

國立交通大學

電信工程研究所

碩士論文

適用於里德所羅門碼之
適應性信念傳遞解碼法則其效能改進之研究

Performance Enhancement of
Adaptive Belief-Propagation Decoding
for Reed-Solomon Codes

研究生：蕭佳容

指導教授：王忠炫

中華民國 101 年 7 月

適用於里德所羅門碼之
適應性信念傳遞解碼法則其效能改進之研究

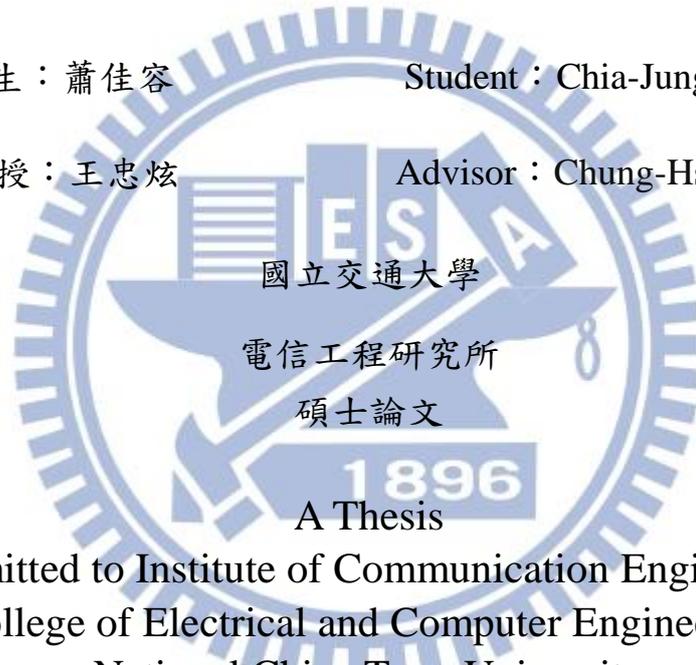
Performance Enhancement of
Adaptive Belief-Propagation Decoding
for Reed-Solomon Codes

研究生：蕭佳容

Student : Chia-Jung Hsiao

指導教授：王忠炫

Advisor : Chung-Hsuan Wang



國立交通大學

電信工程研究所

碩士論文

A Thesis

Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Communication Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中華民國 101 年 7 月

適用於里德所羅門碼之 適應性信念傳遞解碼法則其效能改進之研究

研究生：蕭佳容

指導教授：王忠炫博士

國立交通大學

電信工程研究所

摘要

由於里德所羅門碼的良好錯誤更正能力[1]，常被應用在如數位通訊與數位儲存媒體中，為目前被廣泛使用的錯誤控制碼之一。解碼方面，目前已知軟式決策解碼相較於傳統的硬式決策解碼器有明顯的改善。但軟式決策解碼可能會面臨高解碼複雜度的困難，因此有學者針對複雜度與錯誤率的權衡提出了相關的研究，包括基於信任度解碼與疊代解碼等等，其中本論文對疊代解碼進行討論。Jiang 和 Narayanan 於 2006 年提出了一種利用適應性校驗矩陣，對里德所羅門碼進行軟式輸出輸入解碼的演算法（JN 演算法）[2]。而於本篇論文中，我們將討論造成 JN 演算法解碼失敗的原因，包括 JN 演算法中高信任度位置錯誤以及 JN 演算法在疊代過程中收斂至錯誤的合法碼字所造成的影響，並基於討論提出適當的演算法來避免此問題，並以 1995 年由 Fossorier 和 Lin 所提出的 OSD 演算法輔助 JN 在疊代過程中的訊息傳遞機制，並增進其解碼效能。最後綜合論文中提出的演算法和 JN-OSD(1)演算法比較，在可加性白色高斯雜訊通道並使用雙相位鍵移調變之下，大約有 0.5 - 2.0 dB 的效能改善。

Performance Enhancement of Adaptive Belief-Propagation Decoding for Reed-Solomon Codes

Student : Chia-Jung Hsiao

Advisor : Dr. Chung-Hsuan Wang

Institute of Communications Engineering

National Chiao Tung University

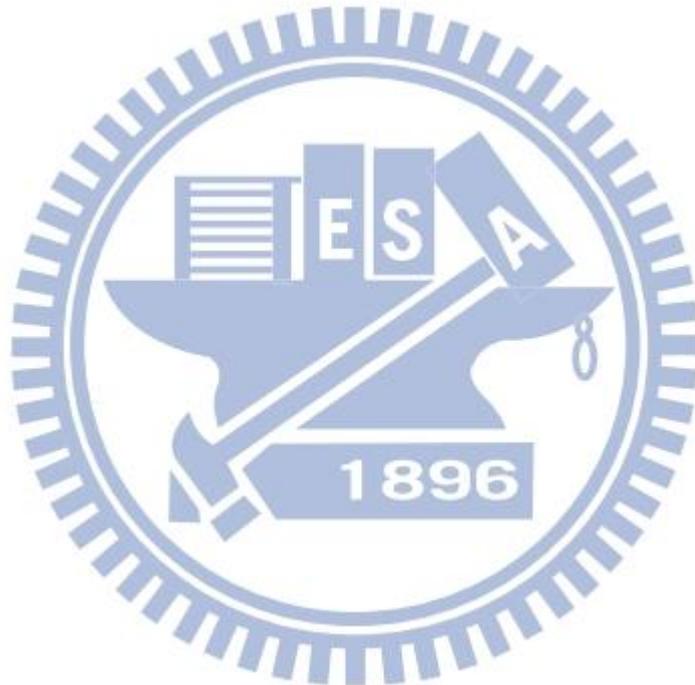
Abstract

Reed-Solomon (RS) codes have been the error-correction codes of choice in many communication systems because of their good correction capability. For the decoding of RS codes, it is known that soft-decision decoding (SDD) of RS codes provides significant performance gain over algebraic hard decision decoding (HDD). There are many researches including reliability-based decoding and iterative decoding which are devoted to better trade-off between the decoding complexity and performance because of the high complexity of SDD. In 2006, Jiang and Narayanan (JN) proposed an iterative soft-in soft-out decoding algorithm of RS codes by adapting the parity check matrix. In this thesis, we discuss the reasons of decoding failure in the JN algorithm. Based on our discussion, we propose different algorithms to avoid the influence of the errors in the most reliable positions and the output of JN convergence to an erroneous valid codeword during iterations. We propose the algorithms using the Ordered Statistic Decoding algorithm which was proposed in 1995 by Fossorier and Lin to modify the message passing rule during the iterations in JN algorithm and helps to improve the decoding performance. Finally, compared with the original JN-OSD(1) algorithm, our design has about a 0.5-2.0dB coding gain while decoding RS codes in the additive white Gaussian noise channel under binary phase shift keying Modulation.

誌 謝

本篇論文得以完成，首先要感謝指導教授王忠炫博士的栽培，在兩年的過程中，教導我研究的態度和方法，並且適時地給予建議和方向。

同時，感謝翁健家與張力仁兩位學長的熱心以及研究上面的幫助，學長總是不厭其煩與我討論研究內容，給了我許多寶貴意見，使論文得以順利完成。兩年的研究生活總有受挫折與不順利時，感謝大學同學黃欣捷小姐與實驗室的同學以及學弟適時的給予我鼓勵與開導，讓我充滿勇氣。最後特別想感謝在我的家人，能讓我沒有後顧之憂專心在研究過程中，謝謝你們。



目錄

中文摘要	I
英文摘要	II
誌謝	III
目錄	IV
圖目錄	V
第一章 概論.....	1
第二章 里德所羅門碼基本定義與相關信念傳遞解碼演算法回顧.....	3
2-1 符號定義.....	3
2-2 里德所羅門碼之二位元映射[11].....	3
2-2.1 二位元映射基底.....	3
2-2.2 二位元映射之代數加法與乘法.....	4
2-2.3 二位元映射之里德所羅門碼奇偶校驗矩陣.....	5
2-3 OSD(w)演算法.....	7
2-4 適應性信念傳遞演算法.....	9
2-4.1 適應性信念傳遞演算法簡介.....	9
2-4.2 幾何解釋與解碼步驟.....	12
2-4.3 錯誤傳遞現象.....	15
2-5 ABP-OSD(1)演算法.....	20
2-6 錯誤傳遞的修正機制.....	22
2-6.1 回饋機制.....	22
2-6.2 事前更正機制.....	22
2-6.2 模擬結果.....	23
第三章 基於信念傳遞解碼之改良演算法.....	25
3-1 初始機制.....	25
3-1.1 演算法基本概念與解碼步驟.....	25

3-1.2 演算法數據觀察.....	28
3-1.3 模擬結果.....	31
3-2 修正型事前更正機制.....	33
3-2.1 演算法基本概念與解碼步驟.....	33
3-2.2 提出演算法之討論.....	34
3-2.3 模擬結果.....	37
3-3 信念傳遞混合列表解碼機制.....	39
3-3.1 演算法基本概念與解碼步驟.....	39
3-3.2 模擬結果.....	49
3-3.3 綜合提出演算法的模擬結果與討論.....	43
第四章 結論.....	48
參考文獻.....	49



圖目錄

圖 2-1、經高斯消去法後，二位元奇偶校驗矩陣內的低密度位置及高密度位置.....	10
圖 2-2、適應性演算法之奇偶校驗矩陣經高斯消去範例.....	15
圖 2-3、位元節點 3 和位元節點 4 為低信任度位置	16
圖 2-4、高信任度位置發生一個錯誤	16
圖 2-5、高密度位置發生一個錯誤時的訊息傳遞情況	16
圖 2-6、經由高斯消去動作位元節點 3 和 4 移至低密度位置.....	17
圖 2-7、高密度位置發生一個錯誤時，經由高斯消去動作後的訊息傳遞情況.....	17
圖 2-8、高信任度位置發生兩個錯誤	18
圖 2-9、高密度位置發生兩個錯誤時的訊息傳遞情況	18
圖 2-10、高密度位置發生兩個錯誤時，經由高斯消去動作後的訊息傳遞情況.....	19
圖 2-11、(15, 7)里德所羅門碼在三種演算法間比較	21
圖 2-12、回饋機制輔佐 ABP-OSD(1)之解碼流程示意圖	22
圖 2-13、事前更正機制輔佐 ABP-OSD(1)之解碼流程示意圖	23
圖 2-14、(15, 7)里德所羅門碼在四種演算法間比較	23
圖 2-15、(15, 7)里德所羅門碼在五種演算法間比較	24
圖 3-1、執行不同次數疊代時 ABP 演算法輸出為錯誤和輸出為錯誤的合法碼字個數.....	25
圖 3-2、初始機制輔佐 ABP-OSD(1)之解碼流程示意圖	26
圖 3-3、(15, 7)里德所羅門碼配合初始機制在不同累積疊代次數初始 LLR 的情況.....	27
圖 3-4、提出演算法和 [12] 在執行 5 次疊代後，OSD 演算法所得到與接收訊號有最大相	28
圖 3-5、提出演算法和 [12] 在執行 10 次疊代後，OSD 演算法所得到與接收訊號有最大相	29
圖 3-6、提出演算法和 [12] 在執行 15 次疊代後，OSD 演算法所得到與接收訊號有最大相	29
圖 3-7、提出演算法和 [12] 在執行 20 次疊代後，OSD 演算法所得到與接收訊號有最大相	30
圖 3-8、比較(15, 7)里德所羅門碼於提出演算法一與不同演算法錯誤率.....	31
圖 3-9、比較(31, 15)里德所羅門碼於提出演算法一與不同演算法錯誤率.....	32
圖 3-10、比較(63, 55)里德所羅門碼於提出演算法一與不同演算法錯誤率.....	32
圖 3-11、提出演算法選擇一個位置翻轉 LLR 值正負號	33
圖 3-12、修正型事前更正機制輔佐 ABP-OSD(1)之解碼流程示意圖	34
圖 3-13、第 I 個校驗節點中，選擇的兩個位置不與之前選擇的重複，共有三種組合	35
圖 3-14、第 I 個校驗節點中，選擇的信任度最低位置與之前選擇的重複，共有兩種組合	35
圖 3-15、第 I 個校驗節點中，選擇的三個位置不與之前選擇的重複，共有七種組合	35
圖 3-16、第 I 個校驗節點中，選擇的信任度最低或次低位置與之前選擇的重複，共有六種	35
圖 3-17、第 I 個校驗節點中，選擇的信任度最低和次低位置與之前選擇的重複，共有四種	36
圖 3-18、比較(63, 55)里德所羅門碼於提出演算法二與不同演算法錯誤率.....	37
圖 3-19、比較(31, 25)里德所羅門碼於提出演算法二與不同演算法錯誤率.....	38
圖 3-20、比較(31, 23)里德所羅門碼於提出演算法二與不同演算法錯誤率.....	38

圖 3-21、示範分割奇偶校驗矩陣 H_b 成較少列的 H'_b	39
圖 3-22、信念傳遞混合列表解碼機制輔佐 ABP-OSD(1)之解碼流程示意圖	40
圖 3-23、比較(15, 7)里德所羅門碼於提出演算法三與不同演算法錯誤率	41
圖 3-24、比較(31, 15)里德所羅門碼於提出演算法三與不同演算法錯誤率	50
圖 3-25、比較(31, 25)里德所羅門碼於提出演算法三與不同演算法錯誤率	42
圖 3-26、比較(31, 15)里德所羅門碼於不同演算法 BER 錯誤率	43
圖 3-27、比較(31, 15)里德所羅門碼於不同演算法 FER 錯誤率	44
圖 3-28、比較(31, 25)里德所羅門碼於不同演算法 BER 錯誤率	44
圖 3-29、比較(31, 25)里德所羅門碼於不同演算法 FER 錯誤率	45
圖 3-30、比較(15, 7)里德所羅門碼於不同演算法 BER 錯誤率	45
圖 3-31、比較(15, 7)里德所羅門碼於不同演算法 FER 錯誤率	46
圖 3-32、比較(63, 55)里德所羅門碼於不同演算法 BER 錯誤率	46
圖 3-33、比較(63, 55)里德所羅門碼於不同演算法 FER 錯誤率	47



第一章

概論

里德所羅門碼是目前被廣泛使用的錯誤更正碼機制，並應用於許多商業用途中，包括數位通訊和數位儲存裝置，如 CD、DVD 和藍光光碟等。廣泛的運用使得里德所羅門碼的解碼成為重要的研究課題。

里德所羅門碼為最大距離分離碼(Maximum Distance Separable Code, MDS Code)和循環碼(Cyclic Code)。最大距離分離碼具有良好的錯誤更正能力，而循環碼在有限場(Finite Field)條件下，任何一個里德所羅門碼字經平移單位個符元仍然是合法碼字。而此特性對於叢集錯誤(Burst Error)和隨機錯誤(Random Error)提供很好的錯誤偵錯能力，且由於里德所羅門碼為非二位元碼字，當更正一個符元時有可能同時更正好幾個位元，因此能有較佳的錯誤率表現。

而目前里德所羅門碼的解碼方式大約分成三類：代數解碼(Algebraic Decoding)、基於信任度解碼(Reliability-Based Decoding)、疊代解碼(Iterative Decoding)。考慮低複雜度，在目前解碼方式中多選擇傳統的硬式決策解碼器(Hard Decision Decoding, HDD)，硬式決策解碼器透過有限場的抽象代數運算解碼。Guruswami 和 Sudan (GS)在有限場下，將碼字形象化成二維平面的一條曲線，每個符元都視作是此線上的一個數據點。在傳輸中，一些值會被破壞，因此期望透過接受信號在二維平面上一定數量的數據點，利用代數結構內插出其餘的數據點，進而得到此碼字在有限場上畫出的曲線，推算出原始的編碼方程式[3]。Koetter and Vardy (KV) 基於 GS 提出代數軟式決策解碼器(Algebraic Soft Decision Decoding, ASD)[4]。然而，軟式決解碼(Soft Decision Decoding, SDD)方面，最大概似解(Maximum Likelihood Decoding, MLD)相較於硬式決策解碼器有明顯的改善，但面臨的是最大概似解碼的高複雜度問題。

而面對最大概似解碼的高複雜度問題，基於信任度解碼分別提出基於最小信任位置(Least Reliable Positions, LRPs)與基於最大獨立信任位置(Most Reliable Independent Positions, MRIPs)的相關解碼方式幫助降低複雜度。前者透過翻轉碼字中 LRPs 位置，試圖找出和接收信號具有相關性最大的碼字。基於信任 LRPs 位置較有可能發生錯誤，以 LRPs 作為可能發生錯誤的位置，事前修正接收信號會較有效率的正確翻轉錯誤位置，Chase[5]與 Forney[6]分別針對接收信號中較容易發生錯誤的 LRPs，翻轉其信任度值並代數解碼，透過修正接收信號中 LRPs 之值，試圖使錯誤個數降至代數解碼的可更正範圍內。後者透過選擇 MRIPs 位置重新編碼，尋找出和接收信號具有相

關性最大的碼字。基於信任 MRIPs 位置比較不會發生錯誤，會較有效率的找出候選碼字集。Fossorier 和 Lin 提出排序統計解碼演算法 (Ordered Statistic Decoding Algorithm, OSD)[7]對接收信號中 MRIPs 重新編碼，由於信任 MRIPs 發生錯誤機率較小，因此只需較少的候選碼字集就有尋找到正確的碼字。

Jiang and Narayanan (JN)提出了基於信念傳遞解碼的疊代演算法，透過將奇偶校驗矩陣依照信任度由小至大做高斯消去，使得 LRIPs 將只連接到一個校驗矩陣，減少發生錯誤對其他位元產生的影響[8]。另外 Jiang and Narayanan 提出了隨機平移解碼演算法(Stochastic Shifting based Iterative Decoding scheme, SSID)[9]，利用里德所羅門碼的循環代數特性，在每次疊代前將接收信號隨機平移後進行更新運算，避免錯誤位置在每次疊代過程傳送錯誤資訊給某些相同位置，幫助減低錯誤傳遞的影響。而 Kothiyal、Takeshita 和 Fossorier[10] 提出了適應性信念傳遞(Adaptive Belief Propagation, ABP)演算法，並提出結合 OSD 演算法相互輔助 ABP 的演算法 ABP-OSD(1)演算法，而後將介紹根據 ABP-OSD(1)提出修正演算法。

基於 JN-OSD(1)演算法，在本論文中提出三種演算法機制:初始機制、修正型事前更正機制與信念傳遞混合列表解碼機制。初始化機制藉由合理的變動收斂的錯誤合法碼使之重新進行解碼，修正型事前更正機制將討論與原始事前更正機制的複雜度與錯誤率表現關係，信念傳遞混合列表解碼機制運用 OSD 演算法的概念修正 ABP 演算法的更新訊息法則。於第二章將會介紹相關的符號定義、里德所羅門碼的二位元映射方法、信念傳遞錯誤傳遞問題以及相關演算法，第三章為所提出之演算法概念並佐以相關模擬數據並加以討論，最後於第四章中提出本論文之結論。

第二章

里德所羅門碼基本定義與相關 信念傳遞解碼演算法回顧

接下來的說明中，於字母底下加一底線代表一個向量，例如 \underline{c} ；粗體的字母代表一個矩陣，如 \mathbf{H} 。本章節一開始先介紹里德所羅門碼的符號定義以及雜訊通道、調變方式等，且由於此章節討論的相關信念傳遞解碼為基於二位元的 tanner 圖解碼，因此在第二節介紹里德所羅門碼的二位元映射以及代數運算後，在接下來的小節將介紹的信念傳遞解碼與其相關的改善機制。

2-1 符號定義

考慮一個 (N, K) 里德所羅門碼，其二位元碼字表示為 $\underline{c} = [c_1, c_2, \dots, c_n]$ ， \underline{c} 向量中每個元素 c_i 均屬於 $GF(2)$ 中的元素。 \mathbf{G} 和 \mathbf{H} 分別為生成矩陣 (Generator Matrix) 以及奇偶校驗矩陣 (Parity-Check Matrix)。

本論文中假設所使用的通道環境均為可加式高斯白雜訊 (Additive White Gaussian Noise, AWGN)；調變方式為二位元移鍵 (Binary Phase Shift Keying, BPSK)，BPSK 調變將 0 映射至 +1 而 1 映射至 -1。假設 $\underline{r} = [r_1, r_2, \dots, r_n]$ 表示經過 BPSK 調變和 AWGN 後的接收信號，而 $\underline{n} = [n_1, n_2, \dots, n_n]$ 表示 AWGN 通道中的雜訊，其雙邊雜訊功率為 $\frac{N_0}{2}$ ，最後經由 AWGN 通道接收向量關係式為

$$\underline{r} = (2\underline{c} - 1) + \underline{n} \quad (2.1)$$

2-2 里德所羅門碼之二位元映射 [11]

