

行動通訊錯誤緩和機制之研究

**A Study of Channel Error Mitigation for GSM  
Networks**

研究生：曹正宏

**Student: Cheng-Hung Tsao**

指導教授：張文輝

**Advisor: Wen-Whei Chang**

國立交通大學

電信工程學系

碩士論文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Communication Engineering

June 2006

Hsinchu, Taiwan, Republic of China

中華民國 九十五年 六月

# 行動通訊錯誤緩和機制之研究

學生：曹正宏

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

合併音源-通道解碼器的研究，主要是在改良向量量化系統的解碼器，加強其錯誤隱蔽功能以維持位元傳輸的強健性。向量量化編碼器經由樣本碼書比對產生一組最近似輸入信號的碼字，再傳送其索引值作為編碼輸出。而解碼器的訊號還原則是依據接收到的索引值查表取得對應碼字。問題是無線通道環境中存在諸多雜訊的干擾，改變碼字索引的接收值進而造成訊號還原的失真。本論文先針對行動通訊錯誤進行模型分析，以通道匹配技術估測馬可夫模型參數，並結合輸出碼字索引序列的殘餘冗息以提升解碼效能，其關鍵在於配合不同的通道錯誤屬性估算所有可能傳輸索引值的後驗機率，同時依據最小均方誤差準則還原其輸出訊號。我們使用行動通訊平台模擬 GSM 通訊環

境，進行有限狀態馬可夫通道的模型分析，再分別發展基於索引層級和基於位元層級解碼演算法。自迴歸訊號源的模擬結果顯示，基於索引層級演算法較複雜但有較好的系統效能，基於位元層級演算法則易於實現但效能較低。最後，我們整合兩者優點而開發一混合層級的解碼演算法。



# A Study of Channel Error Mitigation for GSM Networks

Student: Cheng Hung Tsao

Advisor: Dr. Wen-Whei Chang

Institute of Communication Engineering

National Chiao Tung University



This study is concerned with the joint source-channel decoding algorithms having higher robustness against bursty channel errors. The proposed decoder exploits the residual source redundancy and the channel memory information through the use of minimum mean-square error (MMSE) estimation. It is important to address the importance of matching the real channel behavior to the channel model on which the a posteriori probability computation is based. This task was accomplished by using the exponential curve-fitting and the

iterative gradient-descent techniques to characterize the channel with memory. We use a mobile communication platform to simulate GSM system and develop the index-level and the bit-level MMSE decoding algorithms. Simulation results indicate that the index-level decoding algorithm performs better than the bit-level decoding algorithm, but at the cost of increased complexity.



## 誌謝

本篇論文的完成，首先感謝指導教授張文輝老師的悉心指導，在研究所兩年的教誨，讓我體認到做研究所需具備的態度與方法，更使我在學業與生活上有深刻的體認與成長。另外也要感謝實驗室的學長、同學及學弟們，協助解決課業與研究上所遇到的瓶頸與問題。最後我要感謝父母與家人在這段期間，給予我最大的鼓勵與支持，順利完成研究所學業。僅此將本論文獻給我的指導老師、學長、父母以及所有關心我的好友們。



# 目錄

中文摘要.....	i
英文摘要.....	iii
誌謝.....	v
目錄.....	vi
圖目錄.....	ix
第一章 緒論.....	1
1.1 研究動機與方向.....	1
1.2 章節概要.....	2
第二章 記憶性通道模型分析.....	3
2.1 行動通訊平台.....	4
2.1.1 操作介面.....	4
2.1.2 建立 GSM 系統模組.....	8
2.1.3 GSM 系統模擬的流程.....	10
2.1.4 錯誤序列的提取.....	11
2.2 有限狀態的馬可夫鏈通道模型.....	13
2.2.1 Gilbert Models.....	14

2.2.2 Fritchman Models.....	15
2.3 Gilbert 及 Fritchman 通道模型分析.....	16
2.3.1 錯誤距機率分佈.....	16
2.3.2 模型參數的估測.....	17
<b>第三章 基於索引層級的解碼器.....</b>	<b>21</b>
3.1 向量量化編碼.....	21
3.2 傳輸索引值之後驗機率的計算.....	22
3.2.1 通道移轉機率的計算.....	23
<b>第四章 基於位元層級的解碼器.....</b>	<b>26</b>
4.1 位元後驗機率的疊代演算法.....	26
4.2 索引層級訊源冗息的整合.....	29
4.2.1 索引通道移轉機率之計算.....	29
4.2.2 事前消息的計算.....	30
4.2.3 量化索引事後機率的計算.....	31
4.2.4 參數估測.....	32
<b>第五章 實驗模擬結果與分析.....</b>	<b>33</b>



5.1 通道模型參數的估測.....	34
5.1.1 參數估測之步驟說明.....	34
5.1.2 參數估測結果分析.....	36
5.2 Gilbert 通道模型環境之系統模擬.....	37
5.2.1 系統模擬之步驟說明.....	37
5.2.2 結果分析.....	38
5.3 Fritchman 通道之系統模擬.....	39
5.3.1 系統模擬之步驟.....	39
5.3.2 結果分析.....	41
5.4 GSM 數據通道之系統模擬.....	41
5.4.1 系統模擬之實驗步驟.....	41
5.4.2 結果分析.....	42
<b>第六章 結論與未來展望.....</b>	<b>44</b>
6.1 結論.....	44
6.2 未來展望.....	45
<b>參考文獻.....</b>	<b>46</b>

## 圖目錄

圖 2-1 CCSS 開始畫面.....	5
圖 2-2 建立 Workspace.....	5
圖 2-3 設定 Worksapce 及建立.....	6
圖 2-4 Workspace tree.....	6
圖 2-5 建立 New Library.....	7
圖 2-6 Library 設定及產生.....	7
圖 2-7 New Library in Workspace.....	8
圖 2-8 tch_f96_2.....	9
圖 2-9 cc_tch_f48.....	9
圖 2-10 GSM 系統模組.....	10
圖 2-11 模擬設定檔.....	11
圖 2-12 Code Generation.....	11
圖 2-13 Simulation.....	11
圖 2-14 Davis 工作畫面.....	12
圖 2-15 載入模擬結果.....	12
圖 2-16 儲存錯誤序列.....	13
圖 2-17 Gilbert model.....	14
圖 2-18 Fritchman Channel.....	15

圖 3-1 向量量化系統流程圖.....	22
圖 4-1 系統傳輸流程圖.....	26
圖 5-1 C/I=1 dB.....	35
圖 5-2 C/I=4 dB.....	35
圖 5-3 C/I=7 dB.....	36
圖 5-4 C/I=10 dB.....	36
圖 5-5 Gilbert 通道之系統模擬.....	38
圖 5-6 Fritchman 通道之系統模擬.....	40
圖 5-7 GSM 數據通道之系統模擬.....	42



# 第一章 緒論

## 1.1 研究動機與方向

典型的數位通訊系統中，訊號傳送出去經過通道到接收端，通常會遭受環境雜訊等諸多效應的干擾，致使接收到的訊號無法判別重建出原始傳送訊號的樣貌。所以我們必須設法改進通訊系統，以其增強其對抗通道雜訊之能力。

而在傳統的數位通訊系統裡，通常將訊源編碼(source coding)和通道編碼(channel coding)兩個系統區塊分開考量和獨立設計，此設計概念乃源自於沈農(Shannon)的消息理論。此理論假設在設計個別的編碼架構時，前提假設另一個編碼器是最佳化設計，但這不符合實際的通訊環境。訊源編碼器在編碼處理後無法完全去除輸出訊號間之關連性，即輸出訊號之間彼此不是獨立同等分佈(i. i. d.)，其間存在某種形式的分佈或隱含有記憶性，因此，若能妥善利用訊號源在編碼後之殘餘冗息(residual redundancy)，應能有效提升系統效能，進而對抗通道雜訊之干擾。有鑑於此，合併訊源通道解碼(joint source-channel decoding)相關研究之目的是，在不考慮錯誤修正碼保護位元情況下，仍能維持強健性位元傳輸以對抗通道雜訊。數位通訊系統中的通道環境，由於許多效應影響，會在接收端造成叢發性錯

誤，為通道的記憶特性。若能在解碼端，有效應用通道環境的記憶特性，則系統效能將會再有所提升。

## 1.2 章節概要

第二章介紹了使用 CoCentric System Studio 軟體來製作 GSM 數據通道錯誤序列。第三章介紹基於索引層級的演算法架構，第四章則是基於位元層級的演算法架構以及混和層級演算法。第五章為實驗系統模擬，模擬三、四章所提之演算法架構，並且在不同的通道環境模型下，對模擬結果作效能評估與比較。第六章為結論與未來展望。



## 第二章 記憶性通道模型分析

數位通訊系統中，信號在無線通道環境傳輸的過程，有可能因通道脈衝響應內各個路徑的增益、相位移及傳播延遲造成的信號衰弱，因而在接收端造成叢發性的傳輸錯誤。當傳輸位元的傳送時間遠小於平均衰減所影響的時間，則此傳輸的位元或相鄰的位元，發生錯誤的機率會大於被正確接收的機率。因此一般研究所假設的無記憶性二元對稱通道，並不足以正確反應無線通訊環境中因多路徑衰退所產生的叢發位元錯誤。為了恰當地描述真實通道環境中的錯誤分佈，本章介紹具有描述叢發位元錯誤能力的有限狀態馬可夫通道模型，以及通道模型的分析方法，若能順利完成通道模型的分析，就能結合通道模型的參數以期求得位元錯誤機率的疊代運算。

為了反應真實的通道環境，我們選擇分析 GSM 系統的通道環境，採用國家系統晶片中心提供的 CoCentric GSM 系統模擬平台，模擬製作 GSM 數據通道的位元錯誤序列，其中主要工作是依不同統計模型組合通道脈衝響應內各個路徑的增益、相位移及傳播延遲。至於通道模型分析，採用有限狀態馬可夫鏈數學模型的 Gilbert 以及 Fritchman 模型來描述無線通道內傳輸位元的狀態變遷。

## 2.1 行動通訊平台

CoCentric System Studio (CCSS)是新思(Synopsys)科技所開發的軟體，其主要功能是幫助系統層次晶片的開發設計與模擬。在現今由超大型積體電路(VLSI)踏入系統層級晶片(System On Chip)的時代，CCSS 所內建的許多 libraries，對於發展系統晶片有很大的助益。

CCSS 是利用模組的概念[1]；一個系統通常有好幾個不同功能的電路區塊，而我們可以把一個電路區塊看成一個模組，而在 CCSS 中就會看到好幾個模組(方塊)透過接線，最後組成一個系統的模型，利用這種概念我們可以簡化系統的複雜度。而 CCSS 提供了一個 GSM Design Kit，它包含了兩個模組：gsmeq 與 gsmphy，利用這兩個模組能夠模擬 GSM 標準 05 系列的實體層傳輸路徑，其中我們主要會用到的部分都在 gsmphy 之內。接下來說明如何利用 CCSS 模擬製作 GSM 數據通道的錯誤序列。

### 2.1.1 操作介面

當進入 CCSS 時，將看到如圖 2-1 的畫面：

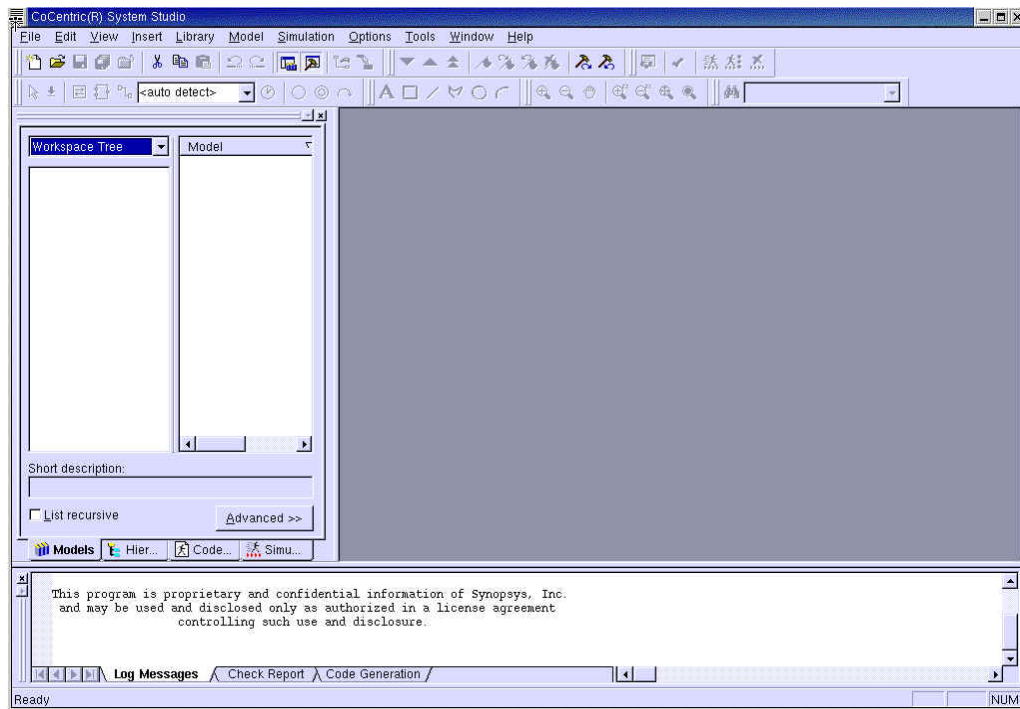


圖 2-1 CCSS 開始畫面

首先必須建立一個 Workspace，建立方法由圖 2-1 所示左上角的 File->New 接著就會出現圖 2-2 的畫面；點選之後將會出現圖 2-3 的設定畫面，Name 決定 Workspace 的名稱，Location 決定工作的路徑。

