tau,k}(\lambda)$ 與 $x_{\tau-1,k}$ 獨立，則 $p(x_{\tau,k}(\lambda) | x_{\tau-1,k}) = p(x_{\tau,k}(\lambda))$ ，而且

$$\begin{aligned}
L(x_{\tau,k}(\lambda) | z_{1,k}^\tau) &= L(x_{\tau,k}(\lambda)) + L(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + \\
&\log \frac{\sum_{x_{\tau,k}, x_{\tau,k}(\lambda)=1} \prod_{m=1, \lambda \neq m}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}, x_{\tau,k}(\lambda)=-1} \prod_{m=1, \lambda \neq m}^K p(z_{\tau,k}(m) | x_{\tau,k}(m)) \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=-1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}
\end{aligned} \tag{3.32}$$

此後驗機率可拆解成 3 項， $L(x_{\tau,k}(\lambda))$ 為事前訊息， $L(z_{\tau,k}(\lambda) | x_{\tau,k}(\lambda))$ 為與通道有關的消息，第 3 項則為訊源解碼輸出的額外訊息。若進一步將通道解碼輸出的額外訊息當作附加事前訊息，利用 $\Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)})$ 替

代 $\prod_{m=1, \lambda \neq m}^K p(z_{\tau,k}(m) | x_{\tau,k}(m))$ ，則(3.32)式中第三項修改

$$\begin{aligned}
L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) &= \log \frac{\sum_{x_{\tau,k}^{(ext)}} \Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}^{(ext)}} \Theta_2(z_{\tau,k}^{(ext)} | x_{\tau,k}^{(ext)}) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=-1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}
\end{aligned} \tag{3.33}$$

第四章 基於迴旋碼的疊代解碼機制

在本章中我們使用迴旋碼為通道編碼，以期比單一同位檢查碼攫取更多的通道額外訊息以提升效能。基於通道解碼器必須能提供軟性輸出值，我們使用 BCJR 當作解碼演算法[6]，因它能依最小化錯誤位元原則作最佳估測。問題是 BCJR 演算法原是根據無記憶性通道假設而推導，但在無線通訊環境中，傳輸信號常因符號間干擾 (intersymbol interference) 及多路徑衰減 (multipath fading) 等因素，使錯誤不再像在加成性白色高斯雜訊通道中獨立發生，而是傾向於叢集性出現。此通道稱具有記憶性，即錯誤之間在統計上具有相關性，無記憶性二元對稱通道將不足以描述通道記憶的特性。所以我們嘗試在迴旋碼的籬柵 (trellis) 結構下，在通道解碼時利用通道記憶性以有效對抗這種叢發性錯誤。我們在 4.1 節簡介通道編碼模型，接著在 4.2 節中先推導無記憶性通道的解碼額外訊息，最後在 4.3 節中計算額外訊息時把通道記憶性納入考量。

4.1 通道編碼模型

迴旋碼為另一種常見的通道編碼機制，不同於區塊碼，它具有記

憶特性，其編碼器是以線性序向電路(linear sequential circuit)實現。考慮一個碼率 $R=1/n$ ，由 J 個暫存器組成的編碼器，假設在時間 t 輸入端有一訊息位元 u_t ，而輸出的碼字位元組合為 $\underline{y}_t = \{y_{t,0}, y_{t,1}, \dots, y_{t,n-1}\}$ 。迴旋編碼機制可利用生成多項式(generator polynomial) $G_l(D)$ 來描述， $l=0, 1, \dots, n-1$ ， D 為一延遲作用子(delay operator)。本論文所採用的迴旋系統碼(recursive systematic code)如圖 4.1 所示， $n=J=2$ ，兩個生成多項式分別為 $G_0(D)=1$ ，

$$G_1(D) = \frac{1+D^2}{1+D+D^2}。$$

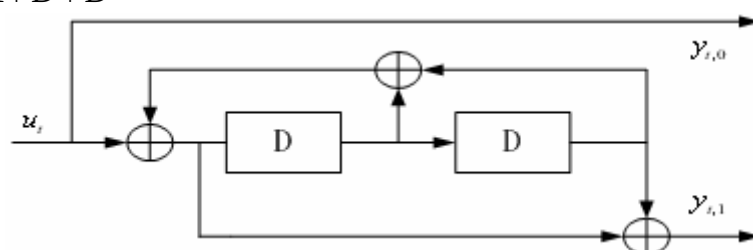


圖 4.1 $R=1/2$ 的遞迴系統碼

此時輸出位元 $y_{t,0}$ 與訊息位元 u_t 相同， $y_{t,1}$ 為同位檢查位元。我們可以使用狀態圖來描述編碼器編碼運作，若以時間為軸將狀態圖做展開即可得到籬柵圖，這將運用在解碼過程，如圖 4.2 所示

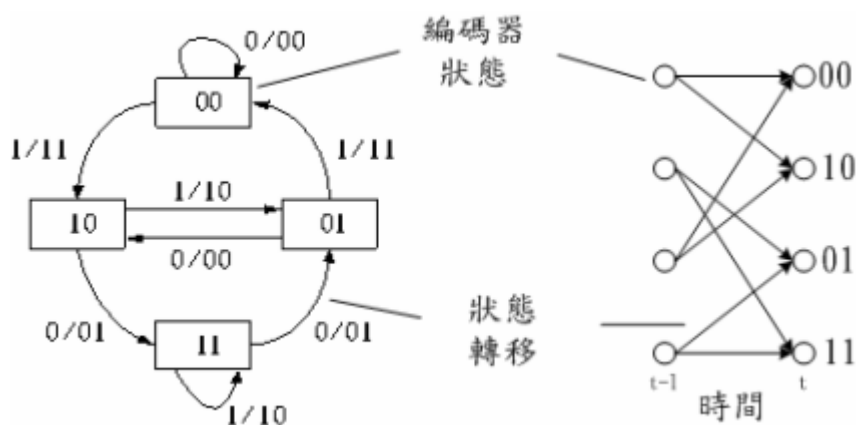


圖 4.2 遞迴系統碼的狀態圖(左)和籬柵圖(右)

1/11 表示訊息位元 u_t 為 1，輸出位元 $y_{t,0} y_{t,1}$ 為 11。此外當編碼結束時會再多輸入非訊息位元的尾巴位元，迫使編碼器狀態回復原來的起始狀態 $S=0$ 。以下為一可行作法，

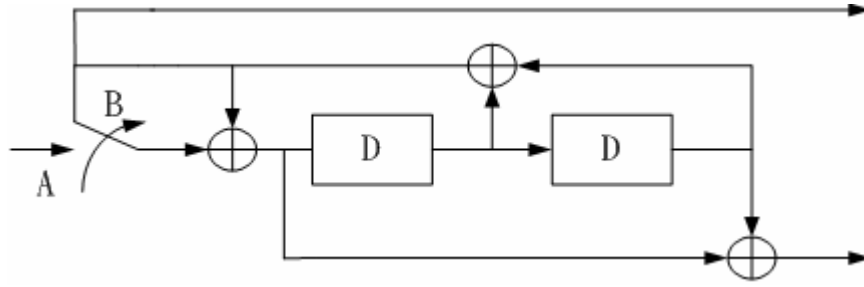


圖 4.3 籬柵終止

在編碼器輸入端做一開關，當在編碼時開關在 A，結束時在 B。

4.2 無記憶性通道的解碼額外訊息推導

假設通道編碼如上一節所提，並假設每次傳送是以訊息位元加上尾巴位元共 M 個為一區塊，則接收的實數序列， $t=0, 1, \dots, M-1$ 。對個別位元言， $q_{t,l} = y_{t,l} + n_{t,l}$ ， $\underline{q}_t \triangleq \{q_{t,0}, q_{t,1}, \dots, q_{t,n-1}\}$ ，其中 $n_{t,l}$ 為一零平均值且變異數 $\sigma^2 = N_0 / 2E_s$ 的高斯雜訊取樣， N_0 為通道的單邊雜訊功率頻譜密度， E_s 為每一編碼符號能量。我們以 BCJR 演算法估測每一訊息位元 u_t 的後驗機率 $p(x_{t,k}(\lambda) | \underline{q}_0^{M-1}) = p(u_t | \underline{q}_0^{M-1})$ ，並利用迴旋碼籬柵結構，則通道解碼器的軟性輸出值為

$$L(\hat{u}_t) = \log \frac{p(u_t = 1 | \underline{q}_0^{M-1})}{p(u_t = -1 | \underline{q}_0^{M-1})}$$

$$= \log \frac{\sum_{(s',s) \in \Omega_t^+} p(s_t = s', s_{t+1} = s, \underline{q}_0^{M-1})}{\sum_{(s',s) \in \Omega_t^-} p(s_t = s', s_{t+1} = s, \underline{q}_0^{M-1})} \quad (4.4)$$

其中 Ω_t^+ 為在時間 t 位元 $u_t=1$ 所對應從狀態 $s_t = s'$ 轉移到時間 $t+1$ 狀態 $s_{t+1} = s$ 的合理途徑集合，同理 Ω_t^- 為在時間 t 位元 $u_t=-1$ 所對應從狀態 $s_t = s'$ 轉移到時間 $t+1$ $s_{t+1} = s$ 的合理途徑集合，結合機率(joint probability)可以表示為

$$p(s_t = s', s_{t+1} = s, \underline{q}_0^{M-1}) = p(s_t = s', s_{t+1} = s, \underline{q}_0^{t-1}, \underline{q}_t, \underline{q}_{t+1}^{M-1}) \quad (4.5)$$