2-2.1 二位元映射基底

假設 $\{1, \alpha, \alpha^2, \dots, \alpha^{m-1}\}$ 為 $GF(2^m)$ 內的一組符元基底，其中 α 為 $GF(2^m)$ 的質元素 (primitive element)，則任意 $A \in GF(2^m)$ 皆可以表示為

$$A = \sum_{i=0}^{m-1} A_b^{(i)} \alpha^i, \text{ 其中 } A_b^{(i)} \in GF(2) \quad (2.2)$$

簡化為二位元係數的向量 $A = [A_b^{(0)}, A_b^{(1)}, A_b^{(2)}, \dots, A_b^{(m-1)}]$ 。

範例：(7, 5)里德所羅門碼於 $GF(2^3)$ 使用質的多項式(Primitive Polynomial)為

$1+x+x^3$ ，而 $\{1, \alpha, \alpha^2, \alpha^3\}$ 基底，其中 $\alpha = x$ 則 $GF(2^3)$ 內每個元素可表示為二位元係數的向量：

$$0 = 0*1 + 0*x + 0*x^2 = [0,0,0]$$

$$1 = 1*1 + 0*x + 0*x^2 = [1,0,0]$$

$$\alpha = 0*1 + 1*x + 0*x^2 = [0,1,0]$$

$$\alpha^2 = 0*1 + 0*x + 1*x^2 = [0,0,1]$$

$$\alpha^3 = 1*1 + 1*x + 0*x^2 = [1,1,0]$$

$$\alpha^4 = 0*1 + 1*x + 1*x^2 = [0,1,1]$$

$$\alpha^5 = 1*1 + 1*x + 1*x^2 = [1,1,1]$$

$$\alpha^6 = 1*1 + 0*x + 1*x^2 = [1,0,1]$$

2-2.2 二位元映射之代數加法與乘法

接下來說明二位元映射的加法與乘法：

每個符元映射至二位元係數的向量後，依然須滿足在 $GF(2^m)$ 的加法與乘法代數特性

範例如下：

(1) 加法：

$$\alpha + \alpha^3 = (0 + 1*x + 0*x^2) + (1 + 1*x + 0*x^2) = [0,1,0] + [1,1,0]$$

$$= [1,0,0] = 1 + 0*x + 0*x^2 = 1$$

$$\alpha^2 + \alpha^3 = (0 + 0*x + 1*x^2) + (1 + 1*x + 0*x^2) = [0,0,1] + [1,1,0]$$

$$=[1,1,1]=1+1*x+1*x^2=\alpha^5$$

(2) 乘法:

$$\alpha^5 * 1 = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} * [1 \ 0 \ 0]^T = [A \ D \ G] = \alpha^5 = [1 \ 1 \ 1]^T$$

$$\alpha^5 * \alpha = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} * [0 \ 1 \ 0]^T = [B \ E \ H] = \alpha^6 = [1 \ 0 \ 1]^T$$

$$\alpha^5 * \alpha^2 = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} * [0 \ 0 \ 1]^T = [C \ F \ I] = \alpha^7 = [1 \ 0 \ 0]^T$$

因此可知經過二位元映射後， $\alpha^5 = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

而由以上的推論方式，可得 $GF(2^3)$ 內每個元素可表示為：

$$0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad 1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \alpha = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \alpha^2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\alpha^3 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \alpha^4 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \alpha^5 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}, \quad \alpha^6 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

2-2.3 二位元映射之里德所羅門碼奇偶校驗矩陣

(N, K) 里德所羅門碼運作於 $GF(2^m)$ 之下時，則長度為 $N=2^m-1$ ，其最小距離為 $d_{min}=N-K+1$ ，而奇偶校驗矩陣定義表示為

$$H_S = \begin{bmatrix} 1 & \alpha & \dots & \alpha^{(N-1)} \\ 1 & \alpha^2 & \dots & \alpha^{2(N-1)} \\ \dots & \dots & \dots & \dots \\ 1 & \alpha^{(d_{\min}-1)} & \dots & \alpha^{(d_{\min}-1)(N-1)} \end{bmatrix} \quad (2.3)$$

其中 α 為 $GF(2^m)$ 的質元素(Primitive Element)。而 H_S 可透過上節介紹的二位元映射方式映射成大小為 $(n-k) \times n$ 的二位元奇偶校驗矩陣 H_b ，其中 $n = m \cdot N$ 且 $k = m \cdot K$

範例：

(7, 5)里德所羅門碼運作於 $GF(2^3)$ 之下時，則長度為 $N = 2^3 - 1 = 7$ ，其最小距離為

$d_{\min} = N - K + 1 = 3$ ，而奇偶校驗矩陣 H_S 表示為：

$$H_S = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \end{bmatrix} \quad (2.4)$$

而經過乘法映射後 $n = 3 \cdot 7 = 21$ 且 $k = 3 \cdot 5 = 15$ ，位元奇偶校驗矩陣 H_b 表示為：

$$H_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (2.5)$$

2-3 OSD(w)演算法

OSD(w)演算法為基於信任度排序的解碼演算法，解碼步驟分為兩個階段：尋找最大信任度獨立位置階段與重新編碼階段。

(1) 尋找最大信任度獨立位置階段：

在此階段，按照所有位元的信任度由大至小進行排序，並依據對應位置將 G 進行高斯消去法，得到一個簡化列梯矩陣(Row Reduced Echelon Form)矩陣 G_b 。此動作由最大信任度的位置開始，利用列運算將該位置正規化後，再對信任度次低的位置進行同樣的動作，若發現該位置無法正規化，則略過該位置進行信任度次低的下一個位置，直到得到 G_b 的簡化梯式矩陣 G'_b 和 k 個成功正規化的位置，稱為最大信任獨立位置 (Most Reliable Independent Positions, MRIPs)，這些位置可以經由重新排序而成為一正規化矩陣(Identity Matrix)。而非最大信任獨立位置的部分於本論文中則稱為最小信任度位置 (LeastReliable Positions, LRP)。

(2) 重新編碼階段：

此階段利用上一階段得到 MRIPs 加上測試錯誤樣本(Test Error Patterns, TEPs)對簡化列梯矩陣 G_b 進行重新編碼。而 OSD(w) 中， w 表示上一階段的 k 個 MRIPs 中分別

有 $0, 1, 2, \dots, w$ 個測試錯誤位置，其中 $w < k$ ，因此 OSD(w) 共產生 $\sum_{i=0}^w \binom{k}{i}$ 個 TEPs。

OSD(w)演算法解碼步驟：

步驟一、 依據信任度大小進行位元節點的排序。

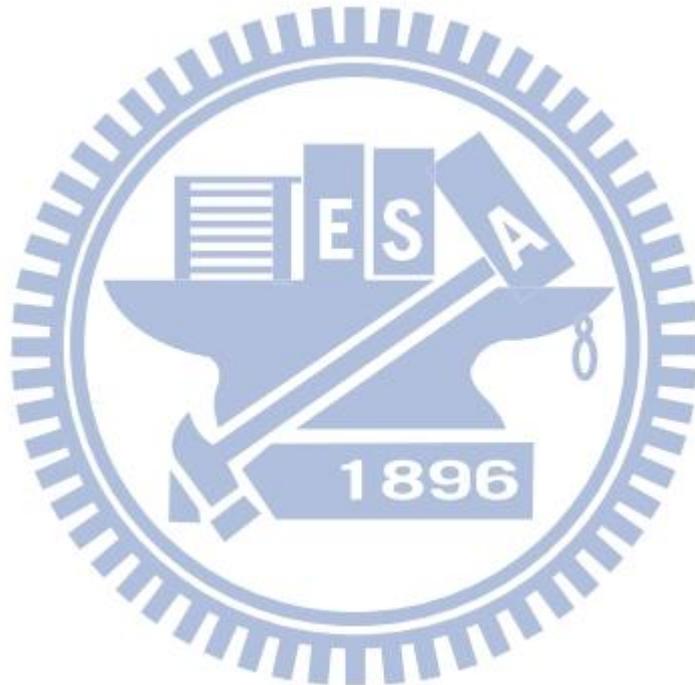
步驟二、 依排序的結果，依信任度由大至小的位置對二位元生成矩陣 G_b 進行高斯消之

動作而得到簡化梯式生成矩陣。若該位置無法經列運算正規化，則跳過該位置至信任度次大的下一個位置，直到得到 k 個 MRIPs 後停止。

步驟三、 製作測試錯誤樣本：假設此 k 個 MRIPs 中分別有 $0, 1, 2, \dots, w$ 個錯誤，($w < k$)，此於演算法之標示為 OSD(w)，則表示此演算法於該 MRIPs 中，最多測試至發生 w 個錯誤。接下來分別產生 $0, 1, 2, \dots, w$ 個錯誤的所有測試錯誤樣本，若該位置發生錯誤，則在樣本中給予 1 的值，否則給 0 的

值，而在 $OSD(w)$ 的情況下，總共產生 $\sum_{i=0}^w \binom{k}{i}$ 個 TEPs。

步驟四、將原來的 MRIPs 向量加上 TEP 向量後，根據高斯消去法所得到的梯式簡化生成矩陣進行編碼，當所有的 TEP 配合 MRIPs 均被重新編碼後，選擇和接收信號相關性最大碼字的做為解碼輸出。或當某個 TEP 配合 MRIPs 所得到的碼字為最大概似碼字時，則直接輸出該碼字。



2-4 適應性信念傳遞演算法

2-4.1 適應性信念傳遞演算法簡介

令 $\underline{c}=[c_1, c_2, \dots, c_N]$ 為里德所羅門碼的二位元碼字，經過 BPSK 調變與 AWGN 通道的接收訊號表示為 $\underline{r}=(2\underline{c}-1)+\underline{n}$ 。獲得接收信號後，碼字中每個位元的初始信任度由對數概似比例(Log Likelihood Ratio, LLR)計算，計算方式為

$$L(c_i) = \log \frac{\Pr(c_i = 0 | r_i)}{\Pr(c_i = 1 | r_i)} = \log \frac{\frac{1}{\sqrt{\pi N_o}} e^{-\frac{(r_i-1)^2}{N}}}{\frac{1}{\sqrt{\pi N_o}} e^{-\frac{(r_i+1)^2}{N}}} = \frac{4r_i}{N_o} \quad (2.6)$$

在 AWGN 通道下，將 LLR 取絕對值 $|L(c_i)|$ 代表此位元解碼的信任度，由於 $L(c_i) \rightarrow \infty$ 表示 $\Pr(c_i = 0)$ 發生機率愈高、於 $\Pr(c_i = 1)$ 發生機率愈低，而 $L(c_i) \rightarrow -\infty$ 表示 $\Pr(c_i = 1)$ 發生機率愈高、於 $\Pr(c_i = 0)$ 發生機率愈低，因此 LLR 絕對值 $|L(c_i)|$ 越大表示 $\Pr(c_i = 0)$ 或者 $\Pr(c_i = 1)$ 發生機率越大錯誤機率愈低，亦表示信任度愈大。反之 LLR 絕對值 $|L(c_i)|$ 愈小表示錯誤機率愈低信任度越低。

適應性信念傳遞演算法可分為兩個階段：更新奇偶校驗矩陣階段與更新位元節點 LLR 階段。

(1) 更新奇偶校驗矩陣階段：

在此階段，按照所有位元的信任度由小至大進行排序，並依據對應位置將 H 進行高斯消去法，得到一個簡化列梯矩陣(Row Reduced Echelon Form)矩陣 H_b 。此動作由最小信任度的位置開始，利用列運算將該位置正規化後，再對信任度次高的位置進行同樣的動作，若發現該位置無法正規化，則略過該位置進行信任度次高的下一個位置，直到得到 H_b 的簡化梯式矩陣 H'_b 和 $n-k$ 個成功正規化的位置，稱為最小信任獨立位置 (least reliable independent positions, LRIPs)，這些位置可以經由重新排序而成為一正規化矩陣 (identity matrix)。而非最小信任獨立位置的部

分於本論文中則稱為最大信任度位置 (most reliable positions, MRPs)，由於 H'_b 中對應到這些 LRIPs 的行向量，均只有一個 1，因此這些部分亦稱為 H'_b 的低密度位置，反之這些位置以外則稱為 H'_b 的高密度位置。

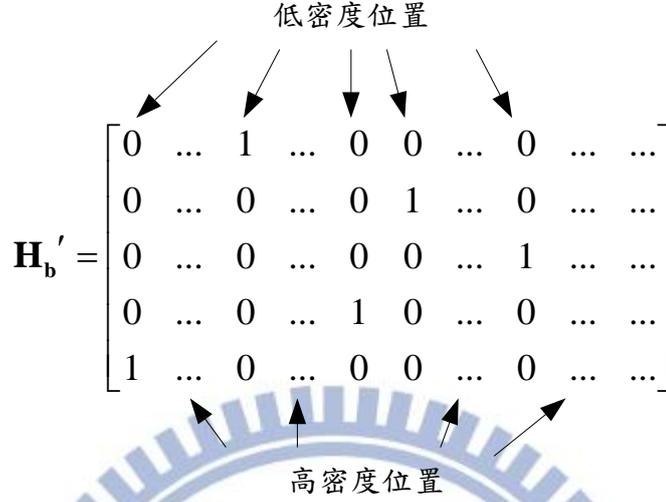


圖 2-1、經高斯消去法後，二位元奇偶校驗矩陣內的低密度位置及高密度位置

(2) 更新位元節點 LLR 階段：

此階段利用更新奇偶校驗矩陣階段所求得 H'_b 進行更新位元節點 LLR。接收信號後，定義 JN 演算法 LLR 向量， $\underline{L}^{(i)}$ 上標表示第 i 次疊代的 LLR 向量

$$\underline{L}^{(i)} = [L^{(i)}(c_1), L^{(i)}(c_2), \dots, L^{(i)}(c_n)] \quad (2.7)$$

而 $\underline{L}^{(0)}$ 為接收信訊號的初始 LLR 向量。基於 $\underline{L}^{(i)}$ 排序進行高斯消去的矩陣表示成：

$$H_b^{(i)} = \Phi(H_b, |\underline{L}^{(i)}|) \quad (2.8)$$

更新位元節點 LLR 階段中，額外消息量 $\underline{L}_{ext}^{(i)}$ 為根據 $\underline{L}^{(i)}$ 在經過更新後的奇偶校驗矩陣

$H_b^{(i)}$ 計算得到，表示成：

$$\underline{L}_{ext}^{(i)} = \varphi(H_b^{(i)}, \underline{L}^{(i)}) \quad (2.9)$$

而每個位元的額外消息量

$$L_{ext}^{(l)}(c_i) = \sum_{\substack{j=1 \\ H_j^{(l)}=1}}^{n-k} 2 \tanh^{-1} \left(\prod_{\substack{p=1 \\ p \neq i, H_p^{(l)}=1}}^n \tanh \left(\frac{L^{(l)}(C_p)}{2} \right) \right) \quad (2.10)$$

JN 演算法解碼步驟:

步驟一、設定抑制係數 α ，最大疊代次數 l_{\max} ，並計算每個位元節點的初始的 LLR 為

$$L^{(0)}(c_i) = (4y_i / N_0)，並以初始 LLR ($L^{(0)}(c_i)$) 開始進行解碼。$$

步驟二、更新奇偶校驗矩陣： $\mathbf{H}_b^{(l)} = \phi(\mathbf{H}_b, |\underline{L}^{(l)}|)$ ， ϕ 表示更新 \mathbf{H}_b 的動作。

(a) 依據 LLR 絕對值 $|\underline{L}^{(l)}|$ 的大小將每個位元位置進行排序，並且記錄排序後的順序。

(b) 利用高斯消去法，按照信任度由小至大的位置將 $\mathbf{H}_b^{(l)}$ 消為簡化梯式矩陣，並得到 $(n-k)$ 個 LRIPs。

步驟三、利用高斯消去法正規化後的 $\mathbf{H}_b^{(l)}$ ，進行梯度下降法得到每個位元節點的額外消

息量： $\underline{L}_{ext}^{(l)} = \phi(\mathbf{H}_b^{(l)}, \underline{L}^{(l)})$ 。 ϕ 表示計算額外消息量之動作。

步驟四、更新每個位元節點的 LLR： $\underline{L}^{(l+1)} = \underline{L}^{(l)} + \alpha \underline{L}_{ext}^{(l)}$ ，在這裡 $0 < \alpha \leq 1$ 。

步驟五、硬式決策： $c_i = \begin{cases} 0, L^{(l+1)}(c_i) > 0 \\ 1, L^{(l+1)}(c_i) < 0 \end{cases}$

步驟六、終止條件：若所有的校驗節點均被滿足，或當到達最大的預設疊代次數時，則停止疊代並且輸出目前經硬式決策後每個位元的決策值。若未達最大疊代數，則令 $l = l + 1$ ，回到步驟二進行下一次的疊代。

2-4.2 幾何解釋與解碼步驟

首先定義兩個函數：

函數定義 3-1

函數 $v: [-\infty, +\infty] \rightarrow [-1, +1]$ 為將 $L(c_i)$ 自 LLR 域映射至 tanh 域之函數：

$$v(L) = \tanh\left(\frac{L}{2}\right) = \frac{e^L - 1}{e^L + 1}. \quad (2.11)$$

此為一對一 (one-to-one) 且映成 (onto) 函數。

函數定義 3-2

v 之反函數 $v^{-1}: [-1, +1] \rightarrow [-\infty, +\infty]$ ：

$$v^{-1}(t) = \ln\left(\frac{1+t}{1-t}\right), \quad t \in [-1, +1]. \quad (2.12)$$

此亦為一對一且映成函數。

從接收信號求得每個位置的 LLR ($L(c_i)$) 後，首先將其利用函數定義 3-1 將每個位置的 LLR 轉換至 tanh 域表示如下：

$$\underline{T} = [T_1, T_2, \dots, T_n] = [v(L(c_1)), \dots, v(L(c_n))]. \quad (2.13)$$

其中 \underline{T} 為經過轉換後的 tanh 域向量。

在 [14] 中，定義 \mathbf{H}_b 中第 j 個校驗方程式的決策信任度 γ_j (decision reliability)

表示如下：

$$\gamma_j = \prod_{p \in \Gamma_j} v(L(c_p)). \text{ 其中 } \Gamma_j \in \{a \mid \forall 1 \leq a \leq n, H_{b,ja} = 1\} \quad (2.14)$$

由上節的討論中得知，LLR 的絕對值較大時，會認為該位置的硬式決策較不易發生錯誤信任度越高，反之 LLR 之絕對值較小時，會認為該位元位置的硬式決策較易發生錯誤信任度越低。同樣的概念應用於此處的決策信任度上，當 $|\gamma_j|$ 愈大表示我們能夠愈信任這個校驗的決策，而當 $|\gamma_j|$ 愈小時，則表示這個校驗的決策愈容易發生錯誤。而由式函數定義 3-1 可知， γ_j 屬於 $[-1, +1]$ 區間內。

接下來定義一位能函數 (potential function) J ，表示式如下：

函數定義 3-3:

$$\text{位能函數 } J: \quad J(\mathbf{H}_b, \underline{T}) = -\sum_{j=1}^{(n-k)} \gamma_j = -\sum_{j=1}^{(n-k)} \prod_{p \in \Gamma_j} T_p. \quad (2.15)$$

其中 $\Gamma_j \in \{a \mid \forall 1 \leq a \leq n, H_{b_{ja}} = 1\}$ ，且 J 是一個同時受 \mathbf{H}_b 和 \underline{T} 所影響的函數。

當收斂至一個合法碼字時，將滿足所有的校驗方程式且決策信任度 $|\gamma_j| = 1$ ，此時位能函數 J 將達到最小值 $J_{\min} = -(n-k)$ 。因此解碼過程可視為，由通道所得到信任度 \underline{T} 為初始起點，使接收信號的 LLR 往位能函數最低點的方向前進，並收斂於位能函數的最低點時，則被認為此過程成功找到了傳送的碼字而成功解碼。因此基於這樣的想法，梯度下降法適合進行這樣的運作過程。首先對位能函數進行偏微分後表示式為：

$$\nabla J(\mathbf{H}_b, \underline{T}) = \left(\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_1}, \frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_2}, \dots, \frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_n} \right). \quad (2.16)$$