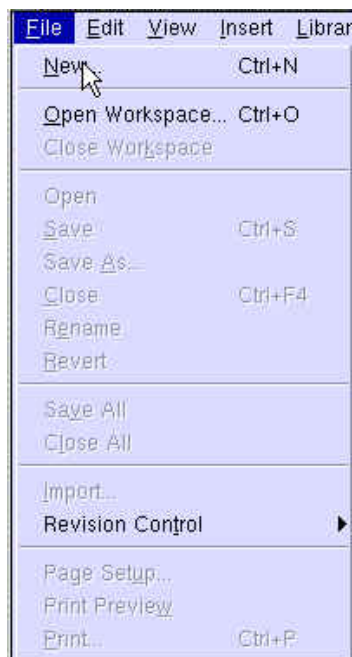


圖 2-2 建立 Workspace



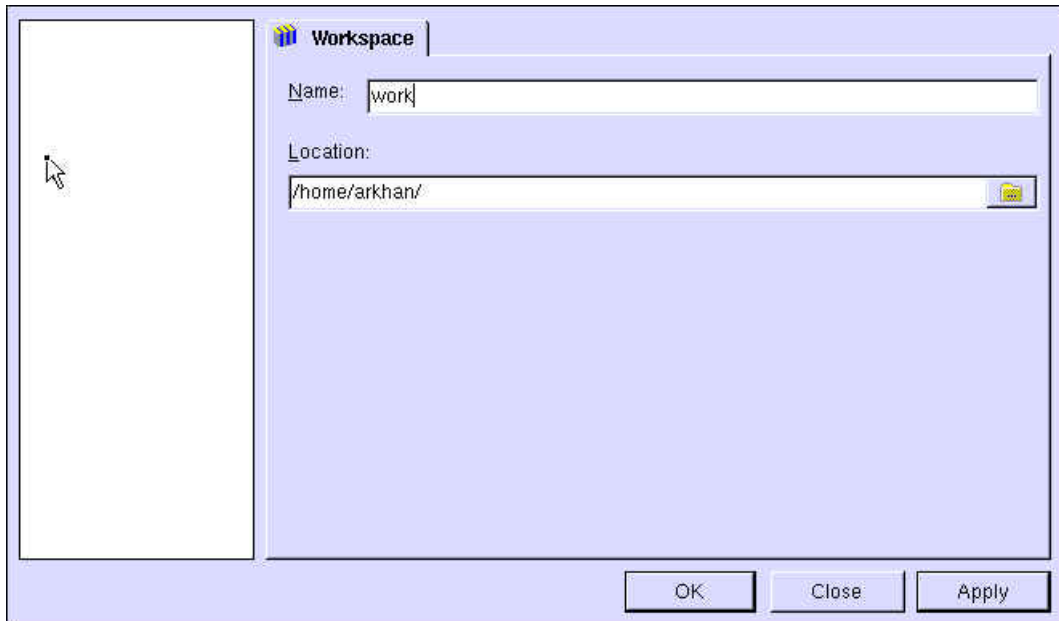


圖 2-3 設定 Worksapce 及建立

當決定好 Name 及 Location 後，建立新的 Worksapce，會發現在一開始的畫面多了一些東西如圖 2-4 所示的內容邊偏上方可以發現到多了一個剛才所打入的 Name 及 cocentric 這二行，和 Name 相同則是 Worksapce，而 cocentric 則是內建好的 Library。只要我們建立一個新的 Worksapce，則會自動產生 cocentric 這個 Library，點一下 + 的符號可以看見 cocentric 中已提供的一些模組。

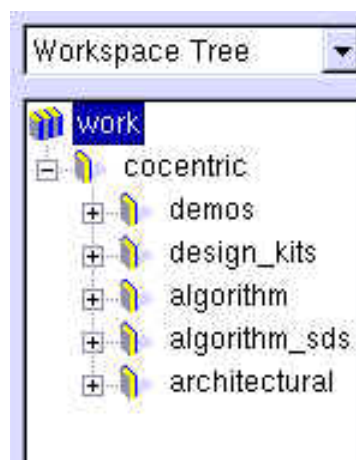


圖 2-4 Workspace tree

接下來則是建立自己的 Library，建立方法為 Library->New Library(如圖 2-5 所示)，接著就會出現圖 2-6 所示要決定 Library 的 Name 及 Location，按下 OK 後在 Workspace Tree 會多一個我們剛剛增加的 Library(如圖 2-7 所示)



圖 2-5 建立 New Library

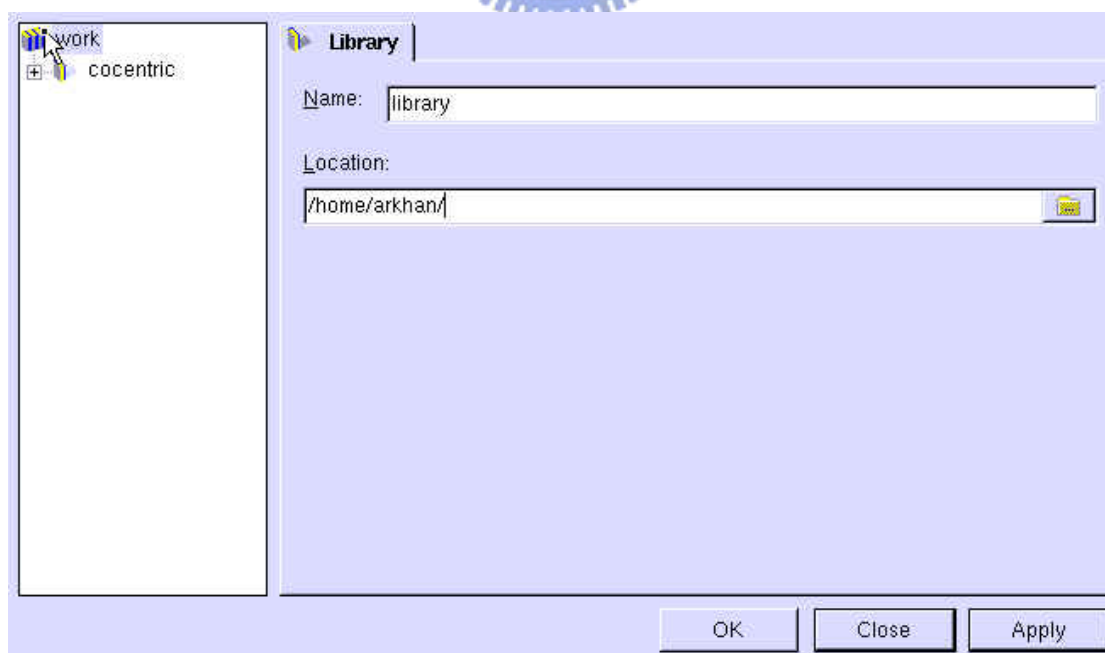


圖 2-6 Library 設定及產生

當 Library 建立好之後就可以把我們在 gsmphy 會用到的模組拖曳到剛剛建立的 Library 之內，這樣我們才能控制修改這些模組功能來執行模擬。

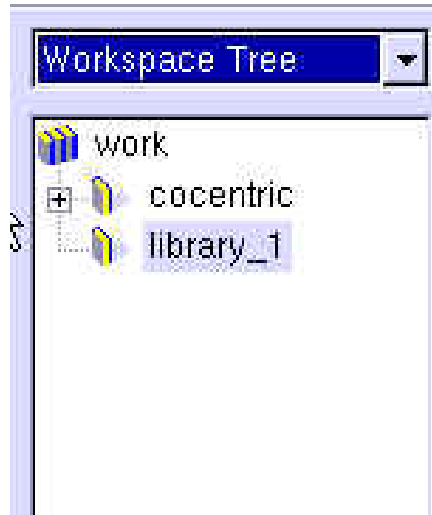


圖 2-7 New Library in Workspace

### 2.1.2 建立 GSM 系統模組

在建立好 Library 之後，接下來則是利用 CCSS 的 GSM Design Kit 所提供的模組來建立出 GSM 系統[2]。我們希望模擬的 GSM 系統環境，是將資料在傳輸速度 full rate 4.8k bits/s 的 traffic channel 中傳送，並受到 cochannel 的影響。首先在 gsmphy 的目錄下有個 gsmphy\_examples 的目錄，裡面提供了模擬 GSM traffic channel 的系統模組，如圖 2-8 所示，我們將會用到的是 tch\_f96\_2 這個模組，將它拖曳到新建的 library 之下。

接著將同樣目錄下的 tch\_fs\_cochannel 內的模組複製到 tch\_f96\_2 中並按照模組內的說明與 tch\_f96\_2 內的模組接線，如此

便可以模擬出 GSM 中 cochannel 的環境。再來從 gsmphy 目錄中，將 cc\_tch\_f48 這個模組(如圖 2-9)複製到 tch\_f96\_2 中，並取代其通道編碼的部分，如此便完成了 Full rate 4.8k bits/s traffic channel 的 GSM 系統，其系統圖如圖 2-10 所示。

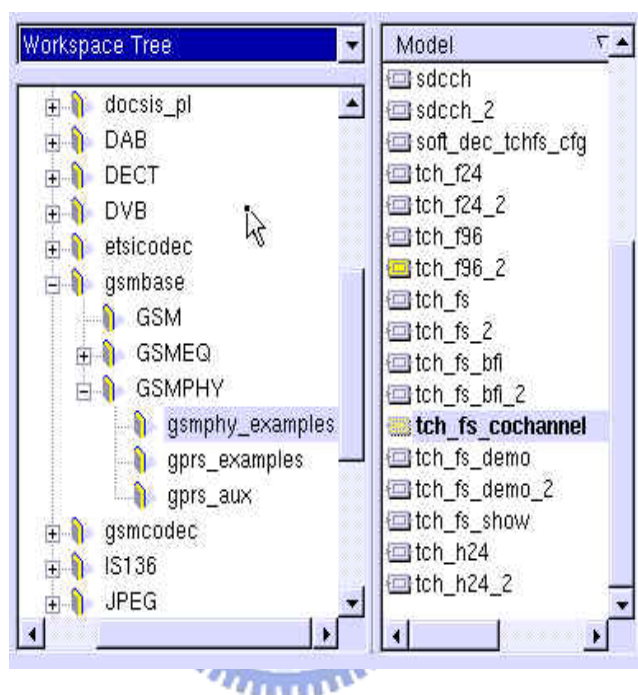


圖 2-8 tch\_f96\_2

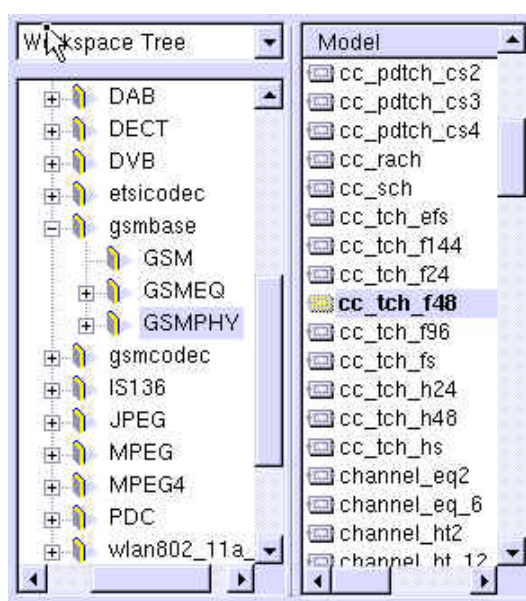


圖 2-9 cc\_tch\_f48

### 2.1.3 GSM 系統模擬的流程

GSM 系統的模組建立好後，接下來則是開始設定模擬所需的參數。CCSS 的模擬功能，提供了一個控制檔的功能，藉由這個控制檔，我們可以用 CCSS 所提供的語法，像寫簡單的 C 語言程式般來控制模擬的變數。詳細如圖 2-11 所示，我們設定 Cochannel 裡的 C/I 值為 1、4、7、10，而其餘參數則皆按照 GSM 05 系列標準來設定。

