

國立交通大學

電信工程學系

碩士論文



寬頻語音的多重敘述編碼系統  
A Study of Multiple Description Coding for  
Broadband Speech

研究生：顏廣儀

指導教授：張文輝博士

中華民國九十七年六月

寬頻語音的多重敘述編碼系統  
**A Study of Multiple Description Coding for  
Broadband Speech**

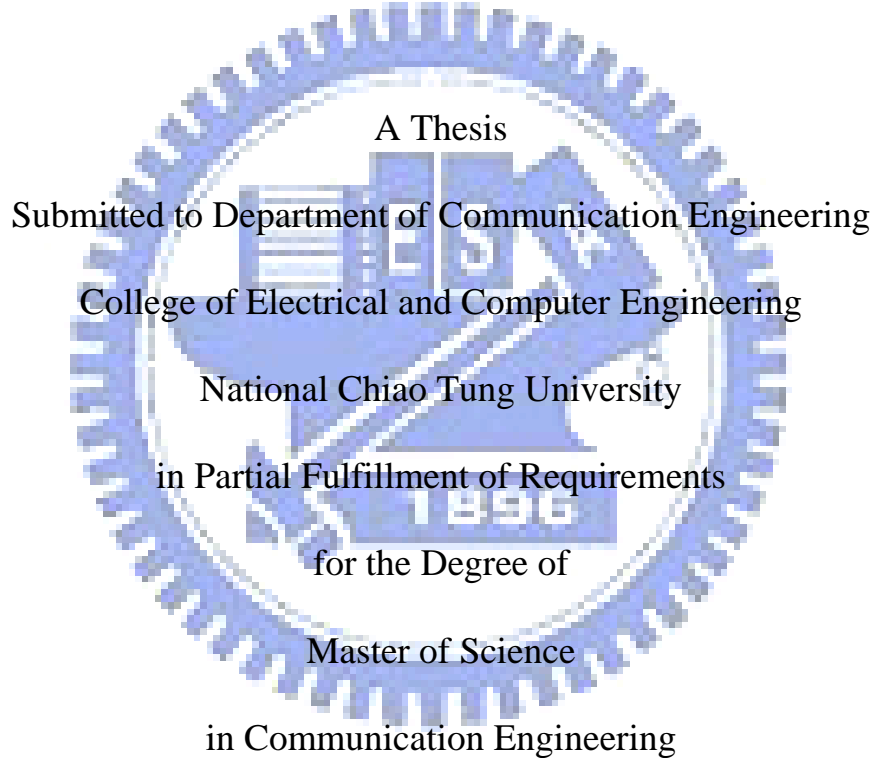
研究生：顏廣儀  
指導教授：張文輝

Student：Kuang-Yi Yen  
Advisor：Wen-Whei Chang

國立交通大學

電信工程學系

碩士論文



June 2008

Hsinchu, Taiwan, Republic of China

中華民國 九十七 年 六 月

# 寬頻語音的多重敘述編碼系統

學生：顏廣儀

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

在數位通訊系統中，多重敘述編碼是一種有效提升冗息保護資訊的機制，我們在該架構下提出基於位元與索引層級的疊代訊源通道解碼演算法。首先探討訊源量化後的殘餘冗息與通道編碼資訊的交叉運用，系統設計藉由納入多重敘述間的相關資訊加強其解碼的可靠性，其效能隨著疊代解碼次數的增加而遞增。為了克服位元間獨立假設的限制，我們更進一步將解碼演算單位由位元改良為索引層級。於實驗模擬中使用自迴歸訊號源與寬頻語音，成功驗證不同敘述間的相關訊息可以有效提高解碼效能。

# **A Study of Multiple Description Coding for Broadband Speech**

Student:Kuang-Yi Yen      Advisor:Dr. Wen-Whei Chang

Department of Communication Engineering

National Chiao Tung University

The logo of National Chiao Tung University is a circular emblem with a gear-like outer border. Inside the circle, there are stylized representations of books and a building. The letters 'E', 'S', and 'A' are prominently displayed in the center, with the year '1956' at the bottom. The word 'Abstract' is overlaid on the logo.

## **Abstract**

For digital communications, multiple description coding (MDC) is an efficient technique to enhance the channel robustness. Based on the MDC structure, we develop iterative source-channel decoding (ISCD) techniques at bit-level and index-level. Our work began with the exploitation of the residual source redundancy and the parity check information of channel coding. By taking into account the correlation among all descriptions, the ISCD schemes have higher decoding reliabilities and the performance gain increases with increasing iterations. Moreover, in order to solve the mismatch between the bit-independence assumption and real environment, we propose an index-based decoding algorithm. Experimental results on AR-source and AMR-WB coded speech indicated that the correlation among multiple descriptions yields significant performance gain.

## 致謝

本篇論文的完成，首先要由衷地感謝我的指導教授張文輝老師，由於老師的細心教導與耐心指正，讓我瞭解到做研究時所應該持有的態度與精神，讓我受益良多。另外也要感謝實驗室的學長，同學以及學弟，不管是在課業或者是在研究上都協助我解決許多問題。同時也要感謝陪伴我的朋友們，尤其在我準備研究所考試以及就讀研究所其間一路上給我鼓勵與加油。最後，僅將此論獻給我的父母以及家人。



# 目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
圖目錄.....	viii
表目錄.....	x
<b>第一章 緒論.....</b>	<b>1</b>
1.1 研究動機與方向.....	1
1.2 章節概要.....	2
<b>第二章 寬頻語音編碼系統.....</b>	<b>3</b>
2.1 多重敘述純量量化器.....	4
2.2 位元與索引層級編碼機制.....	6
2.3 多重敘述編碼在寬頻語音的應用.....	9
2.3.1 AMR-WB 系統簡介.....	10
2.3.2 多重敘述編碼機制的應用.....	13

<b>第三章</b>	<b>位元層級的多重敘述疊代解碼機制</b>	<b>15</b>
3.1	系統解碼流程	15
3.1.1	系統效能評估	15
3.1.2	平行式疊代解碼架構	16
3.1.3	交叉式疊代解碼架構	18
3.2	通道解碼額外訊息的推導	20
3.3	訊源解碼額外訊息的推導	23
3.3.1	平行式架構的相關推導	23
3.3.2	交叉式架構的相關推導	27
<b>第四章</b>	<b>索引層級的多重敘述疊代解碼機制</b>	<b>33</b>
4.1	系統解碼流程	33
4.1.1	系統效能評估	34
4.1.2	平行式疊代解碼架構	34
4.1.3	合併式疊代解碼架構	37
4.2	通道解碼額外訊息的推導	39
4.3	訊源解碼額外訊息的推導	41
4.3.1	平行式架構的相關推導	41
4.3.2	合併式架構的相關推導	44

4.4	對數領域之索引層級合併式疊代演算法.....	47
4.4.1	通道解碼額外訊息的推導.....	49
4.4.2	訊源解碼額外訊息的推導.....	52
<b>第五章</b>	<b>實驗模擬與結果分析.....</b>	<b>55</b>
5.1	位元層級平行式與交叉式解碼之模擬.....	55
5.1.1	系統模擬之步驟說明.....	55
5.1.2	結果分析.....	57
5.1.3	對角線索引配置法之實驗模擬.....	58
5.2	索引層級平行式與交叉式解碼之模擬.....	62
5.2.1	系統模擬之步驟說明.....	62
5.2.2	結果分析.....	65
5.3	對數領域之合併式解碼模擬.....	66
5.3.1	系統模擬之步驟說明.....	66
5.3.2	結果分析.....	68
5.4	寬頻語音的多重敘述解碼實驗.....	68
5.4.1	系統模擬之步驟說明.....	68
5.4.2	結果分析.....	70



第六章 結論與未來展望..... 77

參考文獻..... 79



# 圖目錄

圖 2.1	MDSQ 方塊圖 .....	5
圖 2.2	索引配置矩陣 .....	5
圖 2.3	BSA 與主對角線索引配置矩陣( $Q=16, M=3$ ) .....	6
圖 2.4	位元層級多重敘述編碼器 .....	9
圖 2.5	索引層級多重敘述編碼器 .....	9
圖 2.6	AMR-WB 編碼流程方塊圖 .....	12
圖 2.7	AMR-WB 解碼流程方塊圖 .....	13
圖 2.8	多重敘述編碼系統應用方塊圖 .....	14
圖 3.1	平行式疊代解碼方塊圖 .....	18
圖 3.2	交叉式疊代解碼方塊圖 .....	20
圖 4.1	平行式疊代解碼方塊圖 .....	36
圖 4.2	合併式疊代解碼方塊圖 .....	38
圖 4.3	合併式疊代解碼方塊圖(Log-domain) .....	49
圖 5.1	位元層級的疊代解碼結果[( $Q, M$ )=(16, 3)且 BSA 配置] .....	60
圖 5.2	位元層級的疊代解碼結果[( $Q, M$ )=(32, 3)且 BSA 配置] .....	60
圖 5.3	位元層級的疊代解碼結果[( $Q, M$ )=(16, 3)且對角線配置]	

.....	61
圖 5.4 位元層級的疊代解碼結果 $[(Q, M)=(32, 3)]$ 且對角線配置]	61
.....	61
圖 5.5 索引層級的平行與合併疊代解碼結果 $[(Q, M)=(16, 3)]$	63
.....	63
圖 5.6 索引層級的平行與合併疊代解碼結果 $[(Q, M)=(32, 3)]$	64
.....	64
圖 5.7 位元與索引層級的疊代解碼結果 $[(Q, M)=(16, 3)]$ .....	64
圖 5.8 位元與索引層級的疊代解碼結果 $[(Q, M)=(32, 3)]$ .....	65
圖 5.9 機率與對數領域的索引層級合併解碼結果 $[(Q, M)=(16, 3)]$	67
.....	67
圖 5.10 機率與對數領域的索引層級合併解碼結果 $[(Q, M)=(32, 3)]$	68
.....	68
圖 5.11 位元層級疊代解碼器的平均頻譜失真結果 .....	75
圖 5.12 索引層級疊代解碼器的平均頻譜失真結果 .....	75
圖 5.13 各種疊代解碼器的平均頻譜失真結果 .....	76

# 表目錄

表 2.1	AMR-WB 九種模式的位元配置.....	12
表 2.2	ISF 係數向量量化處理.....	13
表 5.1	ISF 索引所配置的指定矩陣.....	71
表 5.2	各種疊代解碼機制的參數重建結果(iteration 1, -6dB AWGN) .....	72
表 5.3	各種疊代解碼機制的參數重建結果(iteration 1, -2dB AWGN) .....	72
表 5.4	各種疊代解碼機制的參數重建結果(iteration 1, 2dB AWGN) .....	73
表 5.5	各種疊代解碼機制的參數重建結果(iteration 3, -6dB AWGN) .....	73
表 5.6	各種疊代解碼機制的參數重建結果(iteration 3, -2dB AWGN) .....	74
表 5.7	各種疊代解碼機制的參數重建結果(iteration 3, 2dB AWGN) .....	74



# 第一章 緒論

## 1.1 研究動機與方向

多重敘述編碼是一種新的訊源編碼技術，主要是將訊號源透過某種編碼機制轉換成多個具有相關性的敘述，再經由彼此獨立的通道傳送。此種多敘述傳輸架構已廣泛應用於多媒體網路傳輸，如語音、音訊與影像等信號。相較於單通道傳輸的使用，多通道傳輸增加了更多的殘餘冗息可以有效地對抗通道雜訊的干擾。傳統的多重敘述編碼是基於封包漏失通道的模型，其信號的還原品質取決於成功接收的通道個數，當解碼端有較多的接收資訊將提升訊號還原的準確性。問題是封包漏失通道無法模擬真實的無線傳輸環境，在本論文中，我們將延伸多重敘述編碼的概念至高斯白雜訊的通道模型。其關鍵在於接收端使用可靠度(reliability)資訊的解碼方式來達到更好的解碼品質，也就是所謂的軟性輸入輸出解碼。再者，訊號源經多重敘述編碼之後產生的敘述之間並不是互相獨立，解碼端若能引入敘述間的相關性訊息將能有效提升系統效能。

在本論文中，我們的研究重點是多重敘述編碼系統的疊代訊源通道解碼演算法，其相關應用包括有多重敘述純量量化(MDSQ)[1]與多重敘述向量量化(MDVQ)[2]。在編碼端採用多重敘述量化器與迴旋碼

通道編碼器，解碼端即可引用類似渦輪碼疊代解碼[3]的概念達成軟性位元資訊的訊息交換。若進一步將多通道敘述間的相關性資訊納入設計考量，抑或運用訊源量化後相關性強的殘餘冗息，將可有效提升多重敘述傳輸系統的解碼效果。軟性輸入輸出解碼器的傳統設計是針對單一位元來作解碼，事前所作的位元獨立假設與實際環境不符而影響到解碼效果。因此，我們針對此問題提出了索引層級的多重敘述解碼架構。至於應用實務層面，寬頻語音編碼系統為近來最熱門的 3G 標準，我們將寬頻語音信號作為多重敘述編碼的訊號源，以此驗證我們提出架構的系統效能。

## 1.2 章節概要

第二章介紹多重敘述編碼架構及其在寬頻語音傳輸的應用。第三章推導兩種位元層級的多重敘述疊代解碼演算法。第四章介紹兩種索引層級的多重敘述疊代解碼架構。第五章為系統模擬與結果分析。第六章為結論與未來展望。

## 第二章 寬頻語音編碼系統

傳統的多媒體通訊都是設定在單一通道上做傳輸，問題是無線網路傳輸環境存在許多干擾因素，如封包延遲、漏失、多路徑傳輸延遲、白雜訊等。龐大的多媒體資料量作單一通道傳輸而達到良好的重建品質有其限制，因此前人提出了多通道敘述的傳輸系統。多通道傳輸的原理，在於將訊號源經由特定的編碼處理產生數種相關的敘述，然後分別在獨立的通道傳輸。而系統的整體效能與接收端成功接收的敘述個數有關，若全部敘述皆有收到將有最佳的解碼品質，若部分收到則產生次級的解碼品質。

多重敘述編碼是一種新的訊號源編碼模式，結合量化、位元對應與索引配置的流程。我們初步鎖定在雙通道敘述的傳輸系統，而索引的配置規則是透過索引指定矩陣的行列對應關係所產生。至於接收端的多重敘述解碼器，其輸入與輸出端皆是軟性資訊，與迴旋解碼搭配之後就能引用渦輪碼的原理做疊代解碼處理。此外，若進一步將雙通道資訊的相關性納入考量，預期能有效提升解碼系統效能。關於此解碼系統的實用價值，我們將結合新一代寬頻語音的編碼機制與多敘述傳輸系統，然後以實驗模擬驗證之。

## 2.1 多重敘述純量量化器

多重敘述純量量化器(multiple description scalar quantizer, MDSQ)的編碼端由兩個工作方塊所組成，如圖 2.1 所示。第一個方塊是純量量化器與位元對應，分別採用 Lloyd-Max 量化器與自然對應法(natural mapping)。第二個方塊則是索引配置，其流程需要參考圖 2.2 中的索引配置矩陣。觀察量化索引  $\mathbf{u}_i \in \{1, 2, \dots, Q\}$  在矩陣中的所在位置，該位置所對應的列索引  $i \in \{1, \dots, 2^M\}$  與行索引  $j \in \{1, \dots, 2^M\}$ ，即構成雙敘述編碼的輸出資訊。舉例而言，量化索引  $\mathbf{u}_i = 2$  在配置矩陣中位置於第五列第三行，其編碼輸出的敘述即為  $I=5$  與  $J=3$  兩索引值。另外，令  $R_i$  和  $C_j$  分別代表第  $i$  列和第  $j$  行所屬量化索引的集合，如  $R_1 = \{5, 7\}$  和  $C_2 = \{4, 9\}$ 。

至於索引配置的最佳化安排，如果要全面性搜尋  $\binom{2^{2M}}{Q} \cdot Q!$  種不同的配置組合，耗時且不可行。較有效率的有主對角線(diagonal)與二位元置換(Binary Switching Algorithm, BSA)兩種配置演算法。如圖 2.3 所示，主對角線配置法是將量化索引值沿著矩陣的主對角線由左上往右下方向排序。而 BSA 配置法則是一個次佳化的排列方法，計算量降低且有一定程度的系統效能。在文章[1]、[2]、[4]中，對於索引配置的議題有更深入的研究，我們在此則不多加以探討，只沿用其結果於編碼區塊裡。



Index Assignment

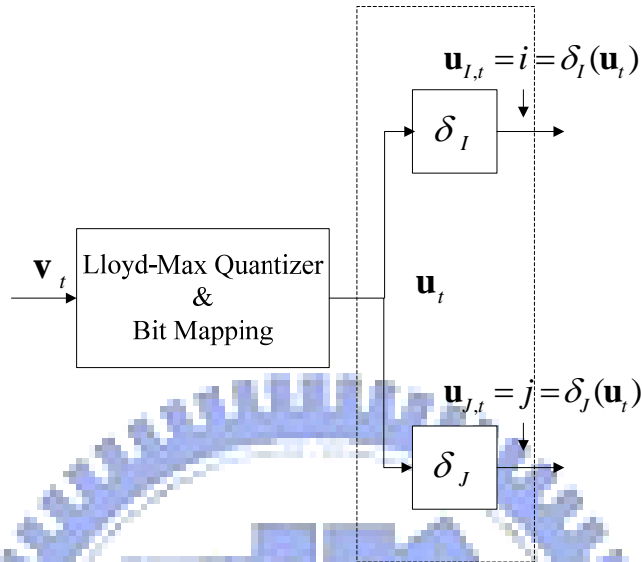


圖 2.1 MDSQ 方塊圖

$j = \delta_J(\mathbf{u}_t)$

	1	2	3	4	5	6	7	8
1	5							7
2		4	3					
3				6		8		
4					13		14	
5			2			10		
6				1	12			
7		9					15	
8	11							16

$i = \delta_I(\mathbf{u}_t)$

$R_1 = \{5, 7\}$

$C_2 = \{4, 9\}$

圖 2.2 索引配置矩陣

5							7
	4	3					
			6		8		
				13		14	
		2			10		
			1	12			
	9						15
11							16

BSA

1							
	4	3					
	2	7	5				
		6	10	9			
			8	13	11		
				12	14		
						15	
							16

主對角線

圖 2.3 BSA與主對角線索引配置矩陣(Q=16,M=3)

## 2.2 位元與索引層級編碼機制

為了要提高對抗通道雜訊的強健性，在多重敘述訊源編碼器之後引入迴旋碼通道編碼區塊。依其交錯器是以單一位元或單一索引為置換單位，會呈現兩種不同的編碼機制。首先定義位元層級編碼器的符號標記。在  $t$  時刻點的取樣實數值  $v_t$ ，經過 Lloyd-Max Quantizer 的純量量化與自然位元對應得到  $\log_2 Q$  位元長度的索引值  $\mathbf{u}_t$ ，通過索引配置函數  $\delta_i$  與  $\delta_j$  分別得到  $M$  位元長度的兩敘述  $\mathbf{u}_{i,t} = \delta_i(\mathbf{u}_t)$  與  $\mathbf{u}_{j,t} = \delta_j(\mathbf{u}_t)$ 。對  $I$  通道敘述的符號標記而言：

$$\mathbf{u}_{i,t} = \{u_{i,t}(1), \dots, u_{i,t}(m), \dots, u_{i,t}(M)\}, u_{i,t}(m) \in \{1, 0\}。$$

其組成序列  $\mathbf{U}_{i,1}^T = \{\mathbf{u}_{i,1}, \dots, \mathbf{u}_{i,T}\}$  經由位元交錯器  $\Phi$  的重新置換，產生新的位元序列定義為：

$$\mathbf{x}_{i,1}^L = \{x_{i,1}, \dots, x_{i,L}\}, L = MT$$

其中每個重新置換後的位元  $x_{l,t}$  對應到原來的特定位元  $u_{l,t}(m)$ ，

$$x_{l,t} = u_{l,t}(m), \quad l=1,2,\dots,L$$

位元序列  $\mathbf{x}_{l,1}^L$  被視為通道編碼器輸入端的資訊位元(information

bit)，我們使用暫存器長度為  $\nu_0$  的  $(n, 1)$  迴授式系統迴旋碼

(Recursive Systematic Convolutional code) 作通道編碼處理。每

個輸入位元  $x_{l,t}$  會產生一個系統位元  $y_{l,t}^s = x_{l,t}$  以及  $(n-1)$  個同位檢查

位元組成的向量  $\mathbf{y}_{l,t}^p$ ，因此位元序列  $\mathbf{x}_{l,1}^L$  對應的編碼輸出序列為

$$\mathbf{Y}_{l,1}^{\hat{L}} = \{\mathbf{y}_{l,1}, \dots, \mathbf{y}_{l,\hat{L}}\}$$

$$\mathbf{y}_{l,t} = \{y_{l,t}^s, \mathbf{y}_{l,t}^p\} = \{x_{l,t}, \mathbf{y}_{l,t}^p\}, \quad l=1,2,\dots,\hat{L}$$

編碼輸出序列的長度為  $\hat{L}$ ，包含  $L$  個資訊位元以及為了要清空暫存器

所需的終止位元(termination bit)。通道編碼輸出的碼字序列  $\mathbf{Y}_{l,1}^{\hat{L}}$ ，

經由二位元相位鍵移調變之後送進白高斯雜訊通道傳送。

接下來，我們介紹索引層級編碼流程的符號標記。在  $t$  時刻的取

樣點  $v_t$ ，經過 Lloyd-Max 純量量化器與位元對應得到量化索引值  $\mathbf{u}_t$ ，

再通過索引配置流程分別輸出兩敘述  $\mathbf{u}_{I,t}$  與  $\mathbf{u}_{J,t}$ 。對  $I$  通道的符號標

記而言：

$$\mathbf{U}_{l,1}^T = \{\mathbf{u}_{l,1}, \dots, \mathbf{u}_{l,T}\}$$

在量化索引序列  $\mathbf{U}_{l,1}^T$  之後串連一索引交錯器，其功用在於對序列中各

個索引作順序上的重新排列，但不變更同一索引所屬  $M$  個位元的次

序，以消除相鄰兩索引之間的殘餘冗息。有別於位元交錯器是以單位元為單位作置換，索引順序重排之後得到的索引序列  $X_{l,1}^T$  為

$$X_{l,1}^T = (\mathbf{x}_{l,1}, \dots, \mathbf{x}_{l,\tau}, \dots, \mathbf{x}_{l,T})$$

而重排後的個別索引  $\mathbf{x}_{l,\tau}$  則會對應到原先的特定索引  $\mathbf{u}_{l,t}$ ，

$$\mathbf{x}_{l,\tau} = \mathbf{u}_{l,t}, \quad \tau = 1, 2, \dots, T$$

$\mathbf{x}_{l,\tau}$  視為資訊索引 (information index) 經由  $(n, 1)$  且暫存器長度為  $U_0$  的迴授式系統迴旋編碼器，產生系統索引  $\mathbf{y}_{l,\tau}^s = \mathbf{x}_{l,\tau}$  與同位檢查索引  $\mathbf{y}_{l,\tau}^p$ ，

$$\mathbf{y}_{l,\tau} = (\mathbf{y}_{l,\tau}^s, \mathbf{y}_{l,\tau}^p) = (\mathbf{x}_{l,\tau}, \mathbf{y}_{l,\tau}^p), \quad \tau = 1, 2, \dots, \hat{T}$$

編碼輸出序列  $\mathbf{Y}_{l,1}^{\hat{T}} = (\mathbf{y}_{l,1}, \dots, \mathbf{y}_{l,\hat{T}})$  的時間長度為  $\hat{T}$ ，包含  $T$  個資訊索引以及清空暫存器所需之終止索引。通道編碼輸出  $\mathbf{Y}_{l,1}^{\hat{T}}$  經由二位元相位鍵移調變之後，再送進白高斯雜訊通道傳送。