其中每個對應 T_i 的偏微分元素可以表示成

$$\frac{\partial J(\mathbf{H}_b, \underline{T})}{\partial T_i} = -\sum_{j \in \Lambda_i} \prod_{p \in \Omega_j} T_p. \quad (2.17)$$

此時 $\Lambda_i = \{q \mid \forall 1 \leq q \leq n-k, H_{b_{qi}} = 1\}$ 且 $\Omega_j = \{s \mid \forall 1 \leq s \leq n, H_{b_{js}} = 1, s \neq i\}$ 。

因此，梯度下降法搜尋位能函數最低點的過程可表示為

$$\underline{T}^{(l)} - \alpha \nabla J(\mathbf{H}_b, \underline{T}^{(l)}) \rightarrow \underline{T}^{(l+1)}. \quad (2.18)$$

這裡 α 為一抑制係數，如同梯度下降法中控制每次疊代的步距 (step size) 係數，上標 (l) 表示目前為第 l 次疊代。由於信任度 T_i 被定義在 $[-1, +1]$ 的區間內，因此經過修正原來的梯度下降法的疊代函數，可得到表示式

$$v^{-1}(T_i^{(l)}) - \alpha \left[-\sum_{j \in \Lambda_i} v^{-1} \left(\prod_{p \in \Omega_j} T_p^{(l)} \right) \right] \rightarrow v^{-1}(T_i^{(l+1)}). \quad (2.19)$$

此時 $\Lambda_i = \{q | \forall 1 \leq q \leq n-k, H_{b_{qi}} = 1\}$ 且 $\Omega_j = \{s | \forall 1 \leq s \leq n, H_{b_{js}} = 1, s \neq i\}$ ， $T_p^{(l)}$ 和 $T_i^{(l)}$ 之上標 (l) 表示目前為第 l 次疊代。

而在上節討論的步驟三中所接收到的額外消息量 (extrinsic information) 總值 $L_{ext}^{(l)}(c_i)$ 可以表示如下

$$L_{ext}^{(l)}(c_i) = \sum_{j \in \Lambda_i} \nu^{-1} \left(\prod_{p \in \Omega_j} T_p^{(l)} \right) = \sum_{j \in \Lambda_i} 2 \tanh^{-1} \left(\prod_{p \in \Omega_j} \tanh \left(\frac{L^{(l)}(c_p)}{2} \right) \right). \quad (2.20)$$

其中 $\Lambda_i = \{q | \forall 1 \leq q \leq n-k, H_{b_{qi}} = 1\}$ 且 $\Omega_j = \{s | \forall 1 \leq s \leq n, H_{b_{js}} = 1, s \neq i\}$ ， $T_p^{(l)}$ 和 $L^{(l)}(c_p)$ 之上標 (l) 表示目前為第 l 次疊代。另外由第 j 個校驗方程式所提供的額外消息量

$L_{ext}^{(l)(j)}(c_i)$ 表示如下

$$L_{ext}^{(l)(j)}(c_i) = 2 \tanh^{-1} \left(\prod_{p \in \Omega_j} \tanh \left(\frac{L^{(l)}(c_p)}{2} \right) \right). \quad (2.21)$$

其中 $\Omega_j = \{s | \forall 1 \leq s \leq n, H_{b_{js}} = 1, s \neq i\}$ 。

2-4.3 錯誤傳遞現象

信念傳遞演算法並不適用於高密度奇偶校驗矩陣(High Density Parity Check Codes, HDPC)，原因是比較多的週期(cycle)使訊息傳遞並不獨立，另外發生錯誤位置則可能會連接至過多的校正節點，而發生錯誤傳遞的問題。

在 AWGN 通道下，接收信號的 LLR 取絕對值 $|L(c_i)|$ 代表此位元解碼的信任度，

$|L(c_i)|$ 愈大被認為較不可能發生錯誤解信任度愈大，反之愈小被認為較有可能發生錯誤且信任度愈小。

因此在適應性信念傳遞演算法中透過高斯消去法，將奇偶校驗矩陣化為一個低信任度位置只連接至一個校驗節點，減少錯誤傳遞。以(7, 5)里德所羅門碼為範例，經高斯消去後的奇偶校驗矩陣 tanner 圖如圖

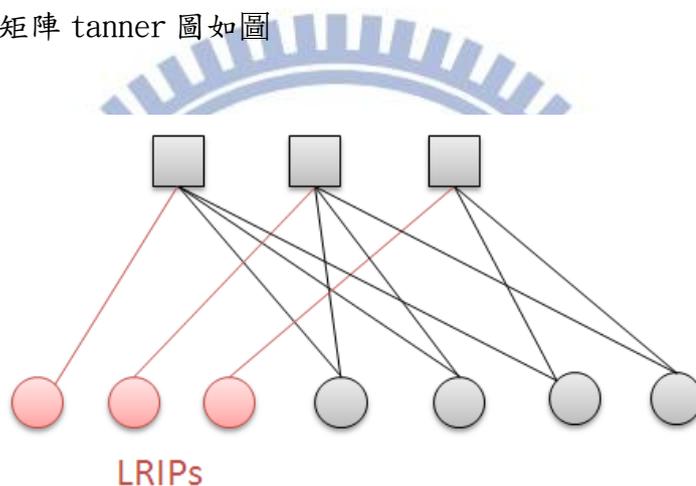


圖 2-2、適應性演算法之奇偶校驗矩陣經高斯消去範例

經過高斯消去後，將可減少當低信任度位置發生錯誤時，經由訊息傳遞機制傳遞錯誤的額外訊息影響其他位元，造成錯誤傳遞現象。

接下來將以圖形化說明錯誤傳遞現象，圖 2-1 是奇偶校驗矩陣之 Tanner 圖，圖中週期 4 (cycle 4) 的部分所代表的是奇偶校驗矩陣中的高密度位置。在以下的討論中，將以正值 (+) 表示正確的 LLR 值，負值 (-) 表示錯誤的 LLR 值，另外在沒有發生錯誤的位置上，高信任度位置 LLR 的絕對值為 V ，低信任度位置的 LLR 的絕對值為 v ，且 $V > v$ ；若在發生錯誤的位置上，高信任度位置 LLR 的絕對值為 $-V$ ，低信任度位置的 LLR 的絕對值為 $-v$ 。在圖 2-1 中，我們假設位元節點 3 和位元節點 4 為低信任度位置，其他位元節點為高信任度位置。

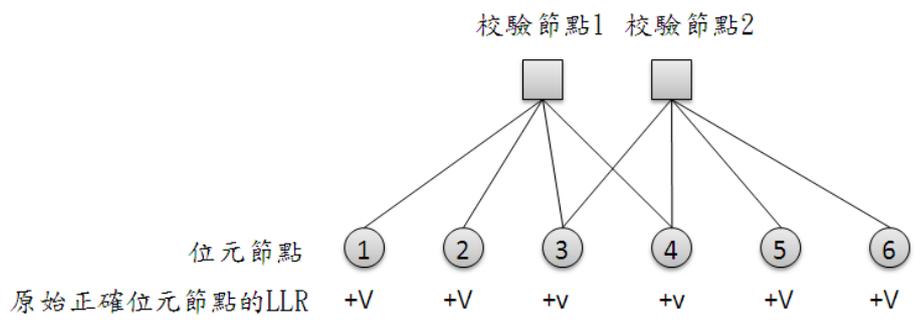


圖 2-3、位元節點 3 和位元節點 4 為低信任度位置

現在假設低信任度位置位元節 3 點發生錯誤，其 LLR 值為 $-v$ ，表示圖如圖：

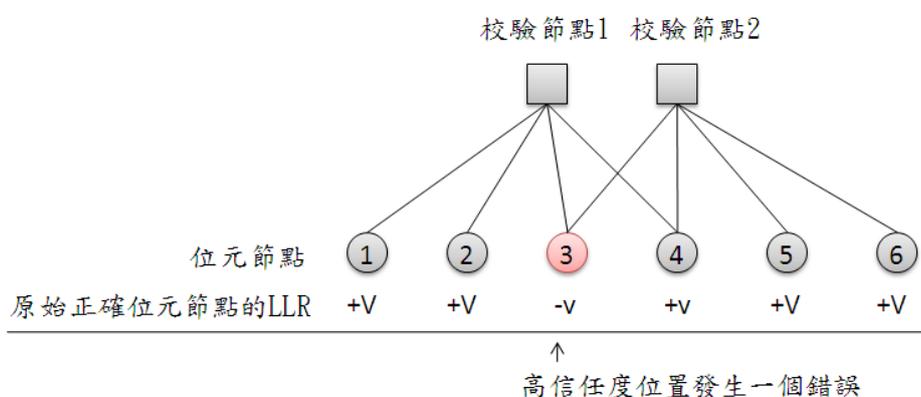


圖 2-4、高信任度位置發生一個錯誤

而經由信念傳遞(Belief Propagation)訊息傳遞機制額外消息量，我們以正值 (+E) 表示校驗節點傳遞正確的額外消息量，負值 (-E) 表示傳遞錯誤的額外消息量，其中 $E > 0$ 。而當信任度位置位元節 3 點發生錯誤時，校驗節點傳遞訊息的機制表示如圖：

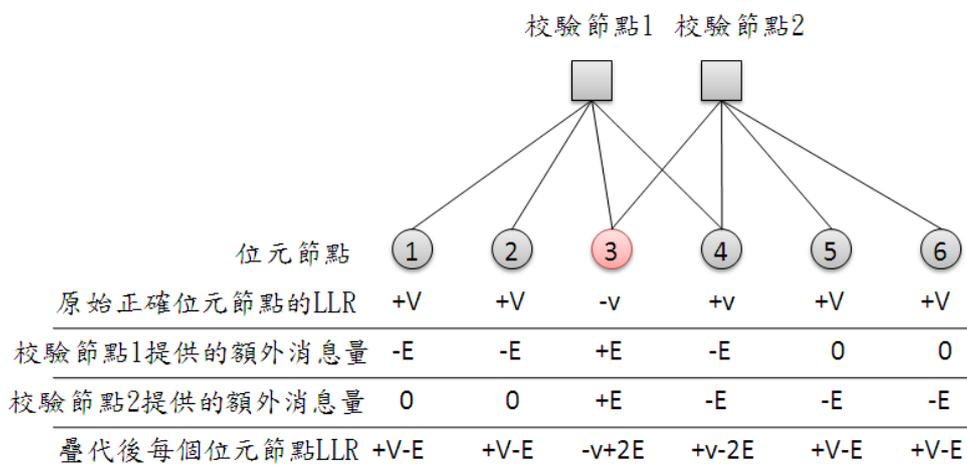


圖 2-5、高密度位置發生一個錯誤時的訊息傳遞情況

由圖 2-5 可知，除了位元節點 3 收到正確的額外消息量以外，其餘位置皆接收到錯誤的額外消息量，此錯誤傳遞可能造成碼字解碼失敗甚至無法收斂。

另外考慮適應性信念傳遞解碼，在低信任度位置位元節點 3 發生錯誤的情況，已知可透過高斯消法的得到簡化梯式校驗矩陣，使得每個校驗節點僅連結至一個低信任度的位元節點，而高密度位置為高信任度的位元節點。經由此動作，位元節點 3 和 4 將被移至只連接到一個校驗節點的低密度位置，而位元節點 1 和 6 將被移至連接到兩個校驗節點的高密度位置，說明圖如圖 2-6:

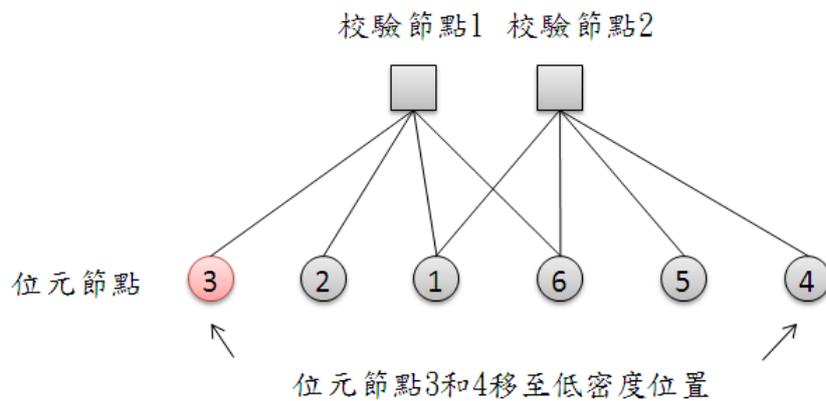


圖 2-6、經由高斯消去動作位元節點 3 和 4 移至低密度位置

位元節點	3	2	1	6	5	4
原始正確位元節點的LLR	-v	+v	+v	+v	+v	+v
校驗節點1提供的額外消息量	+E	-E	-E	-E	0	0
校驗節點2提供的額外消息量	0	0	+E	+E	+E	+E
疊代後每個位元節點LLR	-v+E	+v-E	+v	+v	+v+E	+v+E

圖 2-7、高密度位置發生一個錯誤時，經由高斯消去動作後的訊息傳遞情況

和圖 2-5 相比，可以看出位元節點 3 有機會被更正且高信任度位置僅有位元節點 2 收到錯誤的額外消息量，可以看出錯誤傳遞現象已大幅的減少。

接下來考慮兩個低信任度位置(位元節點 3 和 4)發生錯誤，假設 LLR 值為 -v，說明圖如：圖 2-8

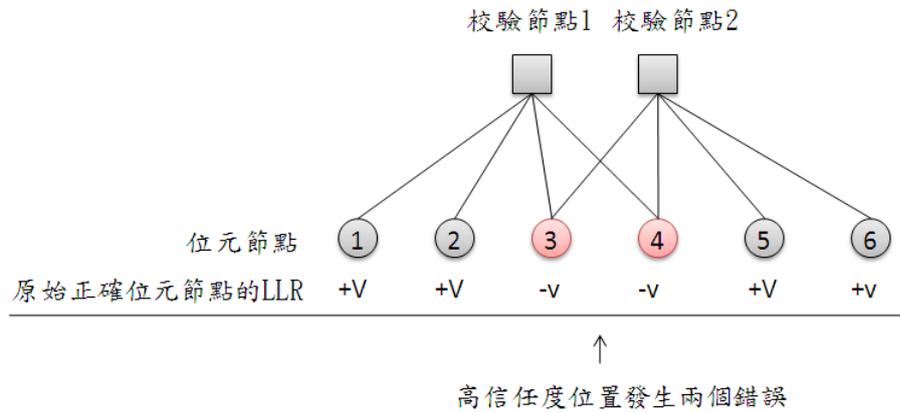


圖 2-8、高信任度位置發生兩個錯誤

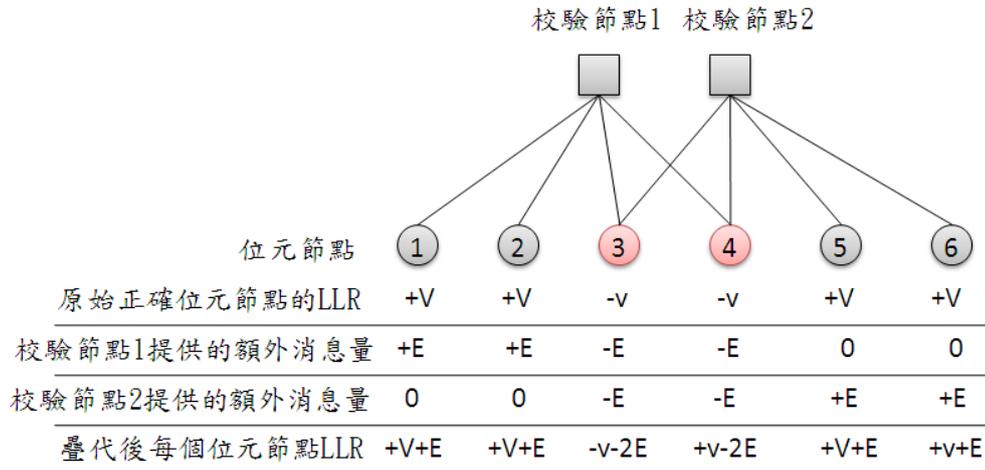


圖 2-9、高密度位置發生兩個錯誤時的訊息傳遞情況

由於週期 4 的關係，錯誤的位元節點 3 和 4 持續收到錯誤的額外消息量，而變得更難正確解碼。同樣地，考慮適應性信念傳遞解碼，在低信任度位置位元節點 3 和 4 發生錯誤的情況，透過高斯消法的得到簡化梯式校驗矩陣，使得每個校驗節點僅連結至一個低信任度的位元節點，而高密度位置為高信任度的位元節點，說明圖如圖 2-10

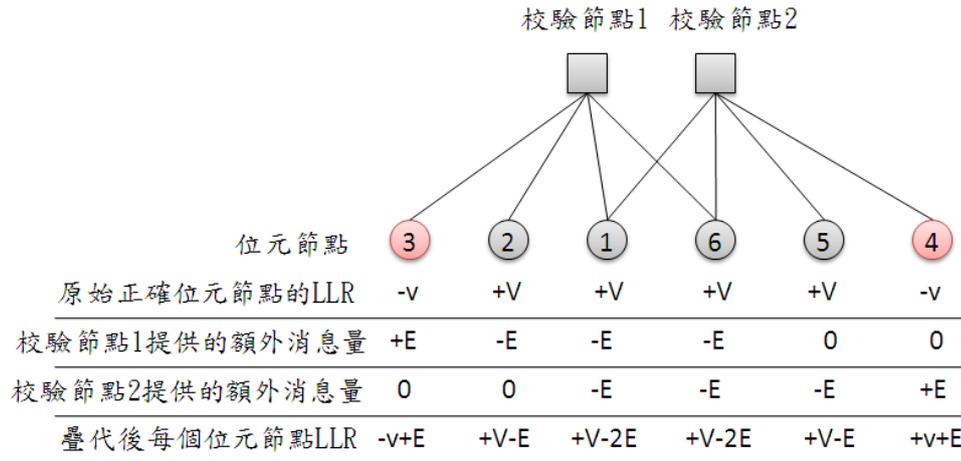


圖 2-10、高密度位置發生兩個錯誤時，經由高斯消去動作後的訊息傳遞情況

和圖 2-9 相比，可以看出低信任度位元節點 3 和 4 有機會被更正，且只要高信任度位置信任度 V 夠大，在疊代過程中便能不受到錯誤的額外消息量影響改變 LLR 的正負號值，相較於圖 2-9，圖 2-10 經過高斯消去動作仍然較能抑制解碼失敗的發生。



2-5 ABP-OSD(1)演算法

JN 演算法透過將奇偶校驗矩陣依照信任度由小至大做高斯消去，將奇偶校驗矩陣化為 $H = [I | P]$ 形式，其經由此動作後 LRIPs 將只連接到一個校驗矩陣，減少發生錯誤時影響較多其他位元。再透過 JN 演算法更新運算，利用 MRPs 傳送額外訊息幫助更正較有可能發生錯的 LRIPs，同時較 MRPs 的錯誤，亦有可能因為 JN 演算法更新運算過程逐漸降低信任度，而在幾次疊代後落在 LRIPs，但假若錯誤位持續存在 MRPs 裡，則仰賴 OSD 演算法機制修正高信任度位置發生的錯誤。

OSD 演算法則透過將生成矩陣依照信任度由大至小做高斯消去，將生成矩陣化為 $G = [I | P]$ 形式得到 MRIPs，OSD 演算法在重新編碼之前事先假設 MRIPs 可能發生了錯誤，並透過 TEPs 事前更正，使得即使 MRIPs 中存在少數的錯誤，OSD 存在仍然能正確解碼，並於 MRIPs 發生錯誤機率較小，因此較少數量的 TEPs 就有可能修正接收的錯誤信號。

ABP-OSD(1)演算法解碼步驟：

步驟一、設定抑制係數 α ，最大疊代次數 l_{\max} ，並計算每個位元節點的初始 LLR 為

$$L^{(0)}(c_i) = \left(\frac{4y_i}{N_0} \right), \text{ 並以初始 LLR } (L^{(0)}(c_i)) \text{ 開始進行解碼。}$$

步驟二、OSD(1)演算法階段：

- (1) 依據 LLR 絕對值 $|L^{(l)}|$ 的大小將每個位元位置進行排序，並且記錄排序後的順序。
- (2) 依排序的結果，依信任度由大至小的位置對二位元生成矩陣 G_b 進行高斯消去之動作而得到簡化梯式生成矩陣。若該位置無法經列運算正規化，則跳過該位置至信任度次大的下一個位置，直到得到 k 個 MRIPs 後停止。
- (3) 此時假設此 k 個 MRIPs 中僅有一個以下的錯誤。因此生成 1 的個數為 0，和 1 的個數為 1 的 TEPs。接下來按照此 $k+1$ 個 TEPs 以高斯消去動作後的生成矩陣進行編碼得到 $k+1$ 組候選碼字 (candidate codewords)。
- (4) 若自候選碼字中找到最大概似碼字，則停止解碼並輸出結果。若到達最大預定疊代次數 l_{\max} 時，則停止解碼並輸出最目前所存相關性最大的碼字。否則保留此次疊代中和接收信號相關性最大的碼字，並進行步驟三。