設定檔完成後，在模擬之前必須先經過 Code Generation 這個步驟，由 CCSS 將這所有的動作轉成程式碼，才能執行模擬的動作，如

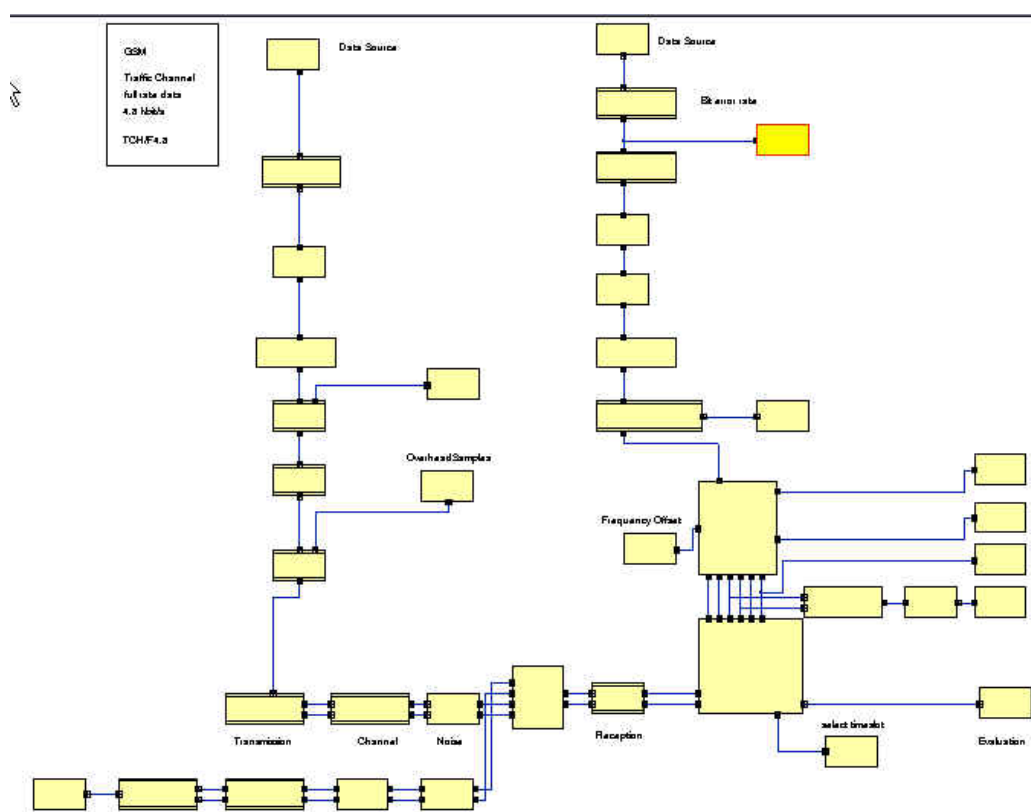


圖 2-10 GSM 系統模組

圖 2-12 所示按下右上方的 Create 鍵之後，便會開始模擬的執行，模擬執行時的畫面如圖 2-13 所示。

```

# System Studio Simulation Control File
# Copyright (c) 1998-1999 by Synopsys, Inc.  ALL RIGHTS RESERVED

set_value CarrierFrequency_Hz [ expr 900e6 ]
set_value RepeatLength [ expr 1354 ]
set_value OverHeadSamples [ expr 15 ]
set_value TSC [ expr 6 ]
set_value BurstType [ expr 0 ]
set_value RangeOfSearch [ expr 2 ]
set_value Display_On [ expr 0 ]
set_value NumBitErrors [ expr 100 ]
set_value /$env(SIM_NAME)/FreqOffset_Hz [expr 50]

foreach C_I_dB { 1 4 7 10 } {
    set_value /$env(SIM_NAME)/C_I_dB [expr ($C_I_dB)]
    run_iteration
}

```

圖 2-11 模擬設定檔

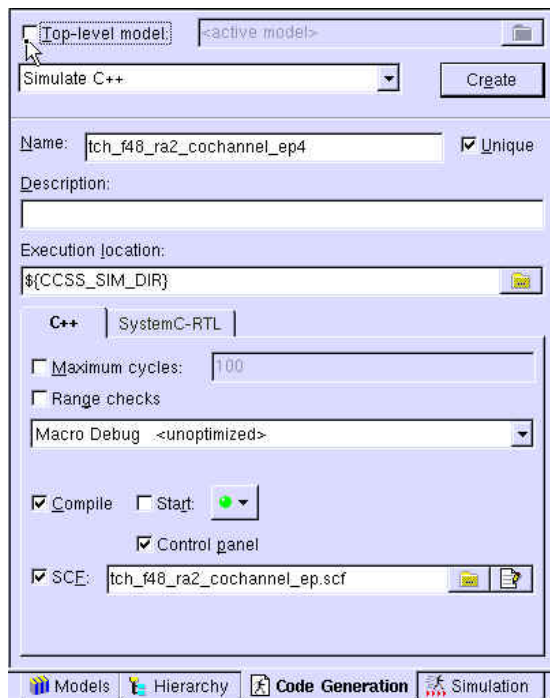


圖 2-12 Code Generation



圖 2-13 Simulation

## 2.1.4 錯誤序列的提取

CCSS 另外還提供了一個觀看模擬結果的程式；Davis[3]，其工作畫面如圖 2-14 所示。

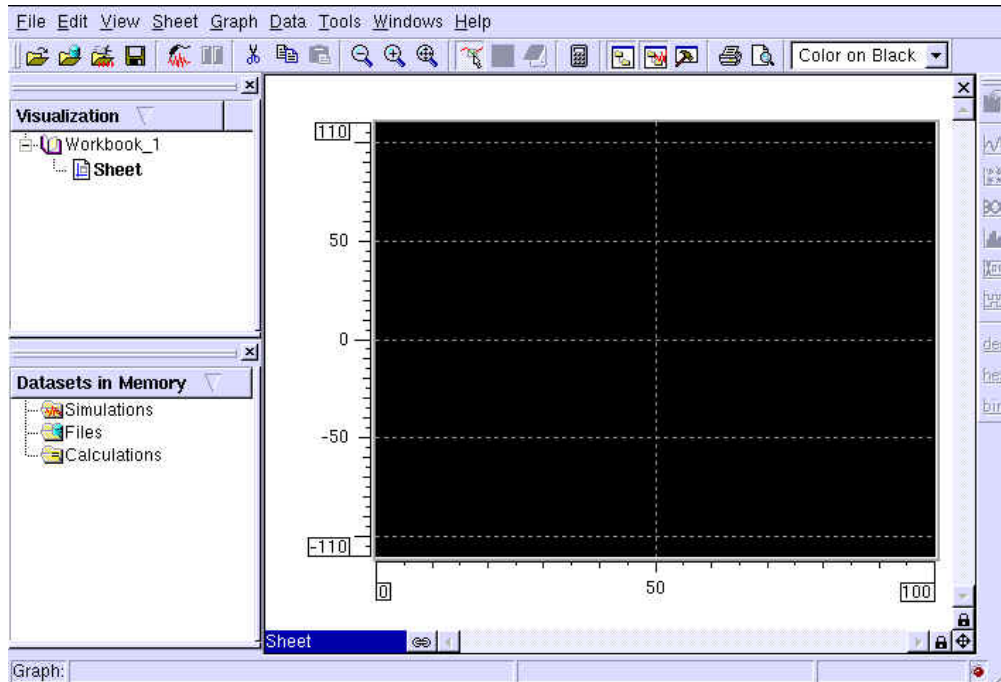


圖 2-14 Davis 工作畫面

開啟 Davis 後，得先連結到剛才所跑的模擬，方法是從左上方的 File->Open simulation，按下後會跑出如圖 2-15 的畫面。這邊可以叫出適才跑模擬時，有存取功能的模組所存入的結果，這邊儲存錯誤序列的模組名稱是 Write Signal，選取後按下 OK，便會出現在圖 2-14 左下方的欄位裡。接著如圖 2-16 所示，在載入的模擬結果上按右鍵

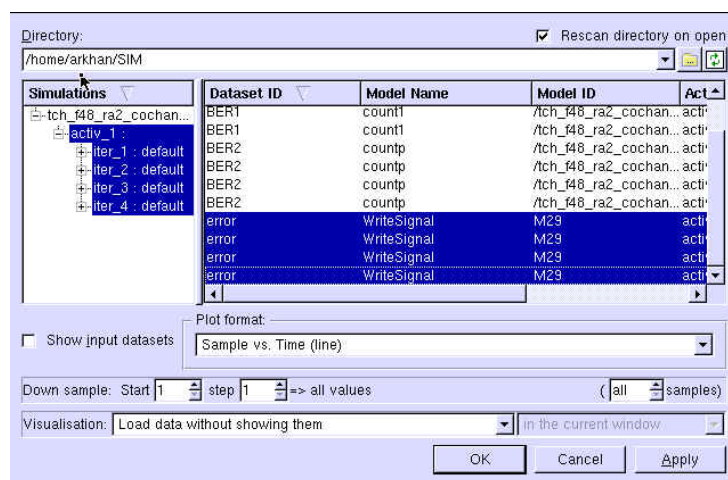


圖 2-15 載入模擬結果

選 Save Dataset As... 選擇儲存的路徑與格式，如此便完成 GSM 數據通道的錯誤序列。

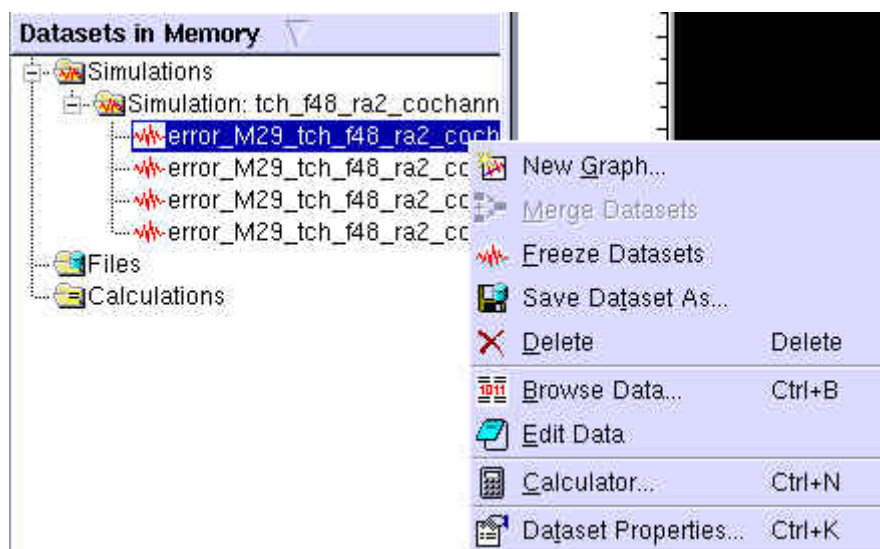


圖 2-16 儲存錯誤序列

## 2.2 有限狀態的馬可夫鏈通道模型

在今日廣泛使用的行動通訊中，假如傳送信號的頻寬遠小於通道的同調頻寬，則被傳輸的信號通常會受到叢發性錯誤的影響，這類型的通道通常歸類為記憶性通道，比較適合採用一有限狀態馬可夫通道模型來描述。其關鍵是將通道模擬為一有限狀態的馬可夫程序，而每一狀態皆有一特定對應的通道品質如錯誤機率或是訊雜比。並且由於都卜勒效應的關係，相鄰的通道狀態互相有關連性，可被描述為一階穩態馬可夫程序。在任一時間點，通道輸出只與目前時間點所在的通道狀態及通道輸入有關，而下一個通道狀態只與目前的通道狀態有關



並與通道輸出獨立。

一般最常用來描述記憶性通道的通道模型包含 Gilbert[4]及 Fritchamn model[5]，此模型最大的優點在於可利用單項的錯誤距機率分佈(error-gap distribution)，即能有效描述出記憶性通道的叢發性錯誤特性。這些通道模型將通道模擬成一階馬可夫程序，將通道描述為有限數目的通道狀態且每一狀態均各有一通道錯誤機率。以下將分別介紹 Gilbert 及 Fritchman 模型。

### 2.2.1 Gilbert Models

如圖 2-17 所示，Gilbert 模型包含兩個狀態，在狀態  $G$  的錯誤發生機率為 0，而狀態  $B$  的錯誤發生機率為  $(1-h)$ 。由狀態  $G$  到狀態  $B$  的轉移機率是  $b$ ，狀態  $B$  到狀態  $G$  的轉移機率是  $g$ 。通道狀態落在狀態  $G$  的機率是  $\frac{g}{g+b}$ ，落在狀態  $B$  的機率是  $\frac{b}{g+b}$ ，而通道的平均位元錯誤機率則是  $(1-h) \cdot \frac{b}{b+g}$