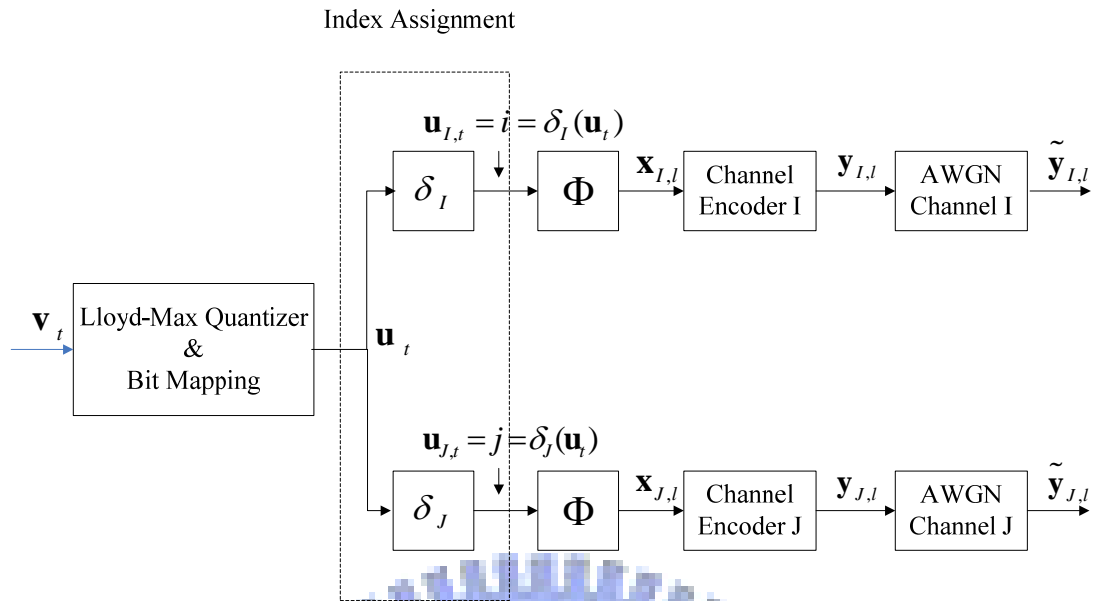


圖 2.4 位元層級多重敘述編碼器

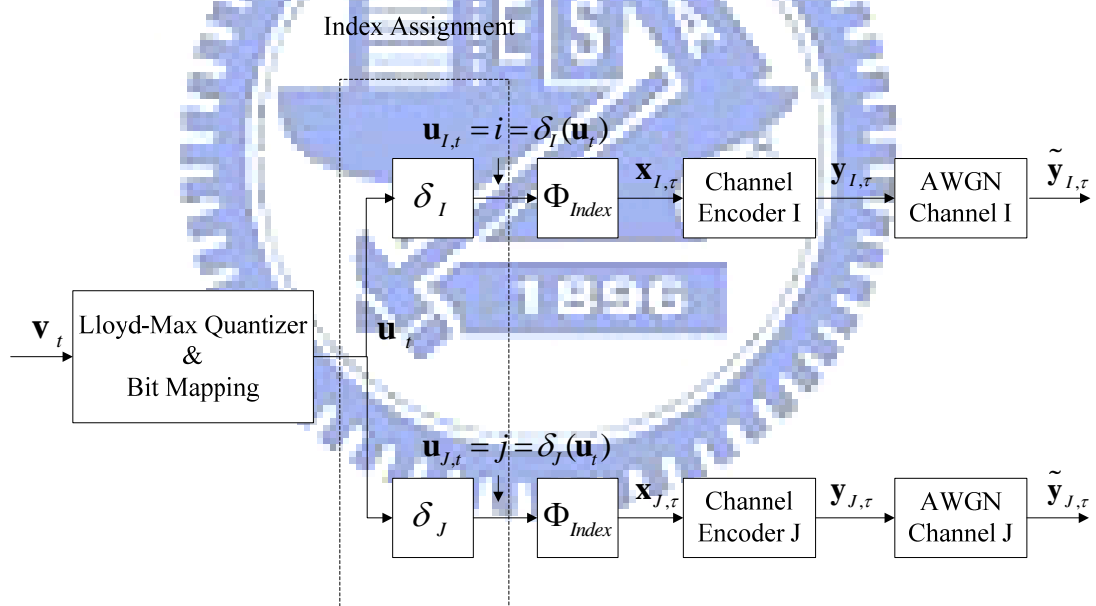


圖 2.5 索引層級多重敘述編碼器

### 2.3 多重敘述編碼在寬頻語音的應用

目前的語音編碼技術開發，大多是設定在窄頻帶(200~3400Hz)頻寬的環境，儘管技術一再改良仍有其瓶頸存在。為了進一步提升通

話品質而原音重現，就必須把語音低頻帶(50~200Hz)與高頻帶(3400~7000Hz)的資訊也一併傳送，這是由於語音的低頻與高頻成分可以增加臨場感與辨識率。但這也代表必須要提高取樣頻率而增加傳輸的資料量，考慮到現有的傳輸頻寬仍然受限制，因此必須要有更好的編碼技術才能有效率的壓縮寬頻語音並傳送之。而 ITU 與 3GPP 組織在前幾年就已朝著這個趨勢發展，也訂定出第三代寬頻語音編碼技術(Adaptive Multi-Rate Wideband, AMR-WB)。由於無線通訊產品的普及率已提高，將來勢必要求更高品質的通話服務，因此 AMR-WB 編碼技術會成為未來的主流是不容懷疑的。

### 2.3.1 AMR-WB 系統簡介

目前所使用的行動通訊服務，普遍是以 2 代的 GSM 與 2.5 代的 GPRS 為主，關鍵技術為碼激發線性預測(Code-Excited Linear Prediction, CELP)編碼演算法。藉由人類口說的語音分析處理以及模擬人聲的合成步驟，得到激發源參數與線性預估係數，以作為接收端重建語音訊號的參考指標，但輸入訊號的頻寬仍然限制在窄頻帶，並無法重現逼真的語音。不過新一代的語音編碼標準 AMR-WB，擴增語音訊號的頻寬，將低頻帶(50~200Hz)與高頻帶(3400~7000Hz)資訊納入考量，低頻帶語音資訊可以加強聲音的臨場感與真實性，高頻帶

資訊則會提升語意的辨識度與語音清晰度，進一步減少合成訊號與原音之間的失真。再者，AMR-WB 具有可調控性的語音編碼速率，從 6.6 到 23.85kbits/sec 共有九種編碼速率可以選擇，可因應通訊環境的不同而作出編碼模式上的調整。

AMR-WB 編碼標準中主要的語音分析合成方法，是採用代數碼激發線性預測演算法(Algebraic Code-Excited Linear Prediction, ACELP)。輸入訊號的取樣頻率為 12.8kHz，以一個音框時間長度 20 毫秒為處理單位，可以分析頻寬達 6.4kHz 的語音訊號。編碼輸出端會產生線性預測係數(LPC)、適應性碼本索引(ACB)、固定性碼本索引(FBC)、與增益(Gain)等分析參數。在 23.85kbits/sec 的編碼模式下，另外提供高頻帶(6.4k~7kHz)能量增益的數據，使其在接收端可重建 50~7000Hz 的寬頻語音。表 2.1 所示為不同編碼模式的位元配置，而系統編碼器與解碼器的方塊圖呈現在圖 2.6 與圖 2.7，其詳細的處理流程可以參考[5]。

經由 AMR-WB 的語音分析合成過程，最重要的一組參數就是導納頻譜頻率(Immittance Spectrum Frequency, ISF)，此組參數由 16 個線性預測係數轉化而來，對於語音重建品質扮演最關鍵的角色。再者，為了要在無線通道環境中傳輸必須作有效率的壓縮動作，而 ISF 的壓縮過程是將 16 個係數進行分次多階向量量化(Split-multistage

Vector Quantization, S-MSVQ), 最後得出七組向量量化索引, 我們

從表 2.2 可看出每組碼向量的維度與位元配置方式。

parameter	Codec Mode(kbps)								
	6.60	8.85	12.65	14.25	15.85	18.25	19.85	23.05	23.85
VAD	1	1	1	1	1	1	1	1	1
LTP flag	0	0	4	4	4	4	4	4	4
ISF	36	46	46	46	46	46	46	46	46
ACB	23	26	30	30	30	30	30	30	30
FCB	48	80	144	176	208	256	288	352	352
GAIN	24	24	28	28	28	28	28	28	28
High-band Gain	0	0	0	0	0	0	0	0	16
Total	132	177	253	285	317	365	397	461	477

表 2.1 AMR-WB 九種模式的位元配置

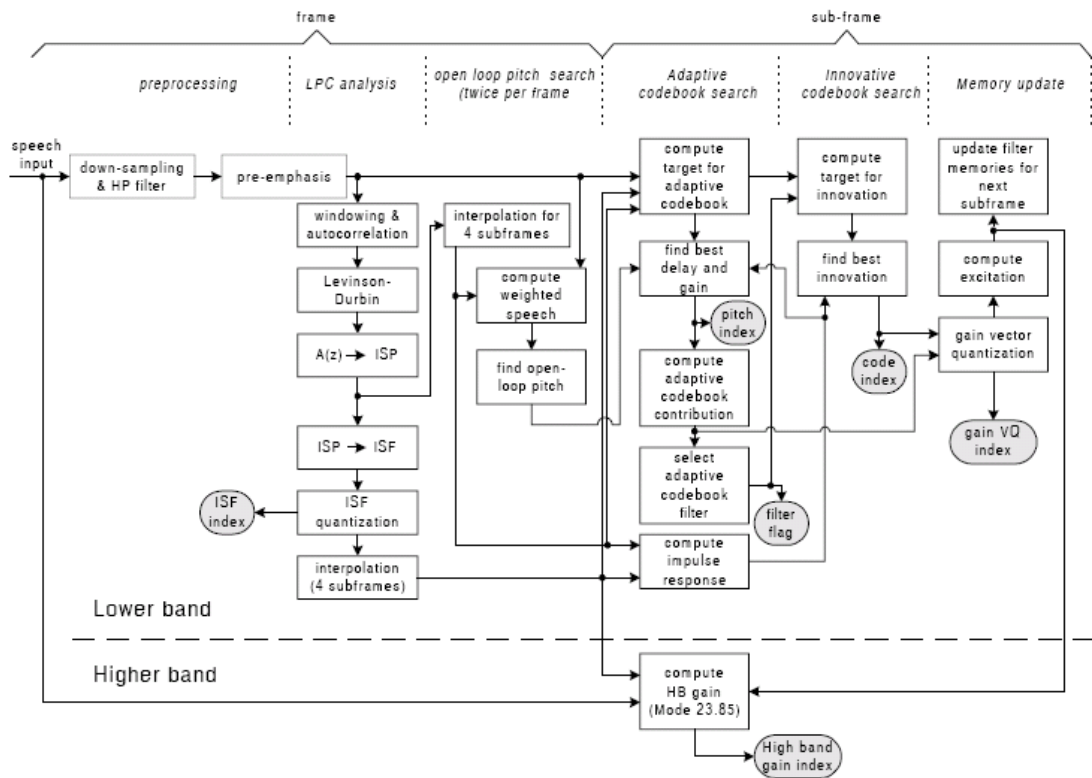


圖 2.6 AMR-WB 編碼流程方塊圖



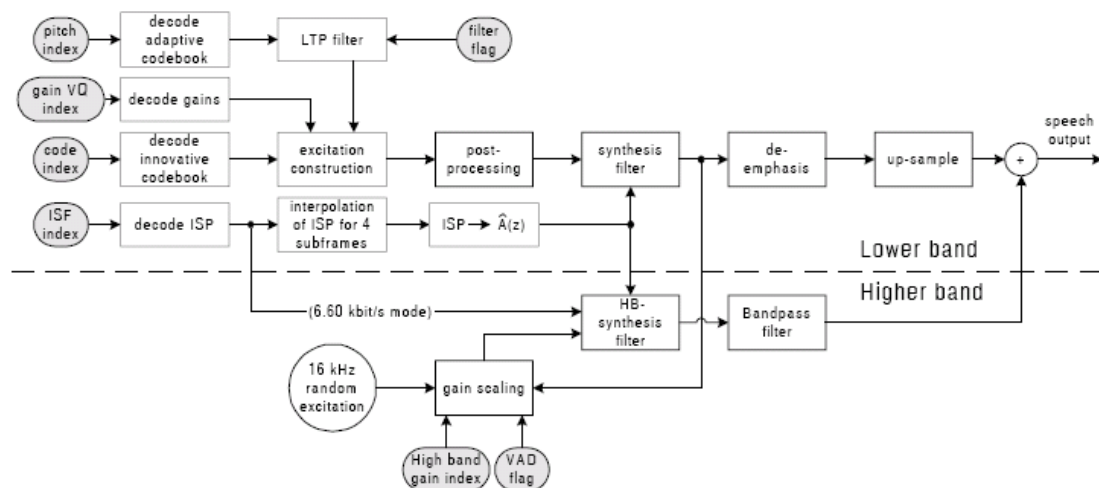


圖 2.7 AMR-WB 解碼流程方塊圖

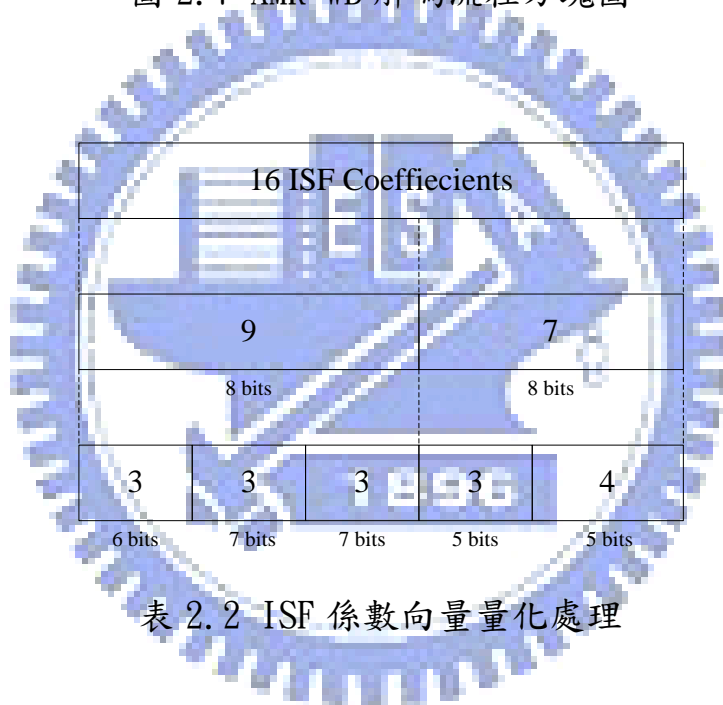


表 2.2 ISF 係數向量量化處理

### 2.3.2 多重敘述編碼機制的應用

由前文可知道 ISF 係數是寬頻語音分析後最重要的資訊，再加上，此係數經編碼處理的結果為 7 個向量量化索引值，因此可將其套用在我們所提出的多重敘述傳輸系統再作驗證。實驗流程方塊圖如圖 2.6 所示，ISF 分次多階向量量化與索引配置兩個區塊可視為多重敘

述向量量化步驟，雙通道輸出的索引值經通道編碼後送進白高斯雜訊通道傳送。接收端收到資訊後，就由本文所提出的多重敘述解碼機制運作之，然後使用估測的索引值經向量量化碼本的查表而得到一組碼字向量作為 ISF 係數的估測值。

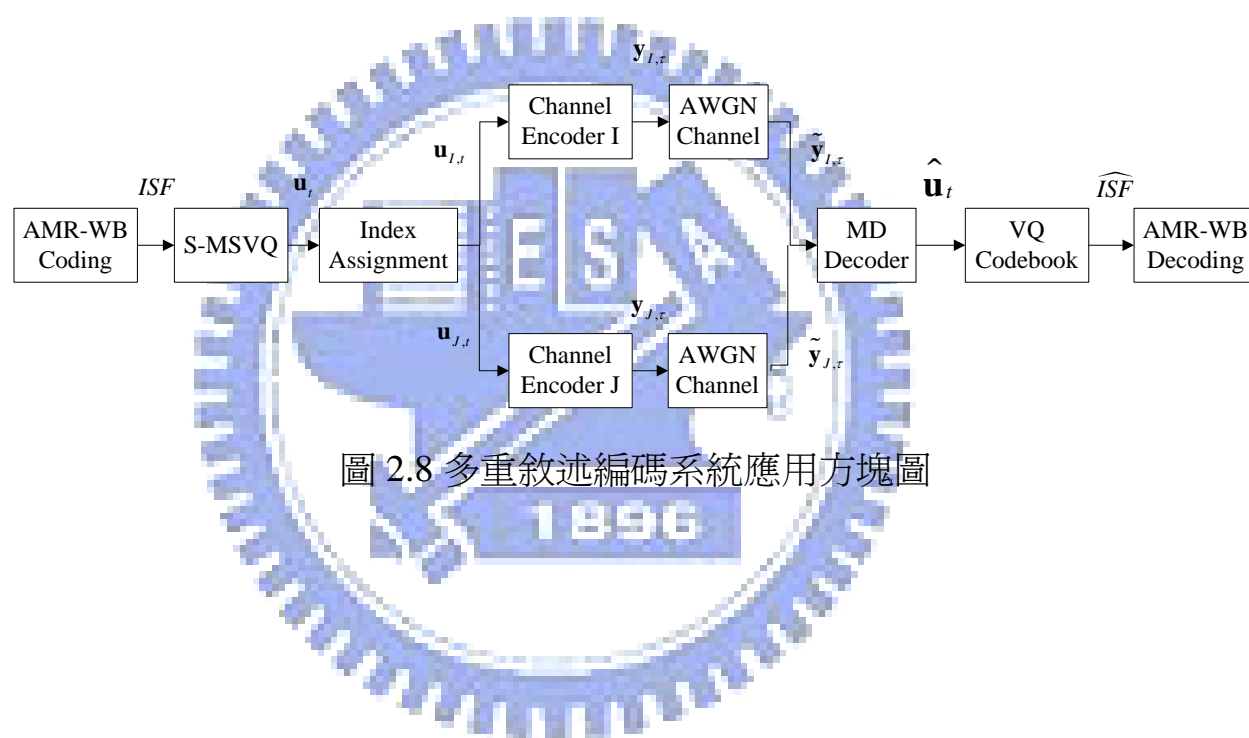


圖 2.8 多重敘述編碼系統應用方塊圖

## 第三章 位元層級的多重敘述疊代解碼機制

渦輪解碼架構是在兩個相同的通道解碼器中間串連一交錯器所共同組成，藉由解碼額外訊息的資訊交換而加強位元判斷的可靠度。訊源通道疊代解碼機制則是根據類似的渦輪碼原理，由通道與訊源兩不同解碼器所組成，加入交錯器是為了消除通道與訊源解碼器輸入訊號的相關性。通道解碼的額外訊息與訊源解碼的額外訊息也作資訊交換的動作，這兩項資訊分別由迴旋碼檢查位元與訊源殘餘冗息所提供。額外訊息的推導將是本章最關鍵的部分，我們將分別介紹 BCJR 演算法以及軟性位元訊源解碼演算法。

本論文的研究課題是兩敘述的傳輸系統，由於解碼元件呈現對稱結構，我們只針對單一通道來探討以簡化其解碼流程的描述。我們提出平行式與交叉式兩種不同的解碼架構，使用 MAP 準則重建訊號並評估兩系統效能上的不同，以下章節將詳細推導過程。

### 3.1 系統解碼流程

#### 3.1.1 系統效能評估

雖然本章所探討的解碼系統是基於位元層級作設計，但系統最終仍需要算出索引後驗機率，以套用在最大後驗機率(maximum a

posteriori probability, MAP) 評判準則，方式如下：

$$\begin{aligned}\hat{i} &= \arg \max_i P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\ \hat{j} &= \arg \max_j P(\mathbf{u}_{J,t} = j | \tilde{\mathbf{U}}_{J,1}^t)\end{aligned}\quad (3.1)$$

接著判斷  $\hat{i}, \hat{j}$  索引值在索引配置矩陣中，是否有對應到一個合法的量化索引或空集合，以便算出  $\hat{v}_t$  訊號。

$$\hat{v}_t = \begin{cases} v_t(\hat{l}), & (\hat{i}, \hat{j}) = \hat{l} \\ \sum_{l=0}^{2^M-1} P(\mathbf{u}_t=l) \cdot v_t(l), & (\hat{i}, \hat{j}) = \phi \end{cases}\quad (3.2)$$

利用 (3.2) 得出來的結果，代進參數訊號雜訊比公式 (3.3) 即可評量系統效能的好壞。

$$SNR_{dB} = 10 \cdot \log \sum_t \left[ \frac{v_t^2}{(v_t - \hat{v}_t)^2} \right]\quad (3.3)$$

### 3.1.2 平行式疊代解碼架構

參考第二章的符號標記方式，可以知道敘述索引是由  $M$  個位元所組成， $\{\mathbf{u}_{I,t} = \{u_{I,t}(1), \dots, u_{I,t}(m), \dots, u_{I,t}(M)\}, u_t(m) \in \{1, 0\}\}$ 。且敘述索引序列  $\mathbf{U}_{I,1}^T = \{\mathbf{u}_{I,1}, \dots, \mathbf{u}_{I,T}\}$  經過交錯器，重新配置其位元順序而得到位元序列  $\{\mathbf{x}_{I,1}^L = \{x_{I,1}, \dots, x_{I,L}\}, x_{I,l} = u_{I,t}(m), l = 1, 2, \dots, L\}$ 。再經過迴旋編碼處理產生序列  $\mathbf{Y}_{I,1}^{\hat{L}} = \{\mathbf{y}_{I,1}, \dots, \mathbf{y}_{I,\hat{L}}\}$ ，其中  $\mathbf{y}_{I,l} = \{y_{I,l}^s, \mathbf{y}_{I,l}^p\} = \{x_{I,l}, \mathbf{y}_{I,l}^p\}$ ，之後經二位元相位鍵移調變再經由一白色高斯雜訊通道傳送。接收端

收到雜訊干擾序列  $\tilde{\mathbf{Y}}_{l,l}^{\hat{L}}$ ，此序列可提供通道相關資訊協助解碼之用。

進一步假設附加性白高斯雜訊(additive white Gaussian

noise, AWGN)通道，可以算出通道相關資訊的機率值如下：

$$P(\tilde{y}_{l,l}^s | x_{l,l}) = \frac{1}{\sqrt{2\pi\sigma_{noise}}} \cdot \exp\left[-\frac{E_s}{N_0} (\tilde{y}_{l,l}^s - x_{l,l})^2\right] \quad (3.4)$$

$$P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p) = \left(\frac{1}{\sqrt{2\pi\sigma_{noise}}}\right)^{(n-1)} \cdot \exp\left[-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_{l,l}^p - \mathbf{y}_{l,l}^p\|^2\right] \quad (3.5)$$

其中雜訊變異數為  $\sigma_{noise}^2 = N_0 / 2E_s$ 。

從圖 3.1 可以清楚看到，敘述 I 的通道解碼器使用通道相關資訊  $P(\tilde{y}_{l,l}^s | x_{l,l})$  及  $P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p)$ ，再整合位元事前消息  $P(x_{l,l})$  計算每一位元的後驗機率  $P(x_{l,l} | \tilde{\mathbf{Y}}_{l,l}^{\hat{L}})$ 。同時也估算通道解碼的額外訊息

$P_{CD}^{[ext]}(x_{l,l})$ ，接著將其經反交錯器處理後得到  $P_{CD}^{[ext]}(u_{l,t}(\lambda))$ ，配合反交錯處理後的通道相關資訊  $P(\tilde{u}_{l,t}(\lambda) | u_{l,t}(\lambda))$  以及索引的一階事前消息  $P(\mathbf{u}_{l,t} | \mathbf{u}_{l,t-1})$ ，一起輸入到軟性位元訊源解碼端(soft-bit source decoder, SBSDD)，分別算出訊源解碼的額外消息  $P_{SBSDD}^{[ext]}(u_{l,t}(\lambda))$  以及索引的後驗機率。索引後驗機率如下表示：

$$\begin{aligned} P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,t}^t) &= P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,t}^{t-1}, \tilde{\mathbf{u}}_{l,t}^t) \\ &= C \cdot P(\tilde{\mathbf{u}}_{l,t} | \mathbf{u}_{l,t} = i) \sum_{\mathbf{u}_{l,t-1}} P(\mathbf{u}_{l,t} = i | \mathbf{u}_{l,t-1}) \cdot P(\mathbf{u}_{l,t-1} | \tilde{\mathbf{U}}_{l,t}^{t-1}) \end{aligned} \quad (3.6)$$