步驟三、JN 演算法階段。

(1)更新奇偶校驗矩陣： $H_b^{(l)} = \phi(H_b, |\underline{L}^{(l)}|)$ ， ϕ 表示更新 H_b 的動作。依據LLR

絕對值 $|\underline{L}^{(l)}|$ 的大小將每個位元位置進行排序，並且記錄排序後的順序。利

用高斯消去法，按照信任度由小至大的位置將 $H_b^{(l)}$ 消為簡化梯式矩陣，並得

到 $(n-k)$ 個LRIPs。

(2)利用高斯消去法正規化後的 $H_b^{(l)}$ ，進行梯度下降法得到每個位元節點的額

外消息量： $\underline{L}_{ext}^{(l)} = \varphi(H_b^{(l)}, \underline{L}^{(l)})$ 。 φ 表示計算額外消息量之動作。

(3)更新每個位元節點的LLR： $\underline{L}^{(l+1)} = \underline{L}^{(l)} + \alpha \underline{L}_{ext}^{(l)}$ ，在這裡 $0 < \alpha \leq 1$ 。

(4)回到步驟二。

接下來為(15, 7)里德所羅門碼在JN-OSD(1)演算法和JN及OSD(1)演算法錯誤比較，JN演算法透過訊號傳遞機制幫助更正錯誤並加強正確位置的信任度後，由OSD(1)利用其高信任度位置編碼，兩者互相輔佐得到更好的模擬結果，其中JN-OSD(1)演算法OSD(1)演算法大約有0.5dB的效能改善。

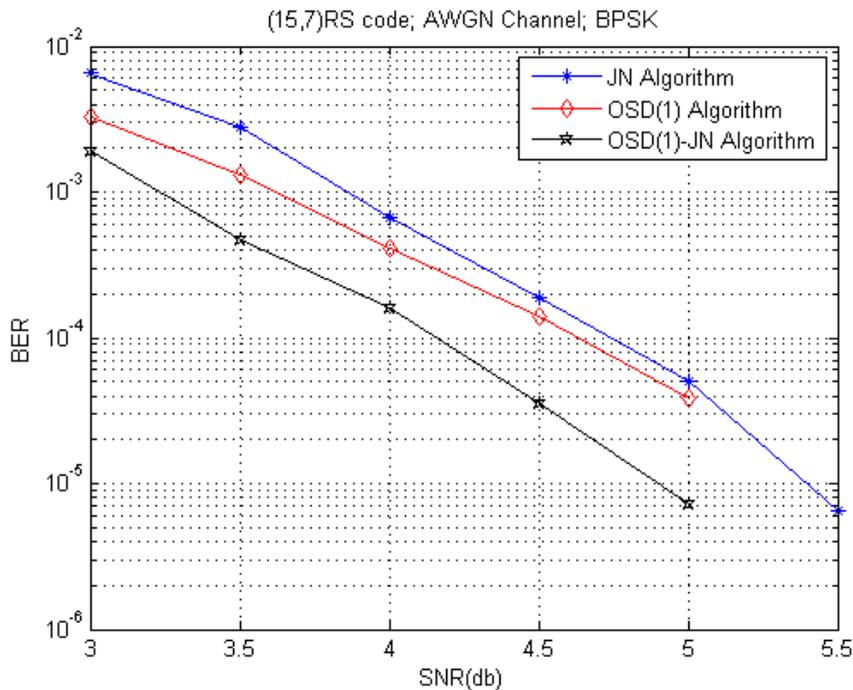


圖 2-11、(15, 7)里德所羅門碼在三種演算法間比較

2-6 錯誤傳遞的修正機制

由於 JN 演算法透過訊息傳遞機制，利用 MRPs 傳送給校驗節點的訊息更正較有可能發生錯的 LRIPs。因此，若有錯誤存在 MRPs 中，經過 JN 訊息傳遞機制將會傳遞錯誤訊息給 LRIPs，造成錯誤傳遞。而為了抑制此現象，[12]和[13]分別在疊代過程中以及疊代前對訊號做處理。

2-6.1 回饋機制

[12]在疊代過程中，利用 OSD(w)演算法所得到與接收信號相關性最大的碼字，於 OSD(1)機制結束後回饋額外的訊息量修正 LLR，使 LRIP 中錯誤位置能較快速被更正，而正確位置能夠加強信任度，使 MRPs 和 LRIPs 進行交換，使得解碼結果能不受到原來 MRPs 錯誤的影響而收斂至正確碼字。

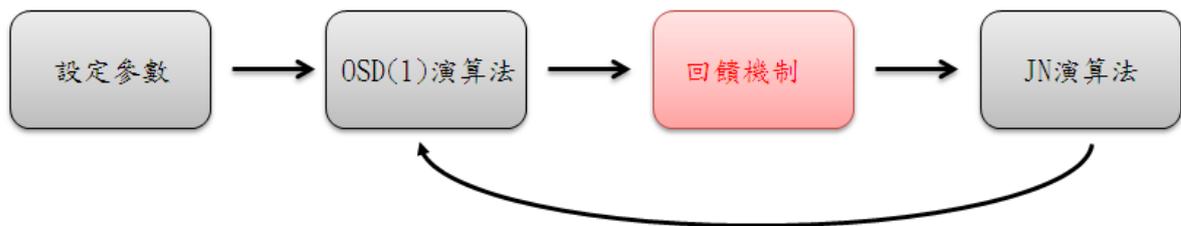


圖 2-12、回饋機制輔佐 ABP-OSD(1)之解碼流程示意圖

回饋機制、

於每次疊代過程中，儲存 OSD(1)機制中所得與接收信號相關性最大的碼字。若於 OSD(1)機制結束後，該碼字於某 LRP 的位置判斷為 0，則於該位置的 LLR 加上 $+A$ 的額外消息量，若該碼字於某 LRP 的位置判斷為 1，則於該位置的 LLR 加上 $-A$ 的額外消息量，其中 A 為可調整之係數，且 $A > 0$ 。

2-6.2 事前更正機制

[13]對於 MRPs 發生錯誤而導致錯誤傳遞提出另一種解決方法，在 JN-OSD(1)之前每次翻轉一個 MRPs 的信任度正負號，產生 K 組額外訊號來幫助解碼，此方法類似 OSD(1)，但效能高於 OSD(1)。原因是，OSD(1)對於 MRPs 中錯誤個數高於 1 個的接收向量無法正確解碼，但[13]可以仰賴 JN 傳遞機制，由其他正確位置更正 MRPs 的錯誤。

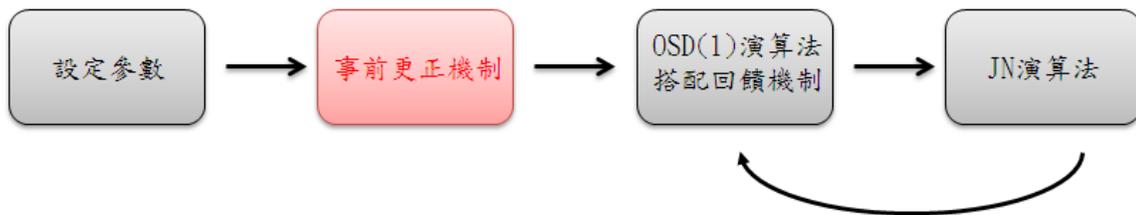


圖 2-13、事前更正機制輔佐 ABP-OSD(1)之解碼流程示意圖

事前更正機制、

- (1) 依據接收信號的 LLR 絕對值 $|L^{(l)}|$ 的大小將每個位元位置進行排序，並且記錄排序後的順序。利用高斯消去法，按照信任度由小至大的位置將 $H_b^{(0)}$ 消為簡化梯式矩陣。
- (2) 對於接收信號每次翻轉第 i 個 MRPs 的 LLR 正負號值，其中 $1 \leq i \leq K$ ，共得到 K 組額外信號進行解碼。

2-6.2 模擬結果

接下來為(15,7)里德所羅門碼於 JN-OSD(1)演算法加以回饋機制輔佐的模擬結果，由圖 2-可以發現，在效能上加以回饋機制輔佐較 JN-OSD(1)演算法的確由於解決了高信任度錯誤位置，而在位元錯誤率 10^{-5} 時，得到 0.45dB 的效能改善。

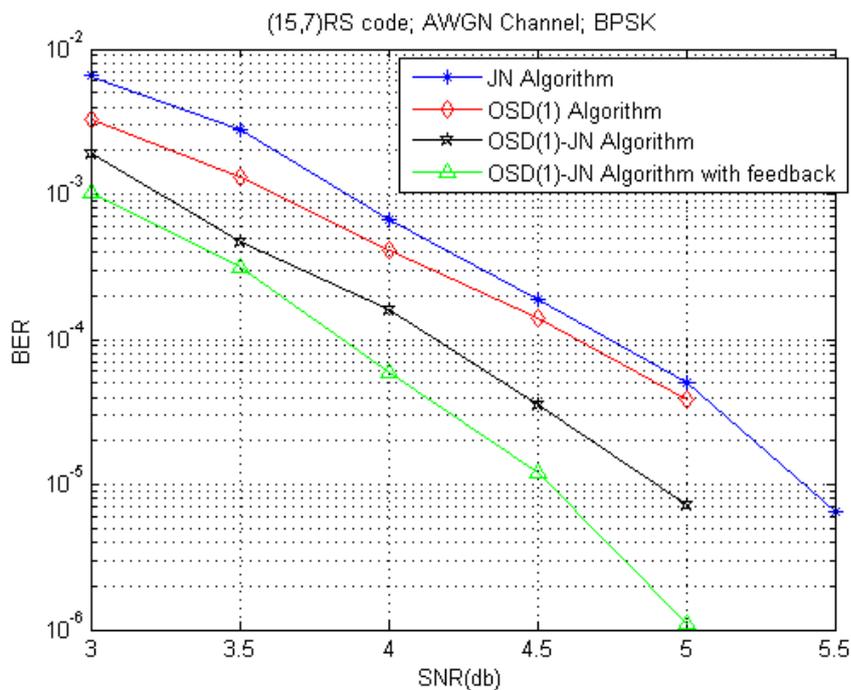


圖 2-14、(15,7)里德所羅門碼在四種演算法間比較

以下為(15,7)里德所羅門碼於 JN-OSD(1)演算法加以事前更正機制輔佐的模擬結果，由於事前更正機制的確在 JN-OSD(1)解碼前，預先更正了高信任度錯誤位置減少了錯誤傳遞，而提供了效能改善。由圖 2-15 可以發現，在效能上加以事前更正機制輔佐較 JN-OSD(1)演算法在位元錯誤率 10^{-5} 時，得到 0.35dB 的效能改善。

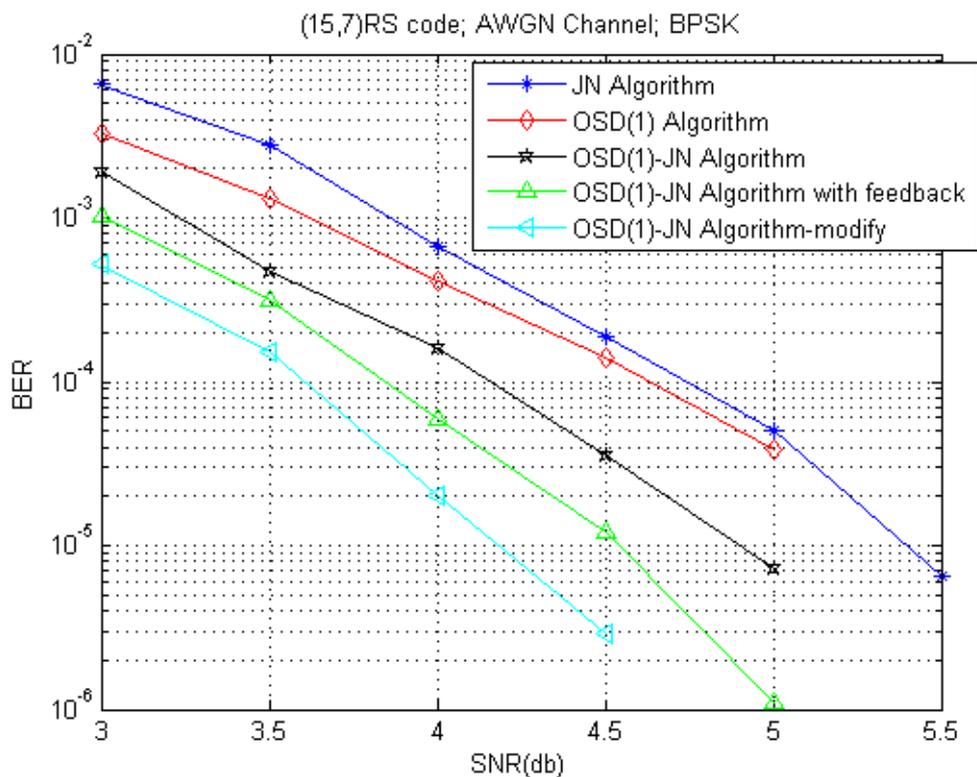


圖 2-15、(15,7)里德所羅門碼在五種演算法間比較

第三章

基於信念傳遞解碼之改良演算法

由於在 JN 演算法中，高度依靠高信任度的正確性，並透過利用高信任度位置經由 JN 演算法更正較容易發生錯誤的低信任度位置。因此，若高信任度位置發生錯誤，即使錯誤個數不多，還是會發生嚴重的錯誤傳遞現象。而為了抑制此現象，[12]和[13]分別在疊代過程中以及疊代前對訊號做處理。

3-1 初始機制

3-1.1 演算法基本概念與解碼步驟

在[12]疊代過程中，(1)JN 演算法輸出逐漸成為合法碼字，但輸出若是錯誤碼字時，因為滿足所有的校驗方程式，使得經過 JN 更新運算後只會增加此位置信任度值，並無法透過 JN 演算法進行更正，並且(2)持續加大信任度值也相對使得 OSD 回饋的影響力變小，亦即 OSD 演算法回饋的值是正確的，但卻沒辦法有效的降低錯誤位置的信任度。

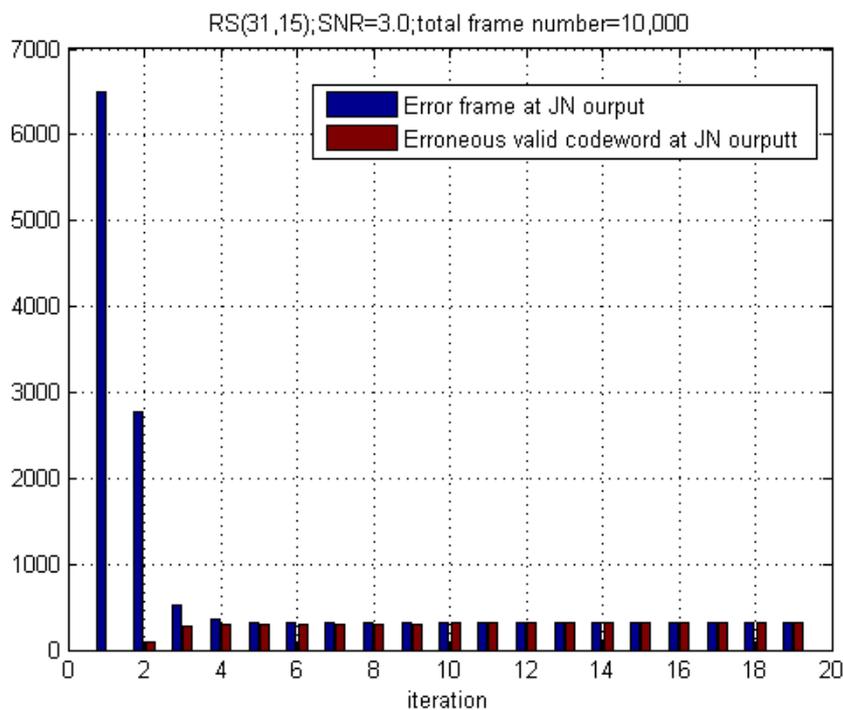


圖 3-1、執行不同次數疊代時 ABP 演算法輸出為錯誤和輸出為錯誤的合法碼字個數

上圖為 (31, 15)里德所羅門碼在 SNR=3.0dB 時，統計在執行不同次數疊代時 JN 演算法輸出為錯誤個數和 JN 演算法輸出為錯誤的合法碼字個數，而橫軸表示執行疊代次數，縱軸表示測式碼字中，發生此兩種情況的錯誤個數。由圖中得知，經過幾次疊代後 JN 演算法輸出為合法的錯誤碼字比例很高，此時若沒有額外機制擾動則無法再透過 JN 演算法進行更正。

因此提出初始化機制針對上述兩點改進

1. 透過初始化機制擾動錯誤的合法碼字，使得 JN 演算法能再度更正錯誤位置，並且有機會收斂至另一個碼字。
2. 初始化後降低 LLR 的信任度大小，使 OSD 的回饋量能夠有效的影響每個位置的信任度值。

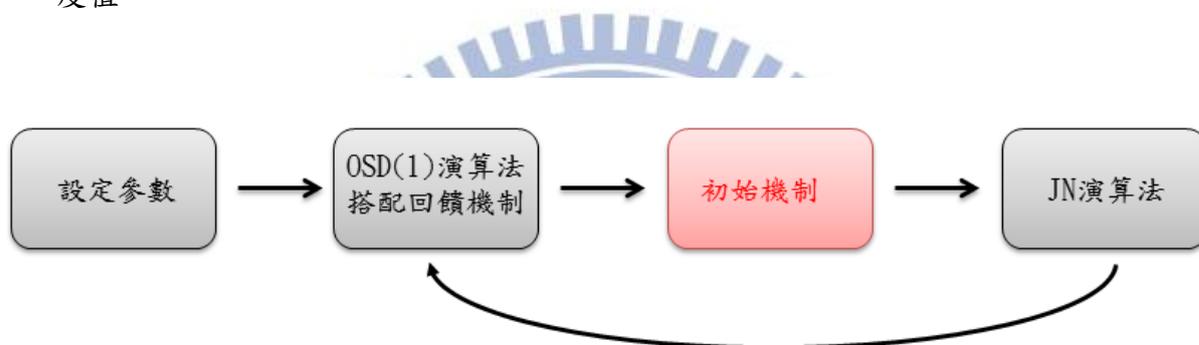


圖 3-2、初始化機制輔佐 ABP-OSD(1)之解碼流程示意圖

初始化機制、

每隔幾次疊代，將當下的 LLR 調整為接收訊號的 LLR 值，再透過 OSD(1)所得到與接收信號相關性最大的碼字。若於 OSD(1)機制結束後，該碼字於某 LRP 的位置判斷為 0，則於該位置的 LLR 加上 $+A$ 消息量，若該碼字於某 LRP 的位置判斷為 1，則於該位置的 LLR 加上 $+A$ 的額外消息量，其中 A 之係數，且 $A > 0$ ，OSD(1)機制回饋修正當下 LLR。表式示如下：

$$\begin{aligned}
 &LLR(c_i) = L^{(0)}(c_i) + A, i \in LRIP \\
 &else \\
 &LLR(c_i) = L^{(0)}(c_i)
 \end{aligned} \tag{3.1}$$