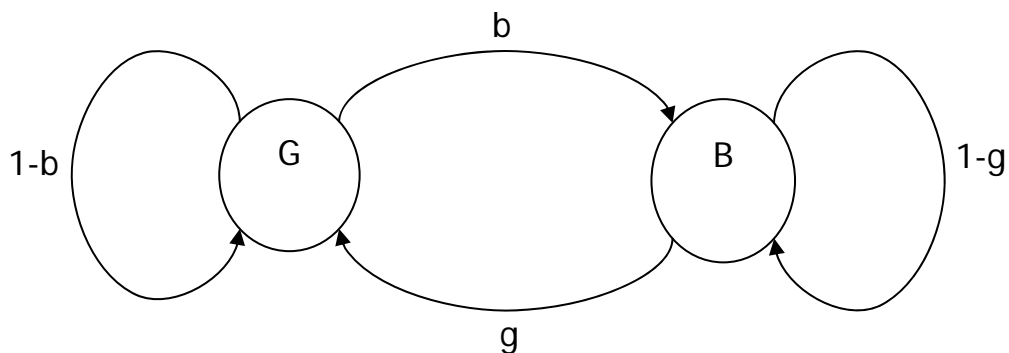


圖 2-17 Gilbert model

## 2.2.2 Fritchman model

Fritchman模型比Gilbert模型要複雜許多，它有 $N$ 個狀態( $N>2$ )，而這 $N$ 個狀態被分為兩組，其中一組的狀態數為 $k$ 且錯誤發生機率都是0，而另一組狀態數為 $N-k$ 並且一定發生錯誤。為了能夠分析Fritchman模型，我們將Fritchman模型簡化，使其有四個狀態 $G_1, G_2, G_3$ ，與 $B$ ；在狀態 $G_1, G_2, G_3$ 中發生錯誤的機率是0，而狀態 $B$ 中的錯誤機率則是1，如圖 2-18 所示。

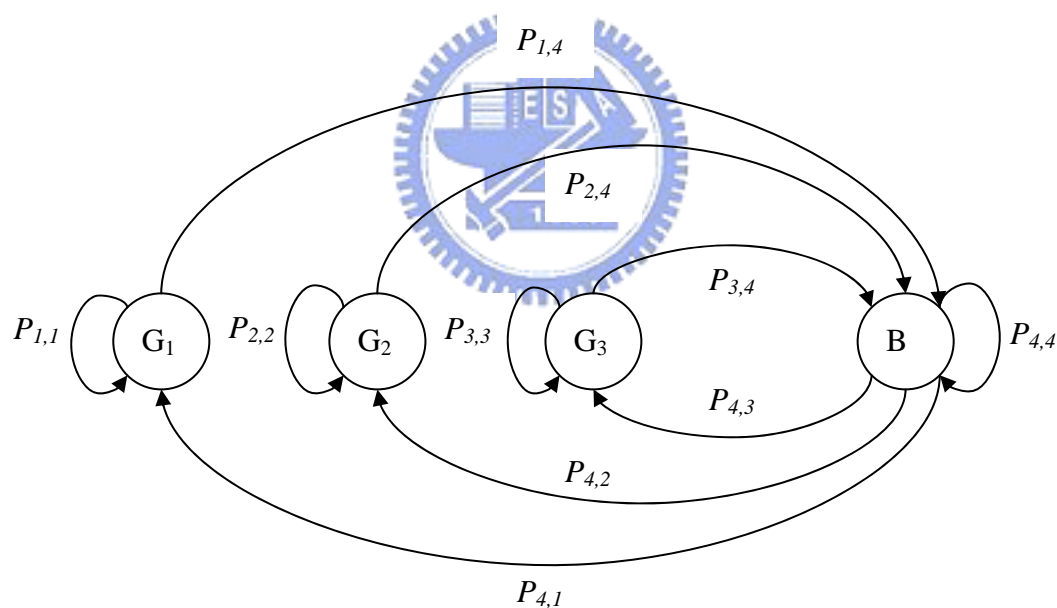


圖 2-18 Fritchman Channel

各狀態的轉移機率如下：

$$\begin{cases} P_{i,i} = \beta_i \\ P_{i,4} = 1 - \beta_i \\ P_{4,i} = \alpha_i \beta_i \\ P_{4,4} = 1 - \sum_{i=1}^3 \alpha_i \beta_i \end{cases}, i = 1, 2, 3 \quad (2.1)$$

其中模型參數 $\{\alpha_i, \beta_i\}$ 的估測將在下一部中詳細介紹。特別說明的是，Gilbert 模型可視為 Fritchman 模型的一種特例，其中 $k=2, N=3$ 。

## 2.3 Gilbert 及 Fritchman 通道模型分析

在分析馬可夫通道模型的參數中，最主要的困難在於狀態轉移機率無法直接觀測。但是若計算錯誤距機率分佈(error-gap distribution)  $P(0^m | 1)$ ，便可利用指數曲線匹配技術[6]合理近似其量所得的測錯誤距機率分佈，特定通道的模型分析便可轉換成一非線性參數最佳化預估問題，以下便介紹通道模型分析的方法。

### 2.3.1 錯誤距機率分佈

錯誤距機率分佈  $P(0^m | 1)$  是在前一個位元發生錯誤情況下，皆下來至少發生了  $m$  個無錯誤位元的機率。在許多應用中，可以證明錯誤距機率可以用一組指數曲線函數的和來近似，式子如下：

$$P(0^m | 1) = \sum_{i=1}^k \alpha_i \beta_i^m \quad (2.2)$$

其中  $k$  為無錯誤狀態的數目，若是 Gilbert model 時  $k=2$ ，Fritchman

model 時  $k=3$ 。在  $k=2$  的情況下，若將錯誤距機率分佈值取對數並繪圖，可發現其曲線可被兩條直線所近似。因此若要做曲線匹配， $\{\alpha_1, \beta_1\}$  便可選擇來近似錯誤距機率分佈中  $m$  較大的區域，而  $\{\alpha_2, \beta_2\}$  則可近似  $m$  較小的區域。於是模型分析便轉換成一種非線性參數最佳化的預估問題，其搜尋變數為  $\{\alpha_i, \beta_i\}$ 。其合適的成本函數則為錯誤距機率與其曲線匹配的平方誤差和，此一求最小匹配誤差的參數估算可以寫成如下：

$$\min_{\alpha, \beta} E = \min_{\alpha, \beta} \left\{ \sum_{m=1}^M \frac{1}{m} \left[ y(m) - \log_{10} \sum_{i=1}^k \alpha_i \beta_i^m \right] \right\} \quad (2.3)$$

其中  $M$  是錯誤序列中兩個接連錯誤間的最長區間， $y(m)$  則是錯誤距機率分佈值量測的對數值。

欲分析 Gilbert 模型，則必須滿足以下條件：

- (1)  $\alpha_1 + \alpha_2 = 1$
- (2)  $0 \leq \alpha_i \leq 1, 0 \leq \beta_i \leq 1, \text{ for } i=1, 2,$

若已估測出最佳的  $\{\alpha_1, \alpha_2, \beta_1, \beta_2\}$ ，則 Gilbert 模型的參數  $\{g, h, b\}$  可計算如下：

$$g = \frac{(1-\beta_1)(1-\beta_2)}{(1-h)} \quad (2.4)$$

$$h = \frac{\beta_1 \beta_2}{\beta_1 - \alpha_1 (\beta_1 - \beta_2)} \quad (2.5)$$

$$b = \alpha_1 (\beta_1 - \beta_2) + \frac{(1-\beta_1)(\beta_2 - h)}{(1-h)} \quad (2.6)$$

### 2.3.2 模型參數的估測

模型參數的估測包含了指數曲線匹配與 iterative gradient-descent 方法，藉由計算成本函數的梯度來自動修正模型參數，以嘗試找出最佳解。根據 steepest-descent 方法，每一個(n+1)次的模型參數修正可由一個簡單的疊代關係式：

$$\begin{aligned} &(\alpha_1(n+1), \alpha_2(n+1), \dots, \alpha_k(n+1), \beta_1(n+1), \beta_2(n+1), \dots, \beta_k(n+1)) \\ &= (\alpha_1(n), \alpha_2(n), \dots, \alpha_k(n), \beta_1(n), \beta_2(n), \dots, \beta_k(n)) - \lambda(n) \nabla E(n) \end{aligned} \quad (2.7)$$

其中  $\nabla E(n)$  為成本函數的梯度， $\lambda(n)$  為 step size。每一次疊代的成本函數梯度及 step size 計算如下：

$$\begin{aligned} \nabla E &= \left( \frac{\partial E}{\partial \alpha_1}, \frac{\partial E}{\partial \alpha_2}, \dots, \frac{\partial E}{\partial \alpha_k}, \frac{\partial E}{\partial \beta_1}, \frac{\partial E}{\partial \beta_2}, \dots, \frac{\partial E}{\partial \beta_k} \right) \quad (2.8) \\ \lambda &= \frac{\sum_{i=1}^k \left( \frac{\partial E}{\partial \alpha_i} \right)^2 + \sum_{j=1}^k \left( \frac{\partial E}{\partial \beta_j} \right)^2}{\sum_{i=1}^k \sum_{j=1}^k \left\{ \left[ \left( \frac{\partial^2 E}{\partial \alpha_i \partial \alpha_j} \right) \left( \frac{\partial E}{\partial \alpha_i} \right) \left( \frac{\partial E}{\partial \alpha_j} \right) \right] + \left[ \left( \frac{\partial^2 E}{\partial \alpha_i \partial \beta_j} \right) \left( \frac{\partial E}{\partial \alpha_i} \right) \left( \frac{\partial E}{\partial \beta_j} \right) \right] \right.} \\ &\quad \left. + \left[ \left( \frac{\partial^2 E}{\partial \beta_i \partial \alpha_j} \right) \left( \frac{\partial E}{\partial \beta_i} \right) \left( \frac{\partial E}{\partial \alpha_j} \right) \right] + \left[ \left( \frac{\partial^2 E}{\partial \beta_i \partial \beta_j} \right) \left( \frac{\partial E}{\partial \beta_i} \right) \left( \frac{\partial E}{\partial \beta_j} \right) \right] \right\} \quad (2.9) \end{aligned}$$

where

$$\frac{\partial E}{\partial \alpha_i} = -2(\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) \right) \times \left( \frac{\beta_i^m}{\sum_{l=1}^k \alpha_l \beta_l^m} \right) \right] \quad (2.10)$$

$$\frac{\partial E}{\partial \beta_i} = -2\alpha_i (\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) \right) \times \left( \frac{m \beta_i^{m-1}}{\sum_{l=1}^k \alpha_l \beta_l^m} \right) \right] \quad (2.11)$$

$$\frac{\partial^2 E}{\partial \alpha_i \partial \alpha_i} = 2(\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) + \log_{10} e \right) \times \left( \frac{\beta_i^{2m}}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \right] \quad (2.12)$$

$$\frac{\partial^2 E}{\partial \beta_i \partial \beta_i} = 2\alpha_i (\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left\{ \left( \frac{m\beta_i^{m-2}}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \times \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) \right) \right. \right. \quad (2.13)$$

$$\cdot \left. \left[ \alpha_i \beta_i^m - \left( (m-1) \left( \sum_{\substack{j=1 \\ j \neq i}}^k \alpha_j \beta_j^m \right) \right) \right] + \left[ (\log_{10} e) (m\alpha_i \beta_i^m) \right] \right\}$$

$$\left( \frac{\partial^2 E}{\partial \alpha_i \partial \beta_j} \right)_{j \neq i} = 2(\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) + \log_{10} e \right) \right. \quad (2.14)$$

$$\left. \times \left( \frac{\beta_i^m \beta_j^m}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \right]$$

$$\left( \frac{\partial^2 E}{\partial \beta_i \partial \beta_j} \right)_{j \neq i} = 2\alpha_i \alpha_j (\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) + \log_{10} e \right) \right. \quad (2.15)$$

$$\left. \times \left( \frac{m^2 \beta_i^{m-1} \beta_j^{m-1}}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \right]$$

$$\frac{\partial^2 E}{\partial \alpha_i \partial \beta_i} = -2(\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left\{ \left( \frac{m\beta_i^{m-1}}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \times \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) \right) \right. \right. \quad (2.16)$$

$$\cdot \left. \left[ \sum_{\substack{j=1 \\ j \neq i}}^k \alpha_j \beta_j^m - \left[ (\log_{10} e) (\alpha_i \beta_i^m) \right] \right] \right\}$$

$$\frac{\partial^2 E}{\partial \alpha_i \partial \beta_j} = 2\alpha_j (\log_{10} e) \sum_{m=1}^M \frac{1}{m} \left[ \left( y(m) - \log_{10} \left( \sum_{l=1}^k \alpha_l \beta_l^m \right) + \log_{10} e \right) \right. \quad (2.17)$$

$$\left. \times \left( \frac{m\beta_i^m \beta_j^{m-1}}{\left( \sum_{l=1}^k \alpha_l \beta_l^m \right)^2} \right) \right]$$

雖然 gradient-descent 方法收斂速度很快，每一次運算函數  $E$  經由計算其梯度來尋找最小值，然而成本函數不僅包含全域極小值，還有一些區域極小值，而且很容易收斂到區域極小值而非全域極小值。因此疊代運算時所設定的初始值就必須充分的靠近成本函數  $E$  的全域極小值，來盡可能避免收斂到區域極小值。