相關機率的詳細推導將在後續的兩節呈現。至於軟性位元訊源解碼的

額外訊息  $P_{SBSDD}^{[ext]}(u_{l,t}(\lambda))$ ，經過交錯器配置後得到  $P_{SBSDD}^{[ext]}(x_{l,l})$ ，將在下一

次疊代過程中結合位元事前消息  $P(x_{l,l})$ ，也就是以其成積

( $P(x_{l,l}) \cdot P_{SBSD}^{[ext]}(x_{l,l})$ ) 此項來當成新的事前消息使用，以加強通道解碼的效果。

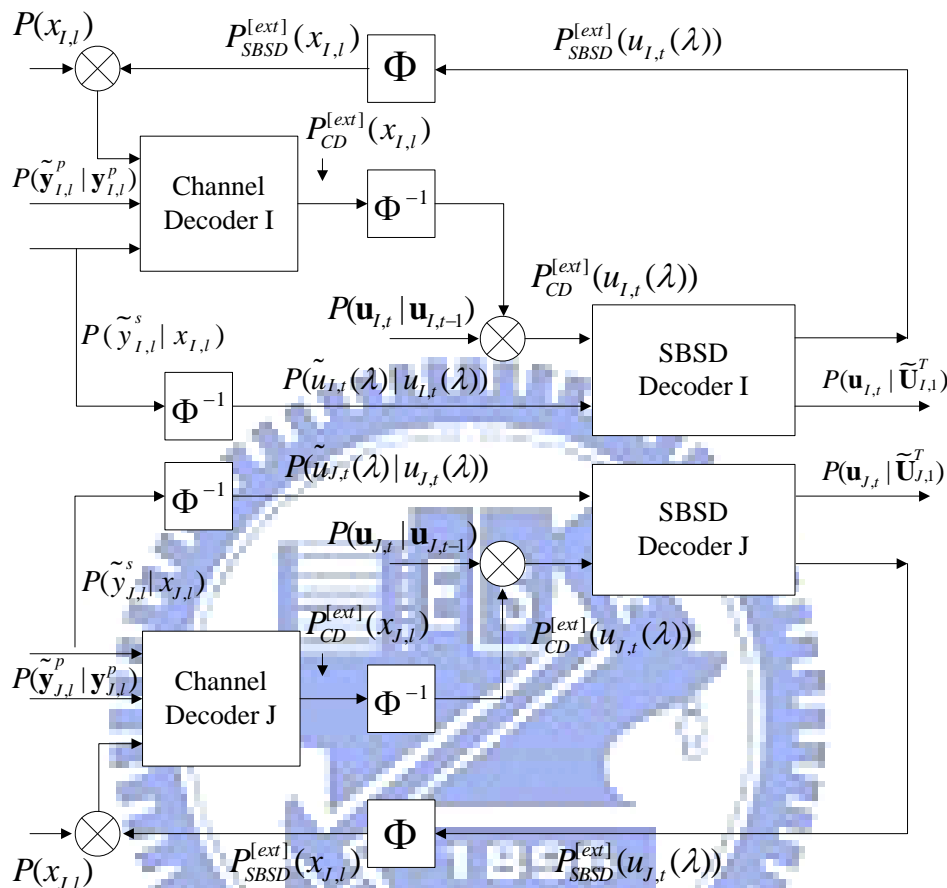


圖 3.1 平行式疊代解碼方塊圖

### 3.1.3 交叉式疊代解碼架構

交叉式與平行式疊代碼編碼架構的最大差異在於訊源解碼器輸入端事前訊息的運用。如同前一小節所示，接收端收到雜訊干擾後的序列  $\tilde{\mathbf{Y}}_{l,l}^l$  後，可以先行計算其通道相關資訊的機率值  $P(\tilde{y}_{l,l}^s | x_{l,l})$  及  $P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p)$ 。將通道相關資訊與位元事前消息  $P(x_{l,l})$  送進通道解碼器，分別計算通道解碼額外訊息  $P_{CD}^{[ext]}(x_{l,l})$  及位元後驗機率

$P(x_{I,t} | \tilde{\mathbf{Y}}_{I,t}^L)$ 。如同平行式解碼架構，交叉式架構的訊源解碼器也利用索引的一階事前消息  $P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1})$  以及反交錯處理的通道解碼額外訊息  $P_{CD}^{[ext]}(u_{I,t}(\lambda))$ 。但最大的不同點是，交叉式架構將整合反交錯配置後的兩項通道相關資訊  $P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda))$  與  $P(\tilde{u}_{J,t}(\lambda) | u_{J,t}(\lambda))$ ，同時考慮到雙通道敘述間的相關性訊息  $P(\mathbf{u}_{J,t}, \mathbf{u}_{I,t})$ ，將這些資訊一併送進訊源解碼器作渦輪交互解碼(turbo cross decoding)使用，最後計算出索引後驗機率值  $P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,t}^t, \tilde{\mathbf{U}}_{J,t}^t)$ 。由於我們額外考慮另一通道的相關資訊與雙通道敘述的結合機率，可以預期能超越基於單一敘述的訊源解碼器，進而增強其訊源解碼額外訊息  $P_{SBSD}^{[ext]}(u_{I,t}(\lambda))$  的可靠度。相關機率的詳細推導將在後續章節呈現。



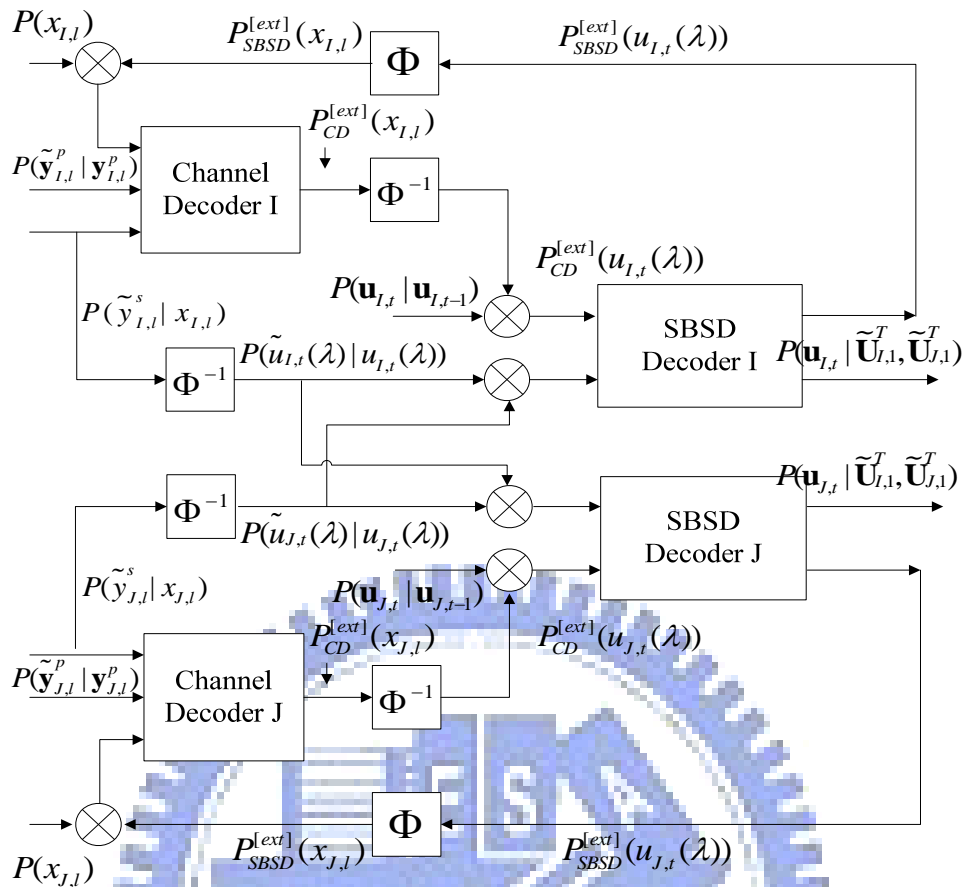


圖 3.2 交叉式疊代解碼方塊圖

### 3.2 通道解碼額外訊息的推導

不論是平行式或交叉式編碼架構，都使用相同的通道解碼器，而其輸入端的事前訊息亦相同，因此本章節有關通道解碼額外訊息的推導可適用於平行式與交叉式架構。BCJR 演算法[6]是一種軟性輸入軟性輸出的解碼方法，常被使用在迴旋碼的通道解碼上。由於編碼的籬柵結構(trellis diagram)，因此解碼流程呈現遞迴的機率形式，可大幅減低計算量與複雜度。此外，經由每個時刻點位元後驗機率的估算，據此判斷其資訊位元為 '1' or '0'。如果要進一步與訊源解碼器



搭配作疊代解碼，則需算出通道解碼的額外訊息，並傳遞到訊源解碼器作為事前消息使用。

我們參考[6]文章中所述的 BCJR 演算法，先從位元的後驗機率模型開始探討，如下呈現：

$$P(x_{l,l} = +1 | \tilde{\mathbf{Y}}_{l,l}^{\hat{L}}) = C \cdot P(x_{l,l} = +1, \tilde{\mathbf{Y}}_{l,l}^{\hat{L}}) = C \cdot \sum_{(s_{l-1}, s_l) \in \Sigma_l^+} P(s_{l-1}, s_l, \tilde{\mathbf{Y}}_{l,l}^{\hat{L}}) \quad (3.7)$$

其中  $C = 1/P(\tilde{\mathbf{Y}}_{l,l}^{\hat{L}})$ ， $\Sigma_l^+$  是輸入位元為  $x_{l,l} = +1$  時從狀態  $s_{l-1}$  轉移至  $s_l$  的轉移集合。接著我們將個別機率拆解成三項並給予定義：

$$\begin{aligned} P(s_{l-1}, s_l, \tilde{\mathbf{Y}}_{l,l}^{\hat{L}}) &= P(s_{l-1}, \tilde{\mathbf{Y}}_{l,l}^{l-1}) \cdot P(s_l, \tilde{\mathbf{y}}_{l,l} | s_{l-1}) \cdot P(\tilde{\mathbf{Y}}_{l,l+1}^{\hat{L}} | s_l) \\ &\triangleq \alpha_{l-1}(s_{l-1}) \cdot \gamma_l(s_{l-1}, s_l) \cdot \beta_l(s_l) \end{aligned} \quad (3.8)$$

其中前向與後向機率的遞迴形式可推導如下：

$$\alpha_l(s_l) = \sum_{s_{l-1}} \gamma_l(s_{l-1}, s_l) \cdot \alpha_{l-1}(s_{l-1}) \quad (3.9)$$

$$\beta_{l-1}(s_{l-1}) = \sum_{s_l} \gamma_l(s_{l-1}, s_l) \cdot \beta_l(s_l) \quad (3.10)$$

遞迴演算法的使用必須設定初始條件，在  $t=0$  的起始狀態為  $s_0 = 0$ ，在  $t = \hat{L}$  的結束狀態為  $s_{\hat{L}} = 0$ ，設定如下：

$$\alpha_0(s_0) = \begin{cases} 1, & s_0 = 0 \\ 0, & s_0 \neq 0 \end{cases} \quad (3.11)$$

$$\beta_{\hat{L}}(s_{\hat{L}}) = \begin{cases} 1, & s_{\hat{L}} = 0 \\ 0, & s_{\hat{L}} \neq 0 \end{cases} \quad (3.12)$$

至於路徑轉移機率  $\gamma_l(s_{l-1}, s_l)$  的推導，基於通道無記憶性的假設，且令

$$(s_{l-1}, s_l) \in \Sigma_l^+,$$

$$\begin{aligned}
\gamma_l(s_{l-1}, s_l) &= P(s_l, \tilde{\mathbf{y}}_{l,l} | s_{l-1}) \\
&= P(s_l | s_{l-1}) \cdot P(\tilde{\mathbf{y}}_{l,l} | s_l, s_{l-1}) \\
&= P(x_{l,l} = +1) \cdot P(\tilde{\mathbf{y}}_{l,l} | \mathbf{y}_{l,l}) \\
&= P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | y_{l,l}^s) \cdot P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p) \\
&= P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p) \\
&\triangleq P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot \gamma_l^{[ext]}(s_{l-1}, s_l) \tag{3.13}
\end{aligned}$$

其中  $\gamma_l^{[ext]}(s_{l-1}, s_l) = P(\tilde{\mathbf{y}}_{l,l}^p | \mathbf{y}_{l,l}^p)$  代表同位檢察位元所屬的通道相關資訊。因

此位元的後驗機率(3.7)可重新改寫成如下形式：

$$\begin{aligned}
P(x_{l,l} = +1 | \tilde{\mathbf{Y}}_{l,l}^L) &= C \cdot \sum_{(s_{l-1}, s_l) \in \Sigma_l^+} \alpha_{l-1}(s_{l-1}) \cdot \gamma_l(s_{l-1}, s_l) \cdot \beta_l(s_l) \\
&= C \cdot \sum_{(s_{l-1}, s_l) \in \Sigma_l^+} \alpha_{l-1}(s_{l-1}) \cdot P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot \gamma_l^{[ext]}(s_{l-1}, s_l) \cdot \beta_l(s_l) \\
&= C \cdot P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \sum_{(s_{l-1}, s_l) \in \Sigma_l^+} \alpha_{l-1}(s_{l-1}) \cdot \gamma_l^{[ext]}(s_{l-1}, s_l) \cdot \beta_l(s_l) \\
&\triangleq C \cdot P(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot P_{CD}^{[ext]}(x_{l,l} = +1) \tag{3.14}
\end{aligned}$$

從上列推導中，可以得知通道解碼額外訊息為

$$P_{CD}^{[ext]}(x_{l,l} = +1) = \sum_{(s_{l-1}, s_l) \in \Sigma_l^+} \alpha_{l-1}(s_{l-1}) \cdot \gamma_l^{[ext]}(s_{l-1}, s_l) \cdot \beta_l(s_l) \tag{3.15}$$

值得注意的是，在疊代的起始過程，訊源解碼器尚未運作而無法提供

其額外訊息，因此通道解碼器輸入端的位元事前消息為  $P(x_{l,l} = +1)$ 。

而在一次疊代完成之後，訊源解碼器輸出的額外訊息  $P_{SBS D}^{[ext]}(x_{l,l} = +1)$ ，

可在後續的疊代過程中，另外結合位元事前消息而得

(  $P(x_{l,l} = +1) \cdot P_{SBS D}^{[ext]}(x_{l,l} = +1)$  )，作為通道解碼輸入端的事前消息使用。

有鑑於此，我們將路徑轉移機率(3.13)式與位元後驗機率(3.14)式分

別更新成為

$$\gamma_l(s_{l-1}, s_l) = P(x_{l,l} = +1) \cdot P_{SBSD}^{[extr]}(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot \gamma_l^{[extr]}(s_{l-1}, s_l) \quad (3.16)$$

$$\begin{aligned} & P(x_{l,l} = +1 | \tilde{\mathbf{Y}}_{l,l}^L) \\ &= C \cdot P(x_{l,l} = +1) \cdot P_{SBSD}^{[extr]}(x_{l,l} = +1) \cdot P(\tilde{y}_{l,l}^s | x_{l,l} = +1) \cdot P_{CD}^{[extr]}(x_{l,l} = +1) \end{aligned} \quad (3.17)$$

### 3.3 訊源解碼額外訊息的推導

#### 3.3.1 平行式架構的相關推導

多重敘述量化器的輸出具有相關性，依其殘餘冗息不同程度的使用模型可概分為 0 階與 1 階的事前資訊。特別強調的是本章所使用的殘餘冗息，只考慮配置矩陣中單一系列索引  $I$  或行索引  $J$  的事前消息，而在 4.3.2 節合併式架構則使用量化索引事前消息。因為量化索引  $\mathbf{u}_l$  比其所屬的兩敘述  $\mathbf{u}_{l,t}$  與  $\mathbf{u}_{j,t}$  具有更豐富的殘餘冗息，實驗證明本章架構的解碼效能有其限制。

平行式架構擁有兩個獨立的訊源解碼器，每一敘述只使用其通道接收到的資訊，各自算出敘述索引 ( $\mathbf{u}_{l,t}$  或  $\mathbf{u}_{j,t}$ ) 的後驗機率值，再據以推得訊源解碼的額外訊息。在接下來的機率推導中，我們將鎖定配置矩陣的列索引  $\mathbf{u}_{l,t}$  作描述，同理可推得行索引  $\mathbf{u}_{j,t}$  的相關機率。

#### [1] 敘述索引 0 階事前訊息的運用

首先基於相鄰兩列索引間互相獨立的假設，我們展開通道  $I$  所傳送列索引  $\mathbf{u}_{l,t}$  的後驗機率如下：

$$\begin{aligned}
& P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,t}^t) \\
& = C \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t} = i)
\end{aligned} \tag{3.18}$$

進一步假設通道的無記憶性，得到

$$P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) = \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \tag{3.19}$$

除了起始疊代之外，疊代過程的額外訊息視為訊源解碼器附加事前消息，也就是將  $P(\mathbf{u}_{I,t} = i) \leftarrow P(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i)$ ，且

$$P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) = \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P_{CD}^{[ext]}(u_{I,t}(m)) \tag{3.20}$$

綜合上述假設，(3.18)將變成：

$$\begin{aligned}
& P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,t}^t) \\
& = C \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \\
& = C \cdot P(\mathbf{u}_{I,t} = i) \cdot \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m))
\end{aligned} \tag{3.21}$$

如同通道解碼流程一樣，我們下一步將根據索引後驗機率

$P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,t}^t)$  先計算出位元的後驗機率，再進一步推導得出位元層級的訊源解碼額外訊息。因此我們必須將(3.21)改成位元層級的後驗機

率形式如下：

$$\begin{aligned}
P(u_{I,t}(\lambda) = +1 | \tilde{\mathbf{U}}_{I,t}^t) & = \sum_{(\mathbf{u}_{I,t}: u_{I,t}(\lambda) = +1)} P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,t}^t) \\
& = C \cdot \sum_{(\mathbf{u}_{I,t}: u_{I,t}(\lambda) = +1)} P(\mathbf{u}_{I,t}) \cdot \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \\
& = C \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot \\
& \quad \sum_{(\mathbf{u}_{I,t}: u_{I,t}(\lambda) = +1)} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \cdot P(\mathbf{u}_{I,t}^{[ext]}(\lambda))
\end{aligned}$$

$$\triangleq C \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot P_{SBSD}^{[ext]}(u_{I,t}(\lambda) = +1) \quad (3.22)$$

在(3.22)公式中，我們假設  $\mathbf{u}_{I,t} = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

的位元間彼此獨立，同時定義  $\mathbf{u}_{I,t}^{[ext]}(\lambda) = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

，其機率為

$$P(\mathbf{u}_{I,t}^{[ext]}(\lambda)) \triangleq \frac{P(\mathbf{u}_{I,t})}{P(u_{I,t}(\lambda) = +1)} \quad (3.23)$$

然後我們從(3.22)公式中，可以得出訊源解碼的額外訊息為：

$$P_{SBSD}^{[ext]}(u_{I,t}(\lambda) = +1) = \sum_{(\mathbf{u}_{I,t}, \mathbf{u}_{I,t}(\lambda) = +1)} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \cdot P(\mathbf{u}_{I,t}^{[ext]}(\lambda)) \quad (3.24)$$

## [2] 敘述索引 1 階事前訊息的運用

基於通道無記憶性與相鄰兩列索引間相關性的假設，推得列索引

的後驗機率公式如下：

$$\begin{aligned} P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) &= P(\mathbf{u}_{I,t} = i, \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{u}}_{I,t}) \\ &= C \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \end{aligned} \quad (3.25)$$

$$\text{其中 } P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) = \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \quad (3.26)$$

接著將通道解碼器輸出的額外訊息當作附加的事前消息使用，也就是

$$P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \leftarrow P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) ,$$

$$\text{且 } P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) = \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P_{CD}^{[ext]}(u_{I,t}(m)) \quad (3.27)$$

(3.25)將變成：

$$\begin{aligned} & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\ &= C \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P_{CD}^{[ext]}(u_{I,t}(m)) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \\ &= C \cdot \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \end{aligned} \quad (3.28)$$

如同 0 階事前訊息運用的前例，先計算出位元的後驗機率，再進一步推導得出位元層級的訊源解碼額外訊息。因此我們必須將(3.28)改成位元層級的後驗機率形式如下：

$$\begin{aligned} P(u_{I,t}(\lambda) = +1 | \tilde{\mathbf{U}}_{I,1}^t) &= \sum_{(\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1)} P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t) \\ &= C \cdot \sum_{(\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1)} \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \\ &= C \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot \\ & \quad \sum_{(\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1)} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t}^{[ext]}(\lambda) | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \\ &\triangleq C \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot P_{SBS D}^{[ext]}(u_{I,t}(\lambda) = +1) \end{aligned} \quad (3.29)$$

(3.29)公式中，我們假設  $\mathbf{u}_{I,t} = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

位元間彼此獨立，同時定義  $\mathbf{u}_{I,t}^{[ext]}(\lambda) = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

，其機率為

$$P(\mathbf{u}_{I,t}^{[ext]}(\lambda) | \mathbf{u}_{I,t-1}) \triangleq \frac{P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1})}{P(u_{I,t}(\lambda) = +1)} \quad (3.30)$$

從(3.29)公式，可以看出訊源解碼的額外訊息為：

$$\begin{aligned}
 & P_{SBSD}^{[extr]}(u_{I,t}(\lambda) = +1) \\
 &= \sum_{(\mathbf{u}_{I,t}, u_{I,t}(\lambda) = +1)} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | \mathbf{u}_{I,t}) \cdot P_{CD}^{[extr]}(u_{I,t}(m)) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t}^{[extr]}(\lambda) | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1})
 \end{aligned} \tag{3.31}$$

### 3.3.2 交叉式架構的相關推導

雖然此系統使用的殘餘冗息仍為行或列索引的事前消息，但預期其解碼效能會優於平行式架構。主要是因為此架構下訊源解碼器有雙通道資訊交互使用的現象，也就是利用另一敘述的通道相關資訊來加強目前敘述的解碼可靠性，遠比平行式架構只使用各自的通道資訊作獨立解碼來得有效。交叉式架構是基於渦輪交互解碼(turbo cross decoding)原理，其相關研究可以參考文章[7]、[8]、[9]。

[1] 敘述索引 0 階事前訊息的運用

根據同時收到的列及行索引序列 $\{\tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t\}$ ，我們先計算出列索引 $\mathbf{u}_{I,t}$ 的後驗機率

$$\begin{aligned}
 P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) &= C \cdot P(\mathbf{u}_{I,t}, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{I,t}, \mathbf{u}_{J,t}, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_t, \tilde{\mathbf{U}}_1^t), \quad [\text{令 } \mathbf{u}_t = (\mathbf{u}_{I,t}, \mathbf{u}_{J,t}), \tilde{\mathbf{U}}_1^t = (\tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)] \\
 &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_t | \mathbf{u}_t, \tilde{\mathbf{U}}_1^{t-1}) \cdot P(\mathbf{u}_t | \tilde{\mathbf{U}}_1^{t-1}) \cdot P(\tilde{\mathbf{U}}_1^{t-1}) \\
 &= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_t | \mathbf{u}_t) \cdot P(\mathbf{u}_t), \quad [\text{令 } C' = C \cdot P(\tilde{\mathbf{U}}_1^{t-1})]
 \end{aligned}$$

$$\begin{aligned}
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_t, \mathbf{u}_t) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t}) \cdot P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t}) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}, \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \tag{3.32}
\end{aligned}$$

$$\text{其中 } P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) = \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \tag{3.33}$$