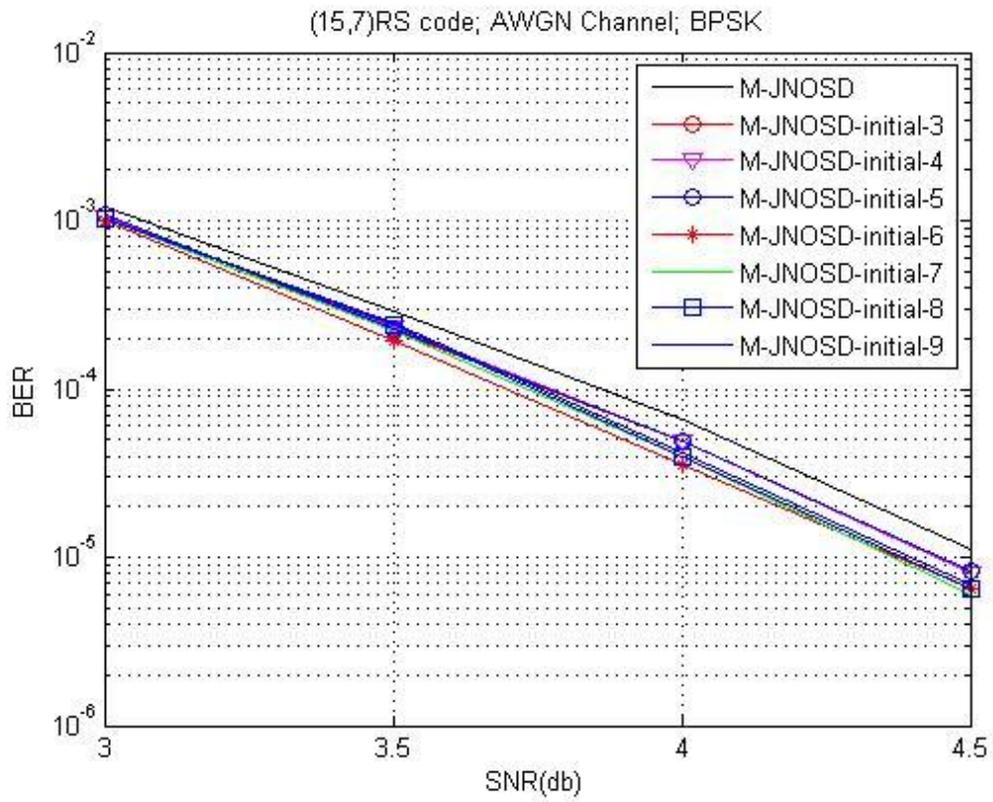


圖 3-3、(15, 7)里德所羅門碼配合初始機制在不同累積疊代次數初始 LLR 的情況



3-1.2 演算法數據觀察

接下來由模擬數據觀察並說明提出演算法和[12]分別在執行 5. 10. 15. 20 次疊代後，OSD 演算法所得到與接收訊號有最大相關性輸出碼字的高信任度位置錯誤分佈。橫軸表示 OSD 演算法輸出碼字在高信任度位置的錯誤個數，而縱軸表示測式碼字中，發生此情況的錯誤個數。此時提出演算法為每隔 6 次疊代初始化 LLR 值至接收訊號的 LLR 值，同時最大疊代次數為 20 次。

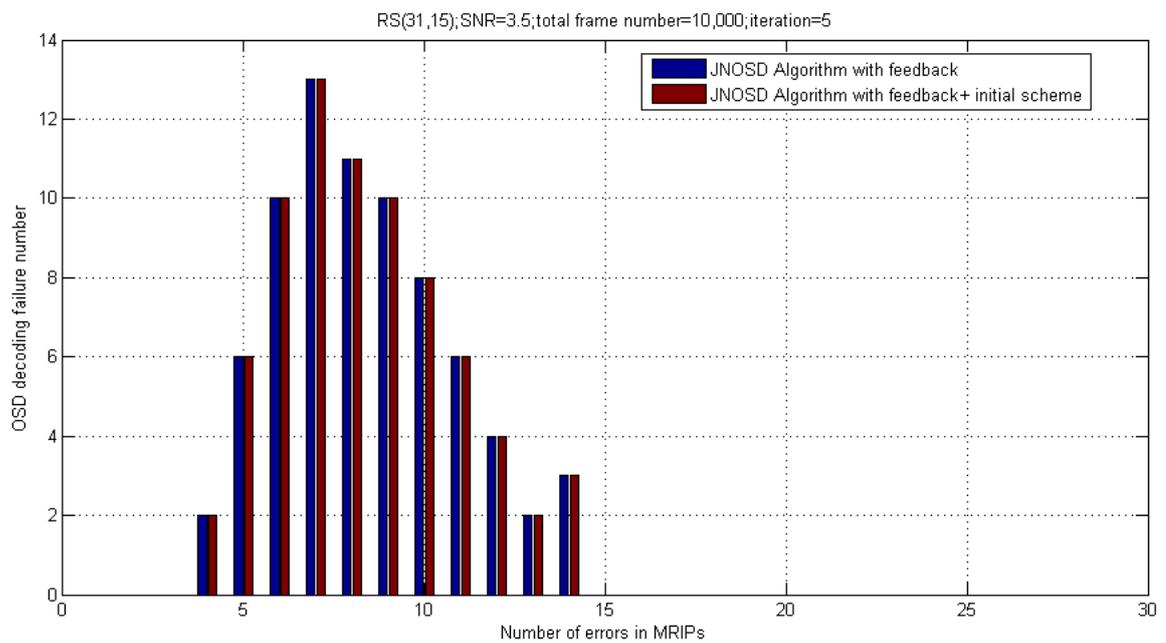


圖 3-4、提出演算法和[12]在執行 5 次疊代後，OSD 演算法所得到與接收訊號有最大相關性輸出碼字的高信任度位置錯誤分佈。

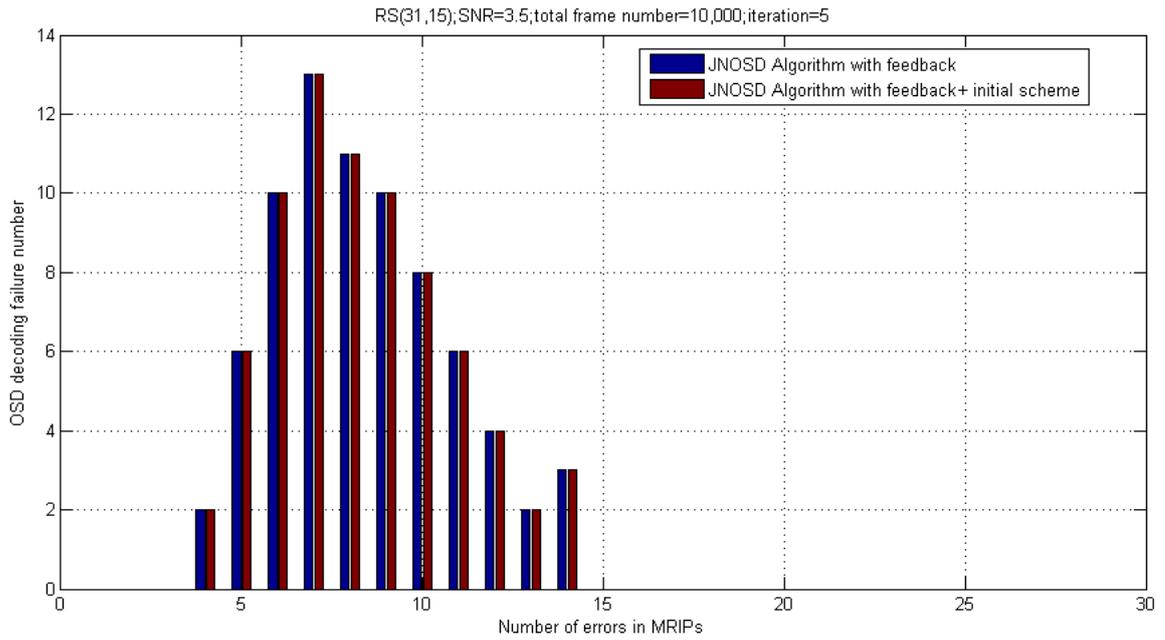


圖 3-5、提出演算法和[12]在執行 10 次疊代後，OSD 演算法所得到與接收訊號有最大相關性輸出碼字的高信任度位置錯誤分佈。

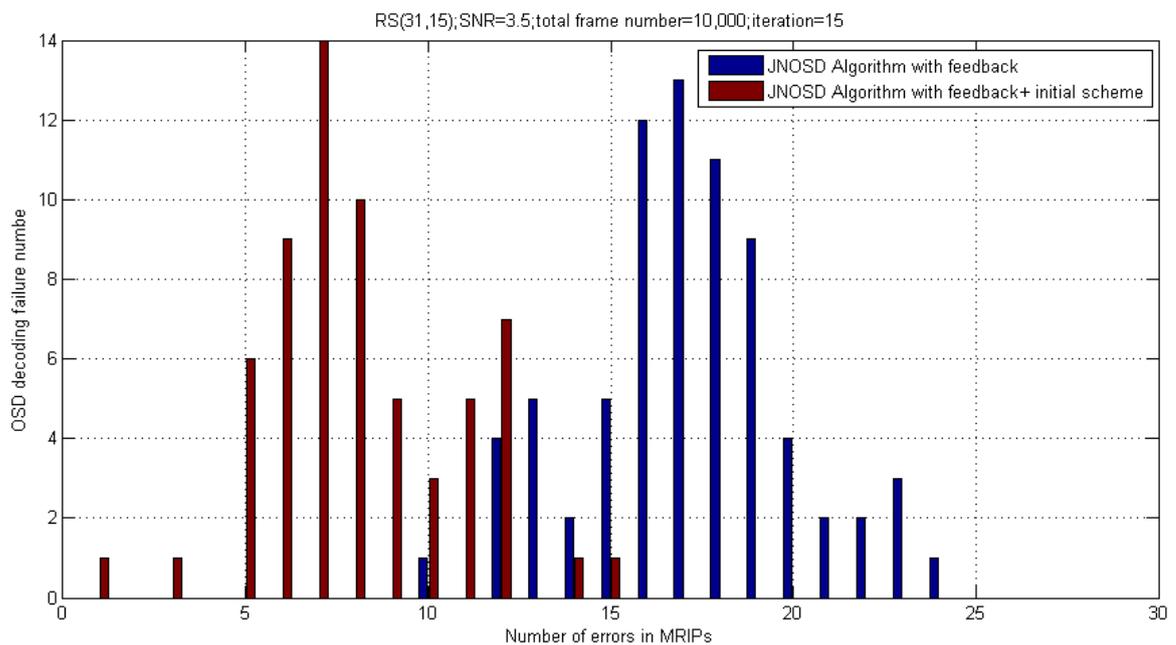


圖 3-6、提出演算法和[12]在執行 15 次疊代後，OSD 演算法所得到與接收訊號有最大相關性輸出碼字的高信任度位置錯誤分佈。

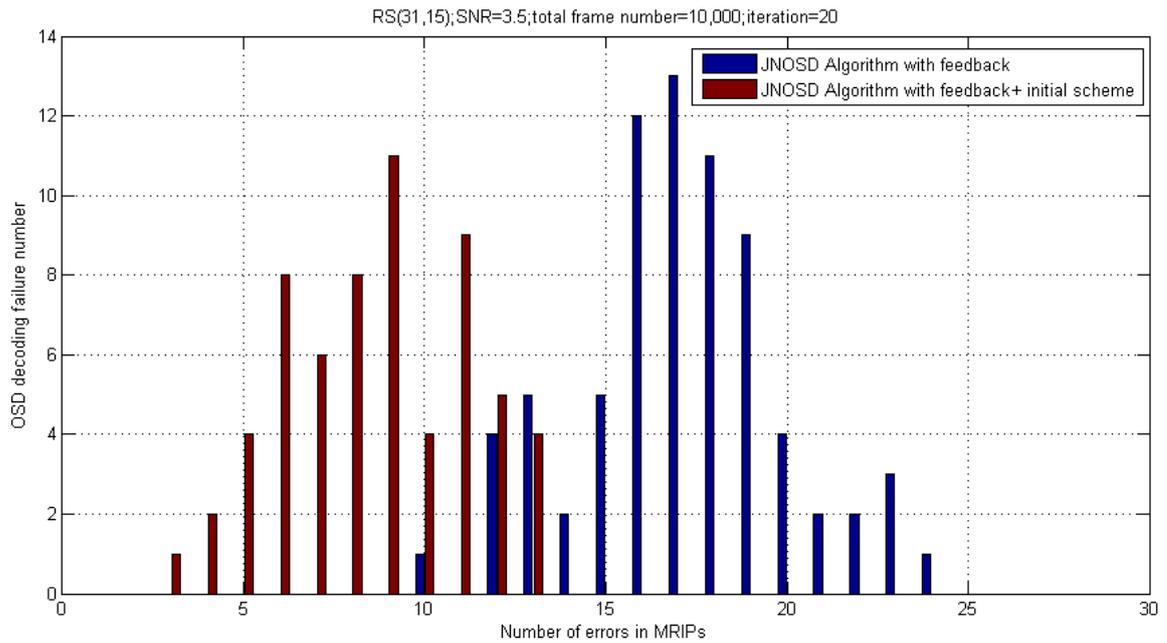


圖 3-7、提出演算法和[12]在執行 20 次疊代後，OSD 演算法所得到與接收訊號有最大相關性輸出碼字的高信任度位置錯誤分佈。

觀察[12]和提出演算法，前者高信任度位置錯誤個數較少的碼字，經過幾次疊代後逐漸被更正，而其餘高信任度位置錯誤個數較多的碼字無法透過[12]更正，因此分布逐漸集中至較多錯誤的區域，而提出演算法的高信任度位置錯誤分布相較於[12]較集中在較少錯誤的部分，這將有利於 OSD(1)演算法找到正確碼字，且 LLR 初始至接收信訊號的初始 LLR 向量後變小，這時 OSD 的回饋也比較有效的更正或降低錯誤位置的信任度。

3-1.3 模擬結果

接下來由模擬數據觀察並說明提出演算法分別在不同碼率下錯誤率分佈，此時提出演算法為每隔 6 次疊代初始 LLR 值至接收信號的 LLR 值，同時最大疊代次數為 20 次。

由於初始機制的確在疊代過程中合理的擾動錯誤的合法碼字之 LLR 值幫助解碼，而提供了效能改善。(15, 7)里德所羅門碼在位元錯誤率 10^{-5} 時，提出演算法和 JN-OSD(1)演算法加回饋機制方式相比得到 0.35dB 的效能改善。(31, 15)里德所羅門碼在位元錯誤率 10^{-3} 時，JN-OSD(1)演算法加上回饋機制方式有 0.3dB 的效能改善，而提出演算法和 JN-OSD(1)演算法加回饋機制方式相比得到 0.2dB 的效能改善。(63, 55)里德所羅門碼在位元錯誤率 10^{-3} 時，提出演算法和 JN-OSD(1)演算法有 0.4dB 的效能改善。

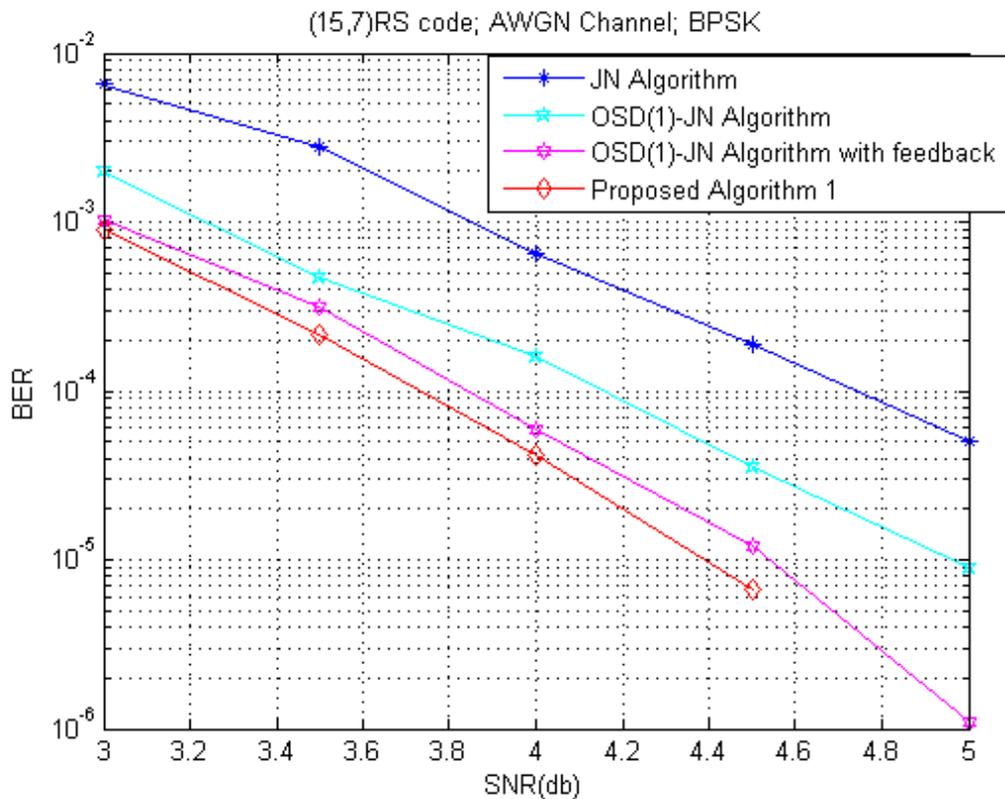


圖 3-8、比較(15, 7)里德所羅門碼於提出演算法一與不同演算法錯誤率

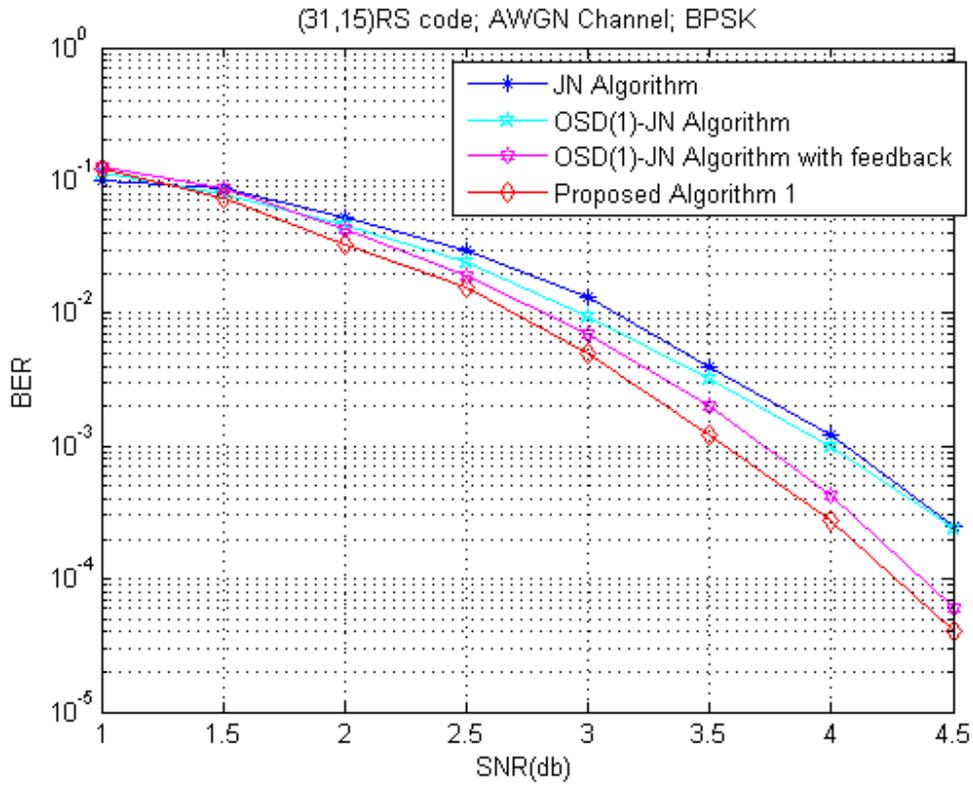


圖 3-9、比較(31, 15)里德所羅門碼於提出演算法一與不同演算法錯誤率

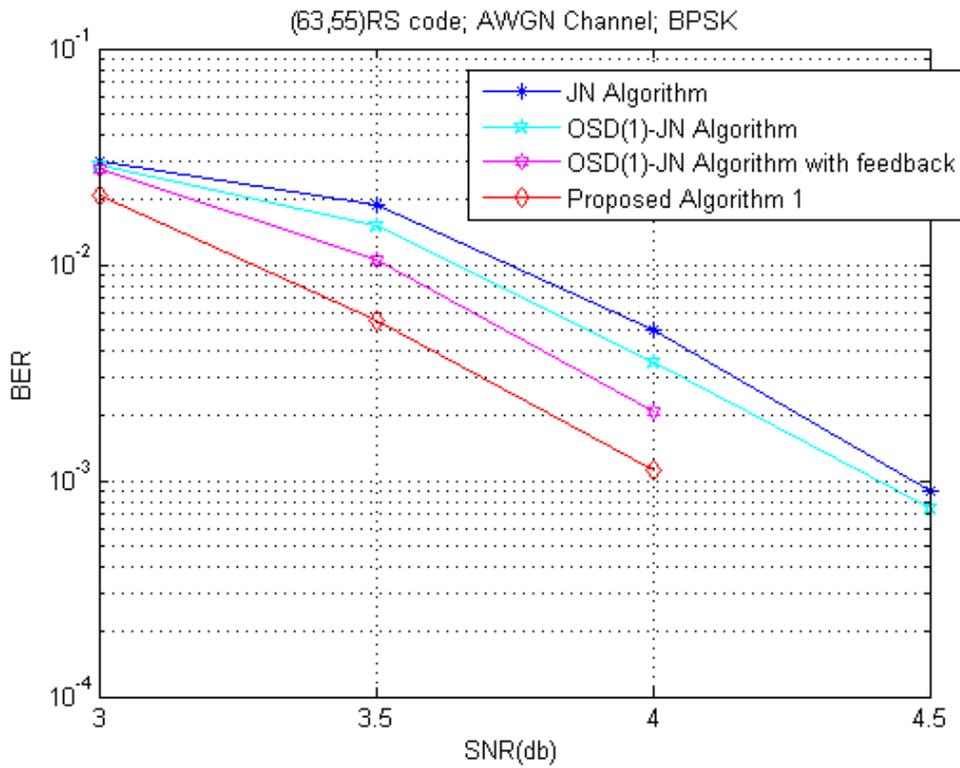


圖 3-10、比較(63, 55)里德所羅門碼於提出演算法一與不同演算法錯誤率

3-2 修正型事前更正機制

3-2.1 演算法基本概念與解碼步驟

為抑制 JN 演算法中，高信任度位置發生錯誤造成的錯誤傳遞現象，[13]對於此提出解決方法，在 JNOSD 之前翻轉每個 MRPs 的信任度，產生 K 組額外訊號來幫助解碼。