### 第三章 基於索引層級的解碼器

向量量化編碼技術目前被廣泛用在影像與聲音多媒體應用領域或是語音傳輸系統方面，因其在將訊號傳輸經過無線通道環境時，能解決傳輸頻寬或儲存容量上的限制。向量量化的編碼模式是先訓練一組能涵蓋其訊號特徵化的樣本碼書，與輸入訊號依序比對擇其中最近似的碼字，再傳送其索引值作為編碼輸出。在本章中，將推導基於索引層級的向量量化解碼器，考慮通道記憶特性及訊號源的殘餘冗息，並使用均方誤差解碼來準確估算其傳輸索引值的後驗機率。在利用上一章介紹的方法完成通道模型的分析後，進而利用估測通道模型所得的參數，推導位元錯誤機率的疊代演算，之後再考慮真實環境同時存在於單一索引內與相鄰索引間位元組合的記憶特性，準確地求得傳輸索引值的後驗機率，再據以估算其最小均方誤差的解碼輸出。

#### 3.1 向量量化編碼

向量量化系統流程如圖 3-1，首先以 LBG 演算法訓練一組能涵蓋訊號特徵變化的樣本碼書，其內容由  $2^M$  個維度為  $k$  之碼字向量  $C = \{c_0, c_1, \dots, c_{2^M-1}\}$  所組成。在編碼端，將在時間  $t$  的輸入訊號  $v_t$  依序與碼書中的碼字做比對，並選擇其中最近似的碼字。再將其索引值  $x_t = i$  轉換為  $M$ -bit 二進位格式的碼字  $b(i) = (b_{m-1}(i), b_{m-2}(i), \dots, b_0(i))$ ，並傳送作



為編碼輸出。當量化索引經由具有雜訊的通道傳送，通道輸入  $x_t$  因為通道錯誤  $e_t$  的干擾而有可能產生錯誤。因此接收到的位元組合  $b_l(y_t) = b_l(x_t) \oplus b_l(e_t), l = 0, 1, \dots, m-1$ ，其中  $\oplus$  為 modulo-2 加法運算子。雖然在一般的無線傳輸系統中，前向錯誤控制(Forward error control)可以用來保護所傳送的資訊，但是若能在不增加多餘的位元即能抵銷通道的影響，會更有效率。因此為了對抗通道雜訊，我們發展一個通道模型，並估測通道模型的參數，利用通道的記憶特性來發展一個演算法。一般解碼端大多使用傳統的碼書比對解碼，我們則使用以兩個次系統組成的最小均方誤差解碼器(SMMSE)。第一個次系統使用接收到的通道輸出序列  $y_t' = \{y_1, y_2, \dots, y_t\}$  來計算每個可能的傳輸索引值之後驗機率  $P(x_t | y_t')$ ， $x_t \in \{0, 1, \dots, M-1\}$ 。而這些機率用於第二個次系統來處理其條件平均值，則輸出的來源訊號估計值  $\hat{v}_t$  為

$$\hat{v}_t = \sum_{i=0}^{M-1} c_i P(x_t = i | y_t') \quad (3.1)$$

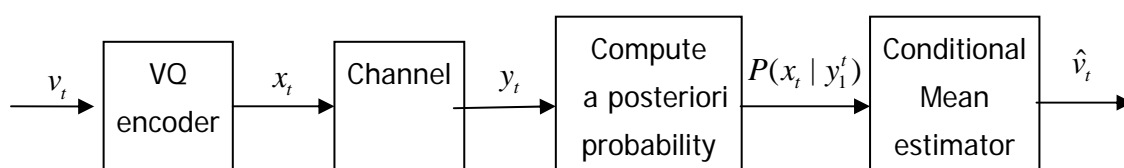


圖 3-1 向量量化系統流程圖

### 3.2 傳輸索引值之後驗機率的計算

為了能準確估算傳輸索引值的後驗機率，我們利用向量量化輸出

序列間的殘存訊息與通道的記憶特性來估算  $P(x_t | y_1^t)$  [7]。首先經由貝氏定理展開  $P(x_t | y_1^t)$ ：

$$P(x_t | y_1^t) = \frac{P(x_t, y_1^t)}{\sum_{x_t=0}^{M-1} P(x_t, y_1^t)} = \frac{\alpha_t(x_t)}{\sum_{x_t=0}^{M-1} \alpha_t(x_t)} \quad (3.2)$$

， $\alpha_t(x_t)$  稱為前向機率。我們假設訊號經過量化編碼後的索引序列，可以用一階馬可夫隨機過程來模擬，其相鄰索引間的移轉機率為  $P(x_t | x_{t-1})$ 。接著我們採用一階馬可夫模型模擬其通道特性，則通道輸出  $y_t$  只受到通道輸入  $x_t$ ， $x_{t-1}$  與通道輸出  $y_{t-1}$  影響，而通道輸入  $x_t$  則與所有之前的通道輸出  $y_1^{t-1}$  無關。如此， $\alpha_t(x_t)$  可被推導為：

$$\begin{aligned} \alpha_t(x_t) &= \sum_{x_{t-1}=0}^{M-1} P(x_t, x_{t-1}, y_t^t, y_1^{t-1}) \\ &= \sum_{x_{t-1}=0}^{M-1} P(y_t | x_t, x_{t-1}, y_1^{t-1}) P(x_t | x_{t-1}) P(x_{t-1}, y_1^{t-1}) \end{aligned} \quad (3.3)$$

其中， $\alpha_{t-1}(x_{t-1}) = P(x_{t-1}, y_1^{t-1})$ ，因此  $\alpha_t(x_t)$  可以被疊代運算，加快了計算的速度。而其前向機率的初始值  $\alpha_0(x_0)$  如下計算：

$$\begin{aligned} \alpha_0(x_0) &= P(y_0 | x_0) P(x_0) \\ &= P(x_0, y_0) \end{aligned} \quad (3.4)$$

SMMSE 解碼的效果取決於如何將通道錯誤特性引入至通道移轉機率  $P(y_t | x_t, x_{t-1}, y_1^{t-1})$  的運算中，而如何選擇一個適當的通道模型來描述真實通道的行為也是一個關鍵。下面，我們將針對 Gilbert 模型與 Fritchman 模型，提出計算  $P(y_t | x_t, x_{t-1}, y_1^{t-1})$  的方法。

### 3.2.1 通道移轉機率的計算

為了計算  $P(y_t | x_t, x_{t-1}, y_1^{t-1})$ ，我們做了以下假設：(i) 通道錯誤  $e_t$  與通道輸入  $x_t$  兩者互為獨立。(ii) 在時間  $t$  的通道錯誤  $e_t$  與在時間  $t-1$  的通道錯誤  $e_{t-1}$  互為獨立。(iii) 在時間  $t$  的通道錯誤  $e_t$ ，其中的位元錯誤有互相的關連性，即  $e_t$  中的第  $l$  個位元  $b_l(e_t)$  與  $\{b_{l-1}(e_t), b_{l-2}(e_t), \dots, b_0(e_t)\}$  有互相關連性。基於以上三點假設，

$P(y_t | x_t, x_{t-1}, y_1^{t-1})$  可被寫為：

$$P(e_t) = \prod_{l=0}^{m-1} P(b_l(e_t) | b_{l-1}(e_t), b_{l-2}(e_t), \dots, b_0(e_t)) \quad (3.5)$$

接著我們假設通道模型可被模擬成一個狀態數為二的馬可夫鏈模型，則通道錯誤  $e_t$  中的位元錯誤  $b_l(e_t)$  將只與前一個位元錯誤  $b_{l-1}(e_t)$  有關。於是通道轉移機率可被寫為：

$$P(e_t) = \pi \prod_{l=1}^m P_e(b_{m-l}(e_t)) \mathbf{1} \quad (3.6)$$

其中  $\pi$  為初始狀態機率， $\mathbf{1}$  為一個向量元素皆為 1 的向量， $P_e(b_{m-l}(e_t))$  則是計算每個通道位元錯誤的可能性機率。從第二章的介紹中，Gilbert 通道模型與 Fritchman 通道模型皆為一狀態數為二的馬可夫鏈模型，當採用 Gilbert 通道模型時，初始狀態機率  $\pi$  如下：

$$\pi = \left( \frac{g}{g+b} \quad \frac{b}{g+b} \right) \quad (3.7)$$

而  $P_e(b_{m-l}(e_t))$  的矩陣則為：

$$P_e(0) = \begin{pmatrix} 1-b & bh \\ g & (1-g)h \end{pmatrix} \quad (3.8)$$

$$P_e(1) = \begin{pmatrix} 0 & b(1-h) \\ 0 & (1-g)(1-h) \end{pmatrix} \quad (3.9)$$

若採用 Fritchman 通道模型時，初始狀態機率  $\pi$ ：

$$\pi = \left( \varepsilon \frac{\alpha_1 \beta_1}{1 - \beta_1} \quad \varepsilon \frac{\alpha_2 \beta_2}{1 - \beta_2} \quad \varepsilon \frac{\alpha_3 \beta_3}{1 - \beta_3} \quad \varepsilon \right) \quad (3.10)$$

而  $P_e(b_{m-l}(e_l))$  的矩陣為：

$$P_e(0) = \begin{pmatrix} \beta_1 & 0 & 0 & 0 \\ 0 & \beta_2 & 0 & 0 \\ 0 & 0 & \beta_3 & 0 \\ \alpha_1 \beta_1 & \alpha_2 \beta_2 & \alpha_3 \beta_3 & 0 \end{pmatrix} \quad (3.11)$$

$$P_e(1) = \begin{pmatrix} 0 & 0 & 0 & 1 - \beta_1 \\ 0 & 0 & 0 & 1 - \beta_2 \\ 0 & 0 & 0 & 1 - \beta_3 \\ 0 & 0 & 0 & 1 - \sum_{i=1}^3 \alpha_i \beta_i \end{pmatrix} \quad (3.12)$$

如此計算每個接收位元的所有可能性之後，完成通道轉移機率的計算，再加入訊號源殘餘資訊，經由解碼器作 MMSE 評估計算來完成解碼。

有別於一般向量量化系統中，因使用插置(interleaving)來假設無線通道沒有記憶性，來設計解碼演算法。本章所介紹的演算法在於將通道環境模型化，利用通道的記憶特性來設計解碼演算法。此方法可以減少一般演算法所帶來大量的延遲，充分利用通道錯誤的資訊。下一章，將介紹基於位元層級的解碼演算法，其運算速度將會比索引層級的演算法更快速。

## 第四章 基於位元層級的解碼器

在上一章所介紹的索引層級解碼演算法，已利用通道的部分記憶特性及訊源的殘餘冗息，但僅能適用於較簡單的 Gilbert 通道模型環境。因此在本章中，我們將推導一個基於位元層級的解碼演算法，充分利用通道的記憶特性來補償通道雜訊所造成的影響，並且能推展應用至較準確的 Fritchman 通道模型。此外我們還可利用事前針對大量訓練語料分析所得的位元層級殘餘冗息，進一步提升解碼器的效能。

#### 4.1 位元後驗機率的疊代演算法

系統傳輸流程如圖 4-1 所示，考慮傳送一個特定參數  $v_t$  至一雜訊通道，首先將參數  $v_t$  進行  $M$  位元的量化處理；逐一比對由  $2^M$  個二維碼字組成的樣本碼書  $C = [c_0, c_1, \dots, c_{2^M-1}]$  找出最近似的碼字  $c_i$ ，再將其索引值  $x_i = i$  以二進位格式表示成  $M$  個位元  $\{x_{t,m}, m = 0, 1, \dots, M-1\}$ 。

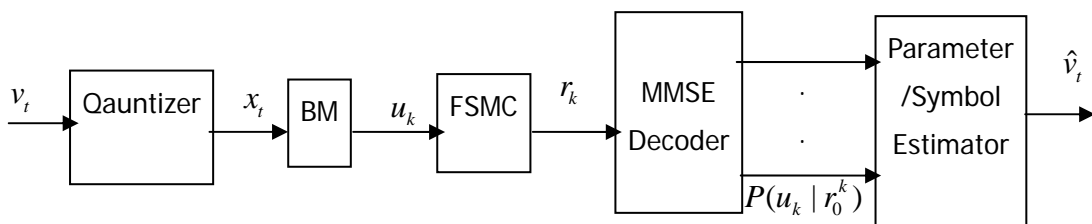


圖 4-1 系統傳輸流程圖

為便於說明，我們定義第  $m$  個位元  $x_{t,m} = u_k$ ，其中  $k = tM + m$ ，

$0 \leq k \leq N_c - 1$ ， $N_c = MT$ ， $T$  為該參數序列的總數。而接收端受到通道錯

誤的影響，可能將原始位元  $u_k$  誤判為  $r_k$ ，我們觀察已接收的位元序列  $r_0^k = \{r_0, r_1, \dots, r_k\}$ ，並估測每個傳送位元的後驗機率  $P(x_{t,m} = \pm 1 | r_0^k)$ ，再據以計算該特定參數的條件平均值為

$$\hat{v}_t = \sum_{x_t=0}^{2^M-1} c_{x_t} P(x_t | r_0^k) \approx \sum_{x_t=0}^{2^M-1} c_{x_t} \prod_{m=0}^{M-1} P(x_{t,m} | r_0^k)。$$

