

國立交通大學

電信工程研究所

碩士論文

應用於 V-BLAST 系統下之以廣度為主做搜尋的多層
排序演算法

Multi-level Sorted Breadth-First Signal Decoder for V-BLAST
Systems

研究生：曾耀葦

指導教授：張文鐘 博士

中華民國 九十九 年 七 月

應用於 V-BLAST 系統下之以廣度為主做搜尋的多層排序演算法

Multi-level Sorted Breadth-First Signal Decoder for V-BLAST
Systems

研究生：曾耀葦

Student : Yao-Wei Tseng

指導教授：張文鐘 博士

Advisor : Dr. Wen-Thong Chang

國 立 交 通 大 學
電 信 工 程 研 究 所
碩 士 論 文



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

Computer and Information Science

July 2010

Hsinchu, Taiwan, Republic of China

中華民國 九十九 年 七 月

應用於 V-BLAST 系統下之以廣度為主做搜尋的多層排序演算法

研究生：曾耀葦

指導教授：張文鐘 博士

國立交通大學電信工程研究所

摘要

在多根傳送天線及多根接收天線的系統下 (multi-input multi-output, 簡稱為 MIMO), 雖然這種架構可以增加傳輸容量, 但對接收端來說, 如何將多根天線間彼此互相干擾的傳輸資料進行估測與解碼是主要的議題, 而學者們近年來已經提出了數個解碼效能接近最佳估測的演算法 (maximum-likelihood detection, 簡稱為 ML detection)。本篇論文著重於此系統中 V-BLAST 架構 (Vertical Bell Laboratories Layered Space-Time) 之接收端的估測, 在錯誤率與複雜度的平衡間, 我們結合了 N-QAM 多層架構搜尋的降低複雜度特性、排序 QR 分解的特定解碼順序與適應性調整門檻來限制搜尋路徑個數的演算法 (breadth-first signal decoder, 簡稱為 BSIDE) [12] 而衍生出一個解碼效能逼近最佳估測且複雜度低的演算法, 稱為 "以廣度為主做搜尋的多層排序演算法" (multi-level sorted breadth-first signal decoder, 簡稱為 MSBSIDE 演算法)。經過數學分析與電腦模擬, 此演算法確實能以很低的複雜度來逼近 ML 估測的效能, 且越高階的調變越能顯現出 N-QAM 多層架構降低複雜度的特性, 所以此演算法能在複雜度與解碼效能之間的取捨取得最佳的平衡。

Multi-level Sorted Breadth-First Signal Decoder for V-BLAST Systems

Student: Yao-Wei Tseng

Advisor: Dr. Wen-Thong Chang

Institute of Communication Engineering
National Chiao Tung University

ABSTRACT

The use of multi-input multi-output (MIMO) architectures promises to achieve higher transmission capacity, but how to detect and decode the transmitted signals from the interference of other antennas for the receiver is the most important topic for discussion. In recent years, a number of detecting schemes have been proposed to perform maximum-likelihood (ML) detection. In this paper, we focus on the detecting algorithm design of the Vertical Bell Laboratories Layered Space-Time (V-BLAST) receiver. Consider the tradeoff between detecting error probability and detecting complexity, we combine the breadth-first signal decoder (BSIDE algorithm) [12], which can limit the number of searching paths by an adaptive threshold, with multi-level searching structure of the N -QAM modulation constellation and specific detecting order of sorted QR decomposition to form a novel detecting algorithm, called “multi-level sorted breadth-first signal decoder (MSBSIDE algorithm)”, and we can let the performance of this novel algorithm approach ML detector. Through analysis and computer simulation, it is shown that the MSBSIDE algorithm has the same bit-error-rate performance as the conventional ML detectors while allowing significantly lower computational complexity. In addition, using multi-level searching structure of the N -QAM modulation constellation for higher level modulation can reduce more computational complexity, so this novel algorithm can achieve a better performance-complexity tradeoff.

誌謝

時光荏苒，兩年的時光匆匆流逝，碩士生涯至此也算圓滿結束，對於我的指導教授張文鐘博士，心中只有滿溢的感謝，老師對於研究的專業，總是能夠在我研究遭遇困難及瓶頸時給予我最精準又最精闢的指導，處事踏實又懂得享受生活的人生哲學，老師給我們研究生做了一個最好的榜樣。同時也感謝口試委員王晉良教授、林銀議教授與林大衛教授，各位口試委員都以精準的眼光和專業能力給了研究上和論文上的建議，讓我的論文能夠更加的完備，很感謝他們。

對於能成為 821 實驗室的一員我滿懷感謝，實驗室的學長、學弟妹和同學們都是我不論研究上或是生活上的好夥伴，博班學長家豪；96 級學長琮壹；98 級和 99 級學弟妹舒評、信妤、維哲、耀駿、詩倩；97 級同學怡如、明穎、雅嵐；助理立杰。有了大家在，讓我的研究生生活精彩非凡，希望大家還能時常相聚。另外，還要謝謝家人特地從遠方寄來的綜合維他命軟膠囊素寶丁，在我趕論文的日子都能給我很多精神上的支持。

最後也是最重要的，我最感謝我的家人，家中父母親總不吝惜長途電話費，讓我能聽到他們聲音，了解家鄉近況，一解思鄉之愁；哥哥元駿像朋友一樣給了我許多建議，他是我的榜樣；女朋友詩薇總是在我低潮的時候陪我度過以及加油打氣；還有要感謝交大土地公的保佑，讓我可以在交大兩年都很平順。沒有大家的支持，就沒有現在的我，我很感謝。

誌於 2010.夏 新竹。交大
耀葦

目 錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	v
圖目錄	vi
一、	緒論.....	1
1.1	背景及動機.....	1
1.2	論文架構.....	3
二、	系統模型.....	4
2.1	V-BLAST 系統架構.....	4
2.2	排序的 QR 分解(sorted QR decomposition).....	6
三、	以廣度為主做搜尋的排序演算法.....	12
3.1	以排序 QR 分解為基礎的 M 演算法(SQRD-M 演算法).....	12
3.2	以廣度為主做搜尋的排序演算法(SBSIDE 演算法).....	17
四、	以廣度為主做搜尋的多層排序演算法.....	22
4.1	N-QAM 調變訊號座標集合的多層架構.....	22
4.2	基本概念.....	24
4.3	演算法描述.....	26
4.4	演算法總結.....	33
4.5	附錄.....	34
4.5.1	計算 $\tilde{n}_{j,L}$ 的平均值和變異數.....	34
4.5.2	兩種計算第 l 層初步估計門檻 d_l 的方法，其中 $l=2\cdots L$	35
4.5.2.1	以最大值為基礎(maximum based)計算 d_l	36
4.5.2.2	以平均值為基礎(mean based)計算 d_l	39
五、	模擬結果.....	44
六、	結論.....	51
參考文獻	52

表 目 錄

表 2.1	排序的 QR 分解演算法.....	8
表 3.1	決定迴授等化估測演算法(DFE 估測演算法).....	17



圖 目 錄

圖 2.1	V-BLAST 系統的傳送端架構.....	5
圖 2.2	V-BLAST 系統的接收端架構.....	5
圖 2.3	在 4×4 的 V-BLAST 系統、調變為 4-QAM 下使用原始 QR 分解與排序 QR 分解於 DFE 估測演算法的錯誤率比較.....	10
圖 3.1	SQRD-M 演算法的流程圖.....	13
圖 3.2	$N_T = 3, N_R = 3$, 運用 SQRD-M 演算法做估測的樹狀圖示.....	15
圖 3.3	SQRD-M 演算法與 modified SQRD-M 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較.....	16
圖 3.4	SQRD-M 演算法與 modified SQRD-M 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的錯誤率比較.....	17
圖 3.5	SBSIDE 演算法的流程圖.....	18
圖 3.6	$N_T = 3, N_R = 3$, 運用 SBSIDE 演算法做估測的樹狀圖示.....	20
圖 4.1	64-QAM 的多層架構.....	22
圖 4.2	64-QAM 第二象限所有訊號的樹狀關係圖.....	24
圖 4.3	MSBSIDE 演算法的流程圖.....	26
圖 4.4	$N_T = 3, N_R = 3$, 運用 MSBSIDE 演算法做估測的樹狀圖示.....	32
圖 4.5	在 64-QAM 訊號座標圖下以最大值為基礎計算估計門檻時, 各層的訊號偏差值大小及訊號值涵蓋的範圍.....	38
圖 4.6	在 64-QAM 訊號座標圖下以平均值為基礎計算估計門檻時, 各層的假想 N-QAM 調變訊號座標圖.....	42
圖 5.1	OSIC-ZF 演算法、OSIC-MMSE 演算法、SGA 演算法、SQRD-M 演算法與 SBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較.....	45
圖 5.2	SQRD-M 演算法與 SBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較.....	46
圖 5.3	SQRD-M 演算法、SBSIDE 演算法、MSQRD-M 演算法和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的錯誤率比較.....	47

圖 5.4	SBSIDE 演算法、MSBSIDE-MAX 演算法和 MSBSIDE-MEAN 演算法在 4×4 的 V-BLAST 系統、調變使用 64-QAM 下的錯誤率比較.....	48
圖 5.5	SQRD-M 演算法($M=16$)、SBSIDE 演算法、MSQRD-M 演算法($M=16$)和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的複雜度比較.....	49
圖 5.6	SBSIDE 演算法、MSBSIDE-MAX 演算法和 MSBSIDE-MEAN 演算法在 4×4 的 V-BLAST 系統、調變使用 64-QAM 下的複雜度比較.....	50



一、緒論

1.1 背景及動機

在最近幾年的研究中，學者已經證明多根傳送天線與多根接收天線的系統架構在多路徑干擾的無線通訊環境下可以提供很高的頻譜效率 (spectral efficiency)，如參考文獻[1]、[2]。而在這些眾多的多根傳送天線與多根接收天線架構中，又以 V-BLAST 系統架構 (Vertical Bell Laboratories Layered Space-Time) 是最簡單來達成高頻譜效率的架構，如參考文獻[13]。V-BLAST 系統並不包含編碼器及解碼器，只是將原始訊號分成多個子資料串，這些子資料串經過調變後，經由不同的傳送天線來負責傳輸不同的子資料串訊號，傳送端與接收端的架構分別如圖 2.1 與圖 2.2。在這樣的系統架構下，由不同傳送天線傳出的訊號彼此間都會互相干擾，所以接收端如何藉由接收到的訊號並運用特定的估測演算法來區分不同根傳送天線傳送之訊號為此系統架構主要著重的議題。

參考文獻[3]、[5]—[12]提出了幾種在 V-BLAST 系統架構下有效率的估測演算法。理論上，ML 估測在此架構下擁有最佳的估測效能，它的做法是將所有天線可能的傳送訊號組合都做估測，再選取最可能發生的傳送訊號組合當成估計解；以四根傳送天線 $N_T=4$ ，使用 QPSK 調變為例子，傳送訊號可能的組合就會有 $4^4=256$ 種，對越高階的調變及越多根傳送天線來說，接收端的複雜度會呈指數遞增，而對現今越來越講究無線通訊的傳輸品質與傳輸速度來說，ML 估測所需負擔的複雜度已不可行。參考文獻[3]提出的 zero-forcing based or minimum mean square error based ordered successive interference cancellation (OSIC-ZF 或 OSIC-MMSE 演算法) 雖然可以降低複雜度，但在估測錯誤率的表現上並不理想。在複雜度與解碼效能間取得較好的平衡為近幾年提出的次序性高斯近似法 (sequential Gaussian approximation, 簡稱為 SGA 演算法[8])、以 QR 分解為基礎的 M 演算法 (QR decomposition based M algorithm, 簡稱為 QRD-M 演算法[5])及 sphere decoder (SD) 演算法。SGA 演算法必須算出通道矩陣的反矩陣，而反矩陣運算量較高，且演算法過程中會讓每根天線間彼此獨立的白色高斯雜訊變成非獨立的雜訊，所以接收端必須付出額外的複雜度來做估測。QRD-M 演算法並不需要計算通道矩陣的反矩陣，只需將通道矩陣進行 QR 分解，其運算量也較低；因此，本篇論文提出的演算法是以 QRD-M 演算法為基礎所衍生出的以廣度為主做搜尋的演算法

(breadth-first signal decoder, 簡稱為 BSIDE [12]), 再結合 N-QAM 的多層架構與排序 QR 分解的特定解碼順序而來。