而[13]的方法在碼率低時較適合使用，隨著碼率越大複雜度呈現線性增加。因此本論文針對碼率高的里德所羅門碼，考慮以相對較少的個校驗節點去選擇要翻轉信任度的位置，選擇產生較少的 $(N-K)$ 組額外訊號來幫助解碼。本論文中在低於[13]複雜度的前提下，討論一個校驗節點選擇多個位置翻轉 LLR 值的情形，以下範例為一個校驗節點選擇一個位置翻轉 LLR 值正負號，說明圖如下：

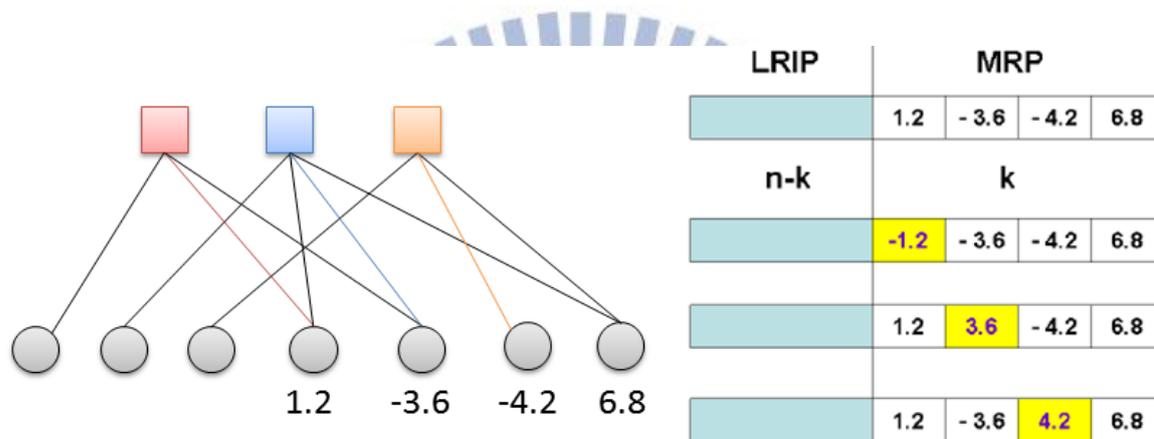


圖 3-11、提出演算法選擇一個位置翻轉 LLR 值正負號

以 $(7, 4)$ 里德所羅門碼為範例，一個以校驗節點所連接的位元節點中，選擇高信任度位置裡信任度最低的位元翻轉 LLR 正負號，產生額外 $N-K=3$ 組額外訊號幫助解碼，少於[13]產生的 $K=4$ 組額外訊號。

而當碼率越高時，圖 3-11、提出演算法選擇一個位置翻轉 LLR 值正負號選擇的 $(N-K)$ 組額外訊號將越小於 K 組額外訊號，因此在複雜度低於[13]的前提下，在一個校驗節點連接 MRPs 中額外選擇更多個位置翻轉 LLR 正負號，因此有機會能在 JN-OSD 解碼之前，事前更正較多個錯誤位置。

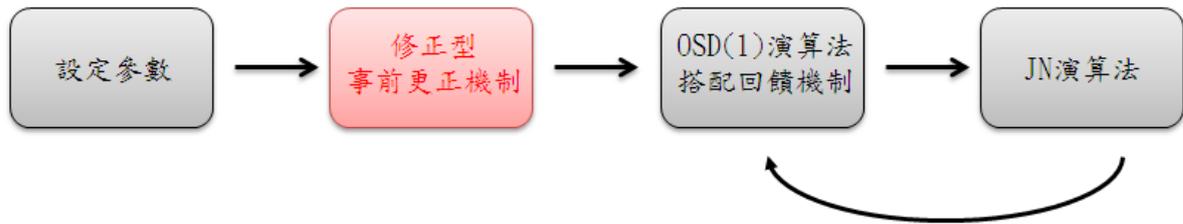


圖 3-12、修正型事前更正機制輔佐 ABP-OSD(1)之解碼流程示意圖

修正型事前更正機制、

(1) 依據接收信號的 LLR 絕對值 $|L^{(l)}|$ 的大小將每個位元位置進行排序，並且記錄排序

後的順序。利用高斯消去法，按照信任度由小至大的位置將 $H_b^{(0)}$ 消為簡化梯式矩陣。

(2) 對於接收信號對應的 tanner 圖中，尋找第 i 校驗節點相連的 MRPs 中信任度最低的位置，並翻轉其 LLR 正負號。若第 i 個校驗節點相連的 MRPs 中，信任度最低的位置與第 $1 \sim i-1$ 個校驗節點相同，則找次低信任度的位置。其中 $1 \leq i \leq N-K$ ，共得到 $N-K$ 組額外信號進行解碼

3-2.2 提出演算法之討論

討論 $(N-K)$ 里德所羅門碼，在低於 [13] 提出方法的複雜度前提下，一個校驗節點選擇多個位置翻轉 LLR 值正負號的情形。以下說明以 Flip(1)、Flip(2)、Flip(3) 統稱一個校驗節點選擇 1 個、2 個、3 個位置翻轉其 LLR 值正負號。

Flip(1): 碼率 $> 1/2$ 時，Flip(1) 複雜度比 [13] 低。

與校驗節點相連的 MRPs 中，選擇信任度最低的位置，翻轉其正負號。若第 i 個校驗節點相連的 MRPs 中，信任度最低的位置與第 $1 \sim i-1$ 個校驗節點相同，則找次低信任度的位置。另外，由於碼率 $< 1/2$ 時，若 K 個位置已經被第 $1 \sim k$ 個校驗節點選過了，則其餘校驗節點不再選擇位置翻轉 LLR 正負號。因此 FLIP 1 的複雜小於等於 Flip all。

Flip(2): 碼率 $> 3/4$ 時，Flip(2) 複雜度比 [13] 低。

與校驗節點相連的 MRPs 中，信任度最低與次低的位置，做 3 種翻轉正負號的組合。其中第 i 個校驗節點中，若最低信任度位置與第 $0 \sim i-1$ 個校驗節點重複，則信任度次低的位置不重複選擇，若次低的位置與第 $0 \sim i-1$ 個校驗節點相同，則找再次低信任度的位置。根據最低信任度位置是否重複選擇，分為兩種例子，其中黑色圈圈表示翻轉信任度

正負號的位置，說明圖如下：

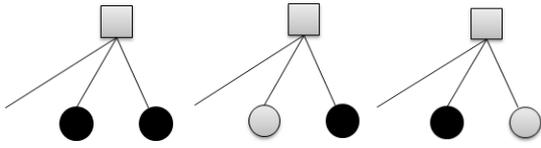


圖 3-13、第 i 個校驗節點中，選擇的兩個位置不與之前選擇的重複，共有三種組合

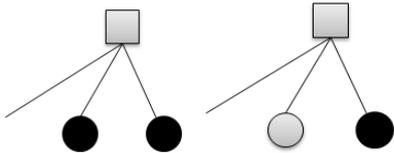


圖 3-14、第 i 個校驗節點中，選擇的信任度最低位置與之前選擇的重複，共有兩種組合

FLIP(3): 碼率 $> 7/8$ 時，FLIP(3) 複雜度比 [13] 低。

與校驗節點相連的 MRPs 中，信任度最低與次低、第 3 低的位置，做 7 種翻轉正負號的組合。其中第 i 個校驗節點中，若最低信任度位置或次低與第 $0 \sim i-1$ 個校驗節點重複，則信任度第 3 低的位置不重複選擇，若第 3 低的位置與第 $0 \sim i-1$ 個校驗節點相同，則找再低信任度的位置。根據最低信任度位置與次低信任度位置是否重複選擇，分為三種例子，其中黑色圈表示翻轉信任度正負號的位置，說明圖如下：

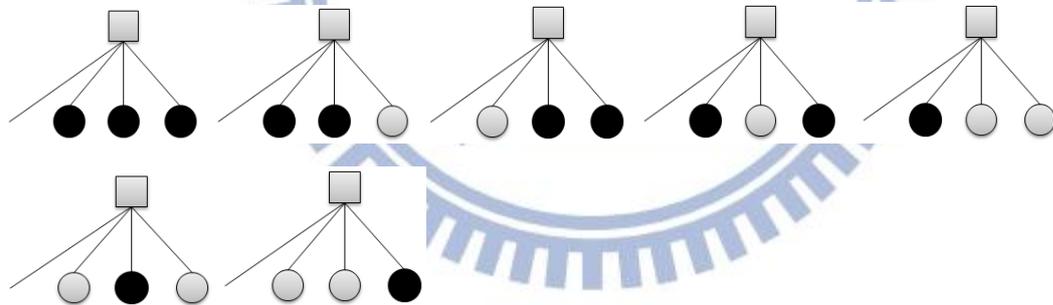


圖 3-15、第 i 個校驗節點中，選擇的三個位置不與之前選擇的重複，共有七種組合

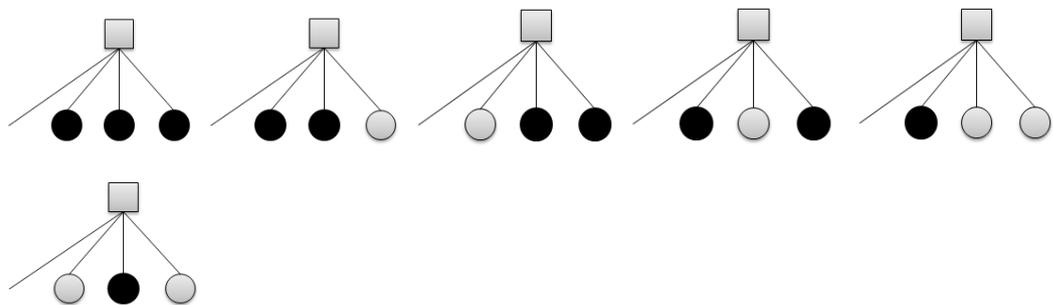


圖 3-16、第 i 個校驗節點中，選擇的信任度最低或次低位置與之前選擇的重複，共有六種組合

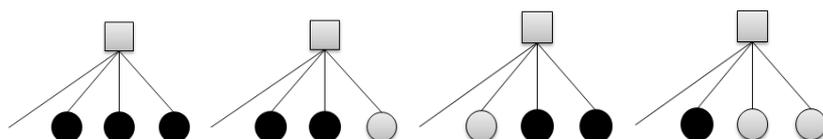


圖 3-17、第 i 個校驗節點中，選擇的信任度最低和次低位置與之前選擇的重復，共有四種組合

由上面討論中，若修正型事前更正機制滿足複雜低於事前更正機制的情況時，必滿足下列式子(3.3)

$$(n - k) \times (2^F - 1) + 1 \leq k + 1 \quad (3.3)$$

其中參數 F 表示每個校驗節點中至多選擇 F 個位置翻轉 LLR 正負號，即修正型事前更正機制 FLIP(F)。假設事前更正機制複雜度為 C ，則修正型事前更正機制相對應複雜度應為式子(3.4)

$$\frac{(n-k) \times (2^F - 1) + 1}{k+1} C \quad (3.4)$$

範例：(63, 55)里德所羅門碼事前更正與修正型事前更正機制複雜度比較

FLIP(1):

$$\frac{(378 - 330) \times (2^1 - 1) + 1}{330 + 1} C = \frac{49}{331} C = 0.148C$$

FLIP(2):

$$\frac{(378 - 330) \times (2^2 - 1) + 1}{330 + 1} C = \frac{145}{331} C = 0.438C$$

FLIP(3):

$$\frac{(378 - 330) \times (2^3 - 1) + 1}{330 + 1} C = \frac{337}{331} C = 1.0181C$$

3-2.3 模擬結果

接下來由模擬數據比較提出演算法在不同碼率下錯誤率分佈和複雜度的關係，其中最大疊代次數為 20 次。以下討論[13]所提出的方法皆以” FLIP(K)” 稱之，模擬曲線為” OSD(1)_JN Algorithm_modify” 。

由於修正型事前更正機制的確在 JN-OSD(1)解碼前，預先更正了高信任度錯誤位置減少了錯誤傳遞，而提供了效能改善。(63, 55)里德所羅門碼模擬中，FLIP(K)共生成 $K=330$ 組額外信號進行解碼，而提出演算法 FLIP(3)則生成 336 組額外信號進行解碼，因此在複雜度相近情況下於接近 ML bound 時仍然有 0.05dB 的效能改善。(31, 25)里德所羅門碼模擬中，FLIP(K)共生成 $K=125$ 組額外信號進行解碼，而提出演算法 FLIP(1)則生成 30 組額外信號、FLIP(2)生成 120 組額外信號進行解碼，因此 FLIP(1)複雜度為 FLIP(K)的 1/4 相近情況下達到相近的錯誤率表現。同樣地，(31, 23)里德所羅門碼模擬中，FLIP(1)以 1/4 的 FLIP(K)複雜度達到相近的錯誤率表現。

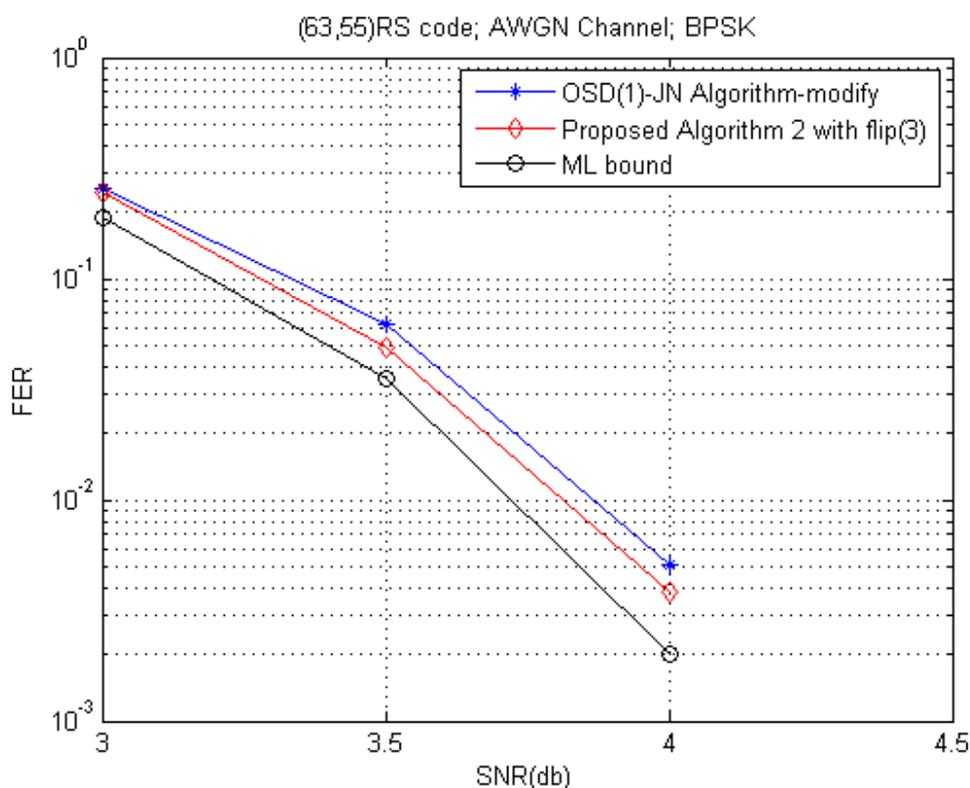


圖 3-18、比較(63, 55)里德所羅門碼於提出演算法二與不同演算法錯誤率

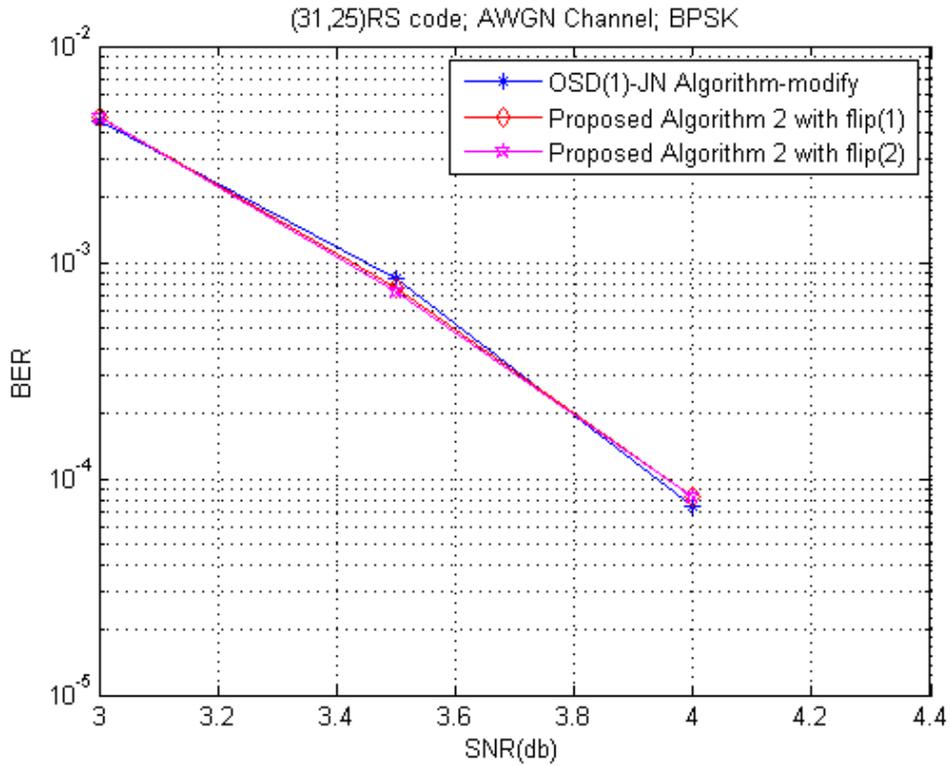


圖 3-19、比較(31, 25)里德所羅門碼於提出演算法二與不同演算法錯誤率

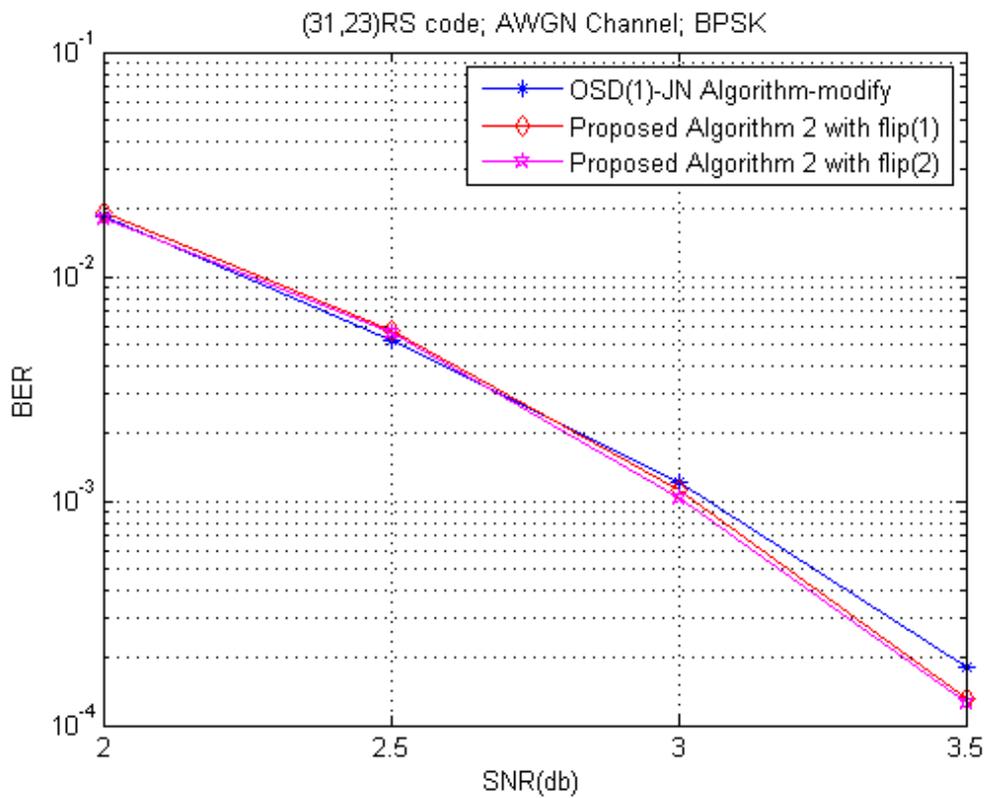


圖 3-20、比較(31, 23)里德所羅門碼於提出演算法二與不同演算法錯誤率

3-3 信念傳遞混合列表解碼機制

3-3.1 演算法基本概念與解碼步驟

將 OSD(1) 利用較不易發生錯誤的高信任度位置重新編碼，以得到較接近接收信號的概念使用在 JN 疊代過程中，幫助加快更正低信任度位置的錯誤。操作方式為將原本奇偶校驗矩陣 H_b 隨機分割成較少列的 H'_b ，由於 H'_b 只連接至一部分位元節點，因此也只會連接至一部分的錯誤位置，而當錯誤個數較少時利用 OSD(1) 演算法重新編碼，較容易找到正確碼字。由原本奇偶校驗矩陣 H_b 中切割較少列 H'_b ，提出演算法為隨機以不重複的兩個校驗節點為一組形成新的 H'_b ，說明圖如下：