此解碼機制的成敗關鍵在於如何有效地整合訊源殘餘冗息與通道記憶特性，以期精確估算其位元後驗機率  $P(x_{t,m} | r_0^k) = P(u_k | r_0^k)$  [8]。

首先利用貝氏定理展開位元後驗機率如下：

$$\begin{aligned} P(u_k | r_0^k) &= \frac{P(u_k, r_0^k)}{P(r_0^k)} \\ &= \frac{\alpha_k(u_k)}{\sum_{u_k} \alpha_k(u_k)} \end{aligned} \quad (4.1)$$

其中  $\alpha_k(u_k) \equiv P(u_k, r_0^k)$  稱為前向機率。若假設每個位元發生機率僅與臨近前一個位元有關聯，則前向機率可簡化為以下的疊代關係：

$$\begin{aligned} \alpha_k(u_k) &= \sum_{u_{k-1}} P(u_k, u_{k-1}, r_k, r_0^{k-1}) = \sum_{u_{k-1}} P(u_k, r_k | u_{k-1}, r_0^{k-1}) P(u_{k-1}, r_0^{k-1}) \\ &= \sum_{u_{k-1}} P(u_k, u_k \oplus r_k | u_{k-1}, u_{k-1} \oplus r_{k-1}) P(u_{k-1}, r_0^{k-1}) \\ &= \sum_{u_{k-1}} P(u_k | u_{k-1}) P(e_k | e_{k-1}) \alpha_{k-1}(u_{k-1}) \end{aligned} \quad (4.2)$$

其中  $e_k = u_k \oplus r_k$  是時間  $k$  時所發生的通道位元錯誤。由於通道的記憶性，相鄰的位元錯誤  $e_k$  與  $e_{k-1}$  有關連性，我們可以藉由引入位元錯誤轉移機率  $P(e_k | e_{k-1})$  更精確地估算其位元後驗機率  $P(u_k | r_0^{N_c-1})$ 。更值得注意的是在(4.2)中，只利用位元層級的訊源冗息於其  $P(u_k | u_{k-1})$  的計算過程，進一步的改良方法將在下一節中介紹。

在上述的  $P(u_k | r_0^k)$  疊代演算法中，能否有效整合通道記憶特性的關鍵在於位元錯誤移轉機率  $P(e_k | e_{k-1})$  的計算。主要是透過通道模型分析，我們可以引入通道模型參數於其  $P(e_k | e_{k-1})$  計算之中，方能補償該特定通道因雜訊所衍生的不良影響。首先針對目前通道狀態  $c_k$  其前一狀態  $c_{k-1}$  做展開，而推導位元錯誤移轉機率如下：

$$\begin{aligned}
 P(e_k | e_{k-1}) &= \frac{\sum_{c_k} \sum_{c_{k-1}} P(e_k, e_{k-1}, c_k, c_{k-1})}{\sum_{c_{k-1}} P(e_{k-1}, c_{k-1})} \\
 &= \frac{\sum_{c_k} \sum_{c_{k-1}} P(e_k | c_k) P(e_{k-1} | c_{k-1}) P(c_k | c_{k-1}) P(c_{k-1})}{\sum_{c_{k-1}} P(e_{k-1} | c_{k-1}) P(c_{k-1})}
 \end{aligned} \tag{4.3}$$

當採用吉爾博(Gilbert) 通道模型時，其中位元錯誤機率  $P(e_k | c_k)$  與通道狀態移轉機率  $P(c_k | c_{k-1})$ ，可用估算所得的通道模型參數  $\{b, g, h\}$  表示為：

$$P(e_k | c_k) = \begin{cases} 1 & \text{if } e_k = 0 \text{ and } c_k = G \\ 0 & \text{if } e_k = 1 \text{ and } c_k = G \\ h & \text{if } e_k = 1 \text{ and } c_k = B \\ 1-h & \text{if } e_k = 0 \text{ and } c_k = B \end{cases} \tag{4.4}$$

$$P(c_k | c_{k-1}) = \begin{cases} 1-b & \text{if } (c_k, c_{k-1}) = (G, G) \\ b & \text{if } (c_k, c_{k-1}) = (B, G) \\ g & \text{if } (c_k, c_{k-1}) = (G, B) \\ 1-g & \text{if } (c_k, c_{k-1}) = (B, B) \end{cases} \tag{4.5}$$

當採用傅立門(Fritcham) 通道模型，其位元錯誤機率與通道狀

態移轉機率則可表示如下：

$$P(e_k | c_k) = \begin{cases} 1 & \text{if } e_k = 0 \text{ and } c_k = G_1, G_2, \text{ or } G_3 \\ 0 & \text{if } e_k = 1 \text{ and } c_k = G_1, G_2, \text{ or } G_3 \\ 1 & \text{if } e_k = 1 \text{ and } c_k = B \\ 0 & \text{if } e_k = 0 \text{ and } c_k = B \end{cases} \quad (4.6)$$

$$P(c_k | c_{k-1}) = \begin{cases} P_{i,i} = \beta_i & \text{if } (c_k, c_{k-1}) = (G_i, G_i) \\ P_{i,4} = 1 - \beta_i & \text{if } (c_k, c_{k-1}) = (B, G_i) \\ P_{4,i} = \alpha_i \beta_i & \text{if } (c_k, c_{k-1}) = (G_i, B) \\ P_{4,4} = 1 - \sum_{i=1}^3 \alpha_i \beta_i & \text{if } (c_k, c_{k-1}) = (B, B) \end{cases}, \text{ 其中 } i = 1, 2, 3. \quad (4.7)$$

據此便可將通道特性引入前向機率之疊代演算法，以便計算位元後驗機率，進而提供量化系統在設計通道匹配的錯誤控制及其性能評估之用。



## 4.2 索引層級訊源冗息的整合

前一節所使用的演算法，僅利用到前後相鄰兩位元間的殘餘冗息，本節則提出一個利用相鄰兩索引間殘餘冗息的方法，以期提高對抗傳輸錯誤的效果。

### 4.2.1 索引通道移轉機率之計算

當我們計算出每個位元的後驗機率  $P(u_k | r_0^{N_c-1})$  之後，可以得到該傳送位元的對數相似比率(log-likelihood ratio, LLR)通常表示為：

$$L_0(u_k) = \ln \frac{P(u_k = 1 | r_0^k)}{P(u_k = 0 | r_0^k)} \quad (4.8)$$



取得  $L_0(u_k)$  值之後，傳送位元的硬性判定為：

$$\tilde{u}_k = \text{sign}[L_0(u_k)] \quad (4.9)$$

而其相對應的位元錯誤機率為：

$$P_{e0}(u_k) = \frac{1}{1 + \exp|L_0(u_k)|} \quad (4.10)$$

根據以上資訊，我們便可得到位元通道移轉機率如下：

$$P(\hat{x}_{t,m} | x_{t,m}) = \begin{cases} 1 - P_{e0}(u_k) & \text{if } \hat{x}_{t,m} = x_{t,m} \\ P_{e0}(u_k) & \text{if } \hat{x}_{t,m} \neq x_{t,m} \end{cases} \quad (4.11)$$

於是索引  $x_t$  的通道轉移機率將為：

$$P(y_t | x_t) = \prod_{m=0}^{M-1} P(\hat{x}_{t,m} | x_{t,m}), \quad x_t \in \{0, 1, \dots, 2^M - 1\} \quad (4.12)$$

此項提供了  $2^M$  種可能的傳送索引值  $x_t$  最後經解碼判定為索引  $y_t = \hat{x}_t$  的機率。



## 4.2.2 事前消息的計算

在一般的情況，通常把量化輸出的索引序列模擬成一個  $N$  階馬可夫程序 ( $M$ th-order Markov process)。量化索引序列若視為  $N$  階馬可夫程序，意謂著現在發生的事件僅與過去剛發生的  $N-1$  個事件存在相關特性，即

$$P(x_t | x_0^{t-1}) = P(x_t | x_{t-N}^{t-1})$$

其中  $x_0^{t-1} \triangleq \{x_0, x_1, \dots, x_{t-1}\}$ 。亦即此序列在時間上有可回溯  $N-1$  個時間點的記憶特性。

為了找出適當的馬可夫階數，通常先以訓練(training)的方式，主要是將語音資料庫的大量語料做量化處理，然後計算每一量化索引的發生次數或相鄰兩量化索引共同發生的次數，進而求得索引序列的相關機率，如  $P(x_i)$ 、 $P(x_i|x_{i-1})$ ，或是更高階的條件機率。我們稱  $P(x_i)$  為零階事前消息 (0th order a priori knowledge)，在統計上的描述為零階的馬卡夫程序，表示為一個沒有記憶性的程序。若假設每一個索引的發生機率相同， $P(x_i)=1/2^M$ ，則稱之為無事前消息 (no a priori knowledge)。除此之外我們稱  $P(x_i|x_{i-1})$  為一階事前消息 (first-order a priori knowledge)，因其代表了一階的馬可夫程序。至於馬卡夫程序階數的最佳設定，必須參考三項要點：1) 量化索引的殘餘冗息，2) 運算複雜度，及 3) 效能和運算複雜度之間的衡量取捨。

### 4.2.3 量化索引事後機率的計算

在接收端，首要任務是準確估測每一個量化索引  $x_i$  的事後機率(a posteriori probabilities)。而其計算方法，又會因為索引序列的馬可夫階數做不同假設而有差異[9]。

假設量化索引序列的索引間仍存在有一階的殘餘相關性，我們可將此一階的索引事前訊息也納入系統設計考量中，在接收端，我們可利用的最大訊息量來自目前已接收到的索引序列  $y'_0 \triangleq \{y_0, y_1, \dots, y_i\}$ 。因

此利用大量語料訓練而得的事前機率  $P(x_t | x_{t-1})$ ，配合索引通道轉移機率  $P(y_t | x_t)$ ，我們可以求得個別索引的事後機率，並以遞迴的形式呈現如下：

$$P(x_t | y_0^t) = C \cdot P(y_t | x_t) \cdot \sum_{x_{t-1} \in J} P(x_t | x_{t-1}) \cdot P(x_{t-1} | y_0^{t-1}) \quad (4.13)$$

其中  $C$  為正規化常數：

$$C = \frac{1}{\sum_{x_t \in J} P(y_t | x_t) \cdot \sum_{x_{t-1} \in J} P(x_t | x_{t-1}) \cdot P(x_{t-1} | x_0^{t-1})} \quad (4.14)$$

其中  $J = \{0, 1, \dots, 2^M - 1\}$ 。常數  $C$  是用來正規化事後機率，而使得

$$\sum_{x_t \in J} P(x_t | y_t) = 1。$$

#### 4.2.4 參數估測



若能順利取得每一個傳送索引的事後機率  $P(x_t | y_0^t)$ ， $x_t \in J$ ，最後一個步驟就是根據事後機率進行參數估測。就一般語音參數而言，通常使用最小均方錯誤準則（minimum mean square error criterion）來估測參數。其估測準則，重建後的參數  $\hat{v}_t$  如下所示：

$$\hat{v}_t = \sum_{x_t \in J} c_{x_t} \cdot P(x_t | y_0^t)$$

其中的  $c_{x_t}$  為索引  $x_t$  所對應的量化參數值。

## 第五章 實驗模擬結果與分析

在前面的章節裡，我們介紹了基於索引層級和基於位元層級的兩種用於向量量化系統的最小均方誤差解碼演算法。在這一章中，進一步將演算法應用於系統模擬，期望藉由模擬的結果驗證演算法之正確性，並比較兩種演算法的優劣。此外在第二章所介紹的通道模型匹配技術，也將在本章節中驗證匹配的效果，證明實驗模擬中所使用的模型參數能夠與 GSM 數據通道的錯誤序列匹配。

本章的內容分為三小節來進行實驗模擬與結果之分析討論。第一節將先進行通道模型參數的估測，將所製作的 GSM 數據通道錯誤序列依通道模型匹配技術估測模型參數，並證明所估測的模型參數與真實通道匹配。第二節將通道環境設定為 Gilbert 通道模型，並使用 Gilbert 通道模擬產生的錯誤序列，來進行基於位元層級、基於索引層級及混合層級解碼演算的系統模擬，並對這三種演算法作效能評比與分析。第三節則將通道環境設定為 Fritchman 通道模型，採用 Fritchamn 通道模擬產生的錯誤序列，來模擬基於位元層級的演算法之系統架構，同時與使用 Gilbert 通道模型來做解碼運算的效能評比與分析。第四節則將 GSM 數據通道錯誤序列使用在系統模擬上，並探討各種演算法的效能與分析。