接著將通道解碼的額外訊息視為訊源解碼器的附加事前消息，也就是

$$P(\mathbf{u}_{I,t}) \leftarrow P(\mathbf{u}_{I,t}) \cdot P_{CD}^{[extr]}(\mathbf{u}_{I,t}), \text{ 且}$$

$$P_{CD}^{[extr]}(\mathbf{u}_{I,t}) = \prod_{\substack{m=1 \\ \mathbf{u}_{I,t}=i}}^M P_{CD}^{[extr]}(u_{I,t}(m)) \tag{3.34}$$

更進一步(3.32)會更新成以下形式：

$$\begin{aligned}
&P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,t}^t, \tilde{\mathbf{U}}_{J,t}^t) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \cdot P_{CD}^{[extr]}(\mathbf{u}_{I,t}) \\
&= C' \cdot \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[extr]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \tag{3.35}
\end{aligned}$$

如前文所述，進一步將(3.35)改成位元層級の後驗機率形式如下：

$$\begin{aligned}
P(u_{I,t}(\lambda) = +1 | \tilde{\mathbf{U}}_{I,t}^t, \tilde{\mathbf{U}}_{J,t}^t) &= \sum_{\mathbf{u}_{J,t}: u_{J,t}(\lambda) = +1} P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,t}^t, \tilde{\mathbf{U}}_{J,t}^t) \\
&= C' \cdot \sum_{\mathbf{u}_{J,t}: u_{J,t}(\lambda) = +1} \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[extr]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}) \\
&= C' \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[extr]}(u_{I,t}(\lambda) = +1) \cdot \\
&\quad \sum_{\mathbf{u}_{J,t}: u_{J,t}(\lambda) = +1} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[extr]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}^{[extr]}(\lambda))
\end{aligned}$$



$$\triangleq C' \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot P_{SBSD}^{[ext]}(u_{I,t}(\lambda) = +1) \quad (3.36)$$

在(3.36)公式中，假設  $\mathbf{u}_{I,t} = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

位元間彼此獨立，同時定義  $\mathbf{u}_{I,t}^{[ext]}(\lambda) = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

，及其機率：

$$P(\mathbf{u}_{I,t}^{[ext]}(\lambda)) \triangleq \frac{P(\mathbf{u}_{I,t})}{P(u_{I,t}(\lambda) = +1)} \quad (3.37)$$

從(3.36)可以清楚得到訊源解碼的額外訊息為

$$\begin{aligned} & P_{SBSD}^{[ext]}(u_{I,t}(\lambda) = +1) \\ &= \sum_{\mathbf{u}_{I,t}: u_{I,t}(\lambda) = +1} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}^{[ext]}(\lambda)) \end{aligned} \quad (3.38)$$

## [2] 敘述索引 1 階事前訊息的運用

基於通道無記憶性與相鄰兩列索引間相關性的假設，推得列索引

的後驗機率如下：

$$\begin{aligned} & P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = C \cdot P(\mathbf{u}_{I,t}, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{I,t}, \mathbf{u}_{J,t}, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t), \text{ [令 } \mathbf{u}_t = (\mathbf{u}_{I,t}, \mathbf{u}_{J,t}), \tilde{\mathbf{U}}_1^t = (\tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)] \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_t, \tilde{\mathbf{U}}_1^t) = C \cdot \sum_{\mathbf{u}_{J,t}} \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t, \mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^t) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t, \tilde{\mathbf{u}}_t | \mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t, \tilde{\mathbf{u}}_t | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t}, \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \end{aligned} \quad (3.39)$$

在(3.39)公式，索引後驗機率可以拆解成三項因子，其中前兩個因子可依通道無記憶性的假設再作簡化，

$$P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t}, \mathbf{u}_{t-1}) = P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \quad (3.40)$$

$$P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{t-1}) = P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}, \mathbf{u}_{J,t-1}) = P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}) \quad (3.41)$$

如前人研究[9]所言，(3.40)強調在同一時刻雙通道資訊的相關性

$P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t})$  會比一階事前資訊  $P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t-1}, \mathbf{u}_{J,t-1})$  來的重要，而(3.41)主

要是強調相同敘述的一階事前消息  $P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1})$  會比不同敘述間的一

階事前消息  $P(\mathbf{u}_{I,t} | \mathbf{u}_{J,t-1})$  更具有相關性。因此(3.39)會變成：

$$\begin{aligned} & P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{I,t-1}} \sum_{\mathbf{u}_{J,t-1}} P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\mathbf{u}_{J,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t}, \tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t-1}) \sum_{\mathbf{u}_{J,t-1}} P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}, \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{I,t-1}} P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}, \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{I,t-1}} P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1}, \tilde{\mathbf{U}}_1^{t-1}) \\ &= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \end{aligned}$$

$$\text{其中 } C' = C \cdot P(\tilde{\mathbf{U}}_1^{t-1}) \quad (3.42)$$

$$P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) = \prod_{\substack{m=1 \\ \mathbf{u}_{J,t}=i}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \quad (3.43)$$

接著將通道解碼器輸出的額外訊息當作附加的事前消息使用，也就是

$P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \leftarrow P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t})$ ，且

$$P_{CD}^{[ext]}(\mathbf{u}_{I,t}) = \prod_{m=1}^M P_{CD}^{[ext]}(u_{I,t}(m)) \quad (3.44)$$

更進一步(3.42)會更新成以下形式：

$$\begin{aligned} & P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C' \cdot \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t}) \cdot \\ & \quad \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= C' \cdot \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot \\ & \quad \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \end{aligned} \quad (3.45)$$

如前文所述，進一步將(3.45)改成位元層級の後驗機率形式如下：

$$\begin{aligned} & P(u_{I,t}(\lambda) = +1 | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = \sum_{\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1} P(\mathbf{u}_{I,t} | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C' \cdot \sum_{\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1} \prod_{m=1}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot \\ & \quad \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= C' \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot \\ & \quad \left\{ \sum_{\mathbf{u}_{I,t}; u_{I,t}(\lambda) = +1} \prod_{\substack{m=1 \\ m \neq \lambda}}^M P(\tilde{u}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot \right. \\ & \quad \left. \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t}^{[ext]}(\lambda) | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \right\} \\ &\triangleq C' \cdot P(u_{I,t}(\lambda) = +1) \cdot P(\tilde{u}_{I,t}(\lambda) | u_{I,t}(\lambda) = +1) \cdot P_{CD}^{[ext]}(u_{I,t}(\lambda) = +1) \cdot P_{SBSBD}^{[ext]}(u_{I,t}(\lambda) = +1) \end{aligned} \quad (3.46)$$

(3.46)公式中，假設  $\mathbf{u}_{I,t} = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$

位元間彼此獨立，定義  $\mathbf{u}_{I,t}^{[ext]}(\lambda) = \{u_{I,t}(1), \dots, u_{I,t}(\lambda-1), u_{I,t}(\lambda+1), \dots, u_{I,t}(M)\}$ ，

及其機率為

$$P(\mathbf{u}_{I,t}^{[ext]}(\lambda) | \mathbf{u}_{I,t-1}) \triangleq \frac{P(\mathbf{u}_{I,t} | \mathbf{u}_{I,t-1})}{P(u_{I,t}(\lambda) = +1)} \quad (3.47)$$

從(3.46)公式中，可以看出訊源解碼的額外訊息為：

$$\begin{aligned} & P_{SBS D}^{[ext]}(u_{I,t}(\lambda) = +1) \\ &= \sum_{\substack{\mathbf{u}_{J,t}: u_{J,t}(\lambda) = +1 \\ m=1 \\ m \neq \lambda}}^M \prod_{m=1}^M P(\tilde{\mathbf{u}}_{I,t}(m) | u_{I,t}(m)) \cdot P_{CD}^{[ext]}(u_{I,t}(m)) \sum_{\mathbf{u}_{J,t}} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t}) \cdot P(\mathbf{u}_{J,t} | \mathbf{u}_{I,t}) \cdot \\ & \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t}^{[ext]}(\lambda) | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \end{aligned} \quad (3.48)$$



## 第四章 索引層級的多重敘述疊代解碼機制

在第三章介紹的位元層級疊代解碼過程中，訊源解碼器的運算是基於索引後驗機率，而通道解碼器的運算是基於位元後驗機率，兩者之間的匹配必須基於位元間彼此獨立的假設，方能藉由位元機率的乘積轉換成索引機率值。但如此一來，與實際上的索引機率值會有相當程度的誤差。為了解決上述問題，我們將通道與訊源解碼器都改成索引後驗機率的運算模式，提出一索引層級 BCJR 演算法替代傳統的通道位元解碼方式，而訊源解碼則為索引層級的軟性位元輸出演算法。除此之外，更進一步據以改良多重敘述疊代解碼的架構，以期有更好的系統效能。由於通道與訊源解碼器是以索引機率值作為資訊交換的媒介，為確保兩解碼器能夠互不影響地獨立運作，串聯於兩者之間之交錯器需針對索引序列的順序重作配置。這章提出兩種不同的解碼架構，為彰顯其系統效能上的差異，有別於前一章的最大後驗機率(MAP)估算法則，我們將使用最小均方誤差(minimum mean square error, MMSE)準則來評估解碼的效果，這更能看出架構上的優缺點以便深入探討。

### 4.1 系統解碼流程

### 4.1.1 系統效能評估

這章所提出的架構旨在計算系統解碼後的後驗機率，接著據以調整不同碼本向量的權重分配，再估算其訊號輸出值。若令  $v_t(l)$  為索引  $l$  的碼本向量，則 MMSE 參數估算準則如下：

$$\hat{v}_t = \sum_{l=0}^{2^M-1} P(\mathbf{u}_t = l | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{J,1}) \cdot v_t(l) \quad (4.1)$$

利用(4.1)得出來的結果代入  $SNR_{dB} = 10 \cdot \log \sum_t \left[ \frac{v_t^2}{(v_t - \hat{v}_t)^2} \right]$  (4.2)

，得其參數信號雜訊比(parameter SNR)，可藉以評量系統效能的好壞差異。

### 4.1.2 平行式疊代解碼架構

這是一個雙敘述通道的傳輸系統，解碼器具有對稱的結構特性，因此如同上一章，我們只針對單一通道探討其解碼演算法。參考第二章所定義的標記方式，資訊索引序列  $\mathbf{U}_{l,1}^T = \{\mathbf{u}_{l,1}, \dots, \mathbf{u}_{l,t}, \dots, \mathbf{u}_{l,T}\}$  經過索引交錯器的重新配置如下： $\mathbf{X}_{l,1}^T = (\mathbf{x}_{l,1}, \dots, \mathbf{x}_{l,\tau}, \dots, \mathbf{x}_{l,T})$ ，其中  $\mathbf{x}_{l,\tau} = \mathbf{u}_{l,t}$ 。將配置後的索引序列經過迴旋編碼過程得到： $\mathbf{Y}_{l,1}^{\hat{t}} = (\mathbf{y}_{l,1}, \dots, \mathbf{y}_{l,\hat{t}})$ ，其中  $\mathbf{y}_{l,\tau} = (\mathbf{y}_{l,\tau}^s, \mathbf{y}_{l,\tau}^p) = (\mathbf{x}_{l,\tau}, \mathbf{y}_{l,\tau}^p)$ 。編碼後的序列透過二位元相位鍵移調變處理，以位元串的形式經由一零平均值變異數為  $\sigma_{noise}^2 = N_0/2E_s$  的白色高斯雜訊(AWGN)通道傳送。接收端收到白高斯雜訊干擾後的序列，利用 AWGN 通道的無記憶性，計算出通道相關資訊的機率值如下：

$$P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau}) = \left(\frac{1}{\sqrt{2\pi}\sigma_{noise}}\right)^M \cdot \exp\left(-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_{I,\tau}^s - \mathbf{x}_{I,\tau}\|^2\right) \quad (4.3)$$

$$P(\tilde{\mathbf{y}}_{I,\tau}^p | \mathbf{y}_{I,\tau}^p) = \left(\frac{1}{\sqrt{2\pi}\sigma_{noise}}\right)^{(n-1)M} \cdot \exp\left(-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_{I,\tau}^p - \mathbf{y}_{I,\tau}^p\|^2\right) \quad (4.4)$$

從通道 I 接收端的觀點來看，通道解碼器使用通道相關資訊

$P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau} = i)$  及  $P(\tilde{\mathbf{y}}_{I,\tau}^p | \mathbf{y}_{I,\tau}^p)$ ，配合索引事前消息  $P(\mathbf{x}_{I,\tau} = i)$ ，估算出索引後驗機率  $P(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^t)$  及其通道解碼的額外訊息  $P_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i)$ 。接著將此項額外訊息與 I 索引的一階事前消息  $P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1})$ ，再加上經過反交錯器處理的通道相關資訊  $P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i)$ ，同時送入軟性位元訊源解碼器 (SBSD) 的輸入端。軟性位元訊源解碼器利用輸入的資訊，計算出其額外訊息  $P_{SBS D}^{[ext]}(\mathbf{u}_{I,t} = i)$ ，及其列索引 I 的後驗機率值  $P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t)$ 。至於索引後驗機率公式的詳細推導，將在後續的章節中呈現。圖 4.1 所示則為資訊交換的流程。軟性位元訊源解碼的額外訊息  $P_{SBS D}^{[ext]}(\mathbf{u}_{I,t} = i)$ ，會在下一次的疊代過程用以取代舊的索引事前消息  $P(\mathbf{x}_{I,\tau} = i)$ ，作為通道解碼端新的可靠資訊。而在每次疊代過程完成時，兩個訊源解碼器估算出來的列及行索引後驗機率則同時送至多重敘述解碼器，進行第二階段量化索引後驗機率  $P(\mathbf{u}_l = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)$  的計算。

首先定義  $P(\mathbf{u}_{I,t} = i)$  為 I 索引的事前機率， $P(\mathbf{u}_{J,t} = j)$  為 J 索引的事前機率， $P(\mathbf{u}_{I,t} = i, \mathbf{u}_{J,t} = j)$  為 I, J 兩索引間的結合機率， $P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t)$  為 I 索引的後驗機率， $P(\mathbf{u}_{J,t} = j | \tilde{\mathbf{U}}_{J,1}^t)$  為 J 索引的後驗機率。若令量化索引  $l$  所對應的列及行索引分別為  $i$  與  $j$ ，基於機率的基本定理，我們推

導出量化索引的後驗機率如下：

$$\begin{aligned}
 & P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{l,1}^t, \tilde{\mathbf{U}}_{j,1}^t) \\
 &= C \cdot P(\mathbf{u}_t = l, \tilde{\mathbf{U}}_{l,1}^t, \tilde{\mathbf{U}}_{j,1}^t) \\
 &= C \cdot P(\mathbf{u}_{l,t} = i, \mathbf{u}_{j,t} = j, \tilde{\mathbf{U}}_{l,1}^t, \tilde{\mathbf{U}}_{j,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{U}}_{l,1}^t | \mathbf{u}_{l,t} = i, \mathbf{u}_{j,t} = j, \tilde{\mathbf{U}}_{j,1}^t) \cdot P(\mathbf{u}_{l,t} = i | \mathbf{u}_{j,t} = j, \tilde{\mathbf{U}}_{j,1}^t) \cdot P(\mathbf{u}_{j,t} = j | \tilde{\mathbf{U}}_{j,1}^t) \cdot P(\tilde{\mathbf{U}}_{j,1}^t) \\
 &= C \cdot [P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,1}^t) \cdot P(\tilde{\mathbf{U}}_{l,1}^t) / P(\mathbf{u}_{l,t} = i)] \cdot P(\mathbf{u}_{l,t} = i | \mathbf{u}_{j,t} = j) \cdot P(\mathbf{u}_{j,t} = j | \tilde{\mathbf{U}}_{j,1}^t) \cdot P(\tilde{\mathbf{U}}_{j,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{U}}_{l,1}^t) \cdot P(\tilde{\mathbf{U}}_{j,1}^t) \cdot P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,1}^t) \cdot \left[ \frac{P(\mathbf{u}_{l,t} = i | \mathbf{u}_{j,t} = j)}{P(\mathbf{u}_{l,t} = i)} \right] \cdot P(\mathbf{u}_{j,t} = j | \tilde{\mathbf{U}}_{j,1}^t) \\
 &= C' \cdot P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,1}^t) \cdot \left[ \frac{P(\mathbf{u}_{l,t} = i | \mathbf{u}_{j,t} = j)}{P(\mathbf{u}_{l,t} = i)} \right] \cdot P(\mathbf{u}_{j,t} = j | \tilde{\mathbf{U}}_{j,1}^t), \quad [\text{令 } C' = C \cdot P(\tilde{\mathbf{U}}_{l,1}^t) \cdot P(\tilde{\mathbf{U}}_{j,1}^t)] \\
 &= C' \cdot P(\mathbf{u}_{l,t} = i | \tilde{\mathbf{U}}_{l,1}^t) \cdot \left[ \frac{P(\mathbf{u}_{l,t} = i, \mathbf{u}_{j,t} = j)}{P(\mathbf{u}_{l,t} = i) \cdot P(\mathbf{u}_{j,t} = j)} \right] \cdot P(\mathbf{u}_{j,t} = j | \tilde{\mathbf{U}}_{j,1}^t) \tag{4.5}
 \end{aligned}$$

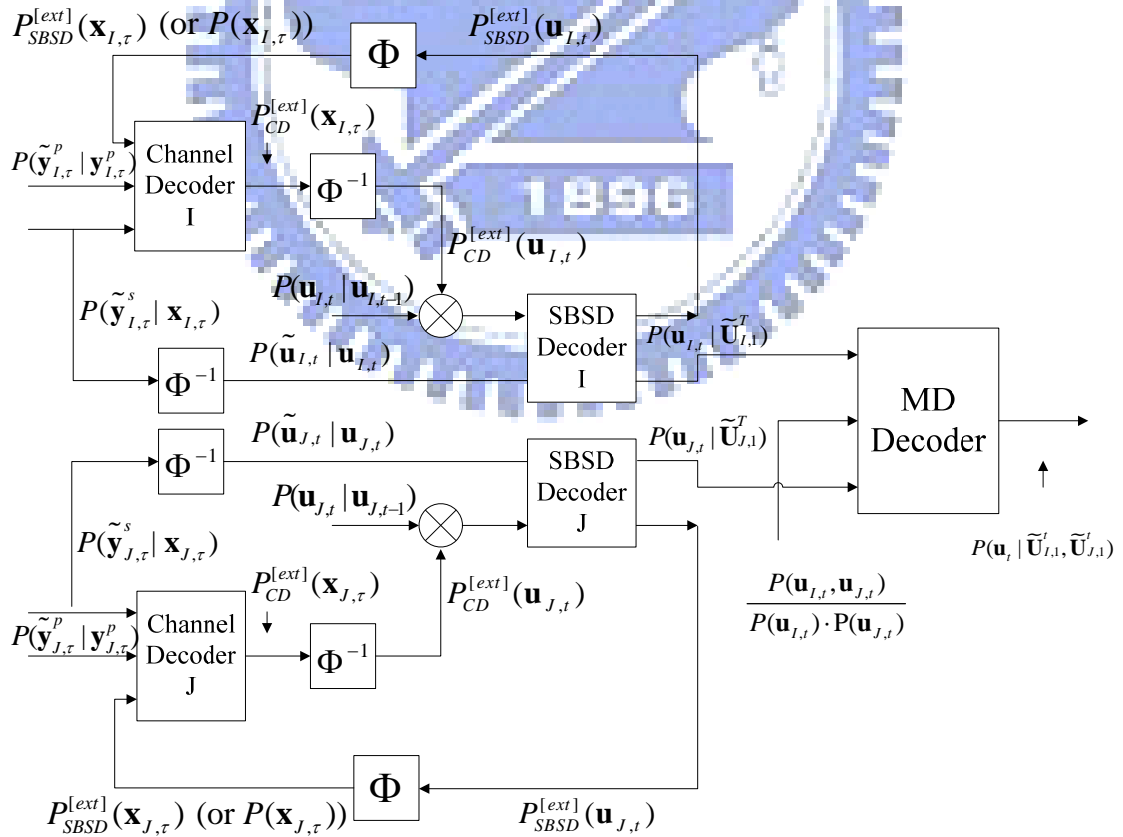


圖 4.1 平行式疊代解碼方塊圖



如圖 4.1 所示，利用前一級訊源解碼器求得的行及列索引後驗機率，配合索引事前機率與索引結合機率等資訊，計算其量化索引的後驗機率值，最後依 MMSE 估算準則來重建其輸出的訊號值。

### 4.1.3 合併式疊代解碼架構

為了簡化解碼演算法的推導，在這一小節當中我們也只呈現單一通道的解碼流程。而傳送端的模型在第二章已介紹，在此則不多加贅述。接收端針對白高斯雜訊干擾的序列，根據(4.3)與(4.4)公式分別計算其通道相關資訊的機率值  $P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau})$  與  $P(\tilde{\mathbf{y}}_{I,\tau}^p | \mathbf{y}_{I,\tau}^p)$ 。

圖 4.2 為合併式解碼的系統方塊圖。通道解碼器依循前一小節所述的解碼方式，使用敘述索引事前消息  $P(\mathbf{x}_{I,\tau} = i)$  或  $P_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau})$  整合通道相關資訊  $P(\tilde{\mathbf{y}}_{I,\tau}^p | \mathbf{x}_{I,\tau} = i)$  及  $P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau} = i)$ ，計算其額外訊息  $P_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i)$ 。再配合量化索引的一階事前消息  $P(\mathbf{u}_t = i | \mathbf{u}_{t-1})$ ，以及經過反交錯器的通道相關資訊  $P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i)$ ，一併輸入多重敘述的訊源解碼器。有別於平行式架構，合併式架構中的多重敘述訊源解碼器不僅接收來自 I 通道解碼之後的資訊，也收到 J 通道解碼過後的訊息。同時利用雙通道資訊所計算出來的訊源解碼額外訊息  $P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i)$ ，可以預期將比只使用單一通道資訊估算所得的額外訊息準確。更重要地是，訊源事前消息是來自量化索引  $\mathbf{u}_t$  的相關性，而非平行式架構中的敘述索引  $\mathbf{u}_{I,t}$  或

$\mathbf{u}_{J,t}$  的相關性，此外，因為合併式架構只使用一個結合式的多重敘述訊源解碼器，我們可以省卻一個解碼步驟而直接算出量化索引的後驗機率  $P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{1,1}^t, \tilde{\mathbf{U}}_{J,1}^t)$ ，其詳細的推導將在 4.3 節描述。多重敘述訊源解碼器輸出的額外訊息  $P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i)$ ，經交錯器後可用於下一次疊代的過程中取代舊的事前消息  $P(\mathbf{x}_{I,\tau} = i)$ ，而進一步加強通道解碼器的效能。