其中 $\underline{q}_0^{t-1}$ 表示在時間 t 以前接收的序列， $\underline{q}_{t+1}^{M-1}$ 表示在時間 t 之後接收的序列。

$$\begin{aligned} p(s_t = s', s_{t+1} = s, \underline{q}_0^{M-1}) &= p(\underline{q}_{t+1}^{M-1} | s_t = s', s_{t+1} = s, \underline{q}_0^{t-1}, \underline{q}_t) \cdot p(s_t = s', s_{t+1} = s, \underline{q}_0^{t-1}, \underline{q}_t) \\ &= p(\underline{q}_{t+1}^{M-1} | s_t = s', s_{t+1} = s, \underline{q}_0^{t-1}, \underline{q}_t) \cdot p(s_{t+1} = s, \underline{q}_t | s_t = s', \underline{q}_0^{t-1}) \cdot p(s_t = s', \underline{q}_0^{t-1}) \end{aligned} \quad (4.6)$$

因為在時間 $t+1$ 之後的路徑僅視狀態 s_{t+1} 決定，而從狀態 s_t 轉移至 s_{t+1} 與接收序列 $\underline{q}_0^{M-1}$ 無關，所以

$$p(\underline{q}_{t+1}^{M-1} | s_t = s', s_{t+1} = s, \underline{q}_0^{t-1}, \underline{q}_t) = p(\underline{q}_{t+1}^{M-1} | s_{t+1} = s) \quad (4.7)$$

$$p(s_{t+1} = s, \underline{q}_t | s_t = s', \underline{q}_0^{t-1}) = p(s_{t+1} = s, \underline{q}_t | s_t = s') \quad (4.8)$$

結合機率則可以化簡為，

$$p(s_t = s', s_{t+1} = s, \underline{q}_0^{M-1}) = p(\underline{q}_{t+1}^{M-1} | s_{t+1} = s) \cdot p(s_{t+1} = s, \underline{q}_t | s_t = s') \cdot p(s_t = s', \underline{q}_0^{t-1})$$

$$= \beta_{t+1}(s) \cdot \gamma_t(s', s) \cdot \alpha_t(s') \quad (4.9)$$

其中

$$\alpha_t(s') \triangleq p(s_t = s', \underline{q}_0^{t-1}) \quad (4.10)$$

$$\beta_{t+1}(s) \triangleq p(\underline{q}_{t+1}^{M-1} | s_{t+1} = s) \quad (4.11)$$

$$\gamma_t(s', s) = p(s_{t+1} = s, \underline{q}_t | s_t = s') \quad (4.12)$$

α ， β 分別為前項(forward)和後項(backward)機率，因此 BCJR 演算法也稱作前後項演算法。此外，前項和後項機率可以各自推導成遞迴形式以降低運算，詳細推導如下。前項機率 $\alpha_{t+1}(s)$ 可展開為

$$\begin{aligned} \alpha_{t+1}(s) &= p(s_{t+1} = s, \underline{q}_0^t) = \sum_{s' \in \sigma_t} p(s_t = s', s_{t+1} = s, \underline{q}_0^t) \\ &= \sum_{s' \in \sigma_t} p(s_{t+1} = s, \underline{q}_t | s_t = s', \underline{q}_0^{t-1}) \cdot p(s_t = s', \underline{q}_0^{t-1}) \end{aligned} \quad (4.13)$$

因為接收序列 $\underline{q}_0^{t-1}$ 與 s_t 至 s_{t+1} 的轉移路徑無關，所以前項機率可寫為

$$\begin{aligned} \alpha_{t+1}(s) &= \sum_{s' \in \sigma_t} p(s_{t+1} = s, \underline{q}_0^t | s_t = s') \cdot p(s_t = s', \underline{q}_0^{t-1}) \\ &= \sum_{s' \in \sigma_t} \gamma_t(s', s) \cdot \alpha_t(s') \end{aligned} \quad (4.14)$$

其中 σ_t 為在時間 t 所有狀態的集合。因編碼器的起始狀態為 $s_0 = 0$ ，

前項遞迴機率在時間 $t=0$ 的初始設定為

$$\alpha_0(s) = \begin{cases} 1, & s=0 \\ 0, & s \neq 0 \end{cases} \quad (4.15)$$

此外，為了避免在計算上出現數值過小的問題，我們必須正常化

$$\alpha_{t+1}(s)$$

$$\alpha_{t+1}(s) = \frac{\sum_{s' \in \sigma_t} \gamma_t(s', s) \cdot \alpha_t(s')}{\sum_{s \in \sigma_{t+1}} \sum_{s' \in \sigma_t} \gamma_t(s', s) \cdot \alpha_t(s')} \quad (4.16)$$

我們利用(4.16)式遞迴計算 $\alpha_{t+1}(s)$ ， $t=0, 1, \dots, M-1$ 。前項遞迴機率

對應的籬柵圖如圖 4.4 所示。

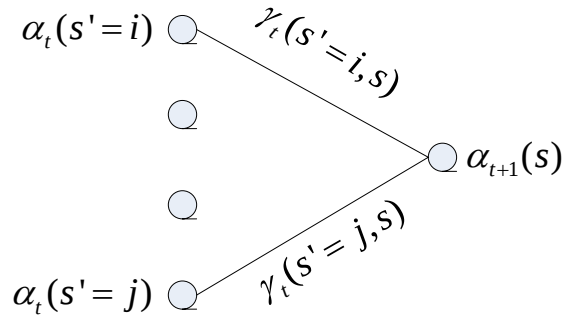


圖 4.4 前項遞迴機率籬柵圖

後項機率 $\beta_t(s')$ 可展開為

$$\begin{aligned} \beta_t(s') &= p(\underline{q}_{t+1}^{M-1} | s_t = s') \\ &= \sum_{s \in \sigma_{t+1}} p(\underline{q}_t^{M-1}, s_{t+1} = s | s_t = s') \\ &= \sum_{s \in \sigma_{t+1}} p(\underline{q}_t, \underline{q}_{t+1}^{M-1}, s_{t+1} = s | s_t = s') \\ &= \sum_{s \in \sigma_{t+1}} p(\underline{q}_{t+1}^{M-1} | \underline{q}_t, s_t = s', s_{t+1} = s) \cdot p(\underline{q}_t, s_{t+1} = s | s_t = s') \quad (4.17) \end{aligned}$$

因在時間 $t+1$ 之後的路徑僅與狀態 s_{t+1} 有關，所以後項機率可化簡為

$$\begin{aligned} \beta_t(s') &= \sum_{s \in \sigma_{t+1}} p(\underline{q}_{t+1}^{M-1} | s_{t+1} = s) \cdot p(\underline{q}_t, s_{t+1} = s | s_t = s') \\ &= \sum_{s \in \sigma_{t+1}} \gamma_t(s', s) \cdot \beta_{t+1}(s) \quad (4.18) \end{aligned}$$

其中 σ_{t+1} 為在時間 $t+1$ 所有狀態的集合。因編碼器的最終狀態

$s_M = 0$ ，後項遞迴機率在時間 M 的起始設定為

$$\beta_M(s) = \begin{cases} 1, & s=0 \\ 0, & s \neq 0 \end{cases} \quad (4.19)$$

為了避免在計算上出現數值過小的問題我們必須正常化 $\beta_t(s')$

$$\beta_t(s') = \frac{\sum_{s \in \sigma_{t+1}} \gamma_t(s', s) \cdot \beta_{t+1}(s)}{\sum_{s' \in \sigma_t} \sum_{s \in \sigma_{t+1}} \gamma_t(s', s) \cdot \beta_{t+1}(s)} \quad (4.20)$$

我們使用(4.20)式遞迴計算 $\beta_t(s')$ ， $t=M-1, M-2, \dots, 0$ 。後項遞迴機率

對應的籬柵圖如圖 4.5 所示。

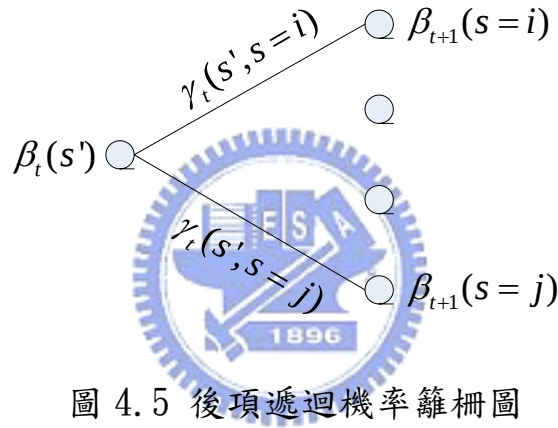


圖 4.5 後項遞迴機率籬柵圖

轉移機率 $\gamma_t(s', s)$ 可展開為

$$\begin{aligned} \gamma_t(s', s) &= p(s_{t+1} = s, \underline{q}_t | s_t = s') = \frac{p(s_{t+1} = s, s_t = s', \underline{q}_t)}{p(s_t = s')} \\ &= \frac{p(s_{t+1} = s, s_t = s')}{p(s_t = s')} \cdot \frac{p(s_{t+1} = s, s_t = s', \underline{q}_t)}{p(s_{t+1} = s, s_t = s')} \\ &= p(s_{t+1} = s | s_t = s') \cdot p(\underline{q}_t | s_{t+1} = s, s_t = s') \\ &= p(u_t) \cdot p(\underline{q}_t | \underline{y}_t) \end{aligned} \quad (4.21)$$

其中 u_t 為訊息位元，而 $\underline{y}_t$ 為相對應於狀態轉移 $s_t = s' \rightarrow s_{t+1} = s$

的輸出位元。假設通道無記憶性，且 $s_t = s' \rightarrow s_{t+1} = s$ 為一合理的狀態

轉移，則

$$\gamma_t(s', s) = p(u_t) \cdot \left(\sqrt{\frac{E_s}{\pi N_0}} \right)^n \cdot e^{-\frac{E_s}{N_0} \|\underline{q}_t - \underline{y}_t\|^2} \quad (4.22)$$

其中 $\|\underline{q}_t - \underline{y}_t\|^2$ 為歐式距離(Euclidean distance)的平方。假如

$s_t = s' \rightarrow s_{t+1} = s$ 不是一合理狀態轉移，則 $p(s_{t+1} = s | s_t = s')$ 與 $\gamma_t(s', s)$