QRD-M 演算法為了解決 ML 估測因全域搜尋所帶來的高複雜度, 接收端在估測傳送訊號前, 先對通道矩陣 \mathbf{H} 做 QR 分解; 由於 \mathbf{Q} 矩陣的正交特性與 \mathbf{R} 矩陣的倒三角形特性, 讓解碼由一次估測全部天線的全域搜尋變成一次估測一根天線的次序性解碼, 且過程中考慮到某些部分天線之傳送訊號組合發生的機率極低, 所以不需做全域搜尋, 次序性解碼過程中每次都只保留 M 組發生機率較高的部分天線之傳送訊號組合。除此之外, 為了提高這 M 組部分天線之傳送訊號組合的正確性, 將 QRD-M 演算法的原始 QR 分解改成排序的 QR 分解, 簡稱為 SQRD-M 演算法, 其中大寫 S 指排序 sorted 的意思, 論文後面所提到的演算法如果使用到排序的 QR 分解, 則演算法前都會加上大寫 S 來表示。SQRD-M 演算法並非依照傳統 QR 分解的解碼順序依序來估測每根天線, 而是依照特定的解碼順序來做估測, 詳細的排序 QR 分解會於第二章說明。

SQRD-M 演算法只適合用於低階的調變, 對越高階的調變來說, M 值的大小對複雜度之影響就越明顯, 以調變使用 64-QAM 的系統為例子, 在演算法過程中, M 的值如果比原來多一就意味著此次被保留下來的部分天線之傳送訊號組合多了一組, 當估測下根天線時, 必須計算與比較價值函數 (cost function, 如式子 (13)) 的個數就多了六十四個, 複雜度會以六十四的倍數成長, 所以不適用於高階的調變系統。針對 SQRD-M 演算法的 M 值, 進一步的降低複雜度而衍生出適應性調整門檻來限制搜尋路徑個數的演算法, 又稱為以廣度為主做搜尋的排序演算法 (SBSIDE 演算法), 其作法與 SQRD-M 演算法類似, 都是根據特定解碼順序依序估測出每根天線的傳送訊號, 但演算法過程中並非每次都保留固定 M 組較可能發生的部分天線之傳送訊號組合, 而是在選取這些要保留的組合前, 必須先算出一個門檻 (threshold), 最後只保留達到此門檻要求的部分天線之傳送訊號組合, 而這些組合的總個數以 E 表示。因為此門檻是一個動態改變值, 所以被保留下來的訊號組合個數 E 也會跟著動態改變, 而我們限制 E 值必須小於或等於 N (N 為調變訊號集合大小的個數), 詳細說明於第三章; 所以, 在兩個演算法錯誤率都接近 ML 估測的情況下, SBSIDE 演算法的複雜度會比 SQRD-M 演算法低。

為了讓 SBSIDE 演算法更適合用於高傳輸速度的系統，所以在高階的調變上必須更進一步的降低複雜度；因此，本篇論文結合了 N-QAM 的多層架構搜尋特性及 SBSIDE 演算法而衍生出以廣度為主做搜尋的多層排序演算法，簡稱為 MSBSIDE 演算法。此演算法與 SQRD-M 演算法比較，除了上述針對 M 值做處理外，同時降低了 N 值的個數，以調變使用 64-QAM 的系統為例子，未使用多層架構搜尋時，估測過程中每進一層，則必須計算與比較價值函數的個數為 $E \times 64$ 個；使用多層架構搜尋時，估測過程中每進一層，則必須計算與比較價值函數的個數只剩 $E \times 4$ 個，唯一不同的地方在於要得到一根傳送天線的估測訊號必須分成 $\log_4 64 = 3$ 個階段來完成，如圖 4.2 的三層架構，以下章節會有詳細的說明。

1.2 論文架構

瞭解了本論文的研究背景及動機之後，第二章會詳細敘述 V-BLAST 系統架構的傳送端及接收端之動作原理，最後說明如何由 ML 估測的一次估測全部天線之全域搜尋變成以 QR 分解為基礎的一次估測一根天線之次序性解碼，並且由次序性解碼的概念衍生出排序的 QR 分解。第三章詳細說明如何從以排序 QR 分解為基礎的 M 演算法（SQRD-M 演算法）衍生出更低複雜度之以廣度為主做搜尋的排序演算法（SBSIDE 演算法）。第四章先詳細定義 N-QAM 調變訊號在座標平面上自然形成的多層架構，接著結合第三章的 SBSIDE 演算法與 N-QAM 的多層架構搜尋特性而衍生出以廣度為主做搜尋的多層排序演算法（MSBSIDE 演算法）。第五章則分別將第三章及第四章所提到的演算法應用於 V-BLAST 系統的接收端，並且利用 Matlab 軟體來模擬並分析各個演算法的錯誤率及複雜度。第六章為論文的總結。

二、系統模型

本論文的系統架構是建立在多根傳送天線與多根接收天線的 V-BLAST 無線通訊系統上，所以在 2.1 節先詳細敘述 V-BLAST 系統架構的傳送端及接收端之動作原理，並於 2.2 節說明當使用此系統架構時，接收端如何由 ML 估測的一次估測全部天線之全域搜尋變成以 QR 分解為基礎的一次估測一根天線之次序性解碼，並且由次序性解碼的概念衍生出排序的 QR 分解。

2.1 V-BLAST 系統架構

在多根傳送天線與多根接收天線的系統中，傳送端可以經由多根天線的傳輸模式來增加整體系統的分集程度 (diversity) 或多工程度 (multiplexing)。在此系統下增加分集程度的傳統方法為使用時空編碼 (Space-time code)，例如 Alamouti code，雖然增加系統的分集程度可以提升接收端在錯誤率的表現，且讓接收端能以簡單的數學運算來區分每根傳送天線的傳送訊號，但此種類型的編碼其傳送端下一時間必須再傳送正交於上一時間的傳送訊號，所以成功傳送一筆資料必須花費兩倍的時間，頻譜效率 (spectral efficiency) 並不高。而在此系統下增加多工程度的方法為使用分層時空編碼 (Layered space-time code)，此架構之所以被稱為分層時空碼是因為傳送端的每根天線間之編碼器及調變器都是獨立的，每一根天線自成一層，且傳送出去的訊號並不需要花費下一時間來傳送正交訊號，所以頻譜效率會較高，但也因為沒有下一時間的正交訊號來幫助接收端估測與解碼傳送訊號，所以使用分層時空編碼的架構時，如何設計接收端的估測演算法讓複雜度與估測錯誤率間取得較好的平衡為主要著重的議題。分層時空編碼除了可以增加系統的多工程度，各層天線的編碼器如果不是獨立的，而是幾根天線的訊號一起進行編碼再輸出給各自獨立的調變器的話，即可增加系統的分集程度，但相對的系統的多工程度就會降低，所以分集程度與多工程度間彼此是呈平衡關係。

本論文使用的系統為分層時空編碼架構中不加入編碼器與解碼器的系統，此種系統因為各層天線都是獨立運作，所以可以達到最高的多工程度，學者們將此架構特別稱為 Vertical Bell Laboratories Layered Space-Time，簡稱為 V-BLAST 系統，其傳送端架構如圖 2.1。原始訊號 (m) 先經過多路傳輸分流後 (S/P)，分成

N_T 個子分流訊號 (sub-stream)，這 N_T 個子分流訊號再經由各自獨立的調變器調變後 (Mod)，最後由 N_T 根傳送天線傳出。這 N_T 個傳送訊號定義為 $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_{N_T}]^T$ ($[\]^T$ 指轉置矩陣)，其中 \mathbf{x} 的元素都是調變訊號集合 $A = \{a_1, a_2, \dots, a_N\}$ 中的其中一個訊號。這些訊號經過通道 \mathbf{H} 的干擾後由 N_R 根接收天線接收 (N_R 必須大於或等於 N_T)， N_R 個接收訊號定義為 $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_{N_R}]^T$ ，而 \mathbf{y} 和 \mathbf{x} 的關係可以用下列式子表示：

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

其中 \mathbf{H} 是一個 $N_R \times N_T$ 的通道矩陣，且假設接收端已經知道通道狀況 (\mathbf{H})， h_{ij} 是 \mathbf{H} 矩陣的第 i 列、第 j 行的元素，也就是從第 j 根傳送天線到第 i 根接收天線間的通道增益，這些元素都是平均值 (mean) 等於零，變異數 (variance) 等於一的 independent circularly symmetric complex Gaussian random variable。 \mathbf{n} 是一個 $N_R \times 1$ 的雜訊向量， $\mathbf{n} = [n_1 \ n_2 \ \cdots \ n_{N_R}]^T$ ，且裡面的元素都是平均值等於零，變異數等於 σ^2 的 independent circularly symmetric complex Gaussian noise。當接收天線收到訊號後，收到的訊號 \mathbf{y} 會在接收端進行傳送訊號的估測與解碼，接收端架構如圖 2.2。 N_R 個接收到的訊號先經由傳送訊號估測 (detector) 估算出原始傳送的 N_T 個子分流訊號，再經由解調器 (Demod) 及解多路傳輸分流 (P/S) 得到估計的原始訊號 (\hat{m})。

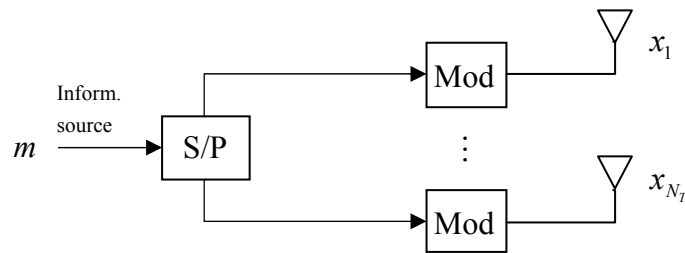


圖 2.1 V-BLAST 系統的傳送端架構

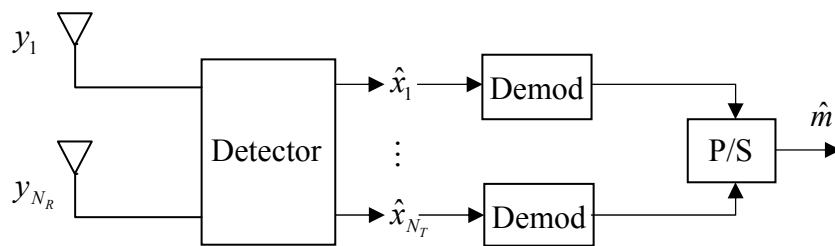


圖 2.2 V-BLAST 系統的接收端架構

2.2 排序的 QR 分解 (sorted QR decomposition)

本篇論文著重於圖 2.2 中估測部分的演算法設計。V-BLAST 架構下的估測，最簡單但複雜度最高的方法為 ML 估測，做法為從所有可能的 \mathbf{x} 訊號組合中找到發生機率最高的組合，此組合即是最佳估測解。而在白色高斯雜訊 (AWGN) 的系統模型下，計算機率值可以等同於計算傳送訊號組合經過通道的增益後與接收到的訊號 \mathbf{y} 之距離，最後再從這些距離中選取最小距離對應的傳送訊號組合即為發生機率最高的最佳估測解。計算距離的方法如下式：

$$distance = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \quad (2)$$

其中， \mathbf{x} 以所有可能的訊號組合代入。這種全域搜尋方法當天線數過多或使用較高階的調變時，計算距離的個數會呈指數遞增，對接收端來說複雜度太高。

QRD-M 演算法為了解決 ML 估測因全域搜尋所帶來的高複雜度，接收端在估測傳送訊號前，必須先對通道矩陣 \mathbf{H} 做 QR 分解，原始的 OR 分解如下式：

$$\mathbf{H} = \mathbf{Q} \times \mathbf{R} \quad (3)$$

$$\Rightarrow \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_T} \\ h_{21} & h_{22} & \cdots & h_{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_R1} & h_{N_R2} & \cdots & h_{N_R N_T} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N_T} \\ q_{21} & q_{22} & \cdots & q_{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N_R1} & q_{N_R2} & \cdots & q_{N_R N_T} \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N_T} \\ 0 & r_{22} & \cdots & r_{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{N_T N_T} \end{bmatrix}$$