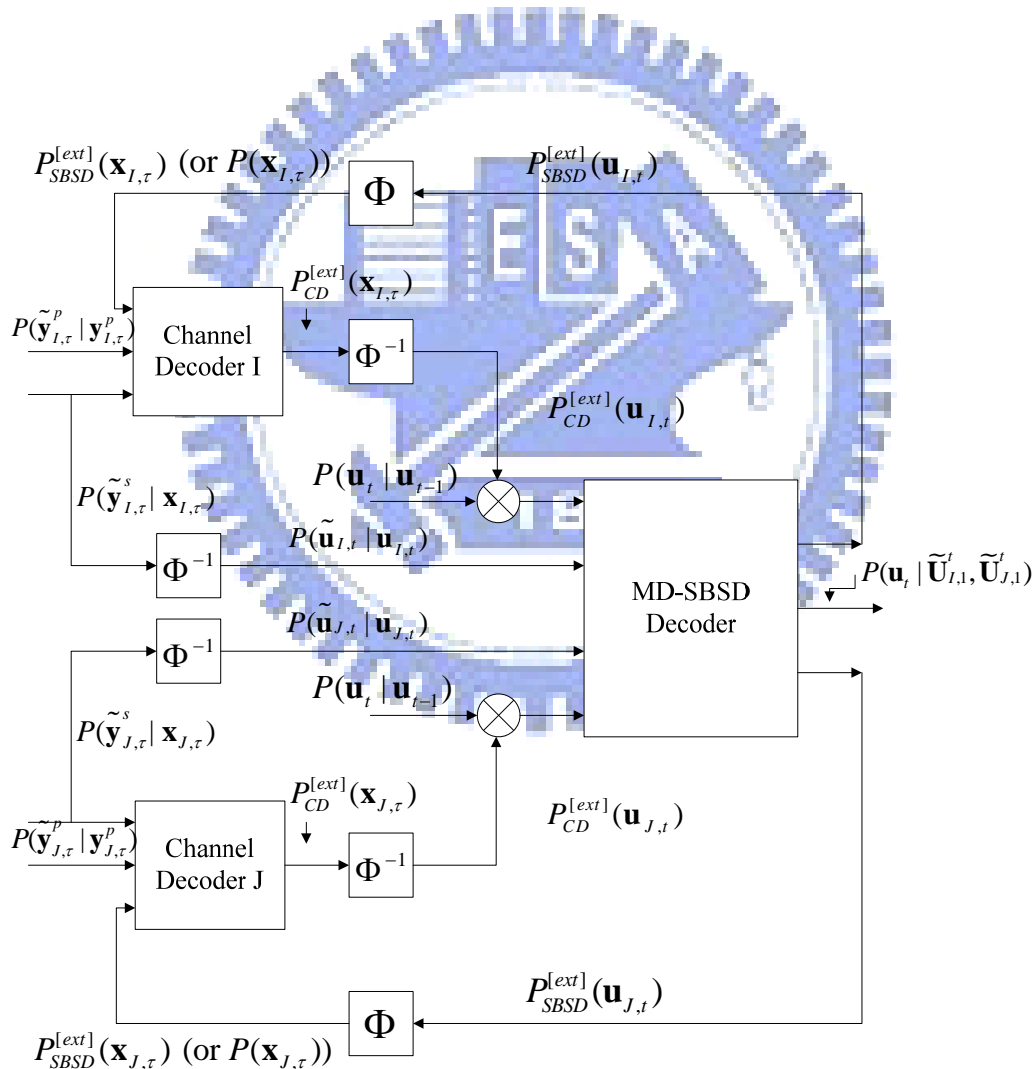


圖 4.2 合併式疊代解碼方塊圖

## 4.2 通道解碼額外訊息的推導

平行式以及合併式兩種疊代解碼架構，都採用相同的通道解碼機制，不同的是訊源解碼機制。傳統的 BCJR 演算法是在收到每個區塊的位元序列之後，估算出每個位元出現” 1” 或” 0” 的機率，再以此判斷資訊位元的傳輸值。而本章節所提出的系統是以索引層級為基本單位，因此必須使用修正後的 BCJR 演算法作解碼處理，索引層級 BCJR 演算法的相關探討可參考[6]。

索引層級 BCJR 演算法主要是計算每個敘述索引值的後驗機率，以及通道編碼的額外訊息。若令  $s_\tau$  代表通道解碼器的狀態，配置矩陣中列索引的後驗機率可展開得：

$$P(\mathbf{x}_{l,\tau} = i | \tilde{\mathbf{Y}}_{l,1}^{\hat{\tau}}) = C \cdot P(\mathbf{x}_{l,\tau} = i, \tilde{\mathbf{Y}}_{l,1}^{\hat{\tau}}) = C \cdot \sum_{s_\tau} P(\mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{Y}}_{l,1}^{\hat{\tau}}) \quad (4.6)$$

而其中

$$\begin{aligned} & P(\mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{Y}}_{l,1}^{\hat{\tau}}) \\ &= P(\mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{Y}}_{l,1}^\tau) \cdot P(\tilde{\mathbf{Y}}_{l,\tau+1}^{\hat{\tau}} | \mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{Y}}_{l,1}^\tau) \\ &\triangleq \alpha_\tau^i(s_\tau) \cdot \beta_\tau^i(s_\tau) \end{aligned} \quad (4.7)$$

進一步推導前向機率  $\alpha_\tau^i(s_\tau)$  與後向機率  $\beta_\tau^i(s_\tau)$  的遞迴公式：

$$\begin{aligned} \alpha_\tau^i(s_\tau) &= P(\mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{Y}}_{l,1}^\tau) \\ &= \sum_{s_{\tau-1}} \sum_k P(\mathbf{x}_{l,\tau} = i, s_\tau, \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^\tau) \\ &= \sum_{s_{\tau-1}} \sum_k P(\mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \cdot P(\mathbf{x}_{l,\tau} = i, s_\tau, \tilde{\mathbf{y}}_{l,\tau} | \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \end{aligned}$$

$$= \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) \quad (4.8)$$

$$\beta_{\tau-1}^i(s_{\tau-1}) = \sum_{s_{\tau}} \sum_k \beta_{\tau}^k(s_{\tau}) \cdot \gamma_{k,i}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) \quad (4.9)$$

兩者皆使用到路徑轉移機率  $\gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1})$ ，因此深入推演此公式如下：

$$\begin{aligned} \gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) &= P(\mathbf{x}_{l,\tau} = i, s_{\tau}, \tilde{\mathbf{y}}_{l,\tau} | \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \\ &= P(\mathbf{x}_{l,\tau} = i | \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \cdot P(s_{\tau} | \mathbf{x}_{l,\tau} = i, \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \cdot \\ &\quad P(\tilde{\mathbf{y}}_{l,\tau} | \mathbf{x}_{l,\tau} = i, s_{\tau}, \mathbf{x}_{l,\tau-1} = k, s_{\tau-1}, \tilde{\mathbf{Y}}_{l,1}^{\tau-1}) \\ &\approx P(\mathbf{x}_{l,\tau} = i) \cdot P(s_{\tau} | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) \cdot P(\tilde{\mathbf{y}}_{l,\tau} | s_{\tau}, \mathbf{x}_{l,\tau} = i) \end{aligned} \quad (4.10)$$

其中  $P(\mathbf{x}_{l,\tau} = i)$  為敘述索引事前訊息， $P(\tilde{\mathbf{y}}_{l,\tau} | s_{\tau}, \mathbf{x}_{l,\tau} = i)$  為接收到的通道相關資訊。若令  $S(\mathbf{x}_{l,\tau} = i, s_{\tau-1})$  代表前一時刻狀態為  $s_{\tau-1}$  而輸入索引值為  $i$  的特定轉移路徑，則

$$P(s_{\tau} | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) = \begin{cases} 1, & s_{\tau} = S(\mathbf{x}_{l,\tau} = i, s_{\tau-1}) \\ 0, & \text{otherwise} \end{cases}。$$

下一個任務是嘗試從路徑轉移機率中，區隔出通道相關資訊與額外訊息，結果如下：

$$\begin{aligned} \gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) &= P(\mathbf{x}_{l,\tau} = i) \cdot P(s_{\tau} | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) \cdot P(\tilde{\mathbf{y}}_{l,\tau} | s_{\tau}, \mathbf{x}_{l,\tau} = i) \\ &\triangleq P(\mathbf{x}_{l,\tau} = i) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i) \cdot \gamma_{i,k}^{[ext]}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) \end{aligned} \quad (4.11)$$

其中同位檢查位元  $\tilde{\mathbf{y}}_{l,\tau}^p$  相關的部分為

$$\gamma_{i,k}^{[ext]}(\tilde{\mathbf{y}}_{l,\tau}, s_{\tau}, s_{\tau-1}) = P(s_{\tau} | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^p | s_{\tau}, \mathbf{x}_{l,\tau} = i) \quad (4.12)$$

將此結果代回(4.6)至(4.8)公式，敘述列索引的後驗機率可得：

$$\begin{aligned}
P(\mathbf{x}_{l,\tau} = i | \tilde{\mathbf{Y}}_{l,1}^{\hat{T}}) &= C \cdot \sum_{s_\tau} \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \cdot \gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau) \\
&= C \cdot P(\mathbf{x}_{l,\tau} = i) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i) \cdot \sum_{s_\tau} \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \cdot \gamma_{i,k}^{[ext]}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau) \\
&\triangleq C \cdot P(\mathbf{x}_{l,\tau} = i) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i) \cdot P_{CD}^{[ext]}(\mathbf{x}_{l,\tau} = i) \tag{4.13}
\end{aligned}$$

可清楚地看出通道解碼的額外訊息為

$$P_{CD}^{[ext]}(\mathbf{x}_{l,\tau} = i) = \sum_{s_\tau} \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \cdot \gamma_{i,k}^{[ext]}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau) \tag{4.14}$$

仔細觀察疊代解碼的過程，在起始疊代階段由於訊源解碼器尚未運作，通道解碼器輸入端的額外資訊只能採用事前消息  $P(\mathbf{x}_{l,\tau} = i)$ 。而在第一次疊代完成之後，即可產生訊源解碼額外訊息  $P_{SBSD}^{[ext]}(\mathbf{u}_{l,t} = i)$ ，或其反交錯處理所得的  $P_{SBSD}^{[ext]}(\mathbf{x}_{l,\tau} = i)$ 。在後續的疊代過程就是將通道解碼器輸入端的事前消息  $P(\mathbf{x}_{l,\tau} = i)$  改用較可靠的  $P_{SBSD}^{[ext]}(\mathbf{x}_{l,\tau} = i)$  取代。因此路徑轉移機率(4.11)與敘述列索引的後驗機率(4.13)就更新成以下形式：

$$\gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) = P_{SBSD}^{[ext]}(\mathbf{x}_{l,\tau} = i) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i) \cdot \gamma_{i,k}^{[ext]}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \tag{4.15}$$

$$P(\mathbf{x}_{l,\tau} = i | \tilde{\mathbf{Y}}_{l,1}^{\hat{T}}) = C \cdot P_{SBSD}^{[ext]}(\mathbf{x}_{l,\tau} = i) \cdot P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i) \cdot P_{CD}^{[ext]}(\mathbf{x}_{l,\tau} = i) \tag{4.16}$$

## 4.3 訊源解碼額外訊息的推導

### 4.3.1 平行式架構的相關推導

由於多重敘述量化器的輸出存有殘餘冗息，可用以提升訊源解碼的可靠程度，依據不同程度殘餘冗息的運用，可分成 0 階事前訊息與

1 階事前訊息的擷取。平行式解碼系統所使用的殘餘冗息，是存在於多重敘述編碼處理後矩陣的列或行索引之相關性。由於此一架構的兩個敘述各自有獨立的訊源解碼器，可直接從敘述索引的後驗機率去推導其專屬的額外訊息。

[1] 敘述索引 0 階事前訊息的運用

通道 I 的列索引後驗機率推導如下：

$$\begin{aligned}
 & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\
 &= C \cdot P(\mathbf{u}_{I,t} = i, \tilde{\mathbf{U}}_{I,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i, \tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1})
 \end{aligned}$$

基於兩通道彼此獨立與無記憶性的假設，後驗機率可簡化為：

$$\begin{aligned}
 & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t} = i) \\
 &= C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t} = i), \quad [\text{令 } C' = C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1})] \tag{4.17}
 \end{aligned}$$

除了起始疊代之外，後續疊代均可將通道解碼的額外訊息視為附加的事前消息，以增強索引值判斷的可靠性。也就是將

$$P(\mathbf{u}_{I,t} = i) \leftarrow P(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i)$$

則(4.17)更新如下：

$$\begin{aligned}
 & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\
 &= C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t} = i) \\
 &= C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{SBS D}^{[ext]}(\mathbf{u}_{I,t} = i) \tag{4.18}
 \end{aligned}$$

其中訊源解碼器產生的額外訊息為

$$P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i) = P(\mathbf{u}_{I,t} = i) \quad (4.19)$$

## [2] 敘述索引 1 階事前訊息的運用

假設相鄰兩列索引間存在相關性，可推導列索引的後驗機率如下：

$$\begin{aligned}
& P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\
&= C \cdot P(\mathbf{u}_{I,t} = i, \tilde{\mathbf{U}}_{I,1}^t) \\
&= C \cdot \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i, \mathbf{u}_{I,t-1}, \tilde{\mathbf{U}}_{I,1}^t) \\
&= C \cdot \sum_{\mathbf{u}_{I,t-1}} P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i, \mathbf{u}_{I,t-1}, \tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}, \tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}) \\
&= C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}) \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \\
&= C'' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}), \quad [\text{令 } C'' = C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1})] \circ
\end{aligned} \quad (4.20)$$

若進一步也將通道解碼的額外訊息視為附加的事前消息，則

$$P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \leftarrow P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i)$$

而更新(4.20)如下：

$$\begin{aligned}
& P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t) \\
&= C'' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \\
&= C'' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i)
\end{aligned} \quad (4.21)$$

從上述公式可以清楚得到訊源解碼的額外訊息：

$$P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i) = \sum_{\mathbf{u}_{I,t-1}} P(\mathbf{u}_{I,t} = i | \mathbf{u}_{I,t-1}) \cdot P(\mathbf{u}_{I,t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}) \quad (4.22)$$

### 4.3.2 合併式架構的相關推導

有別於平行式架構運用單一敘述索引的殘餘冗息，合併式系統則運用訊號源量化索引的殘餘冗息。然而由於此架構下只有一個訊源解碼器，我們必須先估算量化索引的後驗機率，再間接計算單一敘述的後驗機率，之後才能繼續推導個別敘述的額外訊息。

#### [1] 量化索引 0 階事前訊息的運用

首先，我們展開量化索引的後驗機率如下：

$$\begin{aligned}
 & P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C \cdot P(\mathbf{u}_t = l, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_t = l, \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})
 \end{aligned}$$

假設兩通道彼此獨立且無記憶性，量化索引的後驗機率可簡化為

$$\begin{aligned}
 & P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_t = l) \cdot P(\mathbf{u}_t = l) \\
 &= C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t} = i, \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l) \\
 &= C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l), \quad [\text{令 } C' = C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})]。
 \end{aligned} \tag{4.23}$$

為了得到單一敘述的訊源解碼額外訊息，我們進一步推導單一敘述的後驗機率，以列索引 I 為例，

$$\begin{aligned}
 & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
 &= C'' \cdot \sum_{l \in R_t} P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \quad [C'' \text{ is to normalize } \sum_i P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = 1] \\
 &= C'' \cdot \sum_{l \in R_t} C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l)
 \end{aligned}$$



$$= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot C''' \cdot \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l), \quad [\text{令 } C''' = C'' \cdot C'] \quad (4.24)$$

接著整合兩敘述經通道解碼輸出的額外訊息當作附加的事前消息，也就是將  $P(\mathbf{u}_t) \leftarrow P(\mathbf{u}_t) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j)$ ，則單一敘述的後驗機率公式(4.24)變成：

$$\begin{aligned} & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot C''' \cdot \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \\ &= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot C''' \sum_{l \in R_i} P(\mathbf{u}_t = l) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \\ &= C''' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i) \end{aligned} \quad (4.25)$$

經整理可得出單一敘述(列索引 I)的訊源解碼額外訊息為

$$P_{SBSD}^{[ext]}(\mathbf{u}_{I,t} = i) = \sum_{l \in R_i} P(\mathbf{u}_t = l) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \quad (4.26)$$

## [2] 量化索引 1 階事前訊息的運用

假設相鄰兩量化索引間存在相關性，可推導量化索引的後驗機

率：

$$\begin{aligned} & P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C \cdot P(\mathbf{u}_t = l, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C \cdot \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l, \mathbf{u}_{t-1}, \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C \cdot \sum_{\mathbf{u}_{t-1}} P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_t = l, \mathbf{u}_{t-1}, \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\mathbf{u}_t = l | \mathbf{u}_{t-1}, \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \cdot P(\mathbf{u}_{t-1}, \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= C' \cdot \sum_{\mathbf{u}_{t-1}} P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_t = l) \cdot P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \quad [C' = C \cdot P(\tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})] \\ &= C' \cdot \sum_{\mathbf{u}_{t-1}} P(\tilde{\mathbf{u}}_{I,t}, \tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{I,t} = i, \mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \end{aligned}$$

$$= C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \quad (4.27)$$

為了得到單一敘述的訊源解碼額外訊息，我們進一步找出單一敘述的後驗機率，以列索引 I 為例：

$$\begin{aligned} & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= C'' \cdot \sum_{l \in R_i} P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \quad [C'' \text{ is to normalize } \sum_i P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = 1] \\ &= C'' \cdot \sum_{l \in R_i} C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot C''' \cdot \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ & [ \text{令 } C''' = C'' \cdot C' ] \end{aligned} \quad (4.28)$$

接著整合兩敘述經通道解碼輸出的額外訊息當作附加的事前消息，也就是將  $P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \leftarrow P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j)$ 。

因此單一敘述的後驗機率(4.28)則更新為

$$\begin{aligned} & P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\ &= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot C''' \cdot \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \\ & \quad \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot C''' \cdot \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \\ & \quad \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \\ &= C''' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{SBS D}^{[ext]}(\mathbf{u}_{I,t} = i) \end{aligned} \quad (4.29)$$

從上述的公式中，可以清楚看出單一敘述(列索引 I)的訊源解碼額外訊息為

$$P_{SBS D}^{[ext]}(\mathbf{u}_{I,t} = i) = \sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \quad (4.30)$$

此外(4.27)也將更新為

$$P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = C' \cdot P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \quad (4.31)$$

#### 4.4 對數領域之索引層級合併式疊代演算法

對數相似比率函數(log-likelihood ratio, LLR)廣泛應用在疊代解碼演算法，以其有效簡化其運算處理。傳統上仍以位元層級解碼演算法的對數呈現為主，問題是在位元彼此獨立的假設之下列索引  $\mathbf{x}_{I,\tau}$  的後驗機率會有機率相乘的精度誤差與失真，因此這小節推導索引層級解碼演算法的對數運算式。對數領域索引層級疊代演算法的相關推導可以參考文章[6]。

接下來為簡化對數解碼演算法的呈現，我們只針對列索引來推導相關公式，定義列索引的事前消息對數值以自然底數表示，如下：

$$L(\mathbf{x}_{I,\tau} = i) = \log \frac{P(\mathbf{x}_{I,\tau} = i)}{P(\mathbf{x}_{I,\tau} = 0)}, i = 1, 2, \dots, 2^M - 1 \quad (4.32)$$

合併式架構在機率領域的疊代解碼流程如圖 4.2 所示，而其對數領域的流程則如圖 4.3 所示，使用對數表示式可完整地呈現訊號的相關消息，再者可消去機率公式的相同項，並將機率的乘法運算以對數加法

取代之，大大地簡化解碼演算法的運算複雜度。對數訊息交換的區塊

圖如圖 4.3 所示。舉例而言，通道相關訊息的機率公式(4.3)以對數

呈現可簡化為  $L_{ch}(\mathbf{u}_{l,t} = i) = L_{ch}(\mathbf{x}_{l,\tau} = i)$ ，

$$\begin{aligned}
 L_{ch}(\mathbf{x}_{l,\tau} = i) &= \log \frac{P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = i)}{P(\tilde{\mathbf{y}}_{l,\tau}^s | \mathbf{x}_{l,\tau} = 0)} \\
 &= \log \left\{ \exp\left[-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_{l,\tau}^s - \mathbf{x}_{l,\tau}\|^2\right]_{\mathbf{x}_{l,\tau}=i} \right\} - \log \left\{ \exp\left[-\frac{E_s}{N_0} \|\tilde{\mathbf{y}}_{l,\tau}^s - \mathbf{x}_{l,\tau}\|^2\right]_{\mathbf{x}_{l,\tau}=0} \right\} \\
 &= \frac{L_c}{2} \cdot \left[ (\tilde{\mathbf{y}}_{l,\tau}^s \cdot \mathbf{x}_{l,\tau})_{\mathbf{x}_{l,\tau}=i} - (\tilde{\mathbf{y}}_{l,\tau}^s \cdot \mathbf{x}_{l,\tau})_{\mathbf{x}_{l,\tau}=0} \right], \quad i = 1, 2, \dots, 2^M - 1
 \end{aligned} \tag{4.33}$$

其中  $L_c = 4E_s/N_0$ ， $(\tilde{\mathbf{y}}_{l,\tau}^s \cdot \mathbf{x}_{l,\tau})$  代表實數向量  $\tilde{\mathbf{y}}_{l,\tau}^s$  與二位元向量  $\mathbf{x}_{l,\tau}$  的內

積。在疊代演算過程中，系統效能評估的 MMSE 法則需要量化索引的

後驗機率  $P(\mathbf{u}_l = l | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1})$ ，這部分可由其對數值  $L(\mathbf{u}_l = l | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1})$  轉換

成機率形式如下：

$$P(\mathbf{u}_l = l | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1}) = \begin{cases} \frac{1}{1 + \sum_{m=1}^{Q-1} \exp[L(\mathbf{u}_l = m | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1})]}, & l = 0 \\ \frac{\exp[L(\mathbf{u}_l = l | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1})]}{1 + \sum_{m=1}^{Q-1} \exp[L(\mathbf{u}_l = m | \tilde{\mathbf{U}}'_{l,1}, \tilde{\mathbf{U}}'_{j,1})]}, & l = 1, 2, \dots, Q-1 \end{cases} \tag{4.34}$$

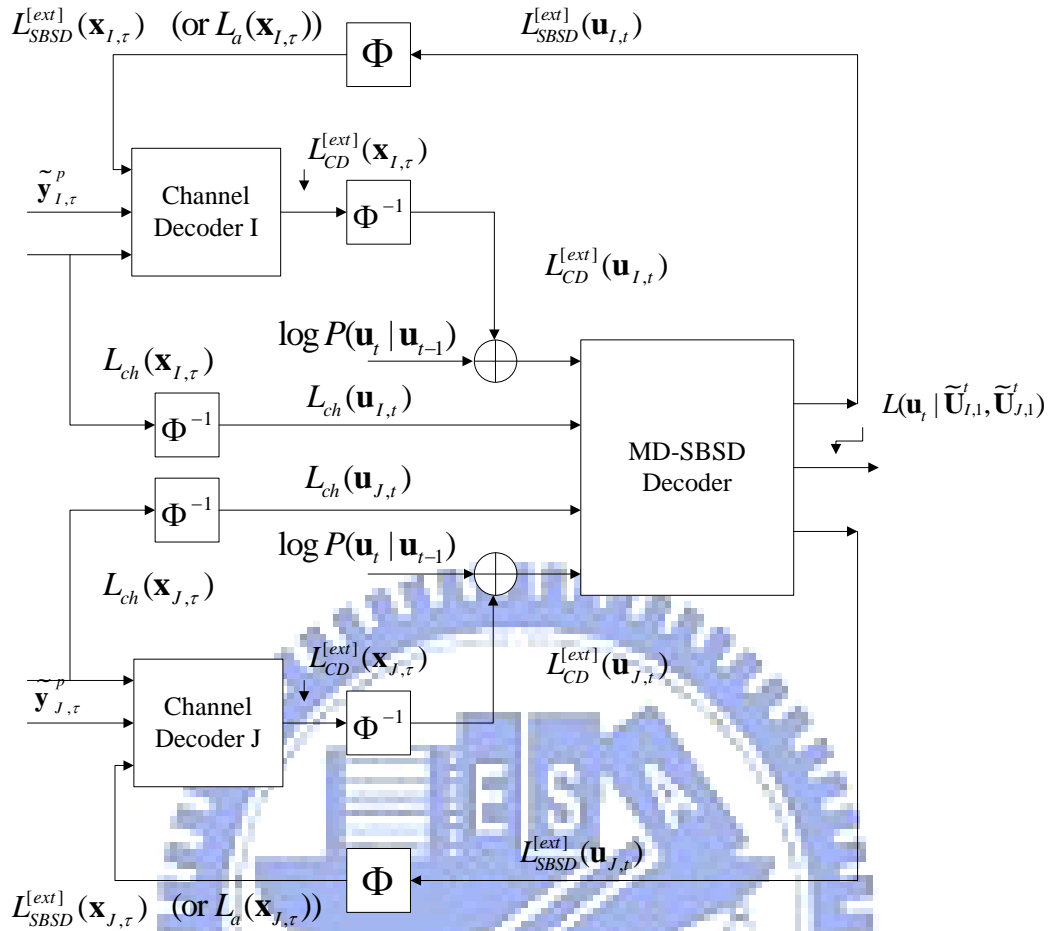


圖 4.3 合併式疊代解碼方塊圖(Log-domain)