皆為零。藉由對數相似比率，事前訊息 $p(u_t = \pm 1)$ 可表示為

$$p(u_t = \pm 1) = \frac{e^{\pm L(u_t)}}{1 + e^{\pm L(u_t)}} = \left(\frac{e^{-L(u_t)/2}}{1 + e^{-L(u_t)}} \right) \cdot e^{u_t L(u_t)/2} \quad (4.23)$$

此外，(4.22)式第三項為

$$\begin{aligned} e^{-\frac{E_s}{N_0} \|\underline{q}_t - \underline{y}_t\|^2} &= e^{(2E_s/N_0)(\underline{q}_t \cdot \underline{y}_t) - \|\underline{q}_t\|^2 - \|\underline{y}_t\|^2} \\ &= e^{-(\|\underline{q}_t\|^2 + \|\underline{y}_t\|^2)} \cdot e^{(L_c/2)(\underline{q}_t \cdot \underline{y}_t)} \end{aligned} \quad (4.24)$$

其中 $L_c = 4E_s / N_0$ 稱為通道可靠因子(channel reliability factor)。

對系統迴旋碼而言， $e^{(L_c/2)(\underline{q}_t \cdot \underline{y}_t)}$ 可寫成

$$e^{\left(\frac{1}{2} L_c q_{t,0} u_t + \frac{1}{2} L_c q_{t,1} y_{t,1} \right)} \quad (4.25)$$

綜合上述推導，轉移機率 $\gamma_t(s', s)$ 可表示為

$$\gamma_t(s', s) = A_t \cdot B_t \cdot \gamma_t^{(e)}(s', s) \quad (4.26)$$

其中

$$\begin{aligned} A_t &= \left(\frac{e^{-L(u_t)/2}}{1 + e^{-L(u_t)}} \right) \cdot \left(\sqrt{\frac{E_s}{\pi N_0}} \right)^n \cdot e^{-(\|\underline{q}_t\|^2 + \|\underline{y}_t\|^2)} \\ B_t &= e^{\frac{1}{2} u_t (L(u_t) + L_c q_{t,0})} \end{aligned}$$

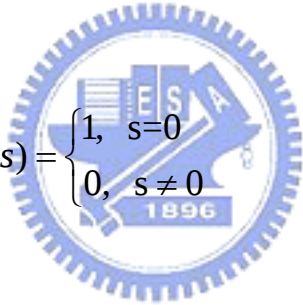
$$\gamma_t^{(e)}(s', s) = e^{\frac{1}{2}L_c q_{t,1} y_{t,1}} \quad (4.27)$$

在(4.4)式累加運算中(4.26)式的 B_t 為一常數，故可提出，而且 A_t 同時出現在(4.4)式的分子與分母，所以可互相抵銷。

$$L(\hat{u}_t) = L_c q_{t,0} + L(u_t) + \log \frac{\sum_{(s', s) \in \Omega_t^+} \gamma_t^{(e)}(s', s) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')}{\sum_{(s', s) \in \Omega_t^-} \gamma_t^{(e)}(s', s) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')} \quad (4.28)$$

通道解碼輸出的對數相似比率可拆解成三項， $L(u_t)$ 為事前訊息， $L_c q_{t,0}$ 為與通道有關訊息，第三項則為通道解碼所得的額外訊息。計算此項額外訊息的步驟以下：

1 初始設定

$$\alpha_0(s) = \begin{cases} 1, & s=0 \\ 0, & s \neq 0 \end{cases} \quad (4.29)$$


2 初始設定

$$\beta_M(s) = \begin{cases} 1, & s=0 \\ 0, & s \neq 0 \end{cases} \quad (4.30)$$

3 計算 $\gamma^{(e)}$

$$\gamma_t^{(e)}(s', s) = e^{\frac{1}{2}L_c q_{t,1} y_{t,1}} \quad (4.31)$$

4 計算轉移機率

$$\gamma_t(s', s) = e^{\frac{1}{2}u_t(L_c q_{t,0} + L(u_t))} \cdot \gamma_t^{(e)}(s', s) \quad (4.32)$$

5 計算前項遞迴機率

$$\alpha_{t+1}(s) = \frac{\sum_{s' \in \sigma_t} \gamma_t(s', s) \cdot \alpha_t(s')}{\sum_{s \in \sigma_{t+1}} \sum_{s' \in \sigma_t} \gamma_t(s', s) \cdot \alpha_t(s')} \quad (4.33)$$

6 計算後項遞迴機率

$$\beta_t(s') = \frac{\sum_{s \in \sigma_{t+1}} \gamma_t(s', s) \cdot \beta_{t+1}(s)}{\sum_{s' \in \sigma_t} \sum_{s \in \sigma_{t+1}} \gamma_t(s', s) \cdot \beta_{t+1}(s)} \quad (4.34)$$

7 計算通道解碼輸出的額外訊息

$$L_{cd}^{(ext)}(u_t) = \log \frac{\sum_{(s', s) \in \Omega_t^+} \gamma_t^{(e)}(s', s) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')}{\sum_{(s', s) \in \Omega_t^-} \gamma_t^{(e)}(s', s) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')} \quad (4.35)$$

在疊代過程中，若進一步將訊源解碼輸出的額外訊息當作附加事前訊息，(4.32)式必須修改為

$$\gamma_t(s', s) = e^{\frac{1}{2}u_t(L_c q_{t,0} + L(u_t) + L_{sbsd}^{(ext)}(u_t))} \cdot \gamma_t^{(e)}(s', s) \quad (4.36)$$

4.3 記憶性通道的解碼額外訊息推導

為了匹配通道的記憶特性，必須採用參數化通道模型以描述位元錯誤發生的相關統計訊息。通常是以有限狀態馬可夫通道(Finite state Markov channel)描述，每一通道狀態描述不同的通道品質且通道狀態之間轉移機率僅與目前的通道狀態有關。Gilbert 模型[7]為一最常見的通道模型，其作法是將位元錯誤序列模擬成一階馬可夫程序，如圖 4.7 所示。Gilbert 模型由兩個通道狀態組成，在狀態 G

時沒有錯誤發生，在狀態 B 時錯誤發生機率為 $1-h$ 。由狀態 G 到狀態 B 的轉移機率為 b ，狀態 B 到狀態 G 的轉移機率為 g 。通道在狀態 G 的機率為

$$p(G) = \frac{p(G|B)}{p(G|B) + p(B|G)} = \frac{g}{b+g} \quad (4.37)$$

通道在狀態 B 的機率為

$$p(B) = 1 - p(G) = \frac{b}{b+g} \quad (4.38)$$

通道的位元錯誤率為

$$p_e = (1-h) \frac{b}{b+g} \quad (4.39)$$

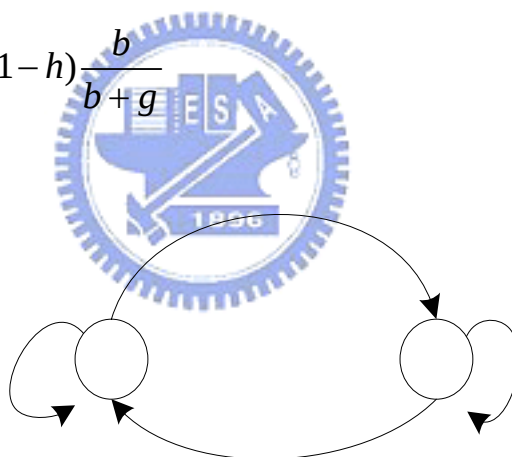


圖 4.6 Gilbert 模型

4.3.1 通道解碼額外訊息的計算

在(4.21)式中所推導的轉移機率

$$\gamma_t(s', s) = p(u_t) \cdot p(\underline{q}_t | \underline{y}_t) \quad (4.40)$$

其中 $p(\underline{q}_t | \underline{y}_t)$ 在有記憶性通道可表示成[8]

$$p(\underline{q}_t | \underline{y}_t) = p(\underline{e}_t) = \pi \prod_{l=0}^{n-1} p_e(e_l) \underline{1} \quad , \quad e_l = q_{t,l} \oplus y_{t,l} \quad (4.41)$$

其中 π 為初始狀態機率， $\underline{1}$ 為一個元素皆為 1 的向量。當採用 Gilbert 通道模型時，初始狀態機率 π 為

$$\pi = \left(\frac{g}{g+b} \quad \frac{b}{g+b} \right) \quad (4.42)$$

而 $p_e(e_l) = 0$ 和 $p_e(e_l) = 1$ 分別為

$$p_e(0) = \begin{bmatrix} 1-b & bh \\ g & (1-g)h \end{bmatrix} \quad , \quad p_e(1) = \begin{bmatrix} 0 & b(1-h) \\ 0 & (1-g)(1-h) \end{bmatrix} \quad (4.43)$$

所以轉移機率 $\gamma_t(s', s)$ 可寫成

$$\gamma_t(s', s) = p(u_t) \cdot p(\underline{e}_t) \quad (4.44)$$

將(4.44)式帶回(4.9)式配合(4.5)式與(4.4)式，則 $L(\hat{u}_t)$ 可表示為

$$L(\hat{u}_t) = L(u_t) + \log \frac{\sum_{(s', s) \in \Omega_t^+} p(\underline{e}_t) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')}{\sum_{(s', s) \in \Omega_t^-} p(\underline{e}_t) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')} \quad (4.45)$$

此值可拆解成兩項，第二項為通道解碼輸出的額外訊息。計算步驟如

4.2 節所提，(4.29)式至(4.34)式不變，接下來使用(4.41)式計算

$p(\underline{e}_t)$ ，再代入

$$L_{cd}^{(ext)}(u_t) = \log \frac{\sum_{(s', s) \in \Omega_t^+} p(\underline{e}_t) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')}{\sum_{(s', s) \in \Omega_t^-} p(\underline{e}_t) \cdot \beta_{t+1}(s) \cdot \alpha_t(s')} \quad (4.46)$$