其中 \mathbf{Q} 是一個 $N_R \times N_T$ 的正交矩陣 (orthogonal matrix)，正交矩陣的特性是它的行向量彼此是正交關係，且每個行向量的長度都為一，因此可以用 $\mathbf{Q}^H \times \mathbf{Q} = \mathbf{I}$ 來表示此特性 ($[\]^H$ 指共軛轉置矩陣， \mathbf{I} 指單位矩陣)； \mathbf{R} 是一個 $N_T \times N_T$ 的上三角矩陣 (upper triangular matrix)。而將 QR 分解應用於解碼端是利用了 \mathbf{Q} 矩陣的正交特性，讓 \mathbf{Q} 的共軛轉置矩陣乘上原始雜訊向量後的統計特性維持不變，依然是白色高斯雜訊的統計特性，以利接收端估測傳送訊號，其用法如下式：

$$\tilde{\mathbf{y}} = \mathbf{Q}^H \times \mathbf{y} = \mathbf{R} \times \mathbf{x} + \tilde{\mathbf{n}} \quad (4)$$

$$\Rightarrow \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N_T} \\ 0 & r_{22} & \cdots & r_{2N_T} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N_T} \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \tilde{n}_2 \\ \vdots \\ \tilde{n}_{N_T} \end{bmatrix}$$

其中， $\tilde{\mathbf{n}} = \mathbf{Q}^H \times \mathbf{n}$ 。由於 \mathbf{Q} 是一個正交矩陣，所以 $\tilde{\mathbf{n}}$ 的統計特性仍然是平均值等於零，變異數等於 σ^2 的 independent circularly symmetric complex Gaussian noise，證明如(5)、(6)式。

$$\because E\{\mathbf{n} \times \mathbf{n}^H\} = \sigma^2 \times \mathbf{I} \quad (5)$$

$$\therefore E\{\tilde{\mathbf{n}} \times \tilde{\mathbf{n}}^H\} = E\{\mathbf{Q}^H \times \mathbf{n} \times \mathbf{n}^H \times \mathbf{Q}\} = \sigma^2 \times \mathbf{I} \times E\{\mathbf{Q}^H \times \mathbf{Q}\} = \sigma^2 \times \mathbf{I} \quad (6)$$

由式子(4)的矩陣架構可知原始 QR 分解自然的讓解碼順序由第 N_T 根天線解回第一根天線。此原因為第 N_T 列的變數最少，所以干擾也最少，較容易進行估測，估測的錯誤率也會較低，所以依次序從第 N_T 根天線解回第一根天線發生錯誤延續 (error propagation) 的機率也會較低。又因為 $\tilde{\mathbf{n}}$ 矩陣裡每個元素彼此都是獨立的，所以接收端在計算價值函數時，可以分開並累積來計算，如下式：

$$\|\tilde{\mathbf{y}} - \mathbf{R} \times \mathbf{x}\|^2 = \sum_{i=1}^{N_T} \left| \tilde{y}_i - \sum_{j=i}^{N_T} r_{ij} \times x_j \right|^2 \quad (7)$$

所以，接收端估測到第 k 根天線的價值函數就是從第 N_T 根天線累積到第 k 根天線的價值函數。而排序 QR 分解 (sorted QR decomposition) 的概念產生是從訊雜比 (SNR) 及解碼錯誤延續的角度而來，以第 x_{N_T} 列為例， $\tilde{y}_{N_T} = r_{N_T N_T} \times x_{N_T} + \tilde{n}_{N_T}$ ，當 $r_{N_T N_T}$ 越大，訊雜比會越大，如(8)式，所以，估算出來的 \hat{x}_{N_T} 錯誤率也會越低。

$$SNR_{N_T} = \frac{|r_{N_T N_T}|^2 \times P}{\sigma^2} \quad (8)$$

其中， P 為每根天線傳送訊號的平均能量。除此之外，其它根天線的估測都會用到 \hat{x}_{N_T} 的值，所以 \hat{x}_{N_T} 的正確性高低也決定了解碼錯誤延續是否發生，因此我們會希望 \mathbf{R} 的對角元素（diagonal）從第一根解碼之天線那列的值到最後一根解碼之天線那列的值由大到小排列，讓整體估測的正確性更高，而有排序 QR 分解的概念產生，詳細的排序 QR 分解演算法如表 2.1。

初始化.

$$order = \phi$$

步驟一.

計算 \mathbf{H} 中除去 $order$ 紀錄的行後其他行向量的長度，取最小長度那行當成此次執行 QR 分解的行（第 k 行），並且把 k 併到 $order$ 裡

$$norm_i = \|h_i\| \quad (h_i \text{ 為 } \mathbf{H} \text{ 的第 } i \text{ 行}), i=1 \cdots N_T \text{ 且 } i \notin order$$

$$k = \arg \min_{i=1 \cdots N_T, i \notin order} (norm_i)$$

步驟二.

$$r_{kk} = norm_k, \text{ 並得到 } \mathbf{Q} \text{ 的第 } k \text{ 行}, q_k = h_k / r_{kk}$$

步驟三.

$j=1 \cdots N_T$ 且 $j \notin order$ ，計算 h_j 在 q_k 基底的投影量，即 r_{kj} ，並從 h_j 裡把 h_j 在 q_k 基底的貢獻度扣掉

$$r_{kj} = q_k \cdot h_j \quad (\text{內積即投影量})$$

$$h_j = h_j - r_{kj} \times q_k$$

步驟四.

如果 $order$ 包含 $1 \cdots N_T$ 所有數字，則結束此演算法，否則回到步驟一.

表 2.1 排序的 QR 分解演算法

排序的 QR 分解可以藉由改變 Gram-Schmidt 演算法來達到。Gram-Schmidt 演算法是用來找向量基底的演算法，而向量基底本身就是長度為一且彼此是正交的向量，符合 \mathbf{Q} 矩陣每行的特性。在排序的 QR 分解演算法中，會先決定要從 \mathbf{H} 矩陣哪一行開始分解，而判斷的方法為 \mathbf{H} 矩陣行向量的長度，從最短長度的行向量開始分解；假設從 \mathbf{H} 矩陣的第 k 行開始分解則得到 \mathbf{Q} 矩陣的第 k 行與 \mathbf{R} 矩陣

的第 k 列，並更新 \mathbf{H} 矩陣未進行分解的行；接著再次決定要分解 \mathbf{H} 矩陣的哪一行，方法與前一步驟相同，直到所有的基底 \mathbf{Q} 矩陣與 \mathbf{R} 矩陣都找到才停止。以一個 3×3 的 \mathbf{H} 矩陣為例子：令一個空矩陣 $order$ ，並算出 \mathbf{H} 矩陣每一行的長度，即 $norm_i = \|h_i\|$ ，其中 $i = 1 \dots 3$ ；假設 \mathbf{H} 矩陣的第二行長度 $norm_2$ 最小，則把 2 併入 $order$ 中，即 $order = \{2\}$ ，並將 \mathbf{R} 矩陣的元素 r_{22} 設成 $norm_2$ ，且計算 \mathbf{Q} 矩陣的第二行 $q_2 = h_2 / r_{22}$ ；接著算出 \mathbf{H} 矩陣的行 h_1 和 h_3 在 \mathbf{Q} 矩陣的第二行 q_2 基底上的投影量，此投影量分別為 \mathbf{R} 矩陣的元素 r_{21} 和 r_{23} ，並且分別從 h_1 和 h_3 裡把 h_1 和 h_3 在 q_2 基底的貢獻度扣掉，也就是 $h_k = h_k - r_{2k} \times q_2$ ，其中 $k = 1, 3$ ；此步驟的結果如(9)式：

$$\begin{array}{c} \text{更新} \\ \swarrow \\ [h_1 \quad h_2 \quad h_3] = [\quad q_2 \quad] \times \begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix} \end{array} \quad (9)$$

由上一步驟找到 \mathbf{Q} 矩陣的第二行與 \mathbf{R} 矩陣的第二列後，因為 \mathbf{H} 矩陣裡的行已被更新過，所以必須再次計算 \mathbf{H} 矩陣裡除了第二行之外每一行的長度，即 $norm_i = \|h_i\|$ ，其中 $i = 1, 3$ ；假設 \mathbf{H} 矩陣的第三行長度 $norm_3$ 最小，則把 3 併入 $order$ 中，即 $order = \{2, 3\}$ ，並將 \mathbf{R} 矩陣的元素 r_{33} 設成 $norm_3$ ，且計算 \mathbf{Q} 矩陣的第三行 $q_3 = h_3 / r_{33}$ ；接著算出 \mathbf{H} 矩陣的行 h_1 在 \mathbf{Q} 矩陣的第三行 q_3 基底上的投影量，此投影量為 \mathbf{R} 矩陣的元素 r_{31} ，並且從 h_1 裡把 h_1 在 q_3 基底的貢獻度扣掉，也就是 $h_1 = h_1 - r_{31} \times q_3$ ；此步驟的結果如(10)式：

$$\begin{array}{c} \text{更新} \\ \swarrow \\ [h_1 \quad h_2 \quad h_3] = [\quad q_2 \quad q_3] \times \begin{bmatrix} r_{21} & r_{22} & r_{23} \\ r_{31} & & r_{33} \end{bmatrix} \end{array} \quad (10)$$

最後只剩下 \mathbf{Q} 矩陣的第一行與 \mathbf{R} 矩陣的第一列還未找到，同上個步驟只需計算 \mathbf{H} 矩陣第一行的長度，即 $norm_1 = \|h_1\|$ ，且把 1 併入 $order$ 中，即 $order = \{2, 3, 1\}$ ，並將 \mathbf{R} 矩陣的元素 r_{11} 設成 $norm_1$ ，且計算 \mathbf{Q} 矩陣的第一行 $q_1 = h_1 / r_{11}$ ；此例子最後結果如(11)式：

$$[h_1 \quad h_2 \quad h_3] = [q_1 \quad q_2 \quad q_3] \times \begin{bmatrix} r_{11} & & \\ r_{21} & r_{22} & r_{23} \\ r_{31} & & r_{33} \end{bmatrix} \quad (11)$$

為了驗證原始 QR 分解與排序 QR 分解在錯誤率上的差異，使用 4×4 的 V-BLAST 系統，調變為 4-QAM，且分別將原始 QR 分解與排序 QR 分解所找到的 Q 矩陣與 R 矩陣代入式子(4)中，而後端的估測演算法為決定迴授等化估測演算法（decision feedback equalization detection，簡稱為 DFE 估測演算法，如表 3.1，即 SQRD-M 演算法將 M 設定成一的情況），結果如圖 2.3。由模擬結果得知，執行演算法前使用排序的 QR 分解在同樣的錯誤率 10^{-3} 下，效能比使用原始 QR 分解大約好 3dB。所以，使用排序的 QR 分解能讓接收端估測出來的結果更準確。

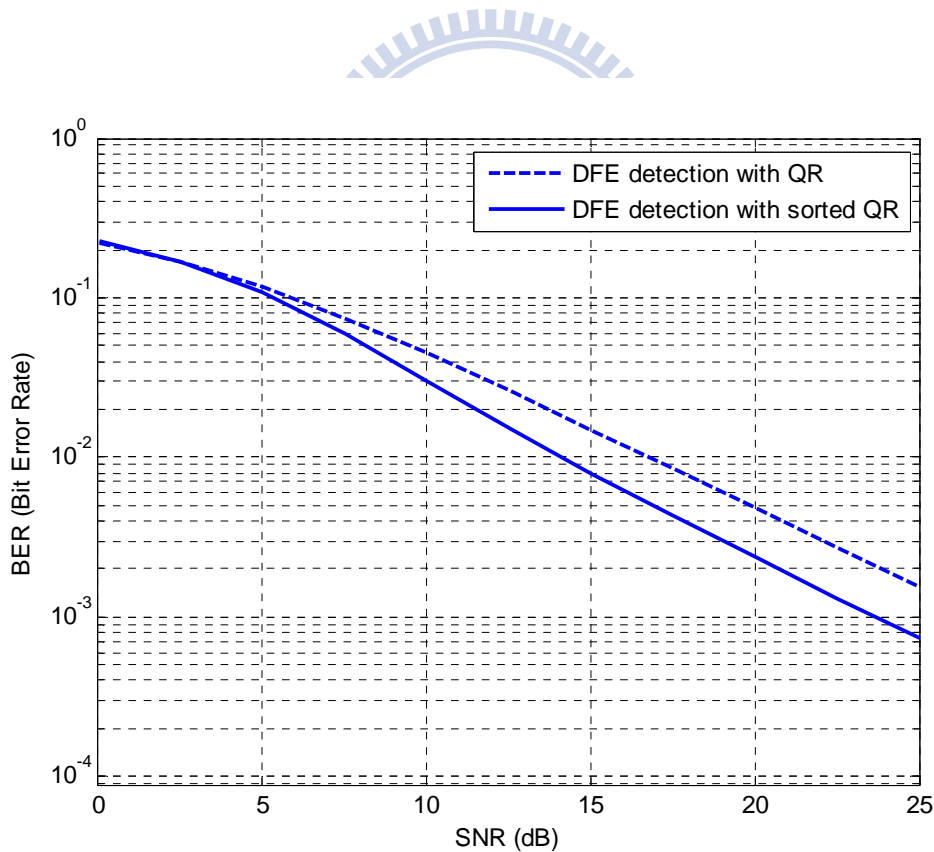
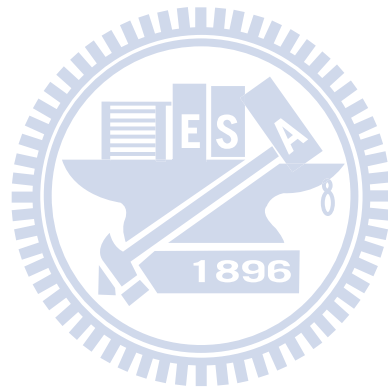