#### 4.4.1 通道解碼額外訊息的推導

為了推導對數領域的索引層級 BCJR 演算法，我們必須定義前向機率、後向機率與路徑轉移機率的對數式。首先從列索引  $\mathbf{x}_{I,\tau}$  經通道解碼的後驗對數值開始推導，定義如下：

$$L(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}}) \triangleq \log \frac{P(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}})}{P(\mathbf{x}_{I,\tau} = 0 | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}})} = \log \frac{\sum_{s_\tau} \alpha_\tau^i(s_\tau) \cdot \beta_\tau^i(s_\tau)}{\sum_{s_\tau} \alpha_\tau^0(s_\tau) \cdot \beta_\tau^0(s_\tau)}$$

$$= \log \frac{\sum_{s_\tau} \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{i,k}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau)}{\sum_{s_\tau} \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{0,k}(\tilde{\mathbf{y}}_{l,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^0(s_\tau)}, i = 1, 2, \dots, 2^M - 1 \quad (4.35)$$

由於無法從(4.34)直接推導得出最終的對數版本，我們需引入基本的對數數學式輔助後續的推導過程，公式呈現如下：

$$\begin{aligned} \max^*(a, b) &\equiv \log(e^a + e^b) = \max(a, b) + \log(1 + e^{-|a-b|}) \\ \max^*(a, b, c) &\equiv \log(e^a + e^b + e^c) = \max^*[\max^*(a, b), c] \end{aligned} \quad (4.36)$$

套用上述公式於(4.8)與(4.9)上，可以分別推導出前向機率與後向機率的對數表示式為

$$\begin{aligned} \bar{\alpha}_\tau^i(s_\tau) &\triangleq \log \alpha_\tau^i(s_\tau) \\ &= \log P(\mathbf{x}_{l,\tau} = i) + \log P(\tilde{\mathbf{y}}_{l,\tau} | \mathbf{x}_{l,\tau}, s_\tau) + \log \left\{ \sum_{s_{\tau-1}} P(s_\tau | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) \sum_k \exp[\log \alpha_{\tau-1}^k(s_{\tau-1})] \right\} \\ &= \log P(\mathbf{x}_{l,\tau} = i) + \log P(\tilde{\mathbf{y}}_{l,\tau} | \mathbf{x}_{l,\tau}, s_\tau) + \max_{s_{\tau-1}}^* \left\{ \log P(s_\tau | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) + \max_k^* [\bar{\alpha}_{\tau-1}^k(s_{\tau-1})] \right\} \end{aligned} \quad (4.37)$$

其中路徑指標函數的對數定義成

$$\log P(s_\tau | \mathbf{x}_{l,\tau} = i, s_{\tau-1}) = \begin{cases} 0, & s_\tau = S(\mathbf{x}_{l,\tau} = i, s_{\tau-1}) \\ -\infty, & \text{otherwise} \end{cases}$$

同理可推得後向機率的對數值為

$$\begin{aligned} \bar{\beta}_\tau^i(s_\tau) &\triangleq \log \beta_\tau^i(s_\tau) \\ &= \log \sum_k P(\mathbf{x}_{l,\tau+1} = k) \sum_{s_{\tau+1}} \beta_{\tau+1}^k(s_{\tau+1}) \cdot P(s_{\tau+1} | \mathbf{x}_{l,\tau+1} = k, s_\tau) \cdot P(\tilde{\mathbf{y}}_{l,\tau+1} | s_{\tau+1}, \mathbf{x}_{l,\tau+1} = k) \\ &= \max_k^* \left\{ \log P(\mathbf{x}_{l,\tau+1} = k) + \max_{s_{\tau+1}}^* \left[ \bar{\beta}_{\tau+1}^k(s_{\tau+1}) + \log P(s_{\tau+1} | \mathbf{x}_{l,\tau+1} = k, s_\tau) + \right. \right. \\ &\quad \left. \left. \log P(\tilde{\mathbf{y}}_{l,\tau+1} | s_{\tau+1}, \mathbf{x}_{l,\tau+1} = k) \right] \right\} \end{aligned} \quad (4.38)$$

接著將(4.36)也套用在(4.35)公式上，可以得到通道解碼的後驗對數

$$\begin{aligned}
L(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}}) &= \log \left\{ \sum_{s_\tau} \exp \left[ \log \left( \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{i,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau) \right) \right] \right\} \\
&\quad - \log \left\{ \sum_{s_\tau} \exp \left[ \log \left( \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{0,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^0(s_\tau) \right) \right] \right\} \\
&= \max_{s_\tau}^* \left[ \log \left( \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{i,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^i(s_\tau) \right) \right] \\
&\quad - \max_{s_\tau}^* \left[ \log \left( \sum_{s_{\tau-1}} \sum_k \alpha_{\tau-1}^k(s_{\tau-1}) \gamma_{0,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) \cdot \beta_\tau^0(s_\tau) \right) \right] \\
&= \max_{s_\tau}^* \left\{ \max_{s_{\tau-1}}^* \left[ \max_k^* \left( \bar{\alpha}_{\tau-1}^k(s_{\tau-1}) + \bar{\gamma}_{i,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) + \bar{\beta}_\tau^i(s_\tau) \right) \right] \right\} \\
&\quad - \max_{s_\tau}^* \left\{ \max_{s_{\tau-1}}^* \left[ \max_k^* \left( \bar{\alpha}_{\tau-1}^k(s_{\tau-1}) + \bar{\gamma}_{0,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) + \bar{\beta}_\tau^0(s_\tau) \right) \right] \right\}
\end{aligned} \tag{4.39}$$

其中由式(4.11)可知，

$$\gamma_{i,k}(\tilde{\mathbf{y}}_{I,\tau}, s_\tau, s_{\tau-1}) = \log P(\mathbf{x}_{I,\tau} = i) + \log P(s_\tau | \mathbf{x}_{I,\tau} = i, s_{\tau-1}) + \log P(\tilde{\mathbf{y}}_{I,\tau} | s_\tau, \mathbf{x}_{I,\tau} = i) , \tag{4.40}$$

而且  $\log P(s_\tau | \mathbf{x}_{I,\tau} = i, s_{\tau-1}) = \begin{cases} 0, & s_\tau = S(\mathbf{x}_{I,\tau} = i, s_{\tau-1}) \\ -\infty, & \text{otherwise} \end{cases}$

此外，我們可將後驗對數拆解成事前消息、通道相關資訊與解碼額外訊息三個對數和，如下呈現：

$$\begin{aligned}
L(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}}) &= \log \frac{P(\mathbf{x}_{I,\tau} = i)}{P(\mathbf{x}_{I,\tau} = 0)} + \log \frac{P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau} = i)}{P(\tilde{\mathbf{y}}_{I,\tau}^s | \mathbf{x}_{I,\tau} = 0)} + \log \frac{P_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i)}{P_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = 0)} \\
&\triangleq L_a(\mathbf{x}_{I,\tau} = i) + L_{ch}(\mathbf{x}_{I,\tau} = i) + L_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i)
\end{aligned} \tag{4.41}$$

由此觀之，通道解碼額外訊息的對數值就可由下式計算：

$$L_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i) = L(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{\tau}}) - L_a(\mathbf{x}_{I,\tau} = i) - L_{ch}(\mathbf{x}_{I,\tau} = i) \tag{4.42}$$

更進一步分析，在下一次的疊代過程中，通道解碼器會接收來自訊源解碼器的額外消息，定義如下：

$$L_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau} = i) \triangleq \log \frac{P_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau} = i)}{P_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau} = 0)}, i = 1, 2, \dots, 2^M - 1 \quad (4.43)$$

若將之用於取代  $\log P(\mathbf{x}_{I,\tau} = i)$ ，則前向機率與後向機率的對數式(4.37)

及(4.38)將更新為

$$\begin{aligned} \bar{\alpha}_\tau^i(s_\tau) = & L_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau} = i) + \log P(\tilde{\mathbf{y}}_{I,\tau} | \mathbf{x}_{I,\tau}, s_\tau) + \max_{s_{\tau-1}}^* \left\{ \log P(s_\tau | \mathbf{x}_{I,\tau} = i, s_{\tau-1}) + \right. \\ & \left. \max_k^* [\bar{\alpha}_{\tau-1}^k(s_{\tau-1})] \right\} \end{aligned} \quad (4.44)$$

$$\begin{aligned} \bar{\beta}_\tau^j(s_\tau) = & \max_k^* \left\{ L_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau+1} = k) + \max_{s_{\tau+1}}^* \left[ \bar{\beta}_{\tau+1}^k(s_{\tau+1}) + \log P(s_{\tau+1} | \mathbf{x}_{I,\tau+1} = k, s_\tau) + \right. \right. \\ & \left. \left. \log P(\tilde{\mathbf{y}}_{I,\tau+1} | s_{\tau+1}, \mathbf{x}_{I,\tau+1} = k) \right] \right\} \end{aligned} \quad (4.45)$$

同理，通道解碼後驗對數的疊代更新可表示成以下形式

$$L(\mathbf{x}_{I,\tau} = i | \tilde{\mathbf{Y}}_{I,1}^{\hat{t}}) = L_{SBSD}^{[ext]}(\mathbf{x}_{I,\tau} = i) + L_{ch}(\mathbf{x}_{I,\tau} = i) + L_{CD}^{[ext]}(\mathbf{x}_{I,\tau} = i) \quad (4.46)$$

#### 4.4.2 訊源解碼額外訊息的推導

類似通道解碼對數式的推導過程，首先分別定義  $i_0$  及  $j_0$  為量化索引  $\mathbf{u}_t = 0$  所對應的列及行索引，亦即  $(i_0, j_0) = 0$ 。根據量化索引一階事前訊息的公式(4.27)，則訊源解碼的量化索引後驗對數式為

$$\begin{aligned} L(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) &= \log \frac{P(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)}{P(\mathbf{u}_t = 0 | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)} \\ &= \log \frac{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})}{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i_0) \cdot P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j_0) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = 0 | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})} \end{aligned} \quad (4.47)$$

接著附加敘述列索引與行索引的通道解碼額外訊息作為事前訊息，



$P(\mathbf{u}_t | \mathbf{u}_{t-1}) \leftarrow P(\mathbf{u}_t | \mathbf{u}_{t-1}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j)$ ，(4.47)將更新成

$$\begin{aligned}
& L(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
&= \log \frac{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i)}{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i_0)} + \log \frac{P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i)}{P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i_0)} + \log \frac{P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j)}{P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j_0)} + \\
& \log \frac{P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j)}{P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j_0)} + \log \frac{\sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})}{\sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = 0 | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})} \quad (4.48)
\end{aligned}$$

進一步套用(4.36)公式，可將(4.48)簡化為

$$\begin{aligned}
& L(\mathbf{u}_t = l | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) \\
&= \left\{ L_{ch}(\mathbf{u}_{I,t} = i) + L_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) + L_{ch}(\mathbf{u}_{J,t} = j) + L_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \right. \\
& \quad \left. + \max_{\mathbf{u}_{t-1}}^* \left[ \log P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) + L(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \right] \right\} \\
& - \left\{ L_{ch}(\mathbf{u}_{I,t} = i_0) + L_{CD}^{[ext]}(\mathbf{u}_{I,t} = i_0) + L_{ch}(\mathbf{u}_{J,t} = j_0) + L_{CD}^{[ext]}(\mathbf{u}_{J,t} = j_0) \right. \\
& \quad \left. + \max_{\mathbf{u}_{t-1}}^* \left[ \log P(\mathbf{u}_t = 0 | \mathbf{u}_{t-1}) + L(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1}) \right] \right\} \quad (4.49)
\end{aligned}$$

此外，我們也需要計算列及行索引的訊源解碼額外消息，前者可以根據(4.29)推導而得列索引的後驗對數式如下

$$\begin{aligned}
& L(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t) = \log \frac{P(\mathbf{u}_{I,t} = i | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)}{P(\mathbf{u}_{I,t} = 0 | \tilde{\mathbf{U}}_{I,1}^t, \tilde{\mathbf{U}}_{J,1}^t)} \\
&= \log \frac{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = i)}{P(\tilde{\mathbf{u}}_{I,t} | \mathbf{u}_{I,t} = 0)} + \log \frac{P_{CD}^{[ext]}(\mathbf{u}_{I,t} = i)}{P_{CD}^{[ext]}(\mathbf{u}_{I,t} = 0)} + \\
& \log \frac{\sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})}{\sum_{l \in R_0} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j^{(0)}) \cdot P_{CD}^{[ext]}(\mathbf{u}_{J,t} = j^{(0)}) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{I,1}^{t-1}, \tilde{\mathbf{U}}_{J,1}^{t-1})} \\
&= L_{ch}(\mathbf{u}_{I,t} = i) + L_{CD}^{[ext]}(\mathbf{u}_{I,t} = i) + L_{SBS}^{[ext]}(\mathbf{u}_{I,t} = i) \quad (4.50)
\end{aligned}$$

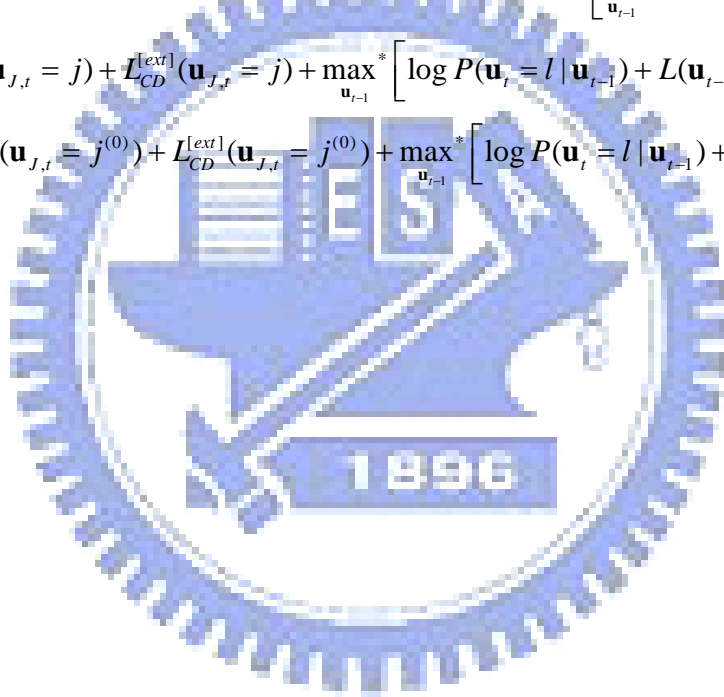
其中  $\mathbf{u}_{J,t} = j^{(0)}$  代表  $R_0$  集合所屬的量化索引之行索引，亦即  $(0, j^{(0)}) \in R_0$ 。

從(4.50)中可清楚看出訊源解碼的額外訊息為

$$L_{SBS D}^{[exr]}(\mathbf{u}_{l,t} = i) = \log \frac{\sum_{l \in R_i} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) \cdot P_{CD}^{[exr]}(\mathbf{u}_{J,t} = j) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1})}{\sum_{l \in R_0} P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j^{(0)}) \cdot P_{CD}^{[exr]}(\mathbf{u}_{J,t} = j^{(0)}) \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1})} \quad (4.51)$$

然後進一步套用(4.36)公式，可簡化(4.51)上述公式的運算如下呈現：

$$\begin{aligned} & L_{SBS D}^{[exr]}(\mathbf{u}_{l,t} = i) \\ &= \max_{l \in R_i}^* \left\{ \log P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j) + \log P_{CD}^{[exr]}(\mathbf{u}_{J,t} = j) + \log \left[ \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1}) \right] \right\} \\ & \quad - \max_{l \in R_0}^* \left\{ \log P(\tilde{\mathbf{u}}_{J,t} | \mathbf{u}_{J,t} = j^{(0)}) + \log P_{CD}^{[exr]}(\mathbf{u}_{J,t} = j^{(0)}) + \log \left[ \sum_{\mathbf{u}_{t-1}} P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) \cdot P(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1}) \right] \right\} \\ &= \max_{l \in R_i}^* \left\{ L_{ch}(\mathbf{u}_{J,t} = j) + L_{CD}^{[exr]}(\mathbf{u}_{J,t} = j) + \max_{\mathbf{u}_{t-1}}^* \left[ \log P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) + L(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1}) \right] \right\} \\ & \quad - \max_{l \in R_0}^* \left\{ L_{ch}(\mathbf{u}_{J,t} = j^{(0)}) + L_{CD}^{[exr]}(\mathbf{u}_{J,t} = j^{(0)}) + \max_{\mathbf{u}_{t-1}}^* \left[ \log P(\mathbf{u}_t = l | \mathbf{u}_{t-1}) + L(\mathbf{u}_{t-1} | \tilde{\mathbf{U}}_{l,1}^{t-1}, \tilde{\mathbf{U}}_{j,1}^{t-1}) \right] \right\} \end{aligned} \quad (4.52)$$



## 第五章 實驗模擬與結果分析

前面章節介紹了位元層級與索引層級的多重敘述傳輸架構，在位元層級之下所屬的平行式與交叉式解碼架構中，交叉式演算法有使用到雙通道敘述之間的相關性資訊，其效能預期會優於沒有使用此一訊息的平行式演算法。至於索引層級的平行式與合併式解碼架構，則強調相鄰索引之間相關性強弱的重要性。合併式解碼使用量化索引的一階事前訊息，而平行式解碼則使用敘述索引的一階事前訊息。由於敘述索引是量化索引經過索引配置後產生的資料，其一階事前相關性將少於量化索引，因此合併式架構的效能將優於平行式架構。在本章中我們將分成三個小節進行系統的模擬實驗與分析討論，來驗證不同解碼架構的效能差別。前兩小節分別針對第三章與第四章所提出的演算法執行模擬實驗，而第三小節則將解碼演算法應用在寬頻語音傳輸，驗證系統效能趨勢與其實用性。

### 5.1 位元層級平行式與交叉式解碼之模擬

#### 5.1.1 系統模擬之步驟說明

平行式與交叉式解碼架構的傳輸端中，訊號源  $u_i$  皆採用一階自迴歸處理來源，變異數設定為  $\sigma_u^2 = 1$ ，且其相關因子設成  $\rho = 0.95$ 。

訊號源關係式表示為

$$u_i = \rho \cdot u_{i-1} + n_i \quad (5.1)$$

其中  $n_i$  為平均值 0 的白色高斯雜訊。

每次模擬中訊號源產生 30000 個取樣值，通過 Q 階的最佳化純量  
量化器 Lloyd-Max Quantizer [10]，位元映射設定為自然對應方式  
(Natural bit-mapping)。針對每個輸出的量化索引值使用 BSA 索引  
配置法產生兩個 M=3 位元的敘述索引，兩敘述索引位元串各經由相同  
的區塊交錯器(block interleaver)重新排列，區塊大小設定為

3x50。然後皆通過一樣的迴旋碼編碼器，編碼率為  $R_c = \frac{1}{2}$ ，記憶體長  
度  $\nu = 2$ ，編碼結構  $G = [1, 1+D^2 / 1+D+D^2]$ ，得到兩倍長度的位元  
序列，接著經過二位元相位鍵移調變後送入白高斯雜訊通道。

[平行式解碼器]

接收端收到來自通道雜訊干擾的兩個實數序列，而接收的資訊是  
採用分開獨立解碼的模式。對通道 I 而言，通道解碼器使用通道相關  
資訊與大量訓練而得的位元事前消息作為解碼資訊，然後基於 BCJR  
演算法求得出通道解碼額外訊息。該額外訊息以及敘述索引 I 的一階  
事前消息，將整合於訊源解碼器輸入端作事前資訊使用，輔以通道相  
關資訊求得訊源解碼額外訊息。此額外訊息再傳回到通道解碼器，作  
為下一次疊代的事前訊息使用，就如同渦輪碼資訊交換的疊代解碼過

程。疊代次數完成時可以計算出敘述索引的後驗機率，然後使用 MAP 法則得到最佳敘述索引，接著反查索引配置矩陣表得出最近似的訊號源數值。

### [交叉式解碼器]

接收端收到雜訊干擾後的兩個實數序列，接收資訊會個別經過通道解碼器解碼處理，之後將產生的通道解碼額外訊息送進訊源解碼器輸入端。訊源解碼器不僅使用來自本身敘述的相關資訊，還使用另一敘述的通道相關資訊以及雙敘述之間的相關訊息，預期可以增加訊源解碼的可靠性並提升整體系統效能。此外，訊源解碼輸出的額外訊息也同樣傳回通道解碼器作疊代解碼。疊代次數完成時可以計算出敘述索引的後驗機率，然後使用 MAP 法則得到最佳敘述索引，接著反查索引配置矩陣表得出最近似的訊號源數值。

### 5.1.2 結果分析

令 MDQB1 與 MDQB2 分別代表位元層級平行式與交叉式解碼系統， $(Q, M)=(16, 3)$ 及 $(32, 3)$ 的實驗結果如圖 5.1、圖 5.2 所示，系統效能評估以及參數訊雜比(parameter SNR)作為衡量標準，圖中縱軸代表系統的參數訊雜比，橫軸則是代表通道環境的訊雜比。

在 0 次疊代時，兩解碼方式皆不考慮通道解碼以及訊源解碼，因

此具有相同的系統效能曲線。而在 1 次疊代之後，兩種解碼效能都有大幅的提升，且效能提升趨勢隨著通道環境逐漸變好而遞增。由於交叉式解碼模式額外引入另一敘述的通道資訊與雙敘述相關資訊，所以效能提升優於平行式解碼，甚至在通道環境極差時，效能曲線仍高過平行式解碼 1.49dB(圖 5.1)與 1.3dB(圖 5.2)。當 2 次疊代完成之後，效能上的提升有限，但仍可以明顯看出曲線變化；而在圖 5.1 與圖 5.2 中極差的通道環境，交叉式解碼曲線分別勝過平行式解碼曲線 1.49dB 與 1.27dB。當疊代次數達 3 次時，平行式解碼的效能曲線幾乎與 2 次疊代重合，代表傳送資訊的殘餘冗息已被完全挖掘，效能無法進一步地提升，然而交叉式解碼的效能曲線仍有 0.1~0.3dB 的提升空間，也快接近飽和狀態。圖 5.1 中的平行式與交叉式的系統效能，於通道環境設定 2dB 時皆達 20.22dB 的理論上限值，而圖 5.2 中兩者的解碼效能則趨向 26.01dB 的理論上限值。

### 5.1.3 對角線索引配置法之實驗模擬

在此繼續針對兩種不同的解碼演算法作實驗模擬，而實驗設定與前述幾乎一樣，唯一不同的地方在於索引配置的方法。有別於前採用的 BSA 演算法，在此考慮採用對角線索引配置方法。沿矩陣對角線作索引配置是前人經驗結果[2]，是已知 MDSQ 索引配置的最佳化設

計。本論文主要重心在探討多重敘述純量量化與其疊代解碼架構，因此後續章節的實驗模擬仍以 BSA 矩陣配置法為主。

0 次疊代不考慮通道解碼與訊源解碼，平行式與交叉式解碼具有一樣的系統效能曲線。1 次疊代有通道與訊源解碼器的作用，平行式與交叉式系統效能曲線皆有大幅的提升，而在通道環境很差時，交叉式效能曲線仍勝過平行式效能曲線 1.18dB 與 0.48dB，分別從圖 5.3 與圖 5.4 中可以看出。2 次疊代之後，兩種解碼效能曲線上升的幅度並不高，這是因為殘餘冗息逐漸被挖掘用完，而從圖 5.3 與圖 5.4 中，在通道環境極差時交叉式效能曲線仍分別高過平行式效能曲線 1.34dB 與 0.55dB。當達 3 次疊代解碼之後，平行式解碼約有 0.1dB~0.46dB 的效能提升，交叉式解碼則有 0.1dB~0.53dB 的提升空間，兩效能曲線皆接近飽和狀態。在圖 5.3 和圖 5.4 中，系統效能曲線在通道環境為 2dB 時皆趨近於 20.22dB 與 26.01dB 的理論上限。