在疊代過程中，若進一步將訊源解碼輸出的額外訊息當作附加事前訊

息，(4.44)式須修改為

$$\gamma_t(s', s) = e^{\frac{1}{2}u_t(L(u_t) + L_{sbsd}^{(ext)}(u_t))} \cdot p(e_t) \quad (4.47)$$

4.3.2 訊源解碼額外訊息的計算

[1] 0 階事前訊息的應用

若考慮記憶性通道，則(3.22)式修改成

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = 1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l))}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = -1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l))} \quad (4.48)$$

其中 $e_{\tau,k}(l) = z_{\tau,k}(l) \cdot x_{\tau,k}(l)$,

$$p(e_{\tau,k}(l)) = \begin{cases} \pi p_e(0) \underline{1}, & e_{\tau,k}(l) = 1 \\ \pi p_e(1) \underline{1}, & e_{\tau,k}(l) = -1 \end{cases} \quad (4.49)$$

若進一步將通道解碼額外訊息當作附加的事前訊息，則修改(4.48)

式如下

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = 1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l)) \cdot p_{cd}^{(ext)}(x_{\tau,k}(l))}{\sum_{x_{\tau,k}^{(ext)}} p(x_{\tau,k}^{(ext)} | x_{\tau,k}(\lambda) = -1) \cdot \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l)) \cdot p_{cd}^{(ext)}(x_{\tau,k}(l))} \quad (4.50)$$

其中

$$p_{cd}^{(ext)}(x_{\tau,k}(l)) = \begin{cases} \frac{1}{1 + \exp(L_{cd}^{(ext)}(x_{\tau,k}(l)))}, & x_{\tau,k}(l) = -1 \\ \frac{1}{1 + \exp(-L_{cd}^{(ext)}(x_{\tau,k}(l)))}, & x_{\tau,k}(l) = 1 \end{cases} \quad (4.51)$$

[2] 1 階事前訊息的應用

若考慮記憶性通道，則(3.33)式修改成

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}^{(ext)}} \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} p(e_{\tau,k}(l)) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=-1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \quad (4.51)$$

若進一步將通道解碼額外訊息當作附加的事前訊息，則修改(4.51)

式如下

$$L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda)) = \log \frac{\sum_{x_{\tau,k}^{(ext)}} \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} (p(e_{\tau,k}(l)) \cdot p_{cd}^{(ext)}(x_{\tau,k}(l))) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})}{\sum_{x_{\tau,k}^{(ext)}} \prod_{x_{\tau,k}(l) \text{ of } x_{\tau,k}^{(ext)}} (p(e_{\tau,k}(l)) \cdot p_{cd}^{(ext)}(x_{\tau,k}(l))) \cdot \sum_{x_{\tau-1,k}} p(x_{\tau,k}^{(ext)} | x_{\tau-1,k}, x_{\tau,k}(\lambda)=-1) \cdot p(x_{\tau-1,k} | z_{1,k}^{\tau-1})} \quad (4.52)$$

第五章 實驗模擬與結果分析

在前面章節裡我們介紹了單一同位檢查碼與迴旋碼，同時結合軟性位元訊源解碼進而推導兩種疊代解碼演算法。在本章中，進一步將演算法實現於系統模擬，以期藉由模擬的結果驗證演算法的優劣性。本章的內容將分為三小節來進行實驗模擬與結果之分析討論。第一節與第二節模擬的通道環境為 AWGN，分別進行單一同位檢查碼與迴旋碼為基礎的疊代解碼演算，並針對不同程度的通道解碼額外訊息作效能評比與分析。第三節模擬通道環境則設定為 Gilbert 位元錯誤通道模型，進行迴旋碼結合軟性位元訊源解碼的疊代解碼演算，並探討運用通道記憶特性及不同交錯器種類對解碼效能的影響。

5.1 基於單一同位檢查碼之疊代解碼實驗

5.1.1 系統模擬之步驟說明

訊號源 $u_{\tau,k}$ 取自一階自迴歸處理(first-order autoregressive process)，變異數設定為 $\sigma^2=1$ ，且其相關因子(correlation factor)採用 $\rho=0.95$ 。關係式表示為

$$u_{\tau,k} = \rho u_{\tau-1,k} + n_{\tau} \quad (5.1)$$

其中 n_r 為均值零的白色高斯雜訊。在每次模擬過程中，由訊號源產生 30000 點的資料作為系統之輸入，接著將訊號源送進純量量化器中，所採用的量化器為 Lloyd-Max 量化器[9]。量化索引使用三個位元表示，其位元映射(bit mapping)則選用折疊二位元編碼方式(folded binary code)。再以九個訊息位元為一區塊進行編碼率 $r=\frac{3}{4}$ 的通道編碼處理，輸出一組由 12 個碼字位元組成的系統碼。接著將其通過二位元相位鍵移(BPSK)調變後，經由白色高斯雜訊通道送至接收端。

在接收端，接收到的序列不再是傳送的位元 $\{1, -1\}$ ，而是經由 AWGN 雜訊擾亂後的實數值。將其輸入至軟性輸出的通道解碼器，解碼輸出的額外訊息再運用於訊源解碼器。此處必須注意在通道解碼過程中，必須以區塊長度作為演算法實現之間隔區塊，每一區塊所解出之通道額外訊息對等於傳送時訊息位元的訊息。疊代過程如第 2 章所描述。

最後進行參數估測時，依疊代過程使用不同程度的訊息而有下列情況。

1) 0 iteration:

$$L(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) = L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L(x_{\tau,k}(\lambda)) \quad (5.2)$$

2) 0^+ iteration:

$$L(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) = L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L(x_{\tau,k}(\lambda)) + L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) \quad (5.3)$$

3) 1 iteration:

$$L(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) = L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L(x_{\tau,k}(\lambda)) + \underbrace{L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) + L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))}_{\text{更新}} \quad (5.4)$$

4) 1⁺ iteration:

$$L(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) = L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L(x_{\tau,k}(\lambda)) + \underbrace{L_{cd}^{(ext)}(x_{\tau,k}(\lambda)) + L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))}_{\text{更新}} \quad (5.5)$$

5) 2 iteration:

$$L(x_{\tau,k}(\lambda) | \underline{q}_{\tau}) = L(q_{\tau,k}(\lambda) | x_{\tau,k}(\lambda)) + L(x_{\tau,k}(\lambda)) + \underbrace{L_{cd}^{(ext)}(x_{\tau,k}(\lambda))}_{\text{更新}} + \underbrace{L_{sbsd}^{(ext)}(x_{\tau,k}(\lambda))}_{\text{更新}} \quad (5.6)$$

最後比照(2.5)式重建參數。其中

$$p(x_{\tau,k} | q_{\tau,k}) = \prod_{\lambda=0}^2 p(x_{\tau,k}(\lambda) | q_{\tau,k}) \quad (5.7)$$

$$p(x_{\tau,k}(\lambda) | q_{\tau,k}) = \begin{cases} \frac{1}{1 + e^{-L(x_{\tau,k}(\lambda) | \underline{q}_{\tau})}}, & x_{\tau,k}(\lambda) = 1 \\ \frac{1}{1 + e^{L(x_{\tau,k}(\lambda) | \underline{q}_{\tau})}}, & x_{\tau,k}(\lambda) = -1 \end{cases} \quad (5.8)$$

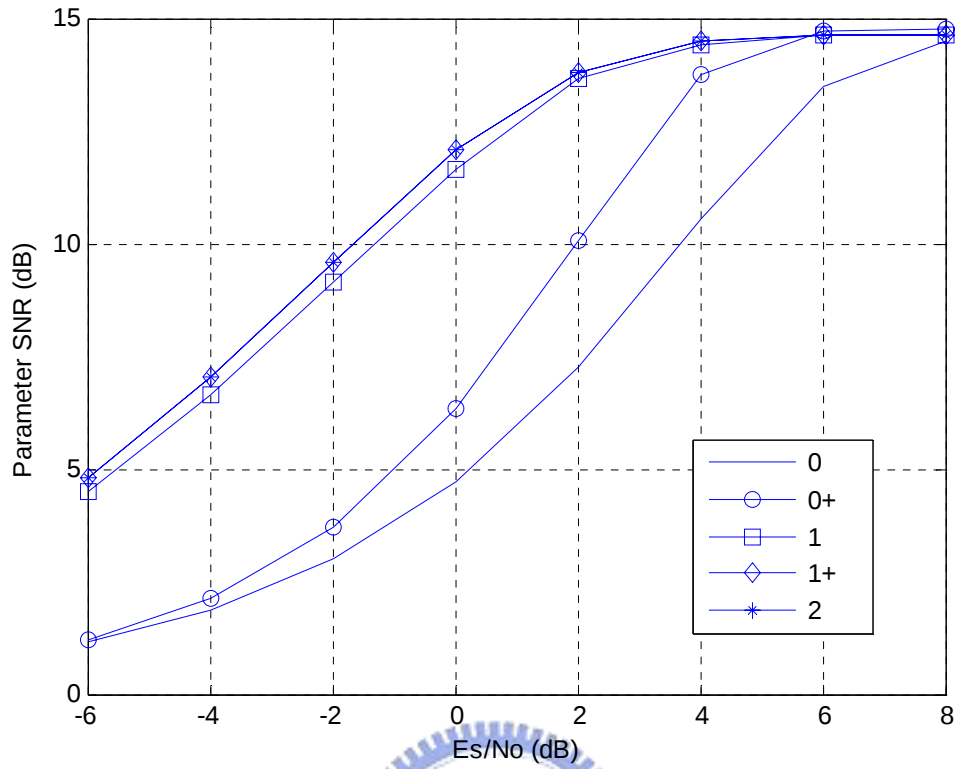


圖 5.1 基於單一同位檢查碼之疊代解碼結果

5.1.2 結果分析

實驗結果如圖 5.1 所示，系統評估使用訊號源參數 $u_{\tau,k}$ 和重建參數 $\hat{u}_{\tau,k}$ 之間的訊號雜訊比(signal-to-noise ratio, SNR)作為衡量：

$$SNR = 10 \log_{10} \frac{u_{\tau,k}^2}{(u_{\tau,k} - \hat{u}_{\tau,k})^2} \quad [\text{dB}] \quad (5.9)$$