## 5.1 通道模型參數的估測

### 5.1.1 參數估測之步驟說明

在製作完成 GSM 數據通道的位元錯誤序列後，由於 CCSS 所儲存檔案格式的相容問題，必須先經過轉換，將原本所存的  $\{0, 1\}$  轉換成  $\{-1, 1\}$ ，才能作為系統模擬之用。接下來則是計算錯誤序列中  $P(0^m | 1)$  的機率，並將之記錄下來。在得到所有情況的  $P(0^m | 1)$  之後，經由第二章所介紹的指數曲線匹配技術，可估測出所設定的通道模型參數。從第二章可以知道在估測時很容易收斂至區域極小值，為避免此情況發生，我們將初始值設定如下：

$$\alpha_1 = 0.9; \quad \beta_1 = 0.6; \quad \beta_2 = 0.9992$$

以上是針對做 Gilbert 通道模型參數估測的初始值。若是做 Fritchman 通道模型參數估測，其初始值設定如下：

$$\alpha_1 = 0.6; \quad \beta_1 = 0.6;$$

$$\alpha_2 = 0.08; \quad \beta_2 = 0.9984;$$

$$\alpha_3 = 0.006; \quad \beta_3 = 0.9974;$$

為觀察估測出的參數與 GSM 位元錯誤序列的錯誤距機率分佈  $y(m)$  是否匹配，我們將估測的參數代入下列式子計算：

$$f_0(m) = \sum_{i=1}^k \alpha_i \beta_i^m \quad (5.1)$$

並將  $f_0(m)$  與  $y(m)$  取對數繪圖觀察，我們設定了 C/I

(Carrier-to-Interference)值分別為1、4、7、10 的四種通道環境，

其匹配結果分別如圖 5-1、圖 5-2、圖 5-3、圖 5-4。

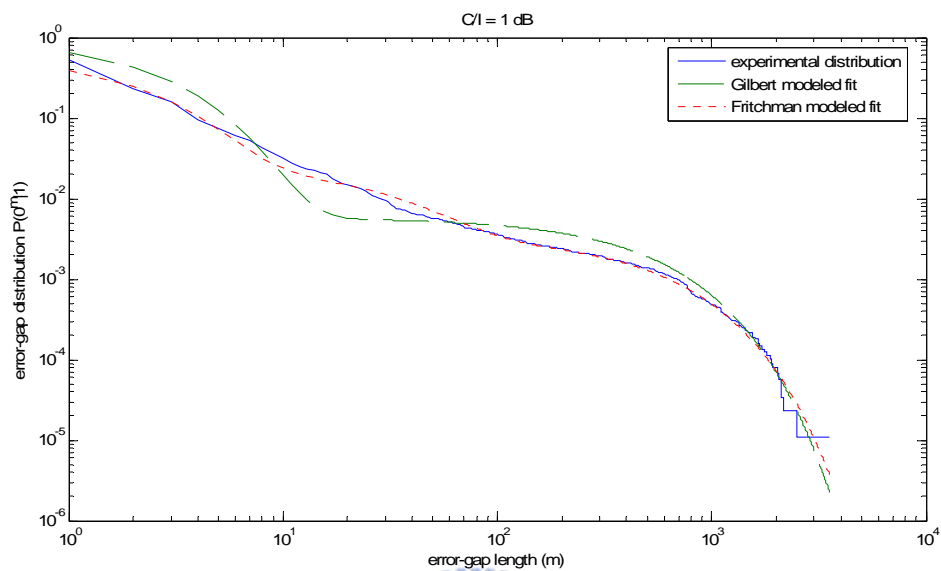


圖 5-1 C/I=1 dB

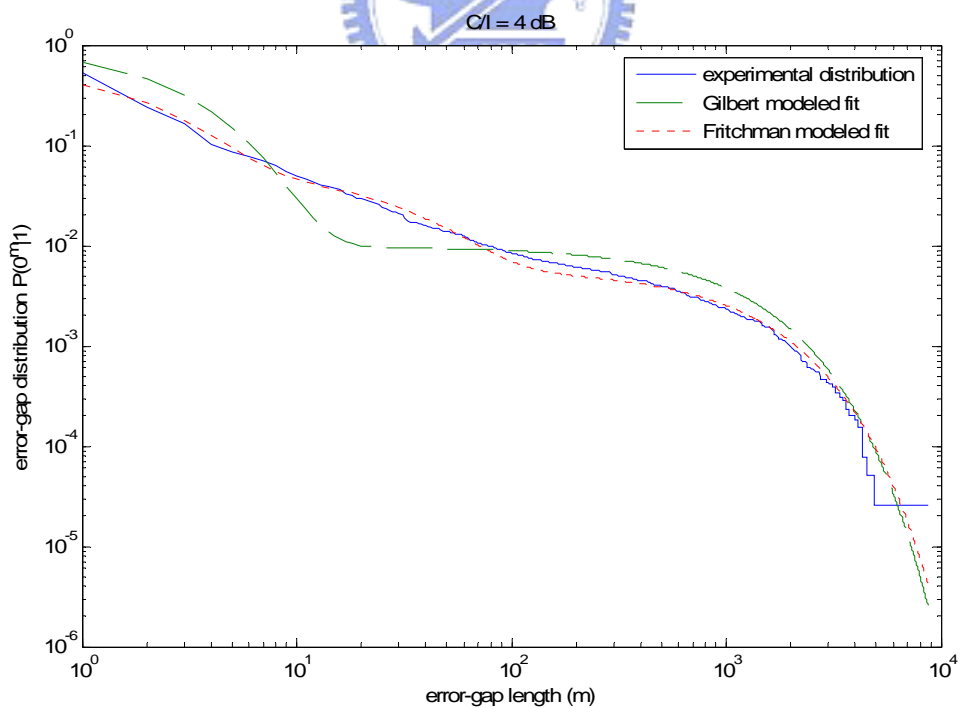


圖 5-2 C/I=4 dB

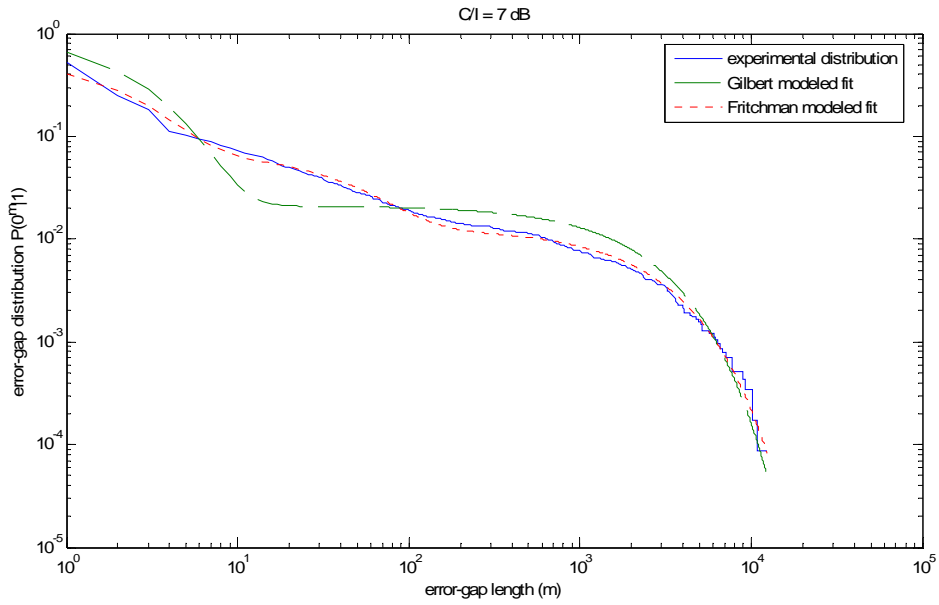


圖 5-3 C/I=7 dB

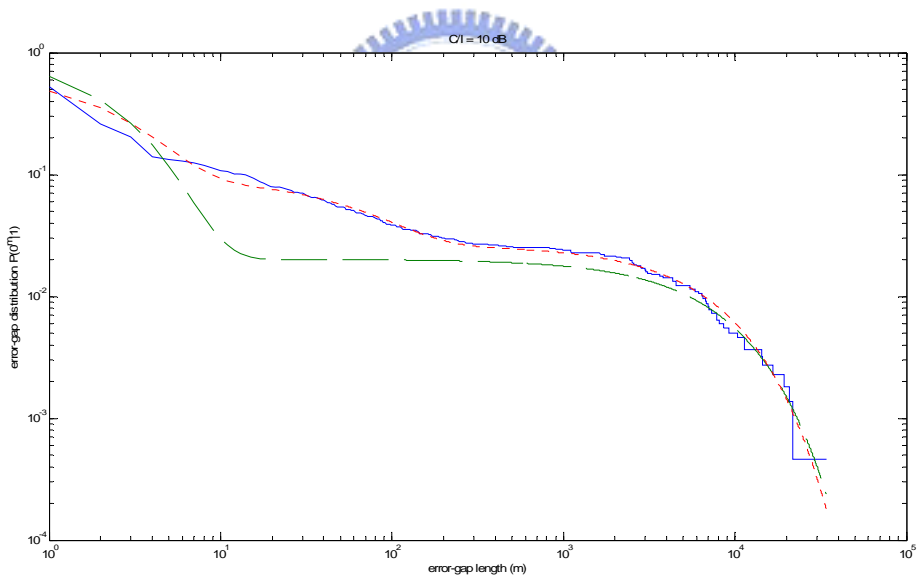


圖 5-4 C/I=10 dB

### 5.1.2 參數估測結果分析

由以上四張實驗結果圖，可看出以Fritchman通道模型去估測GSM數據通道位元錯誤序列，比用Gilbert通道模型來估測還要匹配。可

以預期的是，在作系統模擬時，當通道環境設為 GSM 數據通道時，用 Fritchman 通道模型參數來計算後驗機率，其效能會比使用 Gilbert 通道模型參數來得好。

在估測過程中，會發現成本函數的收斂時常掉入區域極小值中，需要重複的經由上次估測的結果中，調整下一次估測的初始值，方能使成本函數的收斂至全域極小值。

## 5.2 Gilbert 通道模型環境之系統模擬

### 5.2.1 系統模擬之步驟說明

此節的內容中，主要結合第三與第四章所介紹的演算法進行系統模擬與結果分析。系統中訊號源  $v_t$  取自一階自迴歸程序 (first-order autoregressive process)，且其訊號源之相關因子 (correlation factor) 採用  $\rho_v = 0.9$ ，訊號源之變異數則設定為  $\sigma_v^2 = 1$ 。關係式表示為：

$$v_t = \rho_v v_{t-1} + n_t \quad (5.2)$$

其中  $n_t$  表示均值為零的白色高斯雜訊取樣。每次的模擬過程中，由訊號源產生出 400000 點的資料作為系統之輸入，接著將產生出之訊號源送進向量量化器中，每四個點的資料作為一個向量，經由量化器量化後，再以四個位元來傳送其索引值。在得到量化索引的位元資料



後，將其通過二位元相位鍵移(BPSK)調變，把每一位元由原來的 $\{0, 1\}$ 映對成 $\{1, -1\}$ 。接著便將訊號經由通道傳送至接收端，模擬的通道環境使用 Gilbert 通道，即在傳送的過程中，加入由 Gilbert 通道模擬產生的位元錯誤序列做通訊干擾。

在接收端，接收器所接收到的不再是當初傳送時的位元 $\{1, -1\}$ ，而是經由雜訊擾亂後的實數值，將其輸入至解碼器。需注意在計算後驗機率的過程中，需以格子構架之區塊長度為演算法之實行間隔區塊，這裡設定每一百個向量作為一個區塊。其實驗結果如圖 5-5 所示。

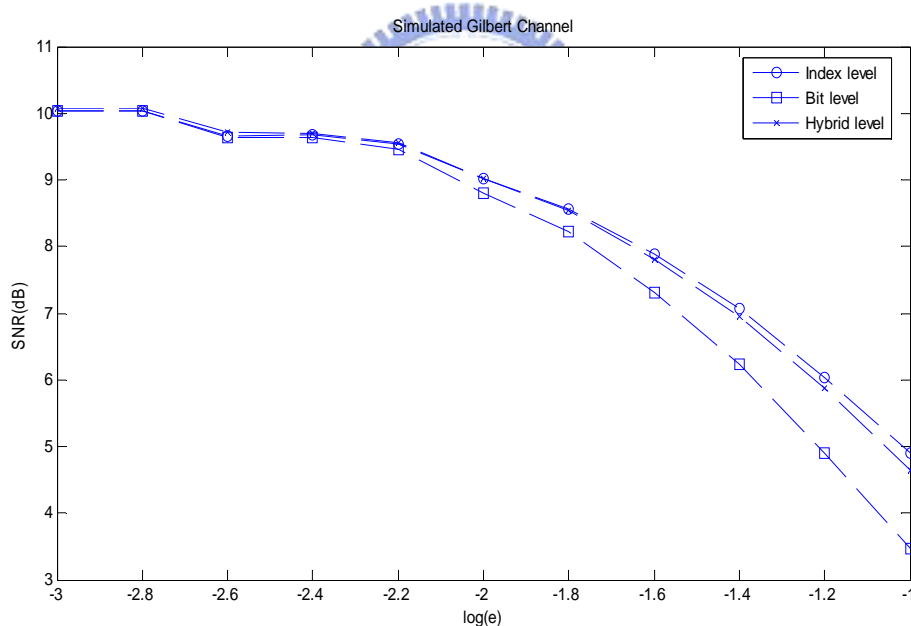


圖 5-5 Gilbert 通道之系統模擬

## 5.2.2 結果分析

圖 5.1 中，系統之評估為使用訊號源參數  $v_i$  和重建參數  $\hat{v}_i$  之間的

訊號雜訊比 (signal-to-noise ratio, SNR) 作為衡量：

$$SNR = 10 \log_{10} \frac{v_t^2}{(v_t - \hat{v}_t)^2} \text{ [dB]} \quad (5.3)$$