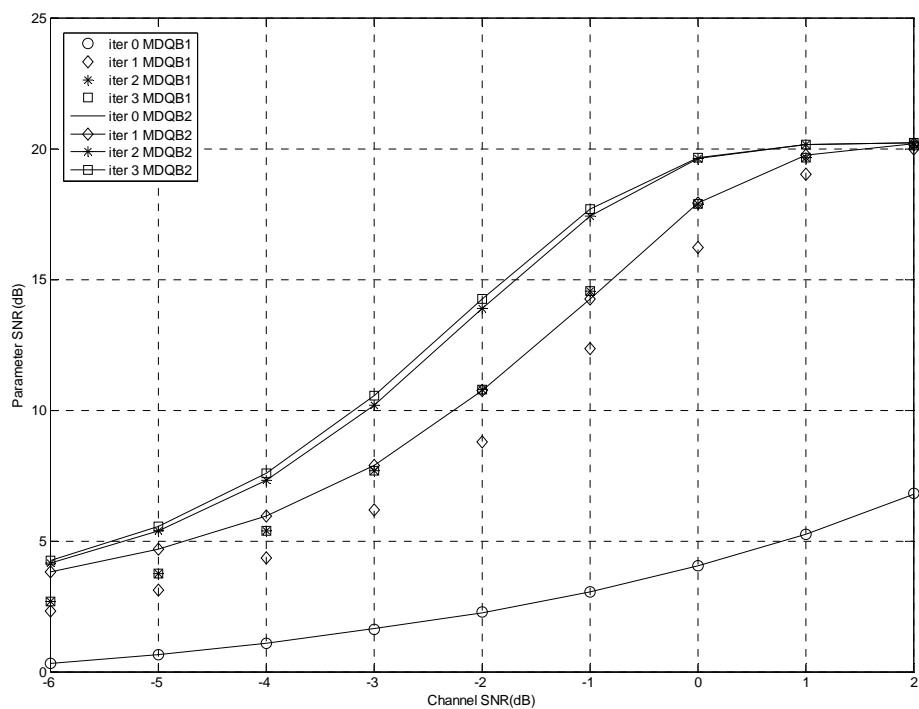


圖 5.1 位元層級的疊代解碼結果 $[(Q, M)=(16, 3)$ 且 BSA 配置]

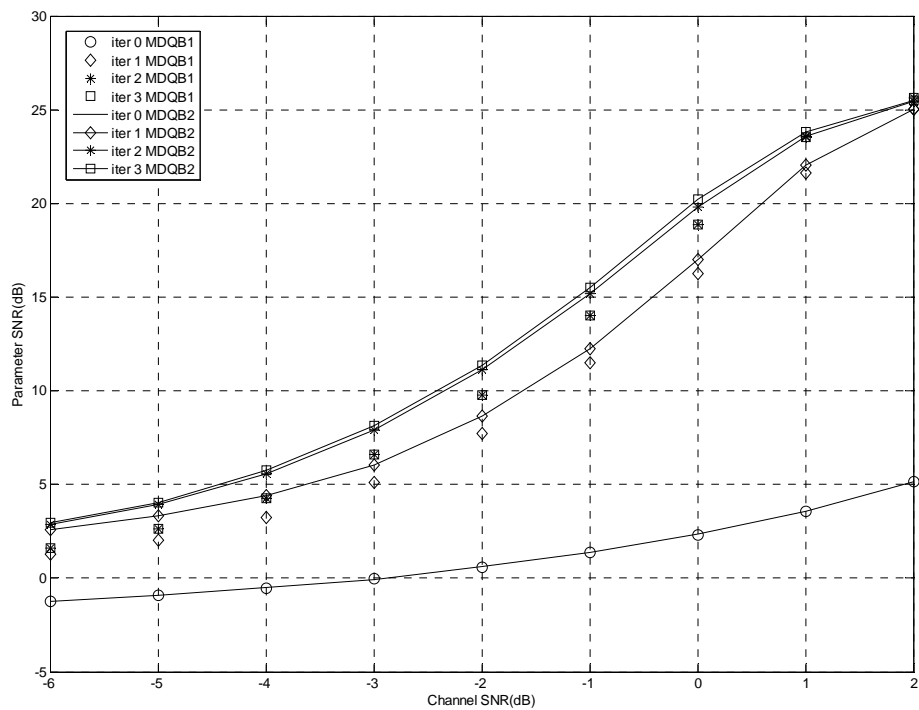


圖 5.2 位元層級的疊代解碼結果 $[(Q, M)=(32, 3)$ 且 BSA 配置]



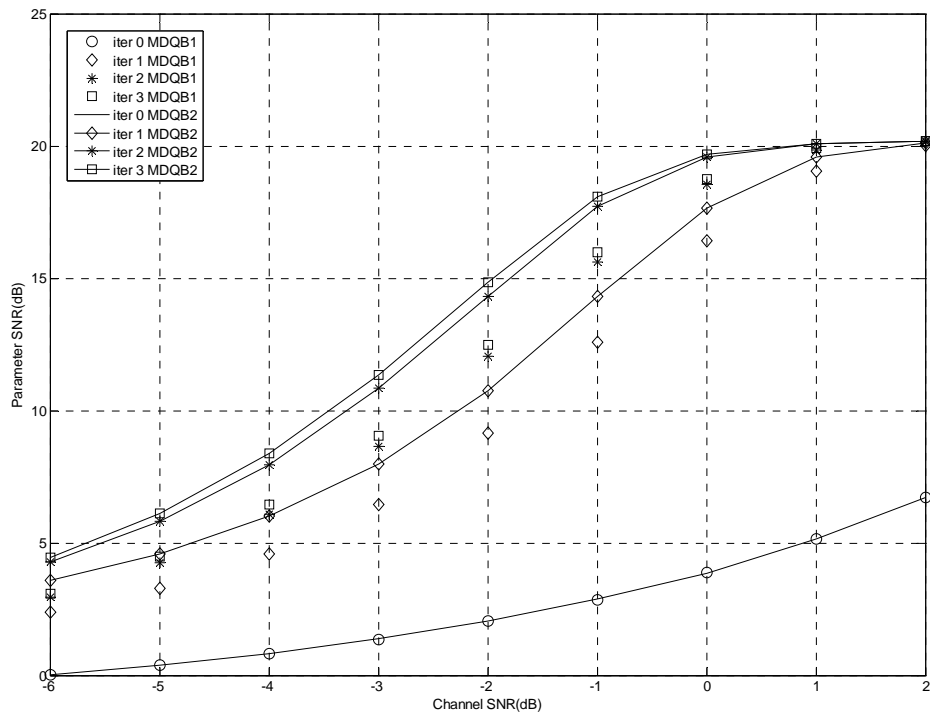


圖 5.3 位元層級的疊代解碼結果 $[(Q, M)=(16, 3)$ 且對角線配置]

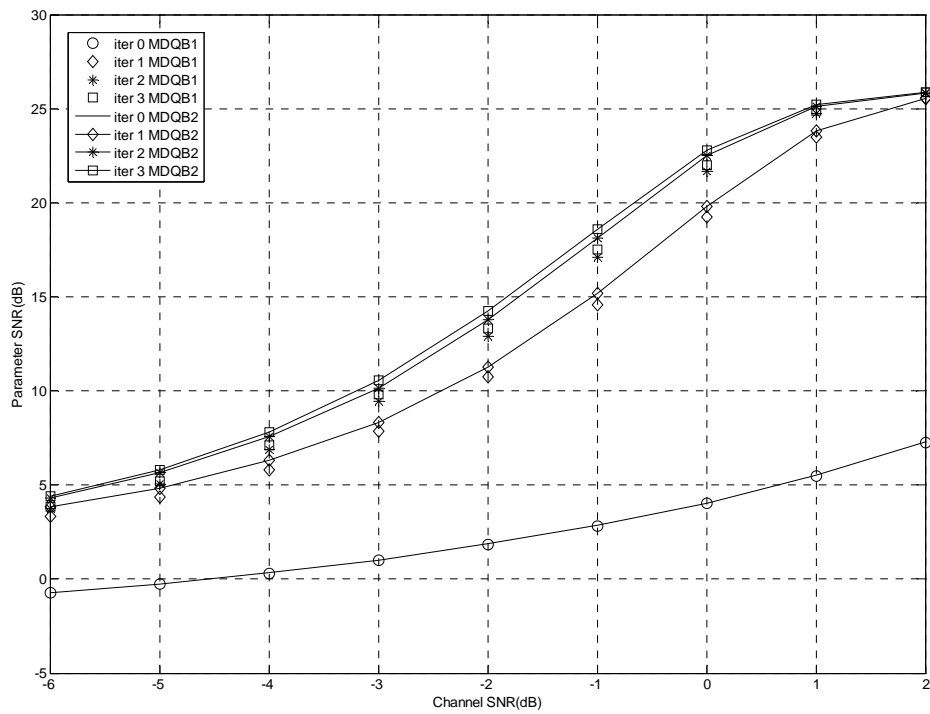


圖 5.4 位元層級的疊代解碼結果 $[(Q, M)=(32, 3)$ 且對角線配置]

## 5.2 索引層級平行式與合併式解碼之模擬

### 5.2.1 系統模擬之步驟說明

平行式與交叉式解碼架構的傳輸端中，訊號源  $u_i$  皆採用一階自迴歸處理來源，變異數設定為  $\sigma_u^2 = 1$ ，且其相關因子設成  $\rho = 0.95$ 。

每次模擬過程中產生 30000 點的訊號源送進 Q 階純量量化器

Lloyd-Max，位元映射選用自然對應編碼方式，產生量化索引後經由 BSA 索引配置法得到 M 位元的兩敘述索引，索引區塊交錯器重新排列敘述索引的順序，區塊大小設定為 10x10，接著兩敘述索引串通過一樣的迴旋碼通道編碼器，設定為  $(n=2, k=1, v=2)$ ，

$G = [1, 1 + D^2 / 1 + D + D^2]$ ，編碼器輸出兩倍長度的訊號位元串，經由二位元相位鍵移調變之後送進白高斯雜訊的通道傳輸。

[平行式解碼器]

接收端收到受通道雜訊干擾的兩個實數序列，分別進行獨立解碼。對 I 通道接收資訊而言，將通道相關資訊與事先訓練的敘述索引事前消息送進迴旋碼解碼器，且使用索引層級 BCJR 演算法作解碼。然後將索引層級的通道解碼額外訊息與敘述索引的一階事前消息送進訊源解碼器，輔以通道相關資訊而得到訊源解碼額外訊息。接著將該額外訊息回傳到通道解碼器的輸入端，依如此疊代方式進行解碼過程。當疊代次數完成後，將兩個敘述索引的後驗機率與敘述間的相關

資訊，一併送進多重敘述解碼器求得量化索引的後驗機率，最後使用 MMSE 法則估算最佳的訊號值。

[合併式解碼器]

與平行式解碼器相似之處在兩個敘述的接收序列是個別送進迴旋碼解碼器，且使用索引層級的 BCJR 演算法計算通道解碼額外訊息。而與平行式解碼器最大不同的地方在於其訊源解碼器只有一個，它接收來自兩個敘述的通道解碼額外訊息以及通道相關資訊，並整合量化索引一階事前消息以計算出量化索引的後驗機率。則是使用 MMSE 法則估算最佳的訊號值。

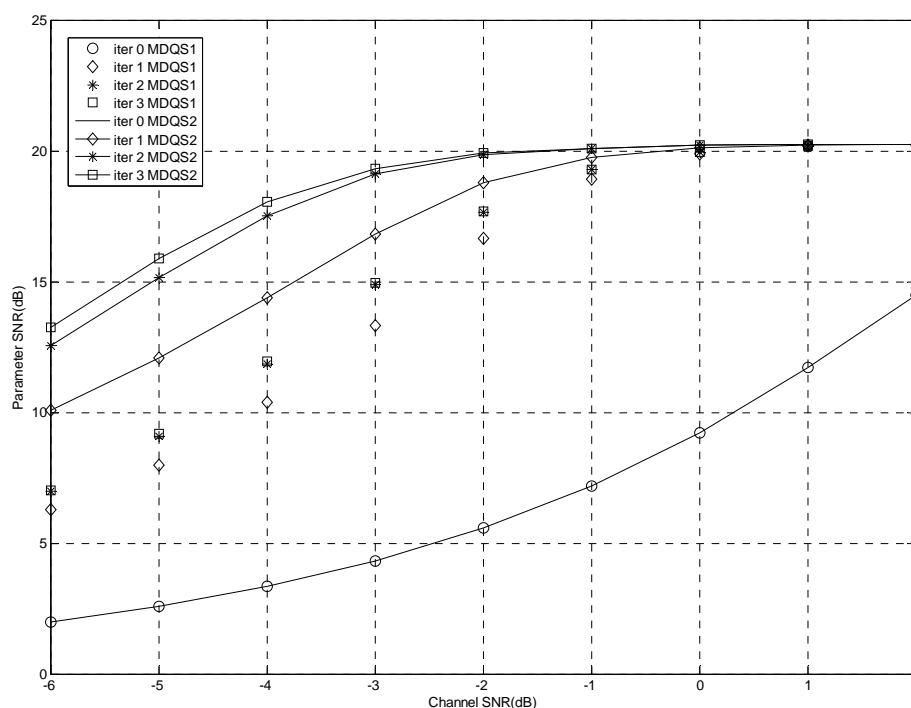


圖 5.5 索引層級的平行與合併疊代解碼結果 $[(Q, M)=(16, 3)]$

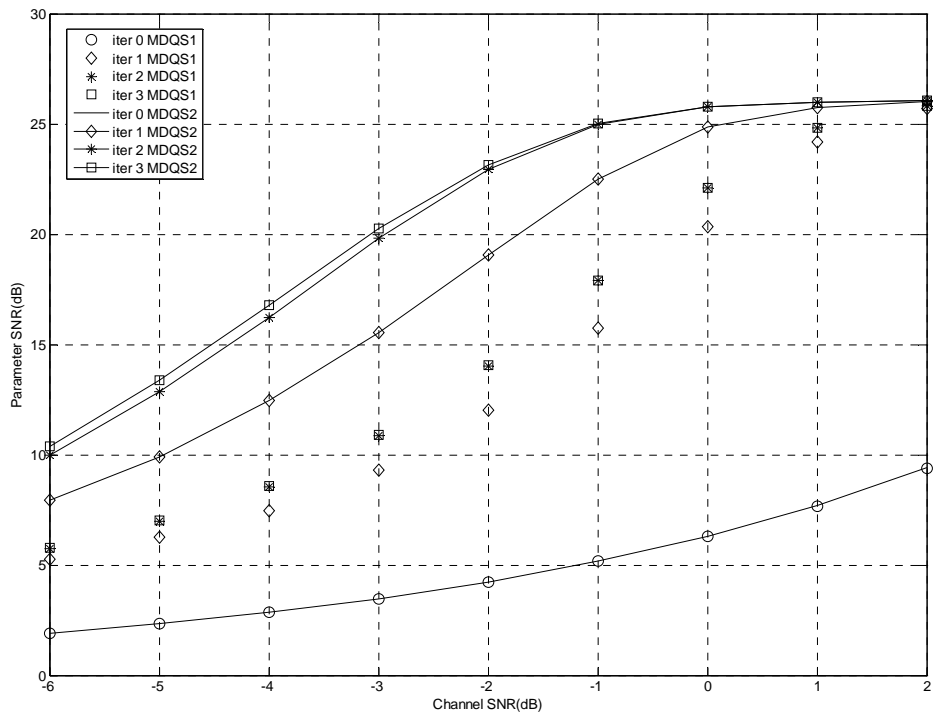


圖 5.6 索引層級的平行與合併疊代解碼結果 $[(Q, M)=(32, 3)]$

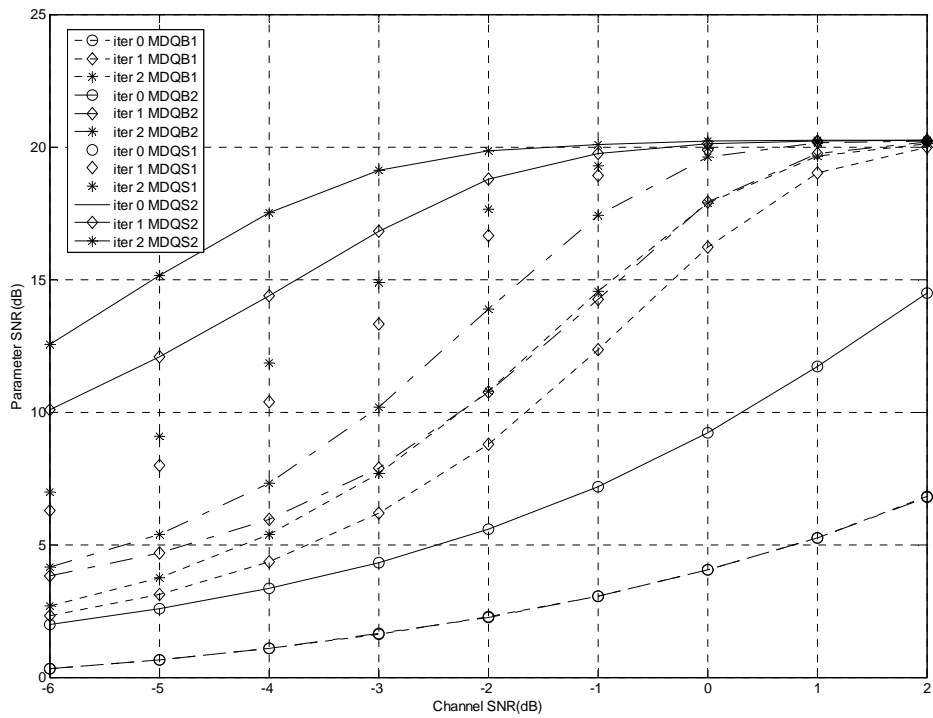


圖 5.7 位元與索引層級的疊代解碼結果 $[(Q, M)=(16, 3)]$

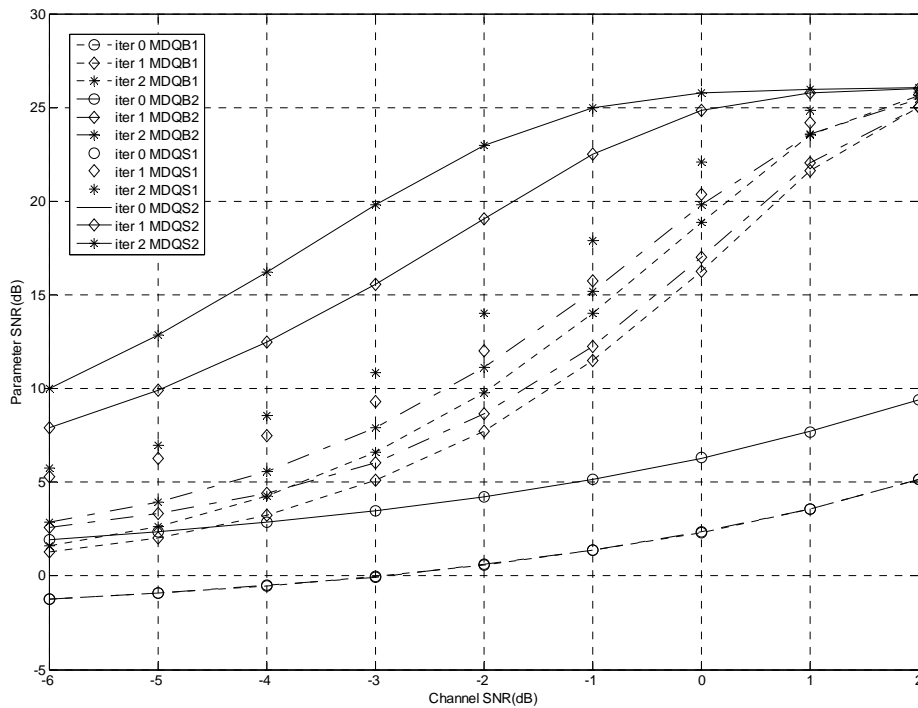


圖 5.8 位元與索引層級的疊代解碼結果 $[(Q, M)=(32, 3)]$

### 5.2.2 結果分析

令 MDQS1 與 MDQS2 分別代表索引層級的平行式解碼與合併式解碼系統效能曲線，其效能衡量仍然使用系統的參數訊雜比值 SNR。平行式解碼與合併式解碼在 0 次疊代時，不考慮通道解碼與訊源解碼資訊，因而具有相同的系統效能曲線。進一步於 1 次疊代完成後，平行式與合併式解碼效能曲線都有大幅度的向上提升，但合併式解碼流程因使用具有較強相關性之量化索引一階事前消息以及使用雙敘述的通道相關資訊，改善幅度遠勝於平行式解碼的效能曲線。於圖 5.5 與圖 5.6 中，在通道環境很差的條件之下，合併式解碼之效能曲線勝過

平行式解碼之效能曲線分別為 3.78dB 與 2.64dB。再者，於 2 次疊代結果來看，平行式與合併式效能曲線皆有些許程度的上升，且合併式之效能曲線上升的幅度比平行式曲線來得大。從圖 5.5 與圖 5.6 中，可以看出在通道極差的條件之下，合併式之效能曲線高於平行式之效能曲線分別為 5.57dB 與 4.24dB。在 3 次疊代結果中，平行式解碼的效能曲線幾乎與 2 次疊代結果重合，代表傳送資訊的殘餘冗息已幾乎被挖掘使用完；但合併式解碼的效能曲線仍有 0.2dB~0.7dB 的提升空間。從圖 5.5 和圖 5.6 中 3 次疊代的結果來看，合併式解碼的效能曲線於通道環境設定 0dB 時就趨近於理論上限值 20.22dB 與 26.01dB，而平行式解碼則要環境設定為 2dB 時才能達到該上限值。從圖 5.6 與圖 5.7 中可以明顯看出索引層級的解碼演算法皆優於位元層級的演算法，且系統效能趨勢大致呈  $MDQB1 < MDQB2 < MDQS1 < MDQS2$ 。

### 5.3 對數領域之合併式解碼模擬

#### 5.3.1 系統模擬之步驟說明

本小節的訊號源與環境條件相關設定，與機率領域的索引層級合併解碼實驗模擬一樣，差別在於解碼端是使用對數式的解碼演算法。0 階與 1 階量化索引事前消息、通道相關資訊以及額外訊息都是用相似比率函數來表示，目的是可以簡化演算法運算的困難度與複雜度，