隨著疊代次數增加，使用更多的額外訊息時，系統效能也跟著提升。0 次疊代不考慮通道解碼與訊源解碼，0⁺ 次疊代則加入通道解碼輸出的額外訊息。當通道情況較差時，0 次與 0⁺ 次疊代效能衰退很多，且彼此之間效能差不多；但當通道情況較好時，0⁺ 次疊代的效能提升比較明顯。在 1 次疊代時，進一步將訊源解碼的額外訊息納入考量，比

0 次與 0^+ 次疊代效能更大幅改善。在 1^+ 次疊代時，因為用了更準確的通道額外訊息，在通道情況較差時大概有 0.4dB 的效能提升。但在 2 次疊代時，系統效能則趨於飽和。所有疊代次數在通道情況很好時，系統效能才會到達 14.62dB 的臨界值。

5.2 基於迴旋碼之疊代解碼實驗

5.2.1 系統模擬之步驟說明

訊號源 $u_{\tau,k}$ 取自一階自迴歸處理，變異數設定為 $\sigma^2=1$ ，且其相關因子採用 $\rho=0.95$ 。在每次模擬過程中，由訊號源產生出 30000 點的資料送進純量 Lloyd-Max 量化器，量化索引使用三個位元。位元映射方法選用折疊二位元編碼方式。接著以 150 個訊息位元為一區塊，由 3×50 的區塊交錯器進行插置。接下來以這 150 個訊息位元，外加 2 個迫使狀態器狀態回歸至零的尾巴位元為一區塊長度送進迴旋編碼器中。將輸出位元組經由二位元相位鍵移調變後，經由白色高斯雜訊通道送至接收端。通道解碼處理必須以一個區塊長度進行，每一區塊解出 150 個訊息位元和 2 個尾巴位元的額外訊息，再將多餘的尾巴位元之額外訊息捨棄。疊代過程使用不同程度的訊息如(5.2)式至(5.6)式所描述。再利用(5.7)式與(5.8)式可得重建參數。實驗結果如圖 5.2 所示

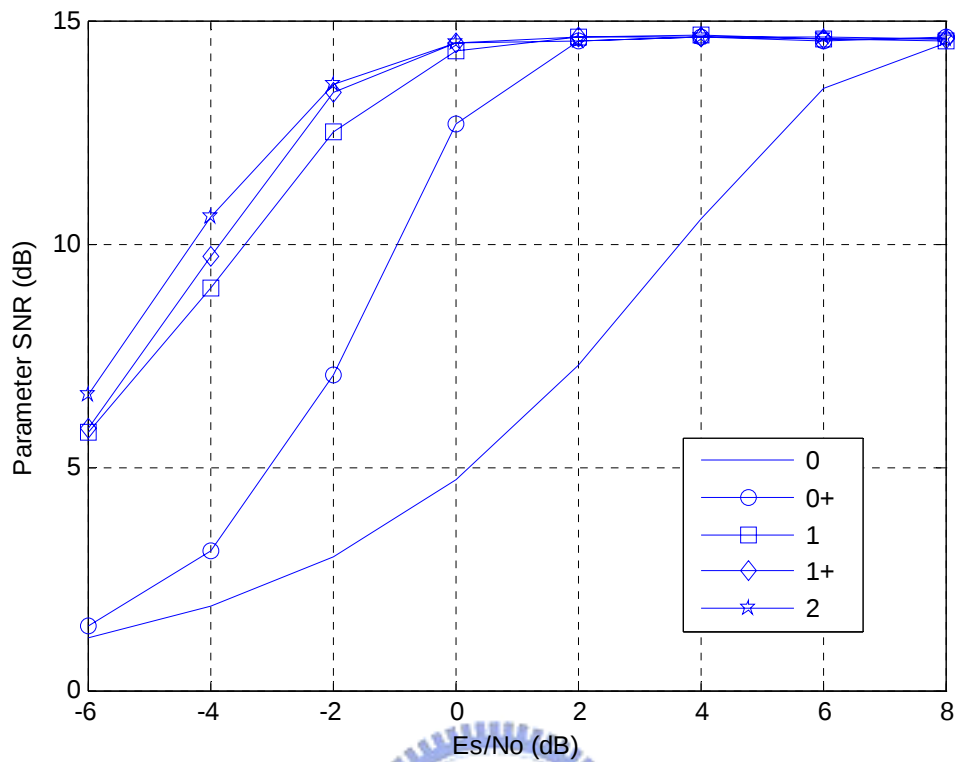


圖 5.2 基於迴旋碼之疊代解碼系統模擬

5.2.2 結果分析

圖 5.2 之系統評估，同樣使用 5.9 式的 SNR 值作為衡量。相較於 0 次疊代，0⁺ 次疊代的解碼效能在通道情況較差時差異不大，但在通道較好時則大幅提升，且 Es/No=2dB 時即已達臨界值。與圖 5.1 的單一同位檢查碼比較，在 0⁺ 次疊代時已充分呈現使用使用迴旋碼的優越性。在 1 次疊代時，解碼效能在通道情況較差時比在通道情況較好時提升更多。不同於圖 5.1 的是，在 2 次疊代時使用迴旋碼的解碼性能仍未趨於飽和。

5.3 記憶性通道之疊代解碼實驗

5.3.1 系統模擬之步驟說明

系統架構基本上與 5.2 節相同，除了通道環境由白色高斯雜訊通道改為 Gilbert 所提出的記憶性通道。在每次模擬過程中，產生 30000 點的一階自迴歸訊號源，送進純量 Lloyd-Max 量化器。量化索引使用三個位元，位元映射方法選用折疊二位元編碼方式。訊息位元以長度 L 的區塊為單位由交錯器進行插置，再以這 L 個訊息位元外加 2 個迫使狀態器狀態回歸至零的尾巴位元為一區塊長度送進迴旋編碼器中。將輸出位元組經由二位元相位鍵移調變後，加入由 Gilbert 通道模擬產生的位元錯誤序列作干擾後送至接收端。通道解碼時必須以一個區塊長度進行，此處每一區塊解出 L 個訊息位元和 2 個尾巴位元的額外訊息，再將多餘的尾巴位元之額外訊息捨棄。我們將考慮 7 種不同的系統設定，進行下列實驗。[1] 不考慮通道記憶性，且加入區塊通道交錯器($L=150$)，其結果在表 5.1 及圖 5.3。[2] 考慮通道記憶性，且加入區塊通道交錯器($L=150$)，其結果在表 5.2 及圖 5.4。[3] 不考慮通道記憶性，且不加入區塊通道交錯器，其結果在表 5.3 及圖 5.5。[4] 考慮通道記憶性，且不加入區塊通道交錯器，其結果在表 5.4 及圖 5.6。[5] 考慮通道記憶性，且加入隨機通道交錯器，其結果在表 5.5 及圖 5.7($L=150$)。[6] 考慮通道記憶性，且加入隨機

通道交錯器(L=1500)，其結果在表 5.6 及圖 5.8。[7]考慮通道記憶性，且加入區塊通道交錯器(L=1500)，其結果在表 5.7 及圖 5.9。

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.34023	4.23029	6.143728	5.959703	6.16104
0.063096	4.84323	5.611873	7.484467	7.31591	7.520231
0.039811	6.02631	7.063601	8.832808	8.683859	8.865108
0.025119	8.12423	8.622126	10.11842	10.00306	10.15852
0.015849	9.18635	9.886368	11.16159	11.07565	11.20516
0.01	10.52125	10.74873	11.76554	11.7142	11.82314
0.00631	11.67632	11.82908	12.65563	12.62485	12.7079
0.003981	12.36125	12.6101	13.15181	13.13428	13.19284
0.002512	12.28785	12.57913	13.07637	13.06075	13.1198
0.001583	13.32312	13.4639	13.78895	13.7804	13.81463
0.001	13.31454	13.49302	13.78145	13.7731	13.80603

表 5.1 使用區塊交錯器(L=150)但不考慮通道記憶性的結果

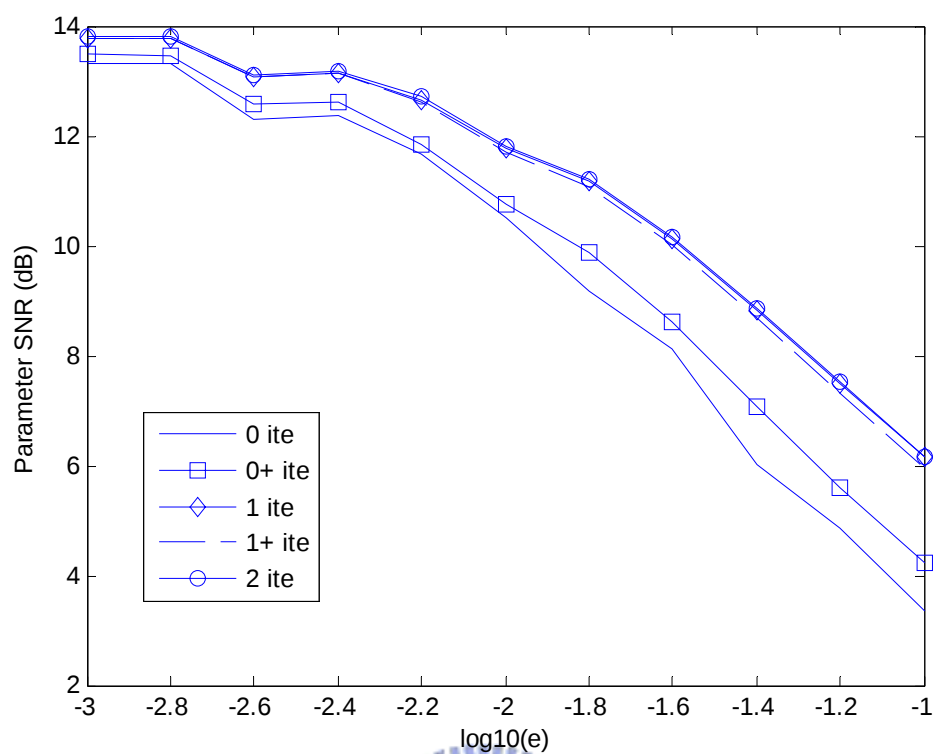


圖 5.3 使用區塊交錯器(L=150)但不考慮通道記憶性之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.34023	4.563882	6.597199	6.490111	6.687642
0.063096	4.84323	5.937123	7.873549	7.787598	7.998222
0.039811	6.02631	7.362225	9.28644	9.2274	9.455355
0.025119	8.12423	8.841343	10.42129	10.35649	10.54187
0.015849	9.18635	10.09377	11.29868	11.25267	11.41782
0.01	10.52125	10.83117	12.26692	12.19603	12.31715
0.00631	11.67632	11.96523	12.76623	12.74993	12.86017
0.003981	12.36125	12.689	13.35156	13.355	13.42404
0.002512	12.28785	12.65687	13.20274	13.19238	13.26973
0.001583	13.32312	13.49719	13.78875	13.78042	13.82875
0.001	13.31454	13.57268	13.84907	13.85148	13.88262