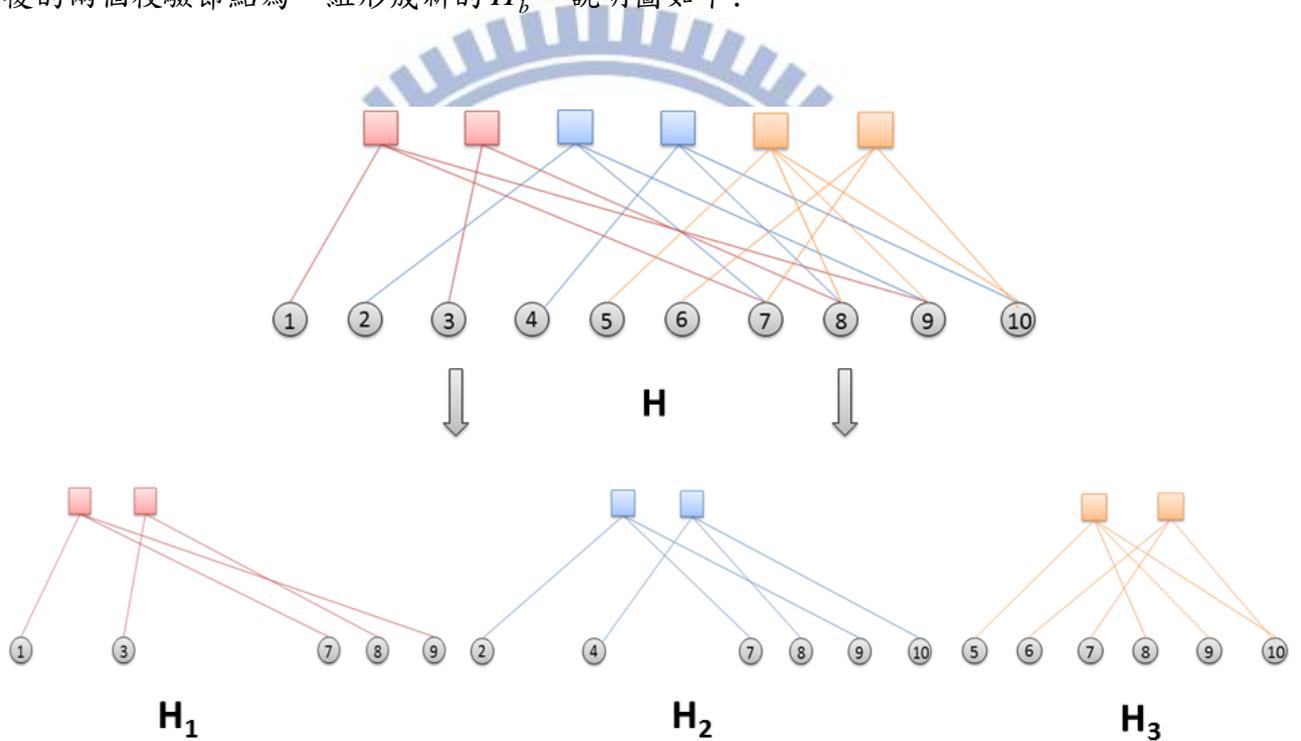


圖 3-21、示範分割奇偶校驗矩陣 H_b 成較少列的 H'_b

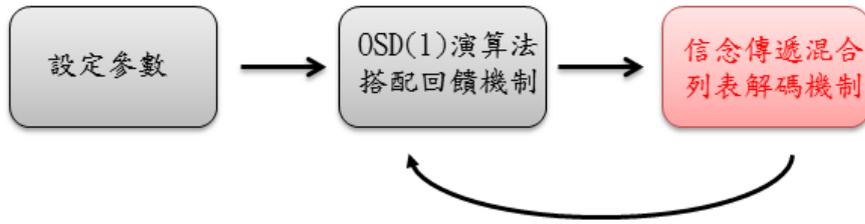


圖 3-22、信念傳遞混合列表解碼機制輔佐 ABP-OSD(1)之解碼流程示意圖

信念傳遞混合列表解碼機制、

- (1) 將 H_b 中不重複的兩列一組，隨機分割成 $\left(\frac{N-K}{2}\right)$ 個 H'_b 。
- (2) 每一個分割後的 H'_b ，若其矩陣大小為 $(2 * M)$ ，則每次翻轉一個 MRP 總共得到 $(M-1)$ 組額外信號並以此編碼得到 $(M-1)$ 組候選碼字。
- (3) 記錄每個 H'_b 所產生候選碼字中相關性最大的碼字後，歸納每個位置被判斷為 0 或 1 的結果，若位元 c_i 判斷為 0 的比例高於 1 的位置，則於經過 JN 更新訊息 LLR 值後再加上 $+\Delta$ 修正量，反之若位元 c_i 判斷為 1 的比例高於 0 的位置，則於經過 JN 更新訊息 LLR 值後加上 $-\Delta$ 修正量

$$LLR(c_i) = \begin{cases} LLR(c_i) + \Delta, & \text{if generated codebit} = 0 \\ LLR(c_i) - \Delta, & \text{if generated codebit} = 1 \\ LLR(c_i) & , \text{else} \end{cases} \quad (3.5)$$

3-3.2 模擬結果

接下來由模擬數據觀察並說明提出演算法和其他演算法分別在不同碼率下錯誤率分佈，最大疊代次數為 20 次。

(15, 7)里德所羅門碼模擬中，在位元錯誤率 10^{-5} 時，提出演算法和 JN-OSD(1)演算法加回饋機制方式相比得到 0.15dB 的效能改善。(31, 15)里德所羅門碼在位元錯誤率 10^{-4} 時，提出演算法和 JN-OSD(1)演算法加回饋機制方式相比得到 0.15dB 的效能改善。(31, 25)里德所羅門碼在位元錯誤率 10^{-5} 時，提出演算法和 JN-OSD(1)演算法有 0.1dB 的效能改善。

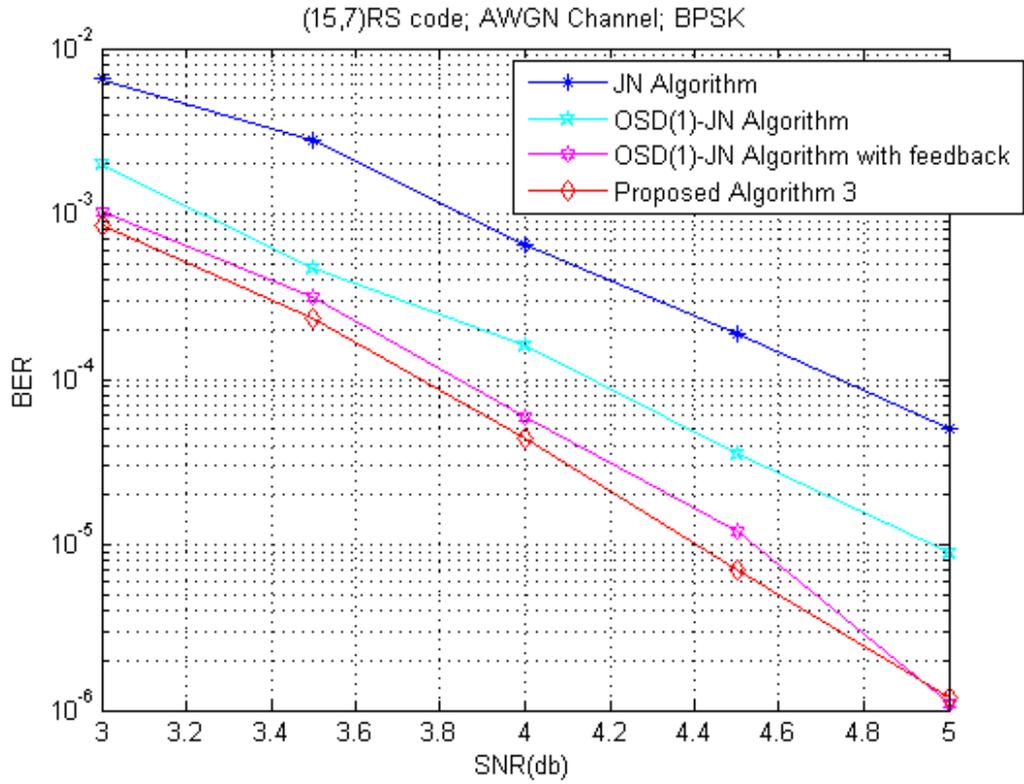


圖 3-23、比較(15, 7)里德所羅門碼於提出演算法三與不同演算法錯誤率

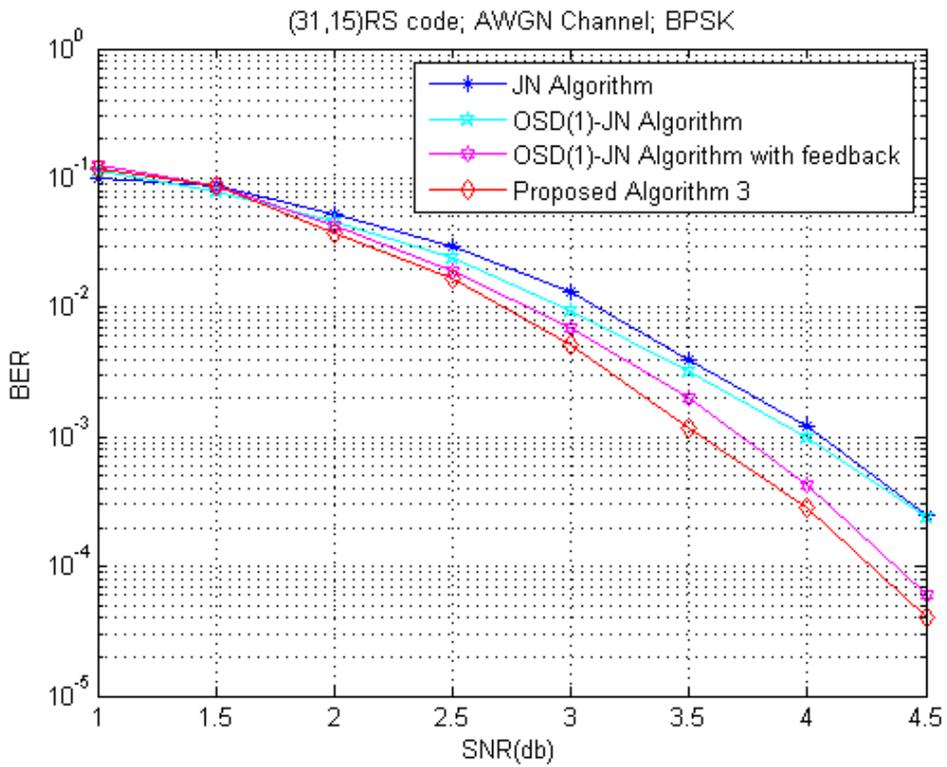


圖 3-24、比較(31, 15)里德所羅門碼於提出演算法三與不同演算法錯誤率

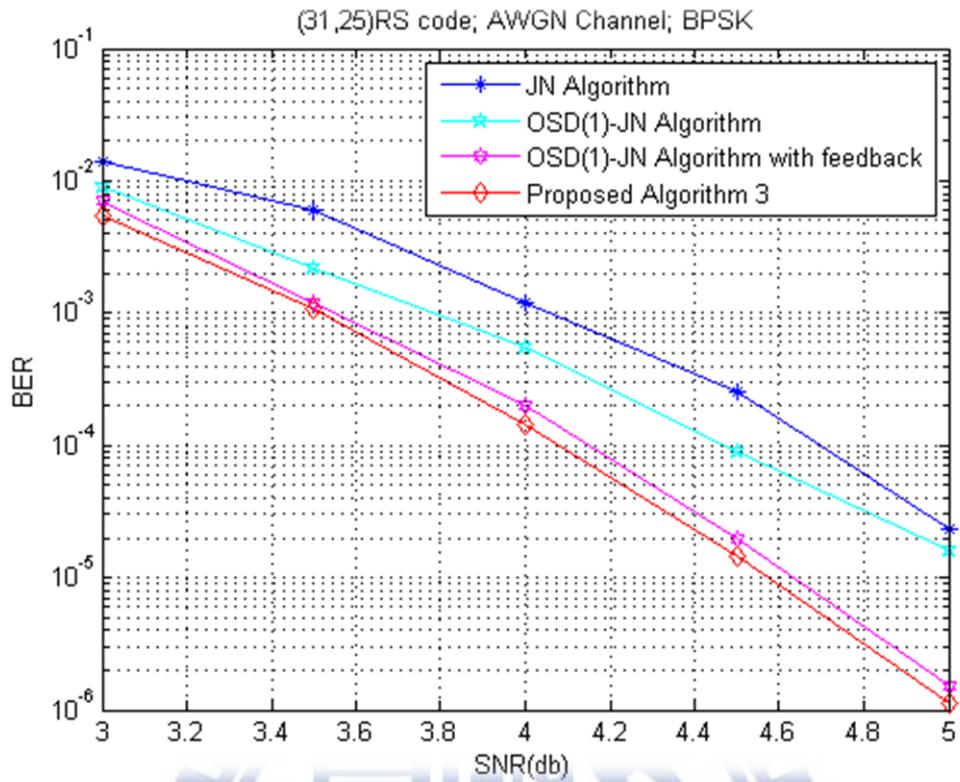


圖 3-25、比較(31, 25)里德所羅門碼於提出演算法三與不同演算法錯誤率



3-3.3 綜合提出演算法的模擬結果與討論

此節將討論混合提出演算法模擬結果，比較(31, 15)里德所羅門碼之提出演算法一、三和 JN-OSD(1)演算法加回饋機制模擬結果，在位元錯誤率 10^{-4} 時分別有 1.5dB 和 1.0 dB 的效能改善，而提出演算法一結合三則有更好的效能改善，和 JN-OSD(1)演算法加回饋機制相比有 1.8 dB 的效能改善。(15, 7)里德所羅門碼模擬圖中，在位元錯誤率 10^{-5} 時，提出演算法一、三和 JN-OSD(1)演算法加回饋機制方式相比皆得到 0.15dB 的效能改善，而同樣地結合提出演算法一和三則效能改善加大，大約得到 0.3 dB 的效能改善。(31, 25)里德所羅門碼在位元錯誤率 10^{-5} 時，綜合提出演算法一和三得到有 0.25dB 的效能改善。(63, 55)里德所羅門碼模擬中，提出演算法二時結合提出演算法一在接近 ML bound 仍然有小幅度的改善。在下列的 FER 模擬圖中，將以[15]中提出的方法計算 ML Union lower bound 並以 ML union LB 表示，而 ML LB 為[15]中的模擬結果。