圖 2.3 在 4×4 的 V-BLAST 系統、調變為 4-QAM 下使用原始 QR 分解與排序 QR 分解於 DFE 估測演算法的錯誤率比較

排序的 QR 分解所得到的 \mathbf{R} 矩陣並不一定是上三角矩陣，但還是符合某列元素只有一個，某列元素只有兩個...某列元素有 N_T 個，而估測順序即是 $order(i)$, $i = N_T \cdots 1$ ，這樣就可以從干擾及變數最少的天線開始估測。排序的 QR 分解並不能保證 \mathbf{R} 矩陣的對角元素依 $order(i)$, $i = N_T \cdots 1$ 的順序由大到小排列，這是因為 \mathbf{H} 矩陣先天性的限制， \mathbf{H} 矩陣的元素是隨機產生的，有些情況下可以達成要求，有些情況下則不可能，但排序的 QR 分解提供一種方法，讓可以達成要求的情況一定能百分之百達成，不可能的情形盡量由大到小排列（也就是說在不可能的情況下並不一定是最佳的分解順序）。從複雜度的角度來看，排序的 QR 分解可以在犧牲一些複雜度的情況下得到很好的效能回報，且錯誤率接近高複雜度最佳分解順序的 QR 分解。



三、以廣度為主做搜尋的排序演算法

在第二章介紹完排序 QR 分解及它的好處後，本章著重於以廣度為主做搜尋的排序演算法（簡稱為 SBSIDE 演算法）如何適應性地調整門檻來限制搜尋路徑的個數，而 SBSIDE 演算法是結合了排序 QR 分解與 BSIDE 演算法而來。SBSIDE 演算法是從以排序 QR 分解為基礎的 M 演算法（簡稱為 SQRD-M 演算法）衍生而來，這兩種演算法都是以廣度為主做搜尋的演算法，它們之所以被稱為以廣度為主做搜尋的原因是在演算法過程中，必須依照排序 QR 分解給定的解碼順序逐步估測每根天線，當最後一根天線的傳送訊號被估測出來後也代表演算法結束；以深度為主做搜尋的做法是先找到一組估計解，此組估計解已經包含了每根天線的估測訊號，再根據演算法提供的方法來更新每根天線的估測訊號，當此組估計解的正確性已經達到預期所設的門檻也代表演算法結束。SBSIDE 演算法為了達到降低複雜度的目的，不採用 SQRD-M 演算法在過程中每次都保留固定 M 組較可能發生的部分天線之傳送訊號組合，因為這固定的 M 組訊號組合中，某些組合發生的機率比起那些發生機率高的組合低很多，所以並不盲從的每次都保留固定 M 組，而是設定一個估計門檻來篩選要保留下來的訊號組合。本章將分成兩小節，3.1 節詳細介紹 SQRD-M 演算法的動作原理及其特性，3.2 節根據 3.1 節所介紹的 SQRD-M 演算法來衍生出 SBSIDE 演算法，詳細說明如下。

3.1 以排序 QR 分解為基礎的 M 演算法（SQRD-M 演算法）

圖 3.1 為 SQRD-M 演算法的流程圖。SQRD-M 演算法在進行訊號估測前必須先執行排序的 QR 分解，並將計算出的 \mathbf{Q} 矩陣及 \mathbf{R} 矩陣代入(4)式，並以(4)式為基礎來做次序性的訊號估測。為了方便說明，將排序的 QR 分解所找到之解碼順序設成從第 N_T 根天線估測到第一根天線，即 $order = \{1, 2, 3 \dots N_T\}$ ， \mathbf{R} 是一個標準倒三角形矩陣。執行演算法會遇到兩種情況，第一種情況是估計第 N_T 根天線的傳送訊號。而篩選是比較機率值的大小，也就是直接將傳送訊號集合 A 所有的訊號代入(12)式中的 x_{N_T} 求出這 N 個延伸路徑的價值函數。接著從這 N 個價值函數 $\psi(x_{N_T})$ 中保留前 M 個最小的價值函數，其餘則刪除，並記錄這 M 個價值函數 $\psi_m(x_{N_T})$ 及所對應的部分天線之傳送訊號組合 $\mathbf{\theta}_{N_T}^{(m)} = [x_{N_T}^{(m)}]^T$ ，其中 $m = 1 \dots M$ ，以便演算法執行下一步驟。SQRD-M 演算法的 M 值是事先給定的且為固定值， M 值

越大被保留下來的價值函數會越多，對估測下根天線的運算量影響很大，但相對的也會影響估測的準確度。

$$\begin{aligned}
 & P(x_{N_T} | \tilde{y}_{N_T}) \\
 & \propto P(\tilde{y}_{N_T} | x_{N_T}) \times P(x_{N_T}) \\
 & \approx \exp\left(-\frac{1}{\sigma^2} |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T}|^2\right) \times P(x_{N_T}) \\
 \Rightarrow & -\log P(x_{N_T} | \tilde{y}_{N_T}) \\
 & \approx \frac{1}{\sigma^2} |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T}|^2 - \log P(x_{N_T}) \\
 \Rightarrow & \psi(x_{N_T}) \equiv |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T}|^2
 \end{aligned} \tag{12}$$

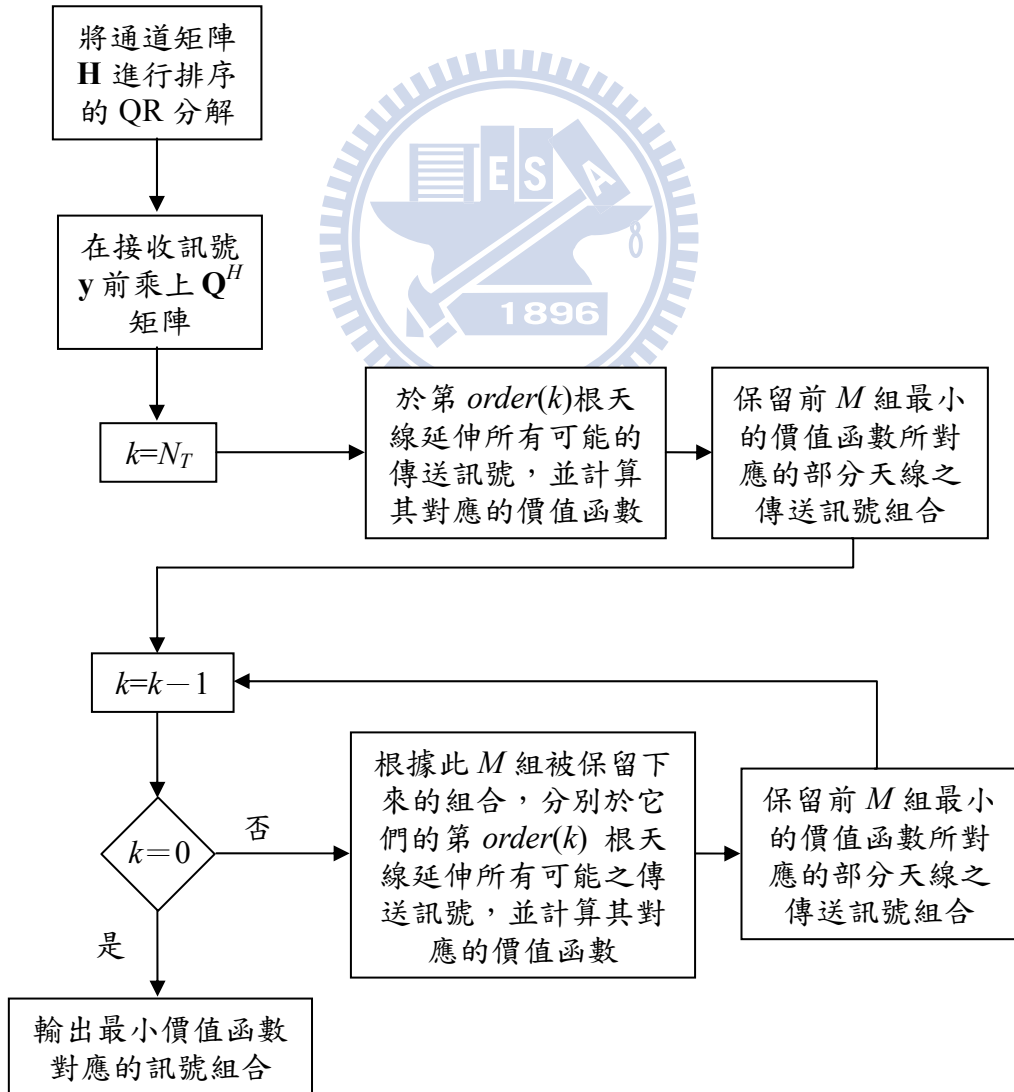


圖 3.1 SQRD-M 演算法的流程圖

第二種情況是開始要估計第 j 根天線的傳送訊號，假設此時已經有第 $j+1 \dots N_T$ 根天線估計出來的 M 組部分天線之傳送訊號組合 $\mathbf{\theta}_{j+1}^{(m)} = [x_{j+1}^{(m)} x_{j+2}^{(m)} \dots x_{N_T}^{(m)}]^T$ 及對應的價值函數 $\psi_m(x_{j+1})$ ，其中 $m = 1 \dots M$ 。估計第 j 根天線的傳送訊號要根據這 M 組部分天線之傳送訊號組合各別去延伸 N 條路徑，也就是將傳送訊號集合 A 所有的訊號代入(13)式中的 x_j 求出這 $M \times N$ 個延伸路徑的價值函數。接著從這 $M \times N$ 個價值函數 $\psi(x_j)$ 中保留前 M 個最小的價值函數，其餘則刪除，並記錄這 M 個價值函數 $\psi_m(x_j)$ 及對應的部分天線之傳送訊號組合 $\mathbf{\theta}_j^{(m)} = [x_j^{(m)} x_{j+1}^{(m)} \dots x_{N_T}^{(m)}]^T$ ，其中 $m = 1 \dots M$ ，以便演算法執行下一步驟。

$$\begin{aligned}
& P(x_j, x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)} | \tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
& \propto P(\tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_j, x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)}) P(x_j) \prod_{k=j+1}^{N_T} P(x_k^{(m)}) \\
& = P(\tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_j, x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)}) \prod_{k=j+1}^{N_T} P(x_k^{(m)}) \\
& \quad \times P(\tilde{y}_j | x_j, x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)}) P(x_j) \\
& \approx P(\tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)}) \prod_{k=j+1}^{N_T} P(x_k^{(m)}) \\
& \quad \times \exp\left(-\frac{1}{\sigma^2} \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_k^{(m)} - r_{jj} x_j \right|^2\right) P(x_j) \\
& \Rightarrow -\log P(x_j, x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)} | \tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
& \quad \approx -\log P(x_{j+1}^{(m)}, \dots, x_{N_T}^{(m)} | \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
& \quad \quad + \frac{1}{\sigma^2} \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_k^{(m)} - r_{jj} x_j \right|^2 - \log P(x_j) \\
& \Rightarrow \psi(x_j) \equiv \psi_m(x_{j+1}) + \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_k^{(m)} - r_{jj} x_j \right|^2, m = 1 \dots M \tag{13}
\end{aligned}$$