表 5.2 利用通道記憶性與區塊交錯器(L=150)之結果

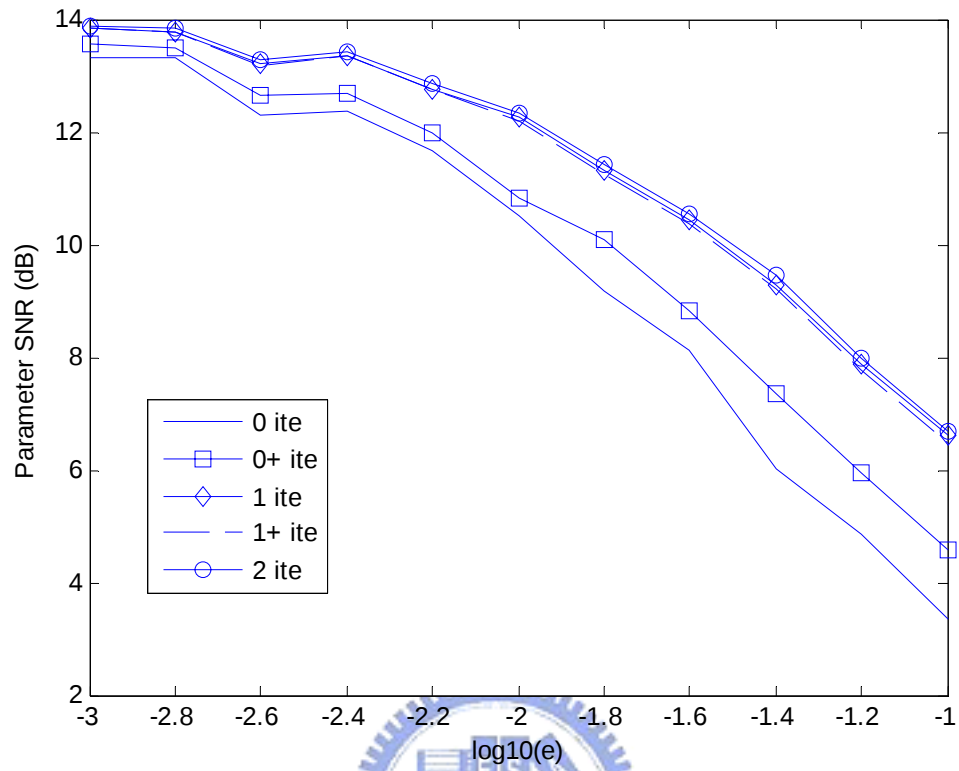


圖 5.4 利用通道記憶性與區塊交錯器(L=150)之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.34023	4.422733	5.018131	4.8735	5.060688
0.063096	4.84323	5.743457	6.446449	6.321988	6.52632
0.039811	6.02631	7.187569	7.873109	7.766199	7.965687
0.025119	8.12423	8.677406	9.318223	9.241679	9.424888
0.015849	9.18635	9.966647	10.47313	10.41191	10.56844
0.01	10.52125	10.79247	11.24199	11.20388	11.3461
0.00631	11.67632	11.89157	12.26967	12.24299	12.35093
0.003981	12.36125	12.63962	12.8638	12.85151	12.93588
0.002512	12.28785	12.61605	12.83553	12.82287	12.90498
0.001583	13.32312	13.48987	13.6405	13.64147	13.69039
0.001	13.31454	13.5030	13.59245	13.58832	13.63644

表 5.3 不考慮通道記憶性與交錯器之結果

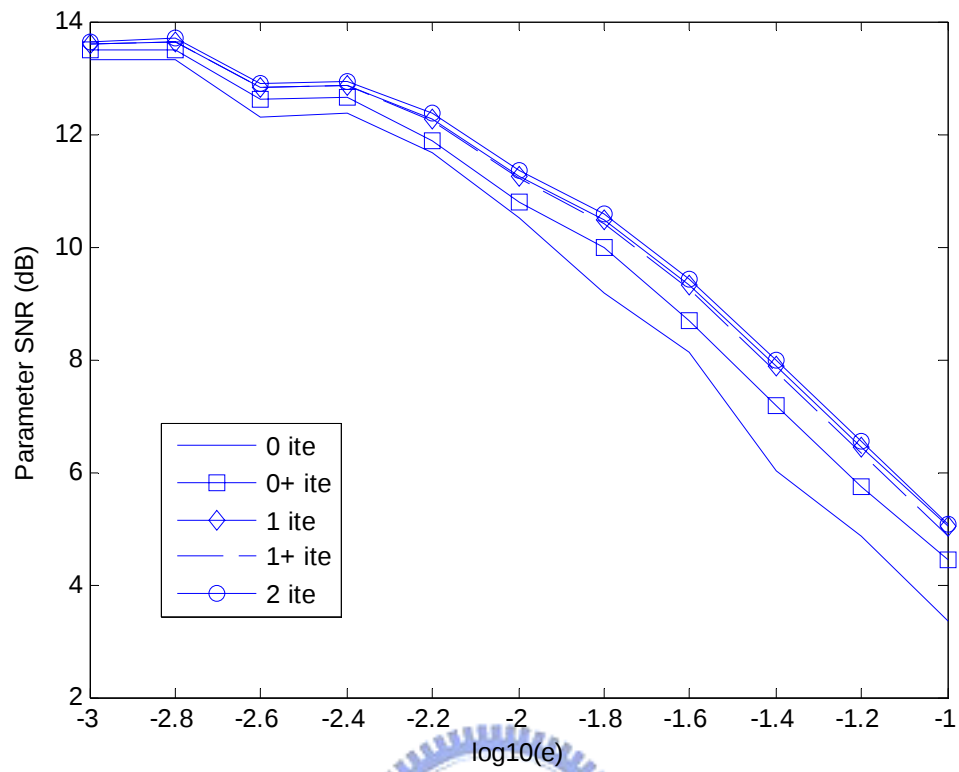


圖 5.5 不考慮通道記憶性與交錯器之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.34023	4.832634	5.747345	5.624446	5.92664
0.063096	4.84323	6.174634	6.992958	6.888797	7.200441
0.039811	6.02631	7.572874	8.30556	8.213173	8.503516
0.025119	8.12423	8.966067	9.616886	9.560227	9.830093
0.015849	9.18635	10.17753	10.8903	10.83709	11.04714
0.01	10.52125	11.0033	11.43019	11.38739	11.58159
0.00631	11.67632	12.08766	12.48347	12.46144	12.60089
0.003981	12.36125	12.75896	13.09319	13.07943	13.17804
0.002512	12.28785	12.67602	12.90435	12.88489	12.98704
0.001583	13.32312	13.52189	13.75813	13.74162	13.79522
0.001	13.31454	13.56514	13.68421	13.67639	13.73238