由實驗結果圖中，可見當通道情況較差時，即訊號能量對通道雜訊能量比率較低時，明顯地基於位元層級演算法衰退的最快。若在基於位元層級演算法中加入額外的訊源殘餘冗息，也就是混合層級演算法，系統效能即明顯的提升接近基於索引層級的演算法。相較於基於索引層級演算法，由於基於位元層級演算法僅利用到前後位元間的關連性，其計算時間會比計算一階事前消息還要少，因此整個系統模擬所需的時間比基於索引層級相對減少。但因為所利用的資訊較少，其效能比基於索引層級要少很多。因此我們利用混合層級的方法，利用一階事前消息，能有效提高效率，但也因此犧牲一些系統模擬的速度。

## 5.3 Fritchman 通道之系統模擬

### 5.3.1 系統模擬之步驟

前一節的系統架構，主要是在 Gilbert 通道環境上模擬。從第一節的通道模型參數估測中，可看出 GSM 數據通道以 Fritchman 通道模型較能匹配其通道環境。因此本節中的通道環境將設定為 Fritchman 通道模型，並分別以 Fritchman 通道模型參數及 Gilbert 通道模型參數來計算後驗機率。系統的架構基本上與前節相同，訊號源產生

100000 點資料，每四點為一個向量送進向量量化器，指定其索引值後，傳送其索引值的二位元表示值進入通道，其表示值在進入通道前已經過 BPSK 調變。傳送的資料經過模擬產生的 Fritchman 通道錯誤序列干擾後，送到接收端。在接收端計算後驗機率時，分別以事前估測模擬產生的 Fritchman 通道錯誤序列，所得到的 Gilbert 通道模型參數以及 Fritchman 通道模型參數來做計算及效果評比。基於位元層級演算法的實驗結果如圖 5-6 所示：

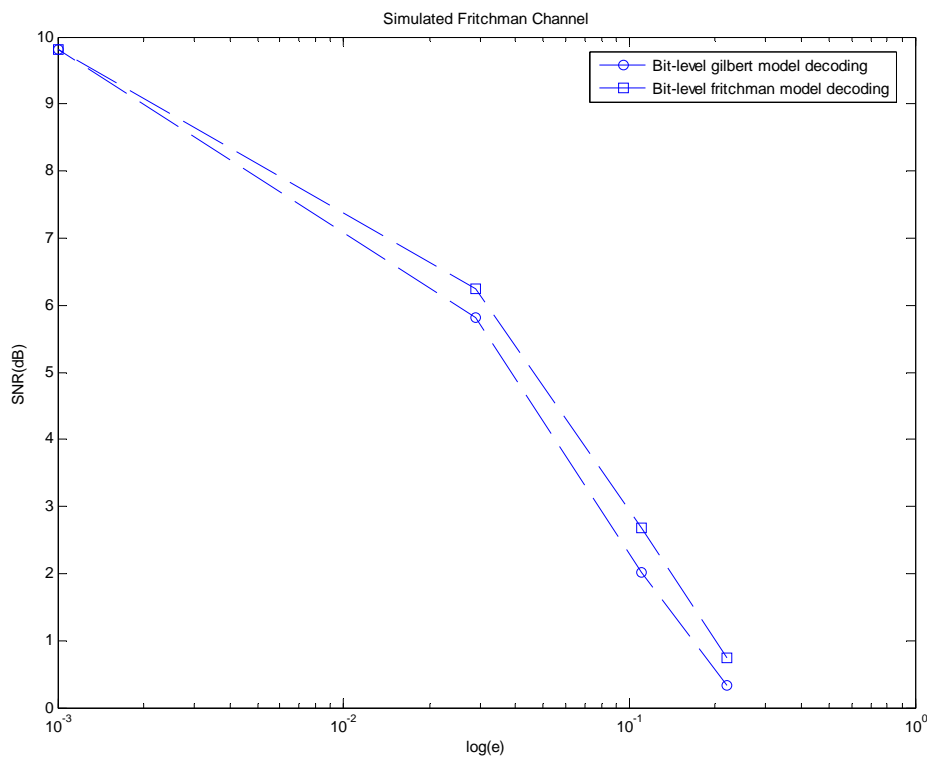



圖 5-6 Fritchman 通道之系統模擬

## 5.3.2 結果分析

圖 5-6 之系統評估，同樣使用 (5.3) 式作為衡量，由圖中結果可知，由於 Gilbert 通道模型較無法匹配 Fritchman 通道環境，使用 Gilbert 模型參數的效能會比 Fritchman 模型參數差。尤其當通道環境越差時，其效果差異更明顯，說明如果估測的模型參數，若能更匹配通道環境，則解碼的效能會更好。當訊號能量對通道雜訊能量比在一定程度以上時，兩種方式的系統效能則顯得差不多。

## 5.4 GSM 數據通道之系統模擬

### 5.4.1 系統模擬之實驗步驟



前兩節中，系統模擬的架構皆是在模擬的 Gilbert 及 Fritchman 通道環境上。本節實驗模擬的系統架構與前兩節相同，而通道環境則設為 GSM 的數據通道，具有代表性的真實環境通道，使用基於位元層級、基於索引層級及混合層級等三種演算法來做系統模擬，且使用 Fritchman 通道模型參數來計算後驗機率。如同 5.2，5.3 節的實驗模擬，訊號源的產生方式、量化與調變皆相同，差別在於通道環境是以 5.1 節所製作的 GSM 數據通道錯誤序列，來對傳送的訊號作雜訊干擾。在接收端，則以 5.1 節中對 GSM 數據通道錯誤序列所估測的 Fritchman 模型參數，計算後驗機率來做解碼還原其訊號。其實驗之

結果如圖 5-7：

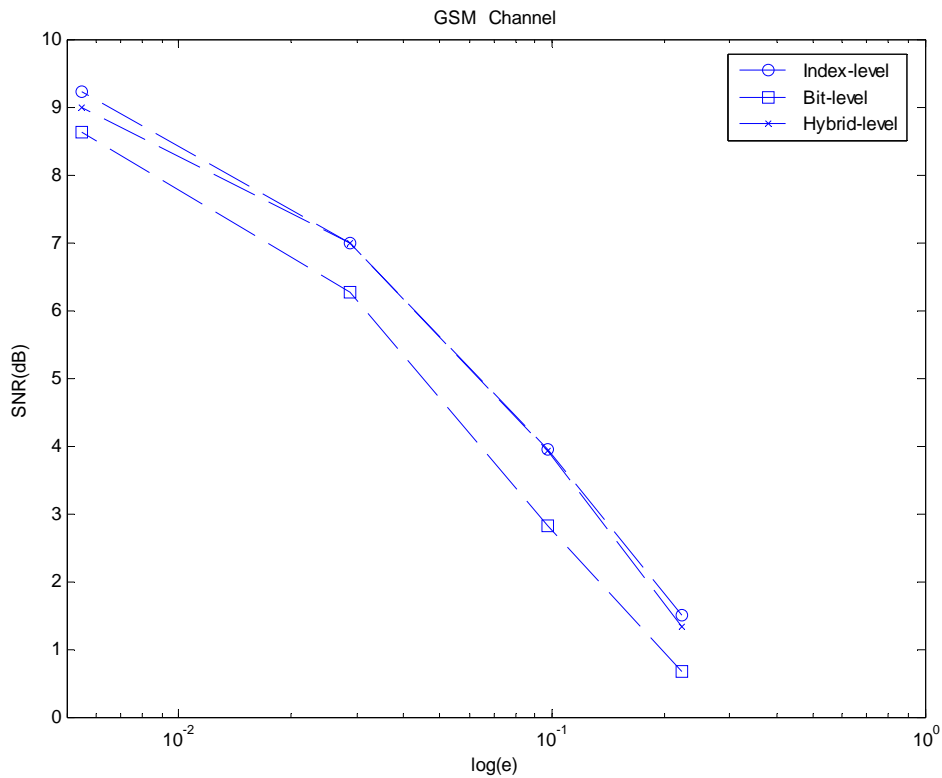


圖 5-7 GSM 數據通道之系統模擬

## 5.4.2 結果分析

為了希望第三、四章所介紹的演算法能應用在真實的通道環境上，於是我們將 GSM 的通道環境使用在系統模擬中。而模擬結果也證實，三種演算法能應用在實際的系統中。實驗結果顯示，基於位元層級演算法中僅利用到位元間的殘餘冗息，效能遜於利用一階事前消息的基於索引層級演算法。不過一階事前消息的計算所花費的時間及記憶體比較多，基於位元層級的系統在執行時間上及記憶體花費上就比

較少。而為了改善基於位元層級的系統效能，讓系統對通道之雜訊干擾更具抵抗性，我們提出利用基於位元層級以及一階事前消息的混合層級演算法。加入利用了一階事前消息，雖然執行時間增加，但是效能明顯的提升了。

這三種演算法在系統模擬的執行時間，以位元層級演算法為最短，而索引層級演算法的執行時間是位元層級演算法的 5.5 倍，混合層級演算法則為 4.5 倍。可看出混合層級雖然犧牲了一些執行速度，但效能接近索引層級演算法卻又比索引層級演算法的執行時間短。



## 第六章 結論與未來展望

### 6.1 結論

本論文主要探討通道匹配的基於索引層級與基於位元層級的解碼機制，在數位通訊系統傳輸過程中，經歷了傳輸通道中不可抗拒的雜訊干擾，致使在接收端因通道的記憶特性而產生叢發性錯誤，使得解碼出現不準確的結果。因此，經過通道匹配的過程後，將通道資訊及狀態紀錄整合於演算法中，挖掘並利用通道記憶特性與量化後索引間的殘餘冗息。我們使用 Gilbert 及 Fritchman 兩種通道模型，來對通道環境作通道匹配，挖掘通道的記憶特性，並且證實 Fritchman 通道模型比 Gilbert 通道模型更能匹配 GSM 系統中的通道環境。最後，在實驗中，分別使用模擬的 Gilbert 及 Fritchman 通道，以及 GSM 系統的通道環境來做系統模擬。基於索引層級演算法的系統效能，經實驗模擬可知最好，但是其利用到量化後索引間的一階事前消息，但是當傳送一個索引值所需的位元數越多時，系統需要的記憶體與計算量就越多，將造成整體系統的緩慢。因此我們提出基於位元層級的演算法，僅利用所傳送序列位元間的事前資訊，不會因所傳送代表索引值的位元數目越多而複雜，因而降低系統需要的記憶體與計算量，但是卻降低了系統效能。為了提升系統效能同時兼顧運算速度，我們提出

混合層級的演算法，加入利用量化索引序列的殘餘冗息來提升系統效能。

## 6.2 未來展望

在此論文中，所提之通道模型運用在所提出的演算法中做系統模擬，確實可看出通道匹配對於改善系統效能的助益。但是面對真實的無線通訊環境，其真實通道是多變的。因此在未來，可以針對通道環境模型化加以改善，尋找更能發揮真實通道特性的通道模型，並結合在文中所提之演算法，盼能以更匹配的通道模型得到更佳之系統效能。





## 參考文獻

- [1] CoCentric System Studio-Getting Started System Studio. Mountain View, CA:Synopsys,Inc., December 2003.
- [2] CoCentric System Studio-Reference Design Kits. Mountain View, CA:Synopsys,Inc., December 2003
- [3] CoCentric System Studio-Davis User Guide. Mountain View, CA:Synopsys,Inc., December 2003.
- [4] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1265, Sept. 1960.
- [5] B. D. Frithman, "A binary channel characterization using partitioned Markov chains," *IEEE Trans. Inform. Theory*, vol. IT-13, no. 2, pp. 221-227, Apr. 1967.
- [6] J. Y. Chouinard, Michel Lecours, Gilles Y. Delisle, "Estimation of Gilbert's and Fritchman's Models Parameters Using the Gradient Method for Digital Mobile Radio Channels," *IEEE Trans. on Vehicular Technology.*, Vol. 30, pp. 158-166, 1988.
- [7] C. L. Lee, W. W. Chang, "Memory-Enhanced MMSE-Based Channel Error Mitigation for Distributed Speech Recognition," *Interspeech 2005-Eurospeech*, Lisbon, Portugal, Sep 2005.
- [8] X. B. Liu, "Joint Source-channel Decoding and its Application to MELP Encoded Speech," *Nanyang Technological University*, 2004
- [9] Tim Fingscheidt and Peter Vary, "Softbit Speech Decoding: A New Approach to Error Concealment," *IEEE Trans. Speech and Audio Processing*, vol. 9 no. 3, pp. 240-251, March 2001.