此演算法會一直執行到部分天線之傳送訊號組合 $\mathbf{\theta}$ 包含所有天線的估計訊號才停止，而此時最小價值函數對應的訊號組合 $\mathbf{\theta}^{(1)}$ 即為使用此演算法得到的最佳估計解 $\hat{\mathbf{x}}$ 。以下用一個簡單的例子來說明演算法步驟：系統為三根傳送天線及三根接收天線的 V-BLAST 系統，調變使用 4-QAM，SQRD-M 演算法的 M 值設為

二，且假設排序的 QR 分解所找到之解碼順序是從第三根天線估測回第一根天線。如圖 3.2，虛線是延伸的路徑，實線是被保留下來的前兩個最小價值函數所對應之路徑，圓形點是可能的傳送訊號，節點旁的數字為路徑延伸到此節點所累積的價值函數。估計第三根天線的傳送訊號時，直接將 4-QAM 的四個調變訊號代入(12)式求出四個價值函數，並且保留兩個最小的價值函數所對應之節點，此例子被保留下來之節點所對應之價值函數分別為 2.3 及 3.0；估計第二根天線的傳送訊號時，必須根據上一步驟保留下來之兩條路徑，分別去延伸四個可能的傳送訊號，即將 4-QAM 的四個調變訊號代入(13)式，求出八個價值函數，並且保留兩個最小的價值函數所對應之節點，此例子被保留下來之節點所對應之價值函數分別為 2.4 及 3.4；估計第一根天線的傳送訊號也是一樣的做法，必須根據上一步驟保留下來之兩條路徑，分別去延伸四個可能的傳送訊號，並且保留兩個最小的價值函數所對應之節點，此例子被保留下來之節點所對應之價值函數分別為 2.5 及 2.7；最後再從這兩個被保留下來之路徑中選取最小價值函數所對應之路徑即為使用 SQRD-M 演算法得到的最佳估計解。

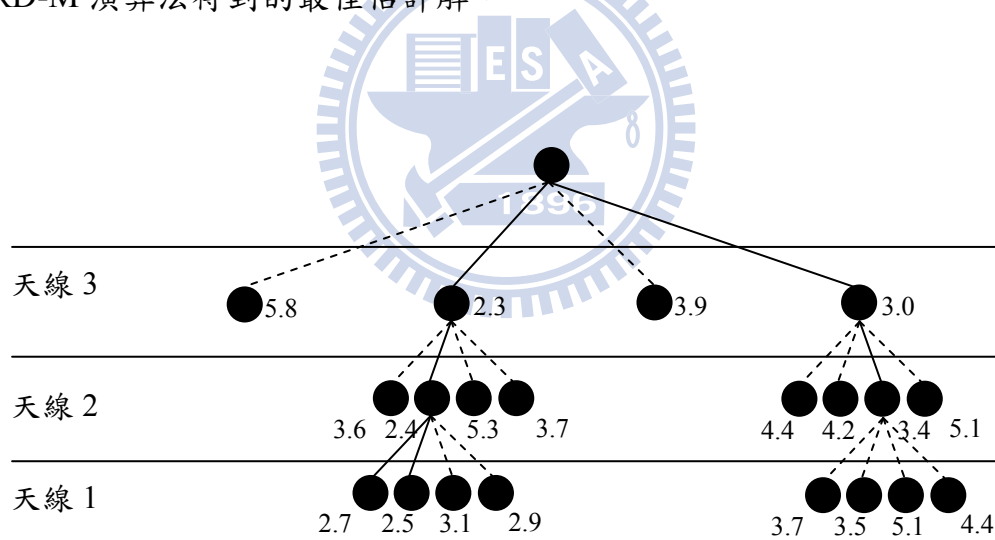


圖 3.2 $N_T=3$ ， $N_R=3$ ，運用 SQRD-M 演算法做估測的樹狀圖示

從模擬結果可以得知 SQRD-M 演算法的 M 值最大只需到 N 即可包含最佳路徑，且讓錯誤率趨近於 ML 估計曲線，其中 N 為傳送訊號集合 A 裡元素的總個數； M 值再大只會增加複雜度，效能改善的程度並不明顯，所以我們限制 $M \leq N$ 。而 M 值最大只需等於 N 的原因如下：SQRD-M 演算法是一次估測一根天線的次序

性解碼，所以第一根天線的估測訊號之正確性會影響其它根天線的估測訊號之正確性，即錯誤延續，所以如果被保留下來的部分天線之傳送訊號組合個數 M 等於調變訊號集合大小 N 表示估測第一根天線時，此根天線所有可能的傳送訊號都被保留下來，才不會發生正確的傳送訊號在估測第一根天線時就被刪除。圖 3.3 和圖 3.4 分別為調變使用 4-QAM 及 16-QAM 於 4×4 的 V-BLAST 系統下，SQRD-M 演算法與 modified SQRD-M 演算法的錯誤率比較圖。由圖可知，SQRD-M 演算法在 M 小於 N 與 M 等於 N 之間有很明顯的錯誤率落差；而 modified SQRD-M 演算法在 M 小於 N 與 M 等於 N 之間的錯誤率差距較不明顯，尤其在更高階的調變下差距更小，其原因是因為 modified SQRD-M 演算法在估測第一根天線時，強迫讓此根天線所有可能的傳送訊號都保留下來，參數 M 的限制只存在其它根天線的估測，所以在相同的 M 值下，經過兩種演算法的錯誤率比較後得到結論如下：第一根天線的估測訊號之正確性是影響 SQRD-M 演算法的效能最主要之因素，尤其在越高階調變下影響越明顯。所以，通常我們限制 SQRD-M 演算法的 M 值最大只需到 N 即可。

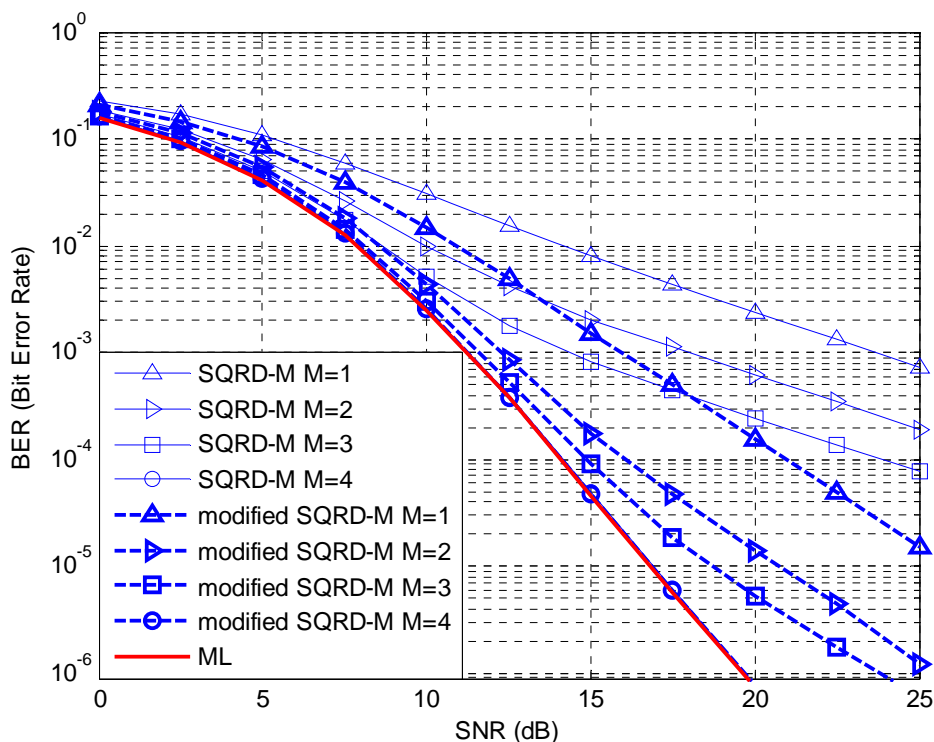


圖 3.3 SQRD-M 演算法與 modified SQRD-M 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較

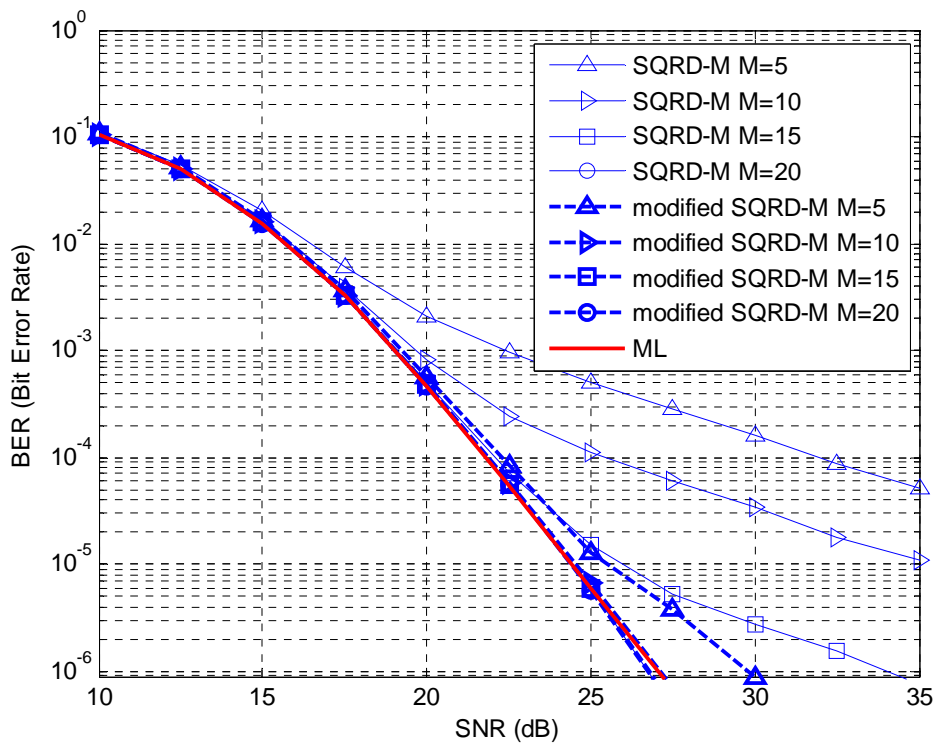


圖 3.4 SQRD-M 演算法與 modified SQRD-M 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的錯誤率比較

3.2 以廣度為主做搜尋的排序演算法 (SBSIDE 演算法)

```

 $\hat{\mathbf{s}} = \mathbf{0}$ 
for  $i = N_T \cdots 1$ 
  if  $i < N_T$ 
    for  $j = i+1 \cdots N_T$ 
       $\tilde{y}_{order(i)} = \tilde{y}_{order(i)} - r_{order(i)order(j)} \hat{s}_{order(j)}$ 
    end
  end
   $\hat{s}_{order(i)} = Q(\tilde{y}_{order(i)} / r_{order(i)order(i)})$ 
end

```

表 3.1 決定迴授等化估測演算法 (DFE 估測演算法)

圖 3.5 為 SBSIDE 演算法的流程圖。SBSIDE 演算法與 SQRD-M 演算法相同，在進行訊號估測前必須先執行排序的 QR 分解，為了方便說明，將排序 QR 分解所找到之解碼順序設成從第 N_T 根天線估測到第一根天線，即 $order=\{1,2,3\cdots N_T\}$ ， \mathbf{R} 是一個標準倒三角形矩陣。而執行演算法前，必須先根據排序的 QR 分解找到之解碼順序加上決定迴授等化估測演算法 (DFE 估測演算法) 估算出一個初步估計解 $\hat{\mathbf{s}}=[\hat{s}_1 \hat{s}_2 \cdots \hat{s}_{N_T}]^T$ ，如表 3.1，其中 $Q(\cdot)$ 指量化，並且算出初步估計門檻 $d=\|\tilde{\mathbf{y}}-\mathbf{R}\hat{\mathbf{s}}\|^2$ 。而使用排序的 QR 分解可以讓初步估計解 $\hat{\mathbf{s}}$ 更準確， $\hat{\mathbf{s}}$ 越準確也會讓初步估計門檻 d 越小，此時篩選的依據就更嚴苛，被保留下來的部分天線之傳送訊號組合就越少，反應在複雜度上也較低。