表 5.4 考慮通道記憶性但不使用交錯器之結果

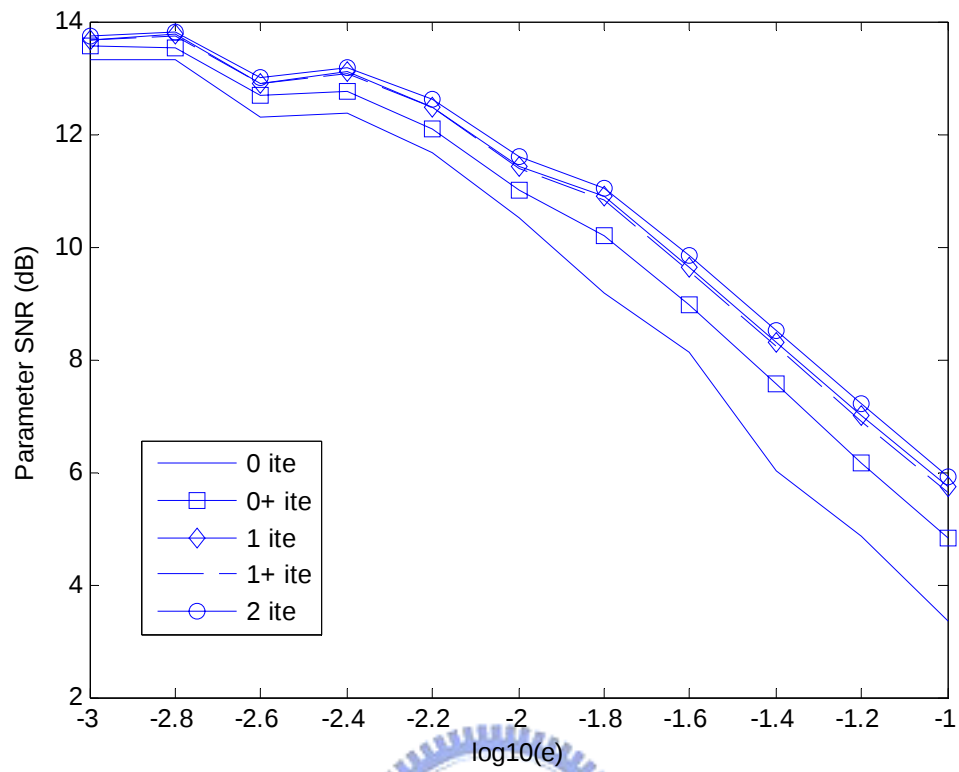


圖 5.6 考慮通道記憶性但不使用交錯器之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.280074	3.891862	6.24413	6.155378	6.303485
0.063096	4.832488	5.355052	7.711722	7.636717	7.788399
0.039811	6.017823	6.828529	9.066071	9.00567	9.154419
0.025119	7.802279	8.379553	10.33891	10.29242	10.42277
0.015849	9.114951	9.677013	11.35532	11.3229	11.42783
0.01	10.23626	10.54127	11.96576	11.93805	12.02958
0.00631	11.38193	11.69281	12.81988	12.80223	12.86669
0.003981	12.31271	12.45916	13.30007	13.28999	13.33776
0.002512	12.39533	12.45119	13.24707	13.23475	13.28245
0.001583	13.31029	13.38901	13.84704	13.8444	13.87077
0.001	13.39782	13.38225	13.82788	13.82329	13.85153

表 5.5 考慮通道記憶性與隨機交錯器(L=150)之結果

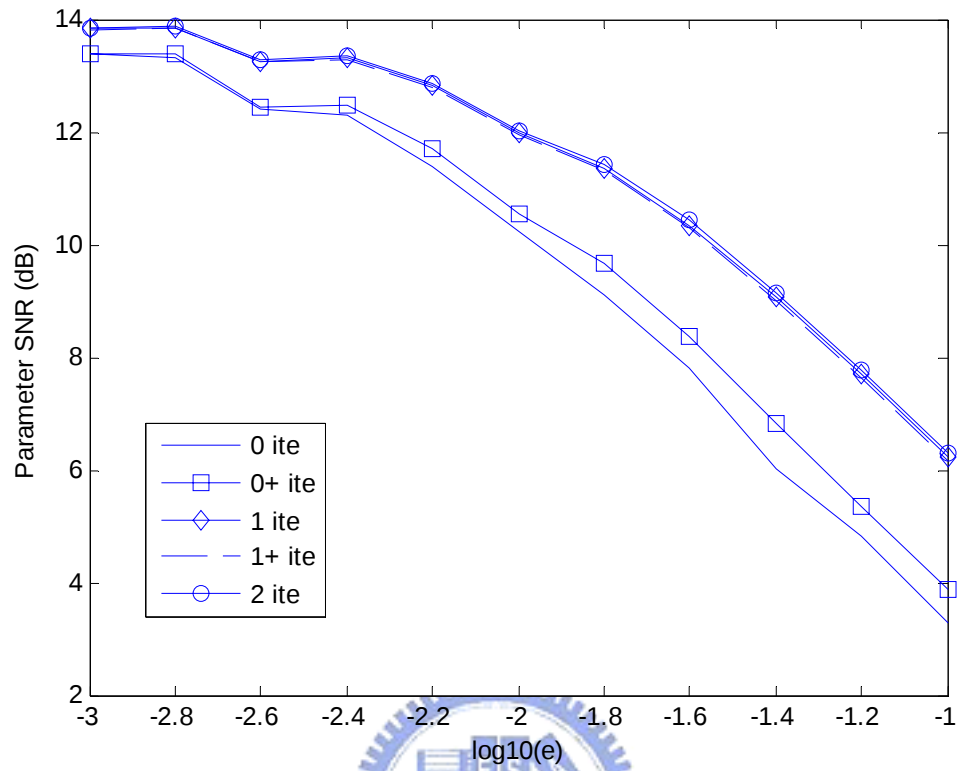


圖 5.7 考慮通道記憶性與隨機交錯器(L=150)之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.159123	3.185125	7.730688	7.732703	8.15125
0.063096	4.457251	4.96989	9.883947	10.01819	10.40223
0.039811	6.227184	6.768756	11.44457	11.65967	11.94138
0.025119	7.702000	8.50987	12.58284	12.81698	12.98913
0.015849	8.955111	9.946305	13.23271	13.43751	13.54499
0.01	10.182538	10.82829	13.55661	13.73288	13.80923
0.00631	11.08539	11.98738	13.94725	14.07501	14.12082
0.003981	11.929747	12.69694	14.15053	14.24724	14.27765
0.002512	12.038657	12.66729	14.11425	14.21155	14.23954
0.001583	13.389939	13.5398	13.34497	14.40108	14.41557
0.001	13.331025	13.54852	13.35336	14.40834	14.42289

表 5.6 考慮通道記憶性與隨機交錯器(L=1500)之結果

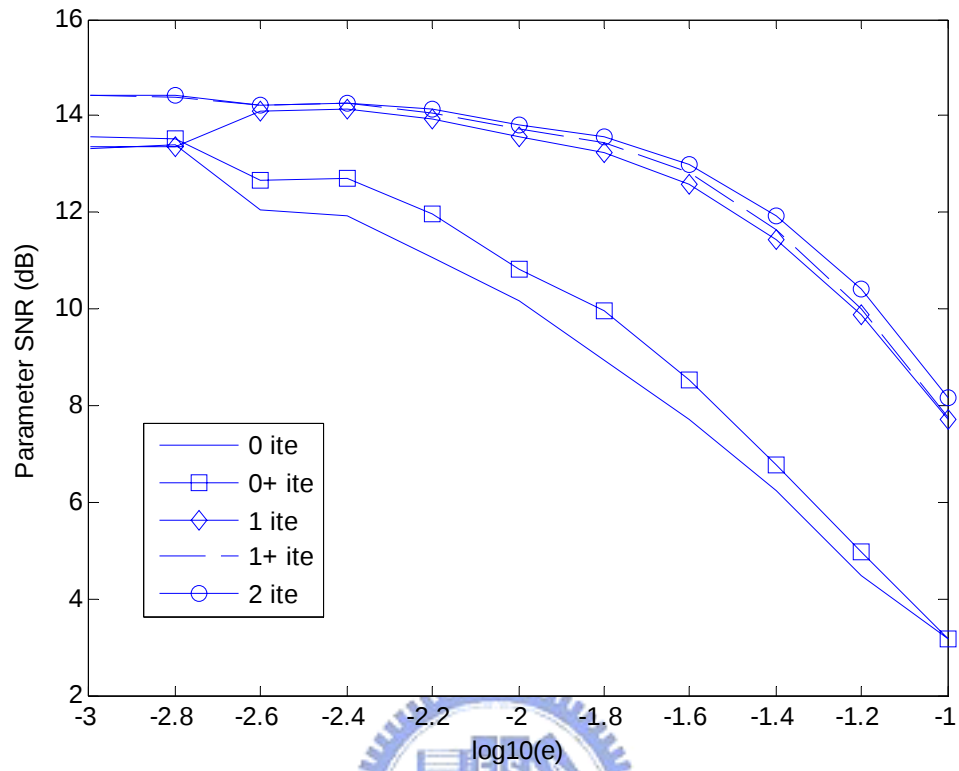


圖 5.8 考慮通道記憶性與隨機交錯器(L=1500)之結果

BER	0 iteration	0 ⁺ iteration	1 iteration	1 ⁺ iteration	2 iteration
0.1	3.521466	4.58132	7.26824	7.067587	7.293906
0.063096	4.86717	5.940222	8.449791	8.279513	8.517638
0.039811	6.38978	7.354542	9.621264	9.481262	9.69871
0.025119	7.89357	8.803742	10.74055	10.62947	10.81735
0.015849	9.40066	10.0296	11.66337	11.57915	11.72525
0.01	10.04822	10.84587	12.18004	12.12095	12.24689
0.00631	11.59933	11.93402	12.90185	12.86244	12.95174
0.003981	12.40761	12.63639	13.34128	13.3115	13.37345
0.002512	12.06808	12.60479	13.29534	13.27826	13.33752
0.001583	13.41113	13.47779	13.87968	13.86768	13.89834
0.001	13.18273	13.47698	13.88169	13.87418	13.90477