以及消除有限精確度的問題，同時仍可達到與機率領域演算法運算相同的系統效能。對數式解碼演算法的內容可參考 4.4 節。

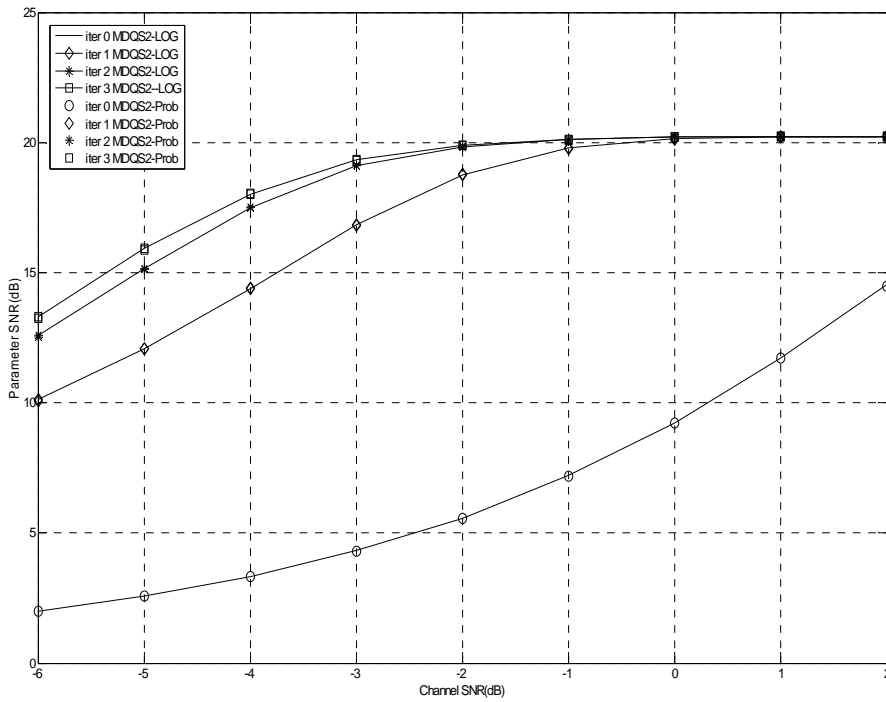


圖 5.9 機率與對數領域的索引層級合併解碼結果 $[(Q, M)=(16, 3)]$

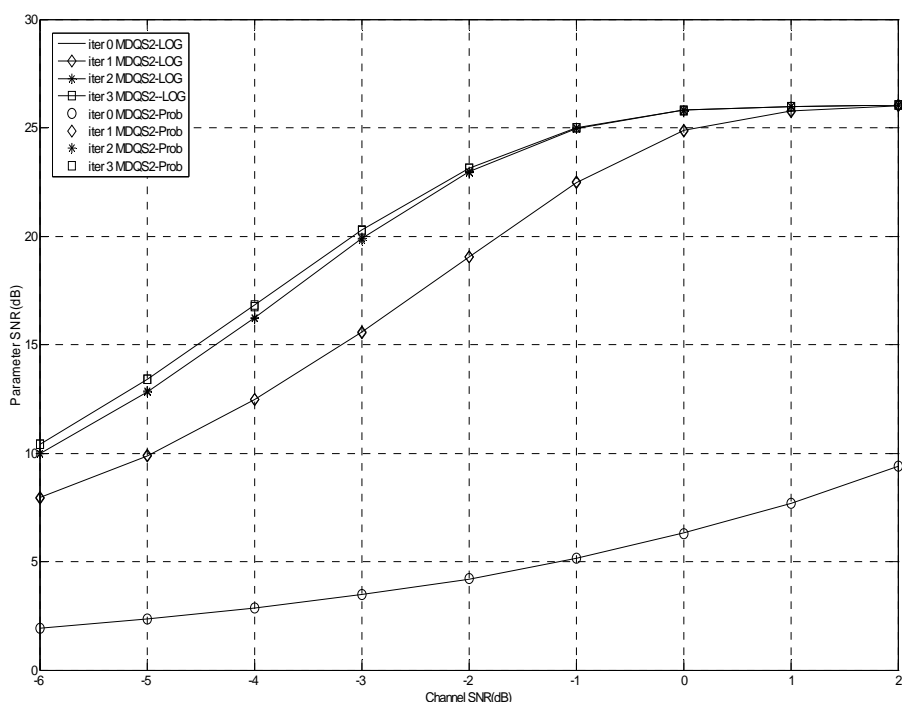


圖 5.10 機率與對數領域的索引層級合併解碼結果 $[(Q, M)=(32, 3)]$

### 5.3.2 結果分析

如同前述預期，對數式的解碼演算法可呈現如機率式演算法的系統效能，從圖 5.9 與圖 5.10 可以看出這種效果。

## 5.4 寬頻語音的多重敘述解碼實驗

### 5.4.1 系統模擬之步驟說明

圖 2.6 為多重敘述系統的模擬架構。實驗設定是參考文章[11]，我們使用大量男聲與女聲的訓練語料經過 AMR-WB 的編碼過程得到 194881 組 *ISF* 索引集合，並記錄其個別的索引事前機率與 1 階事前



消息，以及  $ISF$  參數索引經由索引配置後得到的敘述索引之 0 階與 1 階事前資訊，以供系統解碼時使用。在這些語料當中取出一段 200 秒長度的語音作為訊號源，利用 AMR-WB 編碼方法產生出 10000 組由七個相關索引所構成  $ISF$  索引集合，將這些索引個別視為量化索引  $\mathbf{u}_i^{(n)}$ ， $n \in \{1, 2, \dots, 7\}$ ，分別利用其對應的索引配置矩陣產生兩敘述索引  $I_n$  與  $J_n$ ，索引配置矩陣採用 BSA 配置法，其矩陣配置的大小如表 5.1 所示。兩敘述分別經由迴旋碼編碼處理後送進白高斯雜訊通道。在接收端使用第三章與第四章提出來的位元層級及索引層級解碼演算法，得到預估出來的量化索引  $\hat{\mathbf{u}}_i^{(n)}$ ，最後查詢量化碼書重建出  $\widehat{ISF}_n$  係數。我們使用重建的  $\widehat{ISF}_n$  係數與量化索引  $\mathbf{u}_i^{(n)}$  對應的  $ISF_n$  係數之間的信號雜訊比作為系統效能的衡量標準之一：

$$SNR_{dB} = 10 \cdot \log_{10} \frac{\sum_{m=1}^{10000} ISF_m^2}{\sum_{m=1}^{10000} (ISF_m - \widehat{ISF}_m)^2} \quad (5.2)$$

上述公式是針對單一  $ISF$  係數衡量系統效能。另外，若要考量整體的解碼效果，則可使用平均頻譜失真(Average Spectrum Distortion)公式。對第  $m$  個音框而言，其頻譜失真為

$$SD_m = \sqrt{\frac{1}{f_2 - f_1} \cdot \int_{f_1}^{f_2} \left[ 10 \log_{10}(\tilde{P}_m(f)) - 10 \log_{10}(\hat{P}_m(f)) \right]^2 df} \quad (5.3)$$

其中  $f_1 = 50$ 、 $f_2 = 6400$ 。

$$\tilde{P}_m(f) = \frac{1}{|\tilde{A}_m(\exp(j2\pi f / F_s))|^2} \text{ 與 } \hat{P}_m(f) = \frac{1}{|\hat{A}_m(\exp(j2\pi f / F_s))|^2} \text{ 為量}$$

化後與重建的功率頻譜密度，其中  $\tilde{A}_m(z)$  與  $\hat{A}_m(z)$  為量化後與重建的 LPC 濾波器， $F_s = 12800$ 。

#### 5.4.2 結果分析

表 5.2、表 5.3 與表 5.4 中，分別代表疊代次數為 1 時不同 AWGN 通道環境之下的訊號雜訊比，而表 5.5、表 5.6 與表 5.7 中，則代表疊代次數為 3 時不同通道環境之下的訊雜比。其中 ISF\_1 代表 ISF 第一組係數。位元層級的平行式解碼器(MDQB1)將接收到的雙敘述資料獨立解碼，並使用敘述索引一階事前機率作為事前消息，而交叉式解碼器(MDQB2)則同時考慮到雙通道敘述的資訊與其相關性，因此有較好的效能。再者，索引層級的平行式解碼器(MDQS1)為位元層級平行式解碼器的改良，而合併式解碼器(MDQS2)除了同時考慮雙敘述的接收資訊外，還使用了相關性較強的量化索引一階事前消息，更能有效提升系統解碼效能。由於索引層級的演算法能有效運用相鄰索引間的相關性，因此性能優於位元層級的解碼演算法。疊代次數增加時系統效能也會隨之提升。

至於平均頻譜失真的評估部分，系統效能趨勢也如上述的討論，在通道環境很差的情況之下，合併式演算法有最大的頻譜失真下降幅

度，在通道環境漸佳時也是呈現最低的失真趨勢，且在-1dB 的環境條件下就達到零失真的程度，而另外三種解碼曲線則要到 0dB 時才能完全達到零失真。系統效能基本上大致呈現  $MDQB1 < MDQB2 < MDQS1 < MDQS2$ 。

Index	$\log_2 Q$	M(bits)
1	8	5
2	8	5
3	6	4
4	7	4
5	7	4
6	5	3
7	5	3

表 5.1 ISF 索引所配置的指定矩陣

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	0.9106	1.691	1.8544	2.8973
ISF 2	0.96	1.8886	2.2183	3.7733
ISF 3	0.9772	2.0419	2.3735	3.8999
ISF 4	1.1858	2.0784	2.3028	4.1237
ISF 5	1.4918	2.3234	2.5841	4.5924
ISF 6	1.4144	2.2399	2.5624	4.217
ISF 7	1.5775	2.3019	2.5106	4.2659
ISF 8	1.0217	1.7025	1.8646	3.4707
ISF 9	1.0632	1.8126	2.0582	3.5223
ISF 10	1.782	3.2683	3.532	5.2868
ISF 11	1.9454	3.2639	3.1924	5.1508
ISF 12	2.0189	3.3322	3.2325	4.9065
ISF 13	1.8841	3.1626	3.2311	4.6514
ISF 14	1.7555	3.1591	3.4475	4.6192
ISF 15	1.4997	2.7278	3.3685	4.0553
ISF 16	1.7413	2.6524	2.9087	3.3398

表 5.2 各種疊代解碼機制的參數重建結果(iteration 1, -6dB AWGN)

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	9.5111	11.3028	15.0131	19.521
ISF 2	10.015	12.5991	16.673	22.4493
ISF 3	10.2376	13.2688	17.4786	22.9295
ISF 4	10.3706	13.1011	17.1633	22.9445
ISF 5	10.65	13.1671	17.1061	23.4802
ISF 6	10.5061	13.0593	17.0186	22.1812
ISF 7	10.3288	12.5007	16.1361	21.963
ISF 8	10.075	12.0566	15.7149	21.4539
ISF 9	10.2545	12.6512	16.598	21.6076
ISF 10	11.9945	15.6287	20.8553	25.3414
ISF 11	12.2088	15.3599	19.7176	24.7681
ISF 12	12.201	15.1131	18.6924	23.5123
ISF 13	11.9416	14.788	18.753	23.0013
ISF 14	11.8129	15.1371	19.2865	23.0225
ISF 15	11.5337	14.3875	18.584	21.1498
ISF 16	11.9659	14.4901	18.0207	20.0448

表 5.3 各種疊代解碼機制的參數重建結果(iteration 1, -2dB AWGN)

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	42.3956	45.3601	49.9757	56.0422
ISF 2	44.1926	48.0146	51.975	60.1518
ISF 3	44.3664	47.2766	52.5995	61.6378
ISF 4	45.0454	47.4923	52.6563	59.6475
ISF 5	45.0677	47.4717	53.0667	60.4514
ISF 6	44.804	47.197	53.2625	59.15
ISF 7	43.8996	46.74	51.9929	57.9917
ISF 8	44.1562	46.9985	52.4662	58.6164
ISF 9	44.7265	48.1009	53.7118	58.9691
ISF 10	46.9319	51.1037	59.2345	61.4033
ISF 11	46.5902	50.4927	58.0753	60.4886
ISF 12	46.3973	50.5752	55.6705	58.7718
ISF 13	45.9208	49.8443	57.84	58.3817
ISF 14	46.2341	50.8883	57.0689	58.372
ISF 15	45.9391	48.8297	55.8611	56.3752
ISF 16	45.6684	48.4673	53.7005	55.0338

表 5.4 各種疊代解碼機制的參數重建結果(iteration 1, 2dB AWGN)

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	0.9626	1.8919	1.9491	4.4158
ISF 2	1.0027	2.1119	2.3346	5.6881
ISF 3	1.0178	2.2593	2.5025	5.9353
ISF 4	1.2196	2.2835	2.4134	6.1181
ISF 5	1.5276	2.5337	2.6876	6.6005
ISF 6	1.4546	2.4589	2.6634	6.0044
ISF 7	1.6122	2.532	2.6088	6.1587
ISF 8	1.0611	1.9111	1.9545	5.2721
ISF 9	1.1027	2.0155	2.1487	5.2044
ISF 10	1.8578	3.4703	3.7924	6.7091
ISF 11	2.0184	3.4843	3.4402	6.5603
ISF 12	2.0889	3.569	3.469	6.2519
ISF 13	1.9494	3.3863	3.4572	5.9406
ISF 14	1.8106	3.3661	3.6499	5.8914
ISF 15	1.5576	2.9316	3.5436	5.1689
ISF 16	1.7882	2.8583	3.0885	4.3841

表 5.5 各種疊代解碼機制的參數重建結果(iteration 3, -6dB AWGN)

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	10.548	15.8068	15.6839	28.6691
ISF 2	11.1258	17.254	17.3828	31.9567
ISF 3	11.4038	18.0575	18.2345	33.5826
ISF 4	11.5168	17.7221	17.9359	31.7132
ISF 5	11.7612	17.7755	17.8821	32.1191
ISF 6	11.6116	17.5788	17.7586	30.8066
ISF 7	11.4696	17.1176	16.8983	29.5394
ISF 8	11.1698	16.579	16.4298	29.2682
ISF 9	11.3296	17.0034	17.2658	29.3872
ISF 10	14.082	20.705	22.4727	32.5194
ISF 11	14.3037	20.4543	21.2624	31.6674
ISF 12	14.3121	20.18	20.2447	30.5018
ISF 13	14.1095	19.9363	20.3079	30.4339
ISF 14	13.91	20.0763	20.7274	30.061
ISF 15	13.5517	19.2586	19.8775	27.9437
ISF 16	13.7585	19.1804	19.1744	26.8451

表 5.6 各種疊代解碼機制的參數重建結果(iteration 3, -2dB AWGN)

SNR	MDQB1	MDQB2	MDQS1	MDQS2
ISF 1	43.2561	52.7065	50.2398	62.7487
ISF 2	45.3566	55.5958	52.3132	65.2597
ISF 3	45.7871	56.0317	53.0455	66.8739
ISF 4	46.2872	56.3912	53.116	64.7589
ISF 5	45.9181	56.3005	53.5136	65.3525
ISF 6	45.9322	55.6945	53.7048	64.7216
ISF 7	44.8761	54.5963	52.3293	62.5586
ISF 8	45.2731	54.4853	52.741	61.9843
ISF 9	45.9608	56.0527	53.8112	62.732
ISF 10	49.7514	60.0964	59.6156	64.7507
ISF 11	49.3106	59.6623	58.6961	64.3005
ISF 12	49.0774	59.4549	56.4633	63.1424
ISF 13	48.6054	58.4647	57.9431	63.4353
ISF 14	49.3953	58.0347	57.6186	64.2589
ISF 15	47.7727	55.3074	56.2325	60.9002
ISF 16	47.6215	54.7702	54.6831	59.3123

表 5.7 各種疊代解碼機制的參數重建結果(iteration 3, 2dB AWGN)

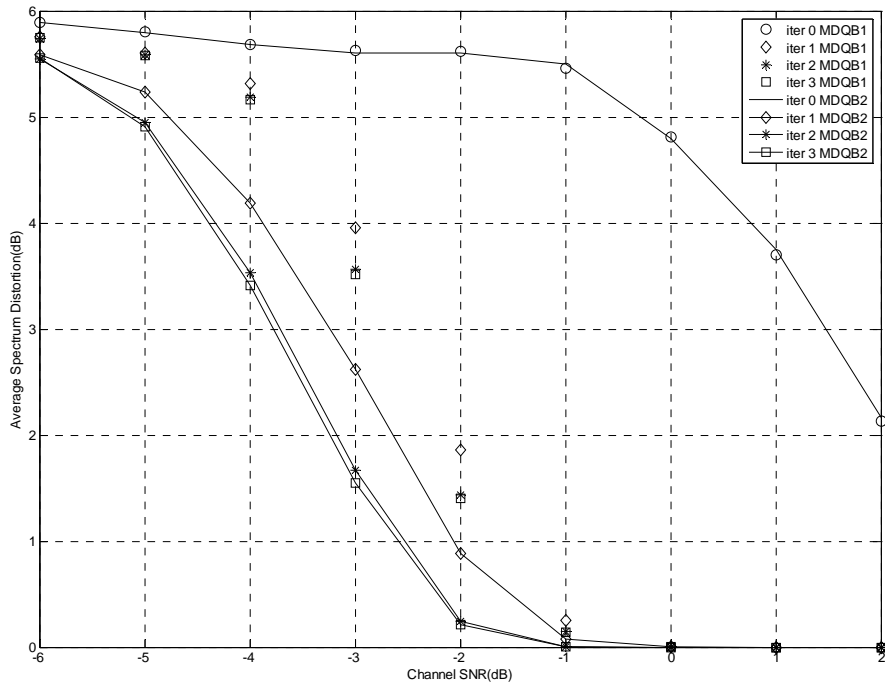


圖 5.11 位元層級疊代解碼器的平均頻譜失真結果

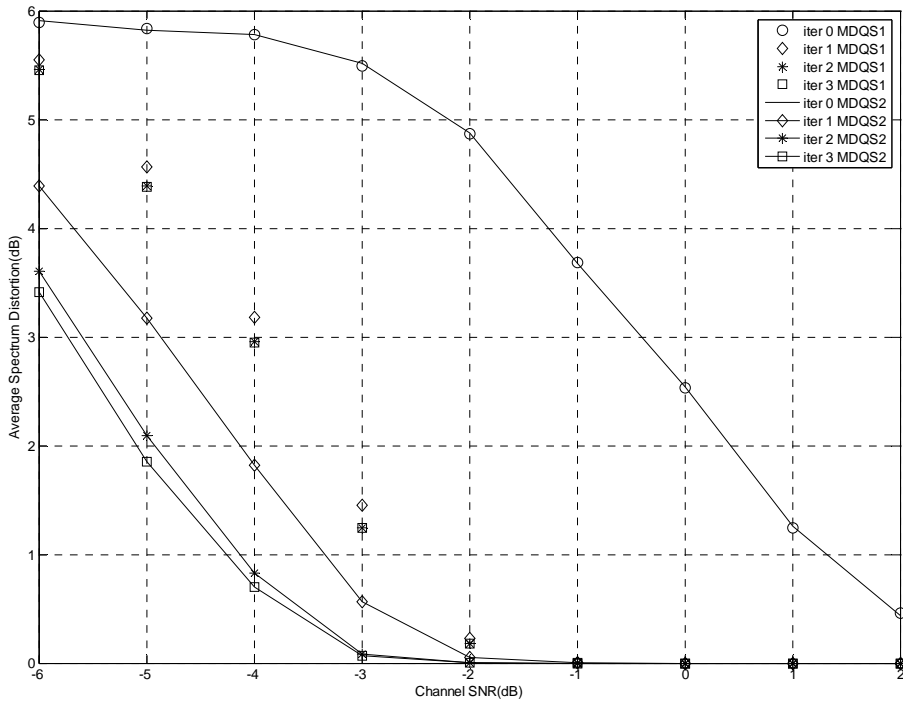


圖 5.12 索引層級疊代解碼器的平均頻譜失真結果

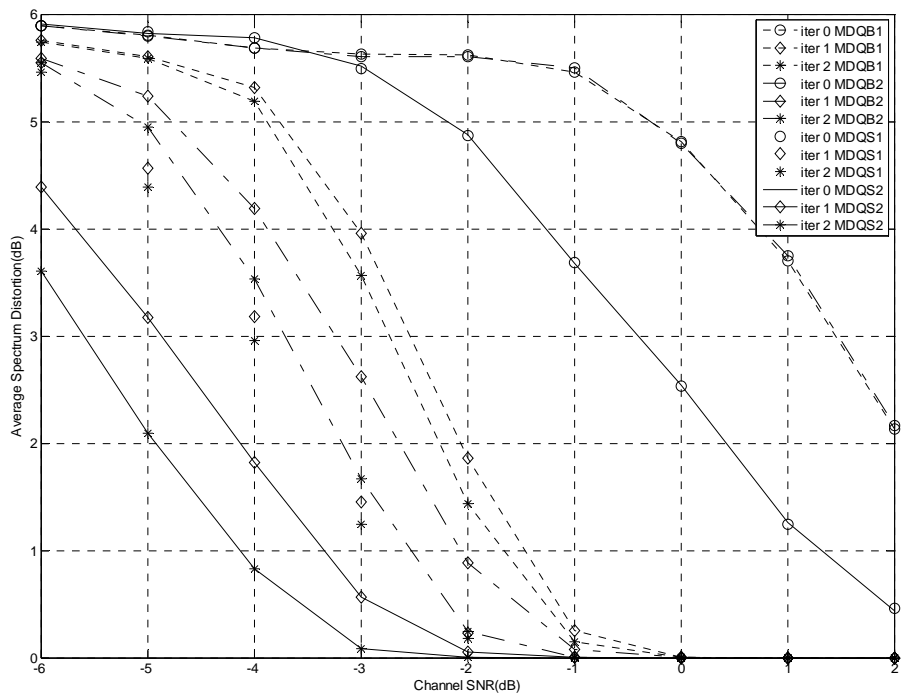


圖 5.13 各種疊代解碼器的平均頻譜失真結果





## 第六章 結論與未來展望

本論文先提出位元層級的疊代訊源通道解碼演算法，分成平行式與交叉式兩種架構。平行式解碼架構對其所接收的雙敘述資訊是採用分開獨立解碼的方式，亦即每個敘述是個別採用傳統的疊代解碼演算法；交叉式解碼架構則考量到雙敘述之間的相關特性，並引入另一敘述的通道資訊提供訊源解碼器解碼之用，可有效提升系統效能。此外，考量到位元層級後驗機率推導所採用的位元獨立假設並不符合實際情況，因此提出索引層級的疊代解碼演算法，其架構又可分為平行式與合併式兩種。平行式解碼架構改良自位元層級平行式解碼演算法，其通道解碼器使用索引層級之 BCJR 演算法，而訊源解碼器採用軟性位元解碼演算法。而合併式解碼架構，其訊源解碼器同時使用收到的雙敘述資訊作解碼，以及考量相關性較強的量化索引一階事前消息，較平行式解碼而言，更能有效地運用雙敘述的接收資訊以達到高解碼效能。

在機率領域的解碼演算法有其運算複雜度與精準度有限的問題，因此我們特別針對性能最佳的索引層級合併式解碼架構，提出其對數領域的解碼演算法，以解決硬體設計與即時製作的困難度。實驗模擬中採用的訊號源包括有仿語音的一階自迴歸模型以及真實語料

經寬頻語音編碼後的資訊，兩者驗證皆符合理論預期。實驗結果顯示索引層級解碼演算法優於位元層級解碼演算法，索引層級中的合併式解碼架構優於平行式解碼架構，而位元層級之交叉式架構優於平行式架構。此外，在索引層級之合併式解碼架構中，對數與機率不同領域的實驗模擬驗證了對數式模型不僅可簡化運算複雜度，且無資訊呈現的失真問題。

目前提出的演算法架構皆基於通道無記憶性的假設前提，但現實的通訊環境是無法以此簡單的機率模型模擬之，以致應用在實際通道的傳輸效能將會有一定程度的衰減。為了解決此應用層面的問題，未來研究必須基於記憶性通道模型來設計解碼器，或者使用更近似於實際通道特性的模型來推導解碼演算法。此外，傳輸架構中的交錯器也扮演影響整體系統效能的關鍵角色，將來可進一步比較不同大小的區塊交錯器與偽隨機交錯器(Pseudo random interleaver)效能上的差異，以便找出最佳的交錯器設定。

## 參考文獻

- [1] V.A. Vaishampayan, “ Design of multiple description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 39, pp. 821-834, May 1993.
- [2] N. Görtz, and P. Leelapornchai, “ Optimization of the index assignment for multiple description vector quantizers,” *IEEE Trans. Commun.*, vol.51, no.3, pp.336-340, Mar. 2003.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima, “ Near Shannon limit error correcting coding:turbo codes,” in *Proc. IEEE International Conference on Communication(ICC)*, pp. 1064-1070, 1993.
- [4] Chi-Shung Tseng, Wen-Whei Chang, “ Channel-Matched Index Assignment Design In Multiple Description Quantization,” *Master thesis*, National Chiao Tung University, June 2008.
- [5] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Mikkola, H. Vainio, and K. Jarvinen, “ The Adaptive Multirate Wideband Speech Codec(AMR-WB).” *IEEE Transactions on speech and audio processing*, vol.10, no.8, pp.620-636, Nov. 2002.
- [6] Yen-Chang Pan, Wen-Whei Chang, “ Index-Based Iterative Source-Channel Decoding,” *Master thesis*, National Chiao Tung University, June 2008.
- [7] J. Barros, J. Hagenauer, and N. Görtz, “ Turbo cross decoding of multiple descriptions,” in *Proc. IEEE ICC*, vol. 3, 2002, pp. 1398–1402.
- [8] M. Srinivasan, “ Iterative decoding of multiple descriptions,” in *Proc. IEEE DCC*, March 1999, pp. 3–22.
- [9] M. Adrat, “ Iterative Source-Channel Decoding for Digital Mobile Communications,” *PhD thesis*, Aachen University of Technology,

July 2003.

[10] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform Theory*, vol. IT-6, pp. 7-12, Mar 1960.

[11] Chih-Chien Tsai, Wen-Whei Chang, "A Study of robust wideband speech coding," *Master thesis*, National Chiao Tung University, June 2007.