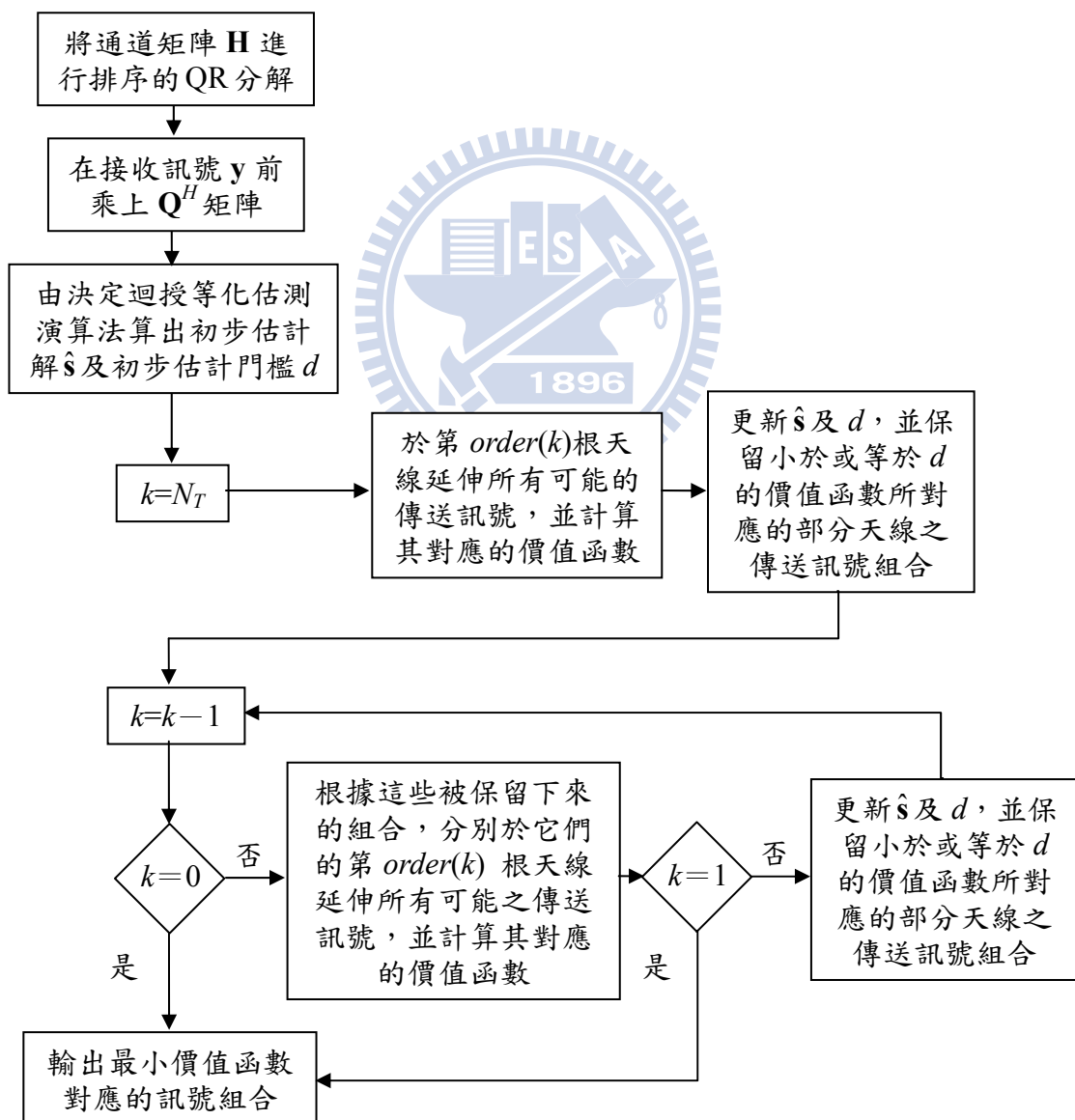


圖 3.5 SBSIDE 演算法的流程圖

執行演算法會遇到兩種情況，第一種情況是估計第 N_T 根天線的傳送訊號。而篩選是比較機率值的大小，與 SQRD-M 演算法計算價值函數的方法相同，不同的地方在於保留價值函數的機制，所以同樣直接將傳送訊號集合 A 所有的訊號代入(12)式中的 x_{N_T} ，求出這 N 個延伸路徑的價值函數。接著從這 N 個價值函數 $\psi(x_{N_T})$ 中取出最小的價值函數（以 $\psi_1(x_{N_T})$ 表示）所對應之部分天線的傳送訊號組合 $\boldsymbol{\theta}_{N_T}^{(1)} = [x_{N_T}^{(1)}]^T$ ，如果 $\boldsymbol{\theta}_{N_T}^{(1)}$ 不等於初步估計解中的 \hat{s}_{N_T} ，則令一個向量 \mathbf{g} ，其中 1 到 N_T-1 位置的值為初步估計解 $\hat{\mathbf{s}}$ 中 1 到 N_T-1 位置的值，而 \mathbf{g} 中 N_T 位置的值為 $\boldsymbol{\theta}_{N_T}^{(1)}$ ，即 $\mathbf{g} = [\hat{s}_1 \cdots \hat{s}_{N_T-1} \boldsymbol{\theta}_{N_T}^{(1)}]^T$ ，並求得 $\hat{d} = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{g}\|^2$ ；如果 \hat{d} 小於初步估計門檻 d ，則讓 d 等於 \hat{d} ，且更新初步估計解 $\hat{\mathbf{s}} = \mathbf{g}$ 。最後從這 N 個價值函數 $\psi(x_{N_T})$ 中保留小於或等於初步估計門檻 d 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_{N_T})$ 及所對應的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{N_T}^{(m)} = [x_{N_T}^{(m)}]^T$ ，其中 $m=1 \cdots E$ ，以便演算法執行下一步驟。

第二種情況是開始要估計第 j 根天線的傳送訊號，假設此時已經有第 $j+1 \cdots N_T$ 根天線估計出來的 E 組部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j+1}^{(m)} = [x_{j+1}^{(m)} x_{j+2}^{(m)} \cdots x_{N_T}^{(m)}]^T$ 及對應的價值函數 $\psi_m(x_{j+1})$ ，其中 $m=1 \cdots E$ ，且已知更新過後的初步估計門檻 d 及初步估計解 $\hat{\mathbf{s}}$ 。估計第 j 根天線的傳送訊號要根據這 E 組傳送訊號組合各別去延伸 N 條路徑，也就是將傳送訊號集合 A 所有的訊號代入(13)式中的 x_j ，且讓(13)式中的 $m=1 \cdots E$ ，來求出這 $E \times N$ 個延伸路徑的價值函數。接著從這 $E \times N$ 個價值函數 $\psi(x_j)$ 中取出最小的價值函數（以 $\psi_1(x_j)$ 表示）所對應之部分天線的傳送訊號組合 $\boldsymbol{\theta}_j^{(1)} = [x_j^{(1)} x_{j+1}^{(1)} \cdots x_{N_T}^{(1)}]^T$ ，如果 $\boldsymbol{\theta}_j^{(1)}$ 矩陣裡的值不等於 $\hat{\mathbf{s}}(j:N_T)$ ，即初步估計解中 j 到 N_T 位置的值，則令一個向量 \mathbf{g} ，其中 1 到 $j-1$ 位置的值為初步估計解 $\hat{\mathbf{s}}$ 中 1 到 $j-1$ 位置的值，而 \mathbf{g} 中 j 到 N_T 位置的值為 $\boldsymbol{\theta}_j^{(1)}$ 矩陣對應的值，即 $\mathbf{g} = [\hat{s}_1 \cdots \hat{s}_{j-1} \boldsymbol{\theta}_j^{(1)}]^T$ ，並求得 $\hat{d} = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{g}\|^2$ ；如果 \hat{d} 小於初步估計門檻 d ，則讓 d 等於 \hat{d} ，且更新初步估計解 $\hat{\mathbf{s}} = \mathbf{g}$ 。最後從這 $E \times N$ 個價值函數 $\psi(x_j)$ 中保留小於或等於初步估計門檻 d 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_j)$ 及對應的部分天線之傳送訊號組合 $\boldsymbol{\theta}_j^{(m)} = [x_j^{(m)} x_{j+1}^{(m)} \cdots x_{N_T}^{(m)}]^T$ ，其中 $m=1 \cdots E$ ，以便演算法執行下一步驟。

此演算法會一直執行到部分天線之傳送訊號組合 $\boldsymbol{\theta}$ 包含所有天線的估計訊號才停止，而此時最小價值函數對應的訊號組合 $\boldsymbol{\theta}^{(1)}$ 即為使用 SBSIDE 演算法得到

的最佳估計解 $\hat{\mathbf{x}}$ 。同樣以一個簡單的例子來說明演算法步驟：系統為三根傳送天線及三根接收天線的 V-BLAST 系統，調變使用 4-QAM，並限制 SBSIDE 演算法過程中經過初步估計門檻篩選後被保留下來的一部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 4$ ，且假設排序的 QR 分解所找到之解碼順序是從第三根天線估測回第一根天線。如圖 3.6，虛線是延伸的路徑，實線是經過初步估計門檻的篩選後而被保留下來之路徑，圓形點是可能的傳送訊號，節點旁的數字為路徑延伸到此節點所累積的價值函數。估計第三根天線的傳送訊號時，直接將 4-QAM 的四個調變訊號代入(12)式求出四個價值函數，假設此時更新過後的初步估計門檻 d 之值為 3.8，則保留小於或等於此初步估計門檻 d 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 2.3 及 3.0；估計第二根天線的傳送訊號時，必須根據上一步驟保留下來的兩條路徑，分別去延伸四個可能的傳送訊號，即將 4-QAM 的四個調變訊號代入(13)式求出八個價值函數，假設此時更新過後的初步估計門檻 d 之值為 3.3，則保留小於或等於此初步估計門檻 d 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 2.4；估計第一根天線的傳送訊號也是一樣的做法，必須根據上一步驟保留下來的一條路徑，去延伸四個可能的傳送訊號，即將 4-QAM 的四個調變訊號代入(13)式求出四個價值函數，最後再從這四個價值函數選取最小價值函數所對應的路徑即為使用 SBSIDE 演算法得到的最佳估計解。

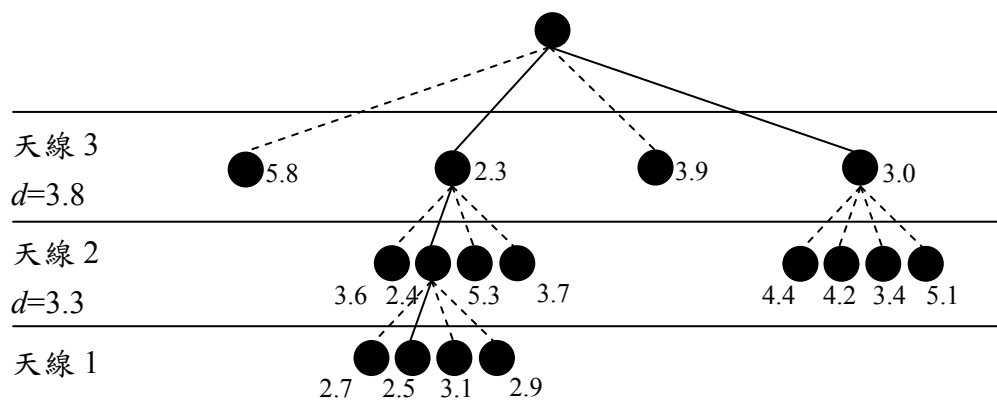


圖 3.6 $N_T = 3$, $N_R = 3$ ，運用 SBSIDE 演算法做估測的樹狀圖示

執行演算法過程中，初步估計解 \hat{s} 會不斷的更新，且越來越準確，也代表初步估計門檻 d 越來越小，但由第(13)式可知價值函數是不斷累加而得到，所以每估測完一根天線後價值函數都會增加；由以上結果得知，當接收端使用 SBSIDE 演算法估測越後面順序之天線的傳送訊號時，其所算出的價值函數經過初步估計門檻 d 的篩選後被保留下來之個數會越來越少，反應在複雜度上也會越低。

由上述演算法的推導過程可知被保留下來的部分天線之傳送訊號組合個數 E 並非定值，此論文中，我們限制 $1 \leq E \leq N$ ，這兩個極端值 1 和 N 得到的方式如下：SBSIDE 演算法在篩選 $E \times N$ 個價值函數時，會根據一個初步估計門檻 d 保留小於或等於此門檻的價值函數，而這些被保留下來的價值函數之個數就是解下一根天線時必須延伸路徑的節點個數 E 。有些情況下 E 值會比 N 還要大，但是從 3.1 節的 SQRD-M 演算法中得到一個結果， M 值最大只需到 N 即可包含最佳路徑，且讓錯誤率趨近於 ML 估測的曲線， M 值再大只會增加複雜度，效能改善的程度並不明顯，所以我們限制 $E \leq N$ ，才不會讓接收端在某些情況下複雜度變很高。另外，在高訊雜比時 E 值有很高的機率會等於一，也就是解下一根天線時需要延伸路徑的節點只有最小價值函數的節點，所以我們也限制 $1 \leq E$ 。