表 5.7 考慮通道記憶性與區塊交錯器(L=1500)之結果

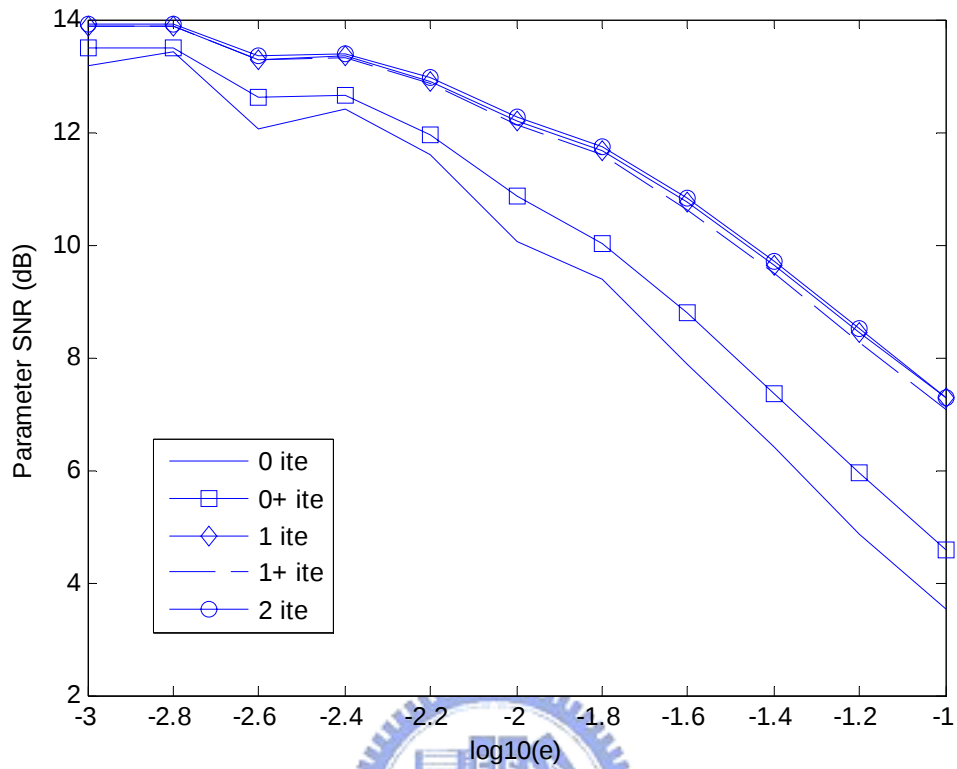


圖 5.9 考慮通道記憶性與區塊交錯器(L=1500)之結果

5.3.2 結果分析

圖 5.3 至圖 5.9 之系統評估，皆使用(5.9)式的 SNR 值作為衡量。首先比較圖 5.3 與圖 5.4，應用通道記憶特性於疊代解碼過程可有效提升系統效能。以 2 次疊代為例，通道錯誤率為 0.1 時，有 0.5dB 的差距，但在通道情形好時則較不明顯。從圖 5.5 與圖 5.6 可得知，在沒有通道交錯器時，考慮通道記憶特性的好處更加明顯。以 2 次疊代為例，當通道錯誤率為 0.1 時，有 0.9dB 的差距，大於使用交錯器時

的 0.5dB 差距。此現象證明了使用通道交錯器可以淡化 Gilbert 通道的記憶特性，所以差距變小。從圖 5.4 與圖 5.9 可得知當區塊交錯器大小變大時系統效能也跟著提升，其原因在於提供了更準確的解碼額外訊息。同理從圖 5.7 與圖 5.8 可得知，在 1 次疊代以後隨機交錯器大小變大可提升系統效能，且比區塊交錯器提升的更多，但在 0 次與 0^+ 次疊代時則沒有提升。比較圖 5.8 與圖 5.9，區塊交錯器在 0 次與 0^+ 次疊代時效能較好，但從 1 次疊代開始時則是隨機交錯器效能較好。最後比較圖 5.4、5.6 及 5.9，我們發現交錯器的區塊長度由 150 增加到 1500 時，Gilbert 通道記憶特性隨之淡化，因此通道記憶用於解碼所得的增益隨之下降。以 2 次疊代為例，當通道錯誤率為 0.1 時，由 0.76dB 減少至 0.61dB。

第六章 結論與未來展望

本論文主要探討利用疊代解碼機制，加強系統強健性以對抗通道雜訊干擾。我們分別針對單一同位檢查碼與迴旋碼，結合軟性位元訊源解碼進行疊代演算。其關鍵在利用渦輪碼原理，在通道解碼器與訊源解碼器之間交叉運用解碼輸出的額外訊息。藉著額外訊息因疊代次數增加而提升的準確性，系統效能也隨之提升。在 AWGN 通道且環境較差時，基於單一同位檢查碼的解碼器在 1⁺ 次疊代時有 0.4dB 的增益，而基於迴旋碼的解碼器在 2 次疊代時效能還有提升。在通道環境有叢發性位元錯誤時，我們根據 Gilbert 通道模型修改解碼演算法以運用通道記憶特性。由迴旋碼實驗發現，通道匹配疊代解碼機制的系統效能最好。當加入通道交錯器時，區塊長度越大效能也跟著變好，其原因為額外訊息更加準確。

在此論文中，當通道有記憶特性時，在通道解碼過程我們僅考慮音框內的記憶特性，但在訊源解碼時則沒有考慮。因此在未來研究，若能在通道解碼與訊源解碼同時考慮音框間的記憶特性，並據以修改本文中所提之疊代解碼演算法，應能進一步改善系統的強健性能。

參考文獻

- [1] C.E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol.27, pp. 379-423, pp. 623-656, 1948.
- [2] 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.
- [3] M. Adrat, "Iterative Source-Channel Decoding for Digital Mobile Communications," *PhD thesis*, Aachen University of Technology, July. 2003
- [4] T. Fingscheidt and P. Vary, "Softbit speech Decoding: A New Approach to Error concealment," *IEEE Trans Speech Audio Processing*, vol. 9, No. 3. Mar. 2001.
- [5] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, no.2, Mar. 1996.
- [6] 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. 20, pp. 284-287, Mar. 1974.
- [7] C.L. Lee, W.W. Chang, "Memory-Enhanced MMSE-Based Channel Error Mitigation for Distributed Speech Recognition," *Interspeech 2005-Eurospeech*, Lisbon, Portugal, Sep 2005.
- [8] E.N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1265, Sept. 1960.
- [9] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform Theory*, vol. IT-6, pp. 7-12, Mar 1960.

附錄 A 對數相似比率代數

假設變數 x 定義在有兩個元素 $\{1, -1\}$ 的 $GF(2)$ 之下，則其對數相似比率定義為

$$L(x) = \log \frac{p(x=1)}{p(x=-1)} \quad (A.1)$$

變數 x_1, x_2 也同樣定義在有兩個元素 $\{1, -1\}$ 的 $GF(2)$ 之下，且互相獨立使用關係式

$$\begin{aligned} p(x_1 \oplus x_2 = 1) &= p(x_1 = 1) \cdot p(x_2 = 1) \\ &+ p(x_1 = -1) \cdot p(x_2 = -1) \end{aligned} \quad (A.2)$$

則 $L(x_1 \oplus x_2)$ 可表示為

$$L(x_1 \oplus x_2) = \log \frac{1 + \exp\{L(x_1)\} \exp\{L(x_2)\}}{\exp\{L(x_1)\} + \exp\{L(x_2)\}} \quad (A.3)$$

進一步由數學歸納法可證

$$L\left(\sum_{j=1}^J \oplus x_j\right) = \log \frac{\prod_{j=1}^J (\exp\{L(x_j)\} + 1) + \prod_{j=1}^J (\exp\{L(x_j)\} - 1)}{\prod_{j=1}^J (\exp\{L(x_j)\} + 1) - \prod_{j=1}^J (\exp\{L(x_j)\} - 1)}$$

使用關係式 $\tanh(x/2) = (\exp\{x\} - 1)/(\exp\{x\} + 1)$ 則

$$L\left(\sum_{j=1}^J \oplus x_j\right) = \log \frac{1 + \prod_{j=1}^J \tanh\{L(x_j)/2\}}{1 - \prod_{j=1}^J \tanh\{L(x_j)/2\}}$$

附錄 B 對數相似比率應用

假設變數 x 定義在有兩個元素 $\{1, -1\}$ 的 $\text{GF}(2)$ 之下，則 x 的對數相似比率表示為

$$L(x) = \log \frac{p(x=1)}{p(x=-1)} \quad (\text{B. 1})$$

由 (B. 1) 式我們可以推導出

$$p(x=1) = \frac{\exp\{L(x)\}}{1 + \exp\{L(x)\}} \quad (\text{B. 2})$$

$$p(x=-1) = \frac{1}{1 + \exp\{L(x)\}} \quad (\text{B. 3})$$

後驗機率 $p(x|z)$ 以對數相似比率表示為

$$L(x|z) = \log \frac{p(x=1|z)}{p(x=-1|z)} \quad (\text{B. 4})$$

由 (B. 4) 式我們可以推導出

$$p(x=1|z) = \frac{\exp\{L(x|z)\}}{1 + \exp\{L(x|z)\}} \quad (\text{B. 5})$$

$$p(x=-1|z) = \frac{1}{1 + \exp\{L(x|z)\}} \quad (\text{B. 6})$$

藉由 (B. 2) 和 (B. 5) 式 $p(z|x=1)$ 可推導成

$$\begin{aligned}
p(z|x=1) &= \frac{p(x=1|z) \cdot p(z)}{p(x=1)} \\
&= p(z) \cdot \frac{1 + \exp\{L(x)\}}{1 + \exp\{L(x|z)\}} \cdot \frac{\exp\{L(x|z)\}}{\exp\{L(x)\}} \\
&= p(z) \cdot \frac{(1 + \exp\{L(x)\}) \cdot \exp\left\{\frac{1}{2} \cdot L(z|x)\right\}}{1 + \exp\{L(x|z)\}} \cdot \exp\left\{\frac{1}{2} \cdot L(z|x)\right\} \\
&= \Psi(x, z) \cdot \exp\left\{\frac{1}{2} \cdot L(z|x)\right\}
\end{aligned} \tag{B.7}$$

藉由(B. 3)和(B. 6)式 $p(z|x=-1)$ 可推導成

$$\begin{aligned}
p(z|x=-1) &= \frac{p(x=-1|z) \cdot p(z)}{p(x=-1)} \\
&= p(z) \cdot \frac{1}{1 + \exp\{L(x|z)\}} \cdot \frac{1 + \exp\{L(x)\}}{1} \\
&= p(z) \cdot \frac{(1 + \exp\{L(x)\}) \cdot \exp\left\{\frac{1}{2} \cdot L(z|x)\right\}}{1 + \exp\{L(x|z)\}} \cdot \exp\left\{-\frac{1}{2} \cdot L(z|x)\right\} \\
&= \Psi(x, z) \cdot \exp\left\{-\frac{1}{2} \cdot L(z|x)\right\}
\end{aligned} \tag{B.8}$$

觀察(B. 7)和(B. 8)式可得知，對於 x 的選取決定了指數函數的正負符號，且有相同的倍數 $\Psi(x, z)$ 。所以兩式可結合為

$$p(z|x) = \Psi(x, z) \cdot \exp\left\{\frac{x}{2} \cdot L(z|x)\right\} \tag{B. 9}$$