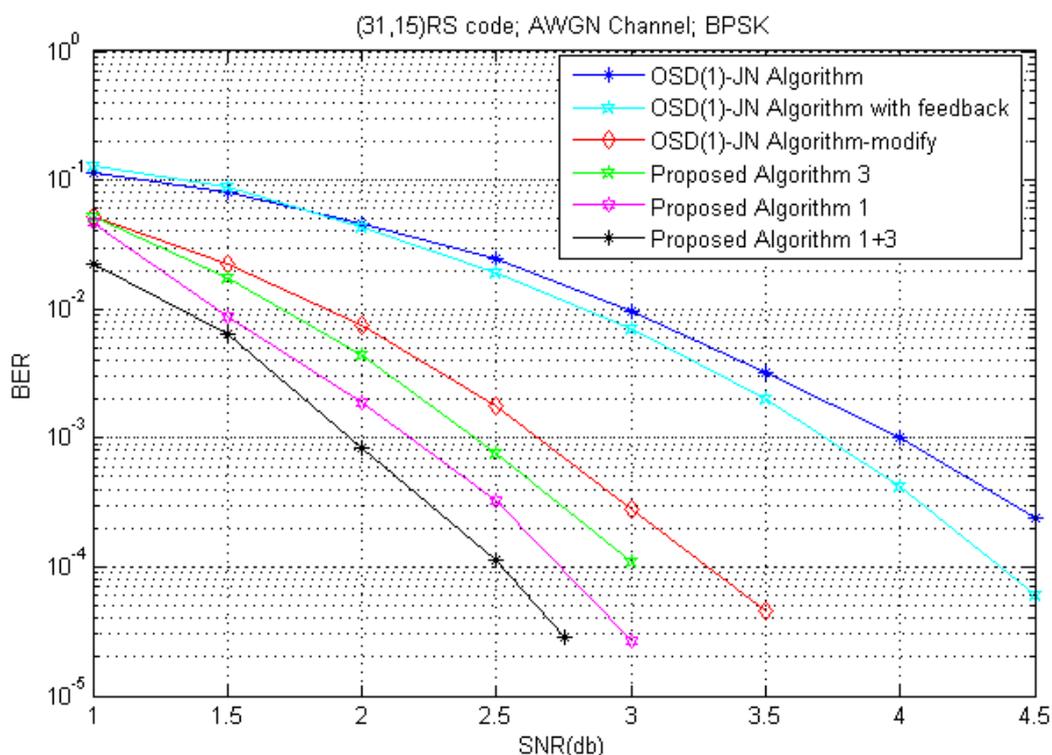


圖 3-26、比較(31, 15)里德所羅門碼於不同演算法 BER 錯誤率

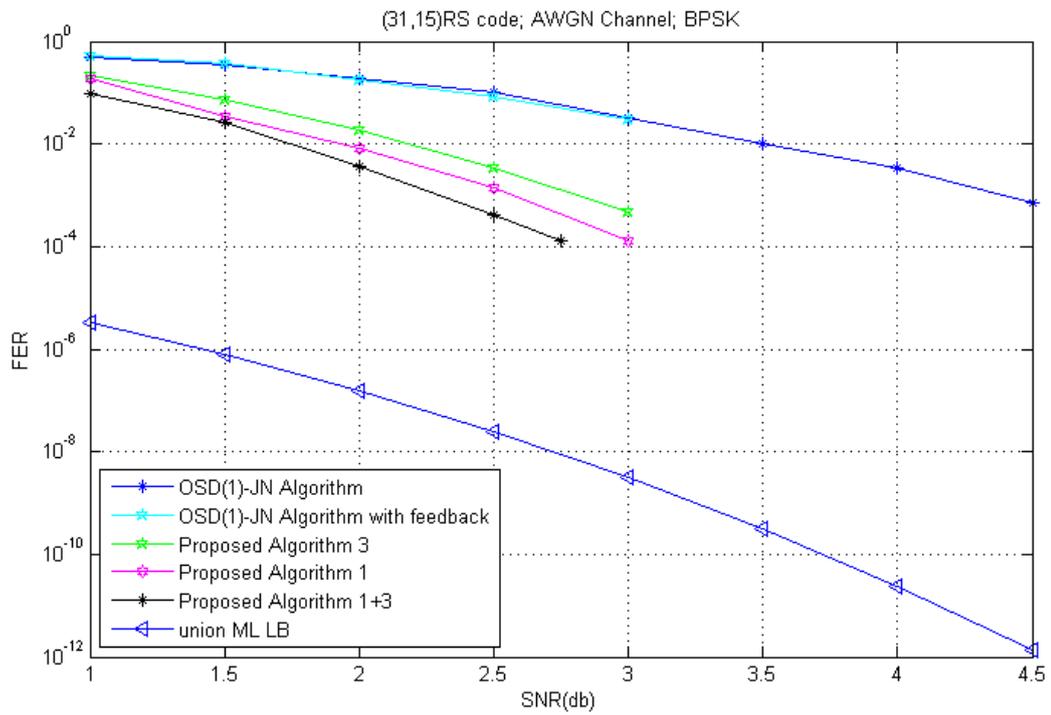


圖 3-27、比較(31, 15)里德所羅門碼於不同演算法 FER 錯誤率

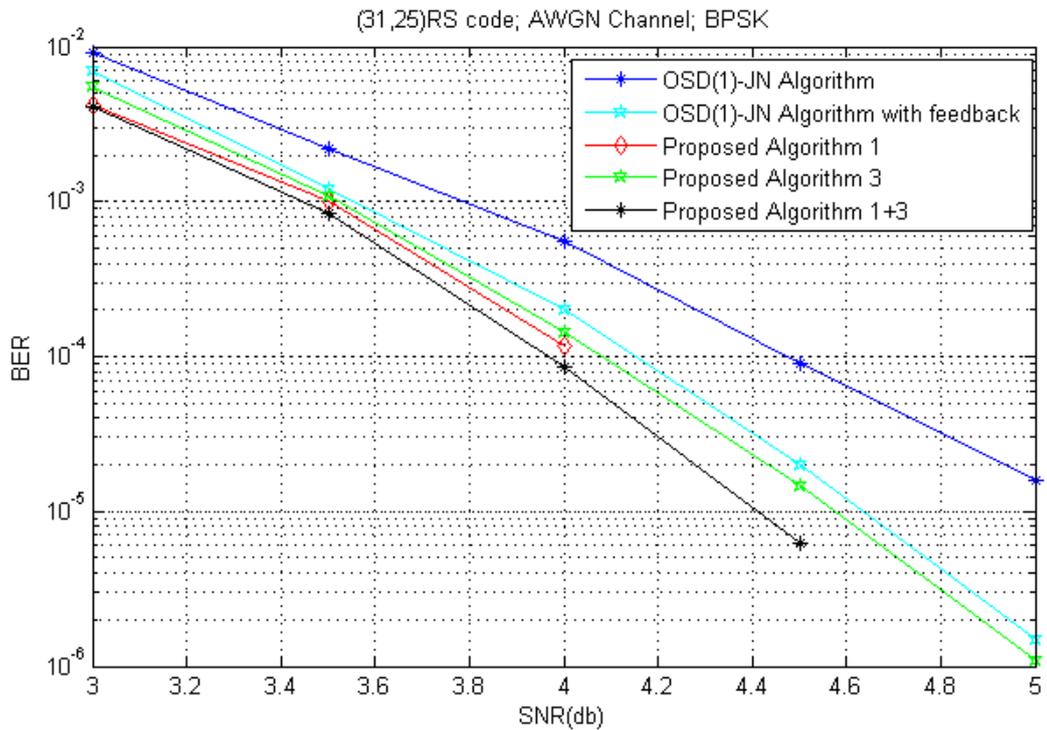


圖 3-28、比較(31, 25)里德所羅門碼於不同演算法 BER 錯誤率

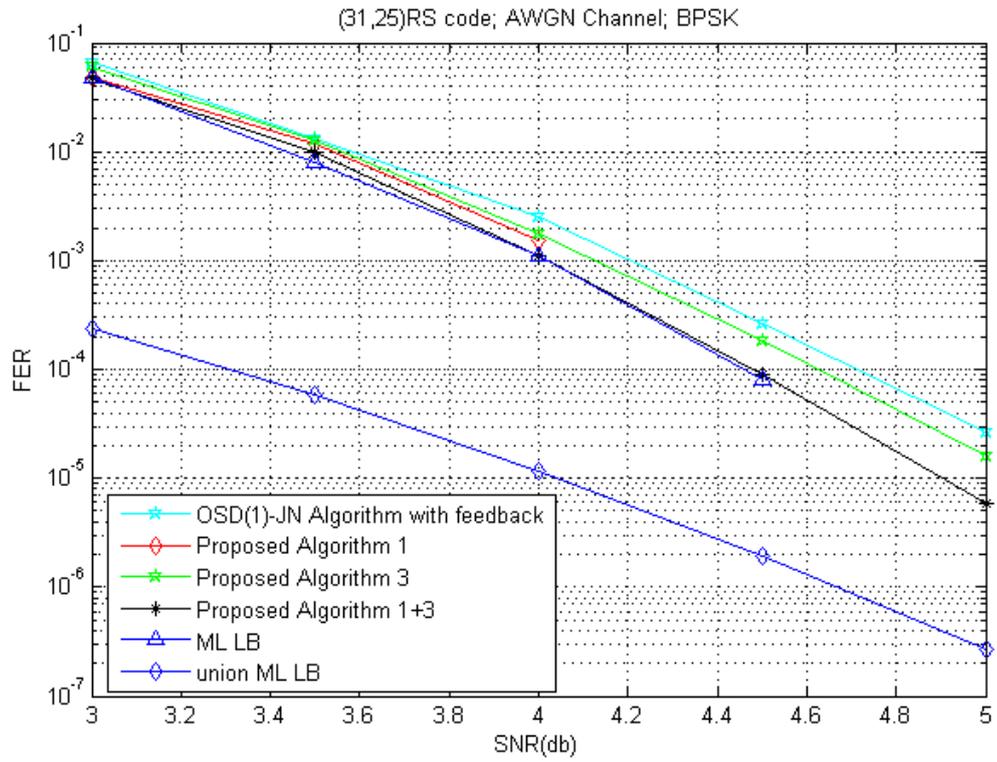


圖 3-29、比較(31, 25)里德所羅門碼於不同演算法 FER 錯誤率

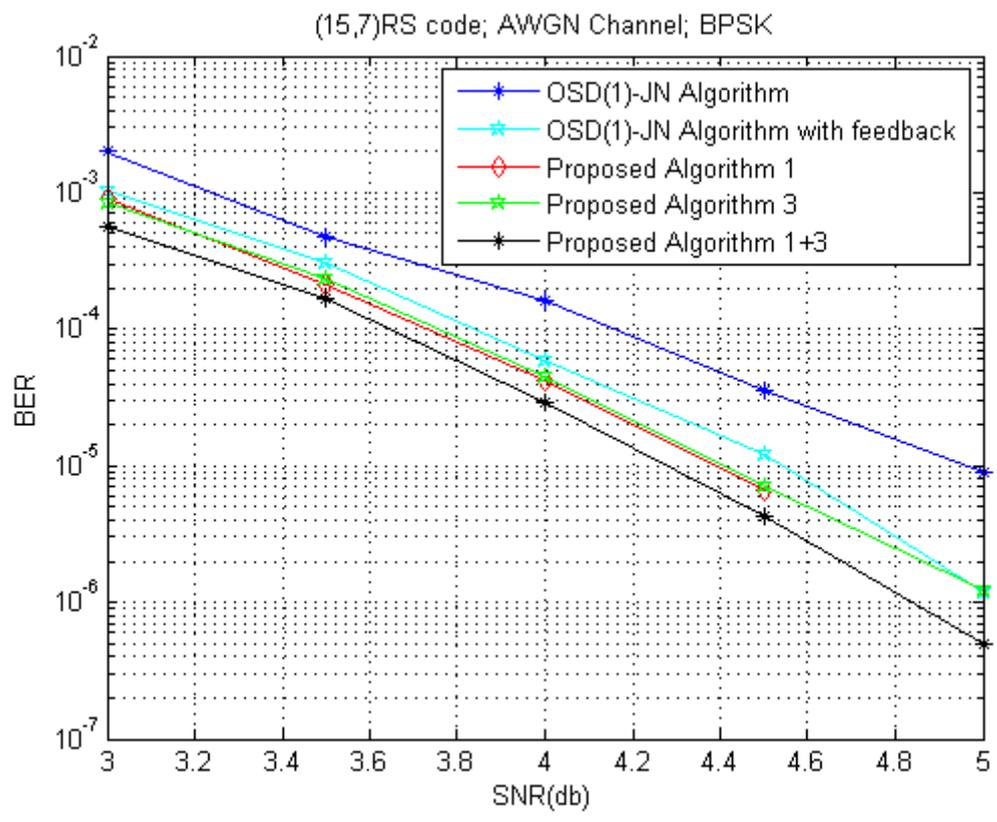


圖 3-30、比較(15, 7)里德所羅門碼於不同演算法 BER 錯誤率

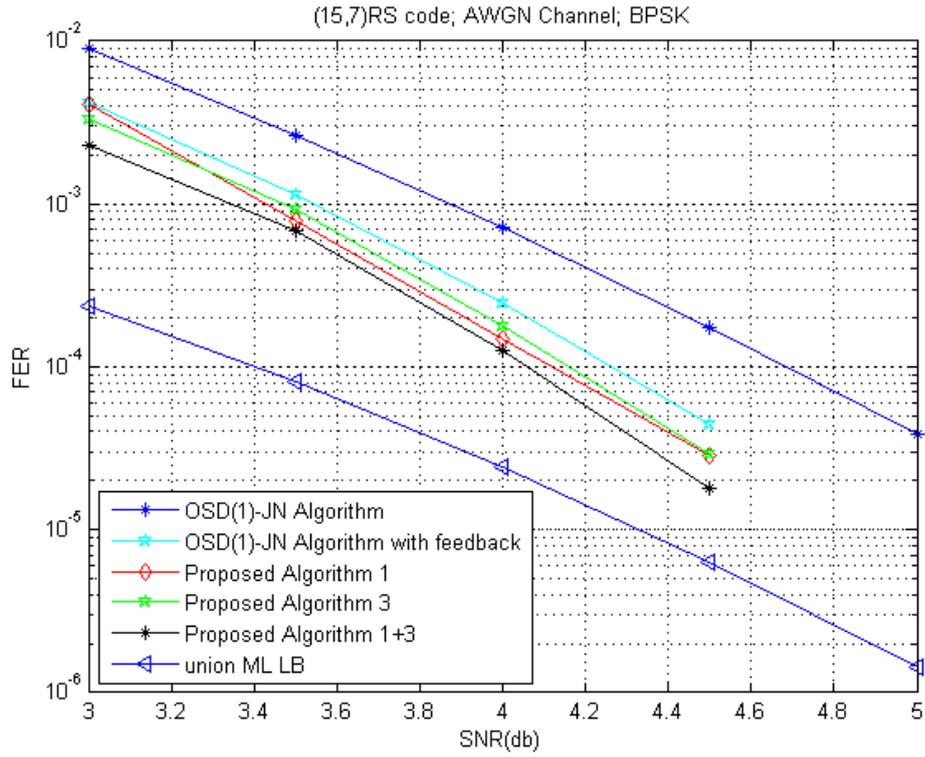


圖 3-31、比較(15, 7)里德所羅門碼於不同演算法 FER 錯誤率

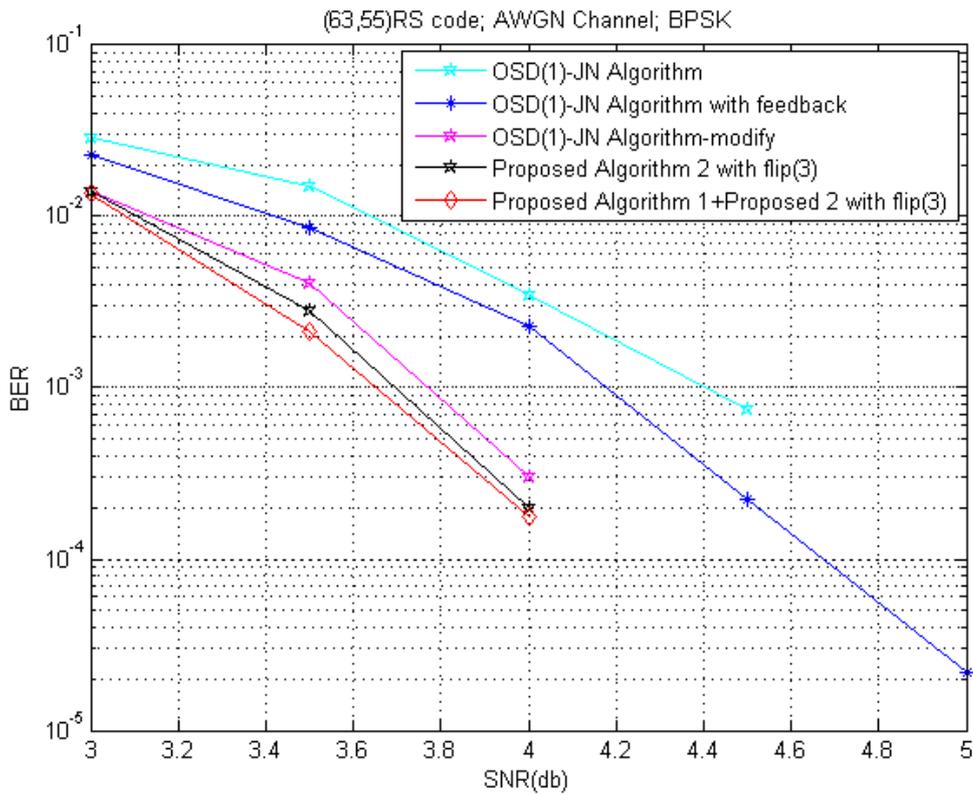


圖 3-32、比較(63, 55)里德所羅門碼於不同演算法 BER 錯誤率

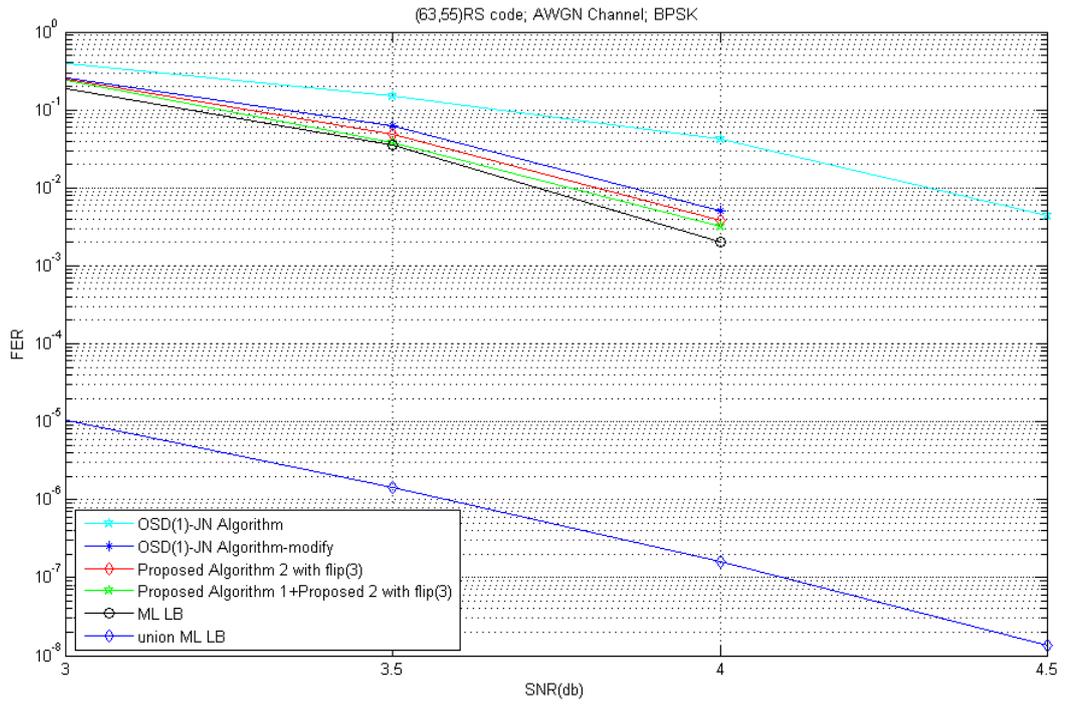


圖 3-33、比較(63, 55)里德所羅門碼於不同演算法 FER 錯誤率



第四章

結論

在本論文中，一開始介紹里德所羅門碼的二位元映射以及運算方式，並以此概念下將里德所羅門碼的解碼方式視為二位元的線性區塊碼(block linear codes)的解碼。接下來介紹 JN 演算法與 OSD(w)演算法，並針對 JN 演算法的錯誤傳遞問題進行討論，以及介紹針對此問題的相關修正機制演算法。

而在本論文中於第三章中提出三種機制，初始機制、修正型事前更正機制與信念傳遞混合列表解碼機制。由模擬數據顯示，針對疊代過程中錯誤的合法碼字進行初始機制的確合理的擾動 LLR 值幫助解碼，此動作使 JN 演算法能再度更正錯誤位置，並且有機會收斂至另一個碼字，而提供了效能改善。而修正型事前更正機制提出在複雜度低於[13]的前提下，在 Tanner 圖裡由一個校驗節點連接高信任度位置中，額外選擇比[13]提中方法中更多個位置翻轉 LLR 正負號，以達到在解碼前更正更多個高信任度位置錯誤位置，以減少錯誤傳遞的影響。而信念傳遞混合列表解碼機制將 OSD(1)演算法的概念運至 JN 演算法疊代過程中，此概念為利用較不容易發生錯誤的高信任度位置重新編碼，以得到與接收信號具有相關性最大的碼字並回饋修正量修正 LLR，以幫助低信任度位置的錯誤加速被更正，且加速提昇低信任度正確位置的信任度，進而和高信任度位置進行交換，並佐以模擬數據顯示了效能改善。最後，綜合提出三種演算法機制，由模擬數據顯示結合不同的提出演算法將能得到最佳的效能改善。

參考文獻

- [1] I.S.Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inf. Theory*, vol. IT-4, pp. 38-39, Sep. 1954.
- [2] J. Jiang and K. R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 52, pp. 3746-3756, Aug. 2006.
- [3] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon codes and algebraic geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1757-1767, Sep 1999.
- [4] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, pp. 2809-2825, Nov. 2003.
- [5] D.Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, pp.170-182, Jan. 1972
- [6] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. IT-12, pp. 125-131, Apr. 1966.
- [7] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, pp. 1379-1396, Sep. 1995.
- [8] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes based on adaptive parity-check matrices," in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, Illinois, Jul. 2004, pp. 261.
- [9] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 244-246, Apr. 2004.
- [10] A. Kothiyal, O.Y. Takeshita, W. Jin and M. Fossorier, "Iterative reliability-based decoding of linear block codes with adaptive belief propagation," *IEEE Commun. Lett.*, vol. 9, pp. 1067-1069, Dec. 2005.
- [11] F.J. MacWilliams and N.J.A.Sloane, *The Theory of Error Correcting Code*. New York: NorthHolland, 1977.
- [12] Y. M. Hsieh, "A Study on Belief-Propagation Based Decoding Algorithms for Reed-Solomon Codes," National Chiao Tung University, Hsinchu, Taiwan, R.O.C., Master Thesis, 2007.
- [13] C. H. Kuo, "An Improved Belief Propagation Based Decoding Algorithm for RS code," National Chiao Tung University, Hsinchu, Taiwan, R.O.C., Master Thesis, 2009.
- [14] J. Hagenauer, E.Oer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 429-445, Mar. 1996

- [15] M. E. Khamy and R. J. McEliece, "Bounds on the average binary minimum distance and the maximum likelihood performance of Reed-Solomon codes," in *Proc. Allerton*, Monticello, IL, Oct. 2004.
- [16] M. El-Khamy and R. McEliece, "Iterative algebraic soft-decision list decoding of RS codes," *IEEE J. Select. Areas Commun.*, vol. 24, pp.481-490, Mar. 2006.