合（圓形點）為第一層訊號， $l=1$ ；十六個正方形點可以視為假想的 16-QAM 訊號點，這十六個正方形點都是在其周圍四個第一層訊號的平均值，並且把這十六個訊號點歸類為第二層訊號， $l=2$ ；四個星形點可以視為假想的 4-QAM 訊號點，這四個星形點都是在其周圍四個第二層假想訊號的平均值，並且把這四個訊號點歸類為第三層訊號， $l=3$ 。從圖 4.1 中，我們可以把這些點依照象限位置來做分類，例如位於第二象限的所有原始訊號與假想訊號都可以定義出一個樹狀圖，如圖 4.2，而有了樹狀圖就會衍生出親代與子代的概念，後面會詳加說明。

更明確的定義這些假想訊號如下，第 l 層的假想訊號集合可以用下面式子表示：

$$A_l \equiv \{a_{l,1}, a_{l,2}, \dots, a_{l,N_l}\} \quad (14)$$

而 $l=1 \dots L$ ，其中 $L \equiv \log_4 N$ ，指 N-QAM 總共會有幾個階層的訊號； $N_l \equiv N4^{1-l}$ ，指第 l 層訊號集合的元素個數。其中必須特別注意的是當 $l=1$ ， A_1 即是 A ，且 $a_{1,k} = a_k$ ， $k=1 \dots N$ 。每一個在 A_l 集合裡的假想訊號 $a_{l,s}$ （這裡稱為親代 parent）都是 A_{l-1} 集合裡其中四個元素的平均值（這裡稱這四個元素為子代 children），且將這四個元素集合稱為 A_{l-1}^s ，所以 A_{l-1}^s 是 A_{l-1} 的子集合，這些元素的關係可以用下列式子表示：

$$a_{l,s} = 0.25 \times \sum_{a_{l-1,k} \in A_{l-1}^s} a_{l-1,k} \quad (15)$$

其中 $l=2 \dots L$ ， $s=1 \dots N_l$ 。 A_l^s 除了是 A_l 的子集合外，還具有 $\cup_k A_l^k = A_l$ 及 $A_l^k \cap A_l^n = \phi$ 的特性，其中 $k, n=1 \dots N_{l+1}$ 且 $k \neq n$ 。

定義完假想訊號和原始訊號的樹狀架構及它們的集合特性後，我們知道一個原始訊號 $a_k \in A$ 在樹狀圖的每層都有屬於它自己的唯一祖先 $a_{l,s}$ ，所以我們可以定義機率模型如下：

$$P(a_k, a_{l,s}) \equiv P(a_k) \times \Pi(a_k \text{ 是 } a_{l,s} \text{ 的子代}) \quad (16)$$

其中 $\Pi()$ 是指示方程式 (如果 $()$ 裡的敘述是正確的, 則 $\Pi()$ 等於一; 如果 $()$ 裡的敘述是錯的, 則 $\Pi()$ 等於零), $k = 1 \cdots N$, $l = 2 \cdots L$, $s = 1 \cdots N_l$; 所以 $P(a_{l,s})$ 可以由以下式子得到:

$$P(a_{l,s}) = \sum_{a_k \in A} P(a_k) \times \Pi(a_k \text{ 是 } a_{l,s} \text{ 的子代}) \quad (17)$$

也就是各層假想訊號的機率是它在第一層的所有子代訊號的機率和。

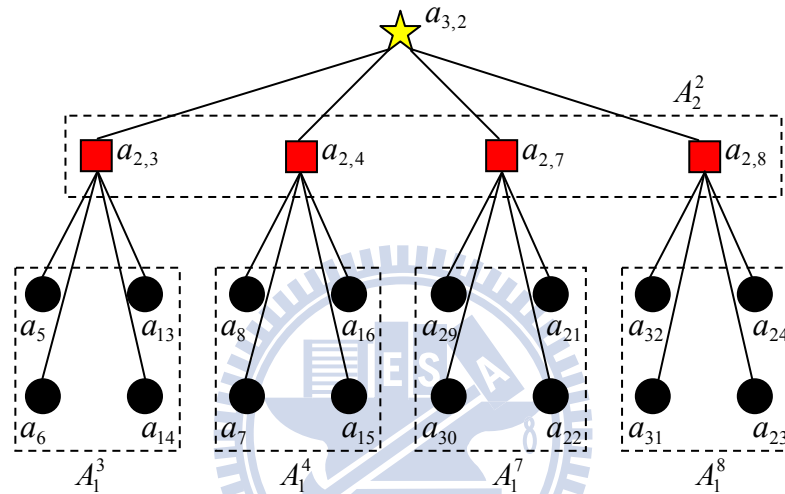


圖 4.2 64-QAM 第二象限所有訊號的樹狀關係圖

4.2 基本概念

與 SBSIDE 演算法相同, MSBSIDE 演算法為了達到降低複雜度的目的, 不採用 SQRD-M 演算法過程中每次都保留固定 M 組發生機率較高的部分天線之傳送訊號組合, 因為這固定的 M 組訊號組合中, 某些組合發生的機率比起那些發生機率高組合低很多, 所以並不盲從的每次都保留固定 M 組, 而是設定一個估計門檻來篩選要保留下來的訊號組合; 這些被保留下來的部分天線之傳送訊號組合個數 E 值與 SBSIDE 演算法得到的結論相同, 我們一樣限制 $1 \leq E \leq N$, 在此限制內, 被保留下來的訊號組合即可包含最佳的估測解, 讓錯誤率逼近 ML 估測且複雜度不會太高。

MSBSIDE 演算法與 SBSIDE 演算法同為以廣度為主做搜尋的演算法，但 MSBSIDE 演算法為了進一步降低計算與比較價值函數的個數而利用 N-QAM 本身多層架構搜尋的特性，將每層在估測一根天線的傳送訊號又分成 L 個階段來完成，其中 $L \equiv \log_4 N$ 。而每個階段（如第 l 階段，其中 $l=1 \cdots L$ ）都是用 4.1 節中對應的第 l 層訊號集合來估測傳送訊號，所以過程中每個階段用來篩選價值函數的估計門檻都是不同的，以第 l 階段的估計門檻 d_l 來表示之，其中 $l=1 \cdots L$ 。需特別注意的是當 $l=1$ 時，第一階段之估計門檻 d_1 的定義及更新方法與 SBSIDE 演算法中估計門檻 d 的定義及更新方法相同，而第 l 階段之估計門檻 d_l 的定義及更新方法會於 4.5 節中詳細說明，其中 $l=2 \cdots L$ 。

MSBSIDE 演算法降低複雜度的基本概念如下：假設接收端開始要估測第 $order(j)$ 根天線的傳送訊號，且已經有前幾次估計出來之天線（第 $order(i)$ 根天線， $i = N_T \cdots j+1$ ）的 E 組可能的部分天線之傳送訊號組合及其對應的價值函數。對高階的調變來說，例如在一個使用 64-QAM 的系統中，SBSIDE 演算法在解此根天線時必須計算與比較 $E \times 64$ 個價值函數，如果此時 E 值很大，對接收端來說複雜度還是太高。而 MSBSIDE 演算法將估測此根天線的傳送訊號分成三個階段，且順序是從第三階段進行到第一階段。第三階段是以第三層假想訊號來估測， $l=L=3$ ，而第三層的假想訊號只有四個（星形），所以只需計算 $E \times 4$ 個價值函數即可，並且檢查是否需要更新第三階段的估計門檻 d_3 ，再從這些價值函數中比較並保留小於或等於此估計門檻的價值函數，假設保留了 E' 個；接著進行第二階段的訊號估測，此階段是以第二層假想訊號來估測， $l=2$ ，根據這 E' 個被保留下來的價值函數所對應的部分天線之傳送訊號組合中的第三層假想訊號，分別去找到它在第二層假想訊號集合中的子代（children），而這些子代只會有四個（方形），所以只需計算 $E' \times 4$ 個價值函數即可，並且檢查是否需要更新第二階段的估計門檻 d_2 ，再從這些價值函數中比較並保留小於或等於此估計門檻的價值函數，假設保留了 E'' 個；最後重覆上一步驟的作法進行第一階段的訊號估測， $l=1$ ，即完成此根天線的估測。由上述過程可知，每成功估測出一根天線的傳送訊號，MSBSIDE 演算法總共計算與比較這些價值函數的個數為 $E \times 4 + E' \times 4 + E'' \times 4$ ，是 SBSIDE 演算法的 $(E + E' + E'') / (E \times 16)$ 倍，其中 $1 \leq E, E', E'' \leq 64$ 。

在使用 MSBSIDE 演算法估算傳送訊號的過程中，必須包含計算與更新各個階段的估計門檻、計算這些假想訊號集合及由親代訊號找到對應之子代訊號的複雜度，以下會有詳細的演算法說明。

4.3 演算法描述

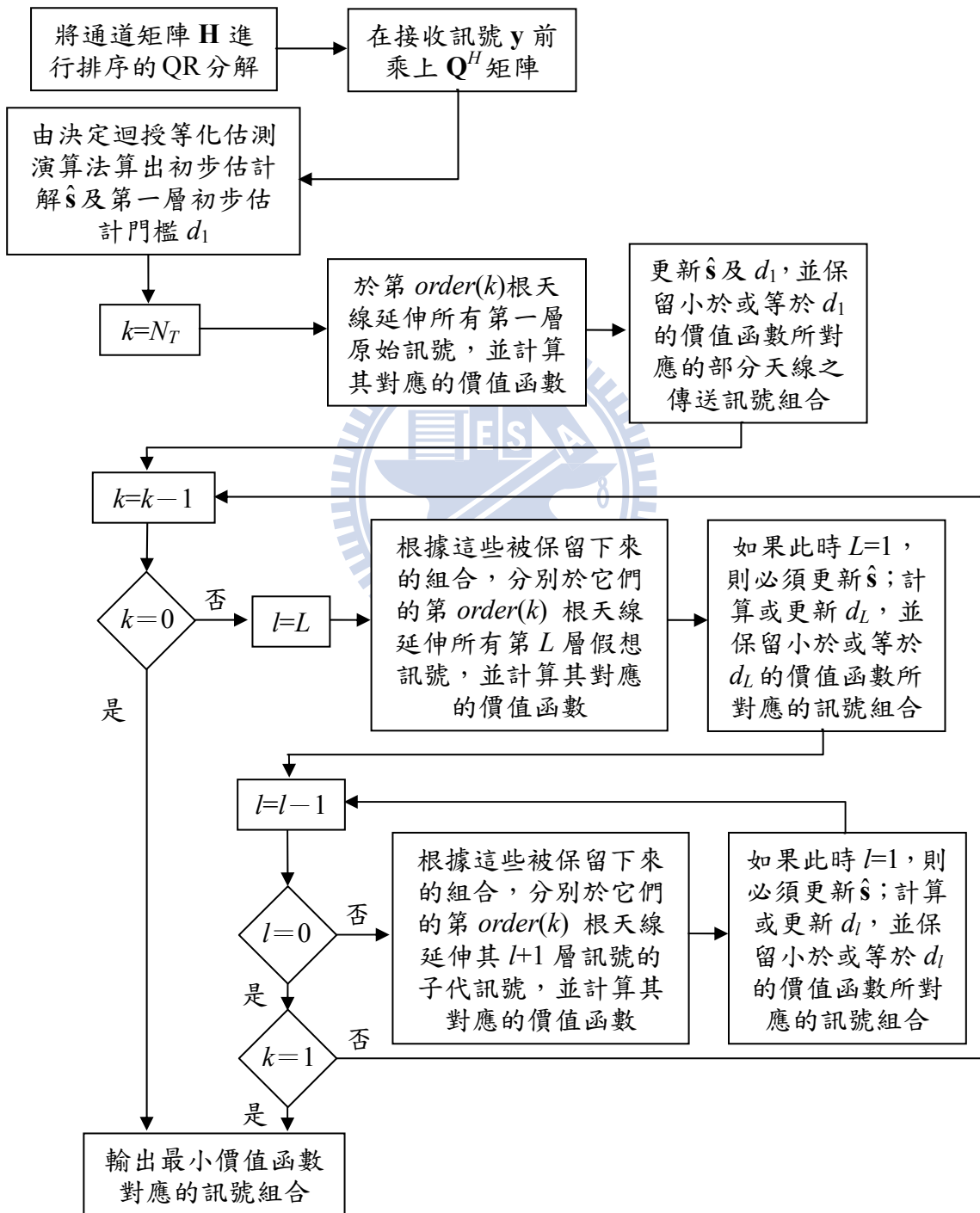


圖 4.3 MSBSIDE 演算法的流程圖

圖 4.3 為 MSBSIDE 演算法的流程圖。MSBSIDE 演算法與 SBSIDE 演算法相同，在進行訊號估測前必須先執行排序的 QR 分解，為了方便說明，將排序的 QR 分解所找到之解碼順序設成從第 N_T 根天線估測到第一根天線，即 $order=\{1,2,3\cdots N_T\}$ ， \mathbf{R} 是一個標準倒三角形矩陣。而執行演算法前，必須先根據排序的 QR 分解找到之解碼順序加上決定迴授等化估測演算法估算出一個初步估計解 $\hat{\mathbf{s}}=[\hat{s}_1 \hat{s}_2 \cdots \hat{s}_{N_T}]^T$ ，如表 3.1，並且算出第一層的初步估計門檻 $d_1=\|\tilde{\mathbf{y}}-\mathbf{R}\hat{\mathbf{s}}\|^2$ 。而使用排序的 QR 分解可以讓初步估計解 $\hat{\mathbf{s}}$ 更準確， $\hat{\mathbf{s}}$ 越準確也會讓第一層的初步估計門檻 d_1 越小，此時篩選的依據就更嚴苛，被保留下來的部分天線之傳送訊號組合就越少，反應在複雜度上也較低。

執行演算法會遇到四種情況，第一種情況是估計第 N_T 根天線的傳送訊號。與 SBSIDE 演算法得到的結論相同，我們限制被保留下來的部分天線之傳送訊號組合個數 E 為 $1\leq E\leq N$ ，且由 SQRD-M 演算法的結論得到第一根天線的估測訊號之正確性是影響演算法效能最主要的因素，因此在估測此根天線時， E 的值落在 N 附近的機率會很高。在此限制下，未使用 N-QAM 的多層架構特性做搜尋時，估計此根天線的傳送訊號所計算的 N 個價值函數最多會全部被保留下來；而使用 N-QAM 的多層架構特性做搜尋時，最多也會將這 N 個價值函數保留下來，所以未使用 N-QAM 多層架構特性來估測此根天線之傳送訊號的複雜度是最低的，只需計算與比較 N 個價值函數，複雜度並未開始成長，因此並不需要額外利用 L 個階段來估測此根天線的傳送訊號。而篩選過程是比較機率值的大小，所以直接將第一層原始傳送訊號集合 A_1 所有的訊號代入(18)式中的 $x_{N_T,1}$ 求出這 N 個延伸路徑的價值函數，與(12)式不同的地方在於這裡不用 x_{N_T} 而改以 $x_{N_T,1}$ 來表示，目的是為了要區分使用不同層訊號集合來做估測，例如這裡以第一層原始傳送訊號集合 A_1 來做估測，就以 $x_{N_T,1}$ 表示。接著從這 N 個價值函數 $\psi(x_{N_T,1})$ 中取出最小的價值函數（以 $\psi_1(x_{N_T,1})$ 表示）所對應之部分天線的傳送訊號組合 $\boldsymbol{\theta}_{N_T,1}^{(1)}=[x_{N_T,1}^{(1)}]^T$ ，如果 $\boldsymbol{\theta}_{N_T,1}^{(1)}$ 不等於初步估計解中的 \hat{s}_{N_T} ，則令一個向量 \mathbf{g} ，其中 1 到 N_T-1 位置的值為初步估計解 $\hat{\mathbf{s}}$ 中 1 到 N_T-1 位置的值，而 \mathbf{g} 中 N_T 位置的值為 $\boldsymbol{\theta}_{N_T,1}^{(1)}$ ，即 $\mathbf{g}=[\hat{s}_1 \cdots \hat{s}_{N_T-1} \boldsymbol{\theta}_{N_T,1}^{(1)}]^T$ ，並求得 $\hat{d}=\|\tilde{\mathbf{y}}-\mathbf{R}\mathbf{g}\|^2$ ；如果 \hat{d} 小於第一層的初步估計門檻 d_1 ，則讓 d_1 等於 \hat{d} ，且更新初步估計解 $\hat{\mathbf{s}}=\mathbf{g}$ 。最後從這 N 個價值函數 $\psi(x_{N_T,1})$ 中保留小於或等於第一層初步估計門檻 d_1 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_{N_T,1})$ 及所對應的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{N_T,1}^{(m)}=[x_{N_T,1}^{(m)}]^T$ ，其中 $m=1\cdots$

E ，以便演算法執行下一步驟。

$$\begin{aligned}
& P(x_{N_T,1} | \tilde{y}_{N_T}) \\
& \propto P(\tilde{y}_{N_T} | x_{N_T,1}) \times P(x_{N_T,1}) \\
& \approx \exp\left(-\frac{1}{\sigma^2} |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T,1}|^2\right) \times P(x_{N_T,1}) \\
& \Rightarrow -\log P(x_{N_T,1} | \tilde{y}_{N_T}) \\
& \approx \frac{1}{\sigma^2} |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T,1}|^2 - \log P(x_{N_T,1}) \\
& \Rightarrow \psi(x_{N_T,1}) \equiv |\tilde{y}_{N_T} - r_{N_T N_T} x_{N_T,1}|^2
\end{aligned} \tag{18}$$

第二種情況是開始要估計第 j 根天線的傳送訊號，假設此時已經有第 $j+1 \cdots N_T$ 根天線估計出來的 E 組部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j+1,1}^{(m)} = [x_{j+1,1}^{(m)} \cdots x_{N_T,1}^{(m)}]^T$ 及其對應的價值函數 $\psi_m(x_{j+1,1})$ ，其中 $m=1 \cdots E$ ，且已知更新過後的第一層初步估計門檻 d_1 及初步估計解 $\hat{\mathbf{s}}$ 。估計第 j 根天線的傳送訊號要從第 L 層的假想訊號開始估計，第 L 層的假想訊號以 $x_{j,L}$ 表示，此時接收端的數學模型如(19)式：

$$\begin{aligned}
\begin{bmatrix} \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} &= \begin{bmatrix} r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} x_{j,1} \\ x_{j+1,1} \\ \vdots \\ x_{N_T,1} \end{bmatrix} + \begin{bmatrix} \tilde{n}_{j,1} \\ \tilde{n}_{j+1,1} \\ \vdots \\ \tilde{n}_{N_T,1} \end{bmatrix} \\
\Rightarrow \begin{bmatrix} \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} &= \begin{bmatrix} r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} x_{j,L} + (x_{j,1} - x_{j,L}) \\ x_{j+1,1}^{(m)} \\ \vdots \\ x_{N_T,1}^{(m)} \end{bmatrix} + \begin{bmatrix} \tilde{n}_{j,1} \\ \tilde{n}_{j+1,1} \\ \vdots \\ \tilde{n}_{N_T,1} \end{bmatrix} \\
\Rightarrow \begin{bmatrix} \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} &= \begin{bmatrix} r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} x_{j,L} \\ x_{j+1,1}^{(m)} \\ \vdots \\ x_{N_T,1}^{(m)} \end{bmatrix} + \begin{bmatrix} \tilde{n}_{j,L} \\ \tilde{n}_{j+1,1} \\ \vdots \\ \tilde{n}_{N_T,1} \end{bmatrix}
\end{aligned} \tag{19}$$

其中 $m=1 \cdots E$ ， $\tilde{n}_{j,L} = \tilde{n}_{j,1} + r_{jj}(x_{j,1} - x_{j,L})$ ；而 $\tilde{n}_{j,L}$ 的平均值和變異數可以由附錄 4.5.1 的(21)及(22)式得到。算出 $\tilde{n}_{j,L}$ 的平均值和變異數後，即可計算延伸路徑的價值函

數。因為第 L 層的假想訊號只有四個，所以估測時會根據前幾次已經估計出的 E 組部分天線之傳送訊號組合各別去延伸四條路徑，也就是將第 L 層假想訊號集合 A_L 所有的訊號代入(20)式中的 $x_{j,l}$ ，並以 $x_{j,L}$ 表示，來計算出這 $E \times 4$ 個延伸路徑的價值函數；與(13)式不同的地方在於傳送訊號必須以 $x_{j,l}$ 表示，來區分此時的傳送訊號為第 l 層的假想訊號，且式子(20)中的雜訊能量並非 σ^2 ，要以 $Var(\tilde{n}_{j,L})$ 表示。接著並不能以第一種情況的方法來計算第 L 層的初步估計門檻 d_L ，因為此時的訊號組合中包含第 L 層的假想訊號，所以並不能與全是第一層原始訊號的初步估計解 \hat{s} 相比較，本論文提供兩種方法來計算第 l 層的初步估計門檻 d_l ，其中 $l=2 \cdots L$ ，如附錄 4.5.2：從結果可知第 l 層的初步估計門檻 d_l 定義為第一層的初步估計門檻 d_1 加上一個偏差值 $bias_l$ ，即 $d_l = d_1 + bias_l$ ，而還未進行估測之天線的可能傳送訊號與其對應的第 l 層假想訊號間的差異值再經過通道的增益後即為 $bias_l$ ；根據此差異值的定義，本論文提供以最大值為基礎 (maximum based) 和以平均值為基礎 (mean based) 兩種方法，且分別將使用這兩種不同方法的 MSBSIDE 演算法命名為 MSBSIDE-MAX 演算法及 MSBSIDE-MEAN 演算法，詳細推導於附錄 4.5.2。由附錄 4.5.2 的方法得到第 L 層的初步估計門檻 d_L 後，接著從這 $E \times 4$ 個價值函數 $\psi(x_{j,L})$ 中保留小於或等於第 L 層初步估計門檻 d_L 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_{j,L})$ 所對應的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j,L}^{(m)} = [x_{j,L}^{(m)} \ x_{j+1,1}^{(m)} \ \cdots \ x_{N_T,1}^{(m)}]^T$ 及其對應從第 N_T 根天線累積到第 $j+1$ 根天線的價值函數 $\psi_m(x_{j+1,1})$ ，其中 $m=1 \cdots E$ ，以便演算法執行下一步驟。

$$\begin{aligned}
& P(x_{j,l}, x_{j+1,1}^{(m)}, \dots, x_{N_T,1}^{(m)} | \tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
& \propto P(\tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_{j,l}, x_{j+1,1}^{(m)}, \dots, x_{N_T,1}^{(m)}) P(x_{j,l}) \prod_{k=j+1}^{N_T} P(x_{k,1}^{(m)}) \\
& = P(\tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_{j,l}, x_{j+1,1}^{(m)}, \dots, x_{N_T,1}^{(m)}) \prod_{k=j+1}^{N_T} P(x_{k,1}^{(m)}) \\
& \quad \times P(\tilde{y}_j | x_{j,l}, x_{j+1,1}^{(m)}, \dots, x_{N_T,1}^{(m)}) P(x_{j,l}) \\
& \approx P(\tilde{y}_{j+1}, \dots, \tilde{y}_{N_T} | x_{j+1,1}^{(m)}, \dots, x_{N_T,1}^{(m)}) \prod_{k=j+1}^{N_T} P(x_{k,1}^{(m)}) \\
& \quad \times \exp\left(-\frac{1}{Var(\tilde{n}_{j,l})} \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_{k,1}^{(m)} - r_{jj} x_{j,l} \right|^2\right) P(x_{j,l})
\end{aligned}$$

$$\begin{aligned}
&\Rightarrow -\log P(x_{j,l}, x_{j+1,1}, \dots, x_{N_T,1}^{(m)} | \tilde{y}_j, \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
&\quad \approx -\log P(x_{j+1,1}, \dots, x_{N_T,1}^{(m)} | \tilde{y}_{j+1}, \dots, \tilde{y}_{N_T}) \\
&\quad \quad \quad + \frac{1}{\text{Var}(\tilde{n}_{j,l})} \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_{k,1}^{(m)} - r_{jj} x_{j,l} \right|^2 - \log P(x_{j,l}) \\
&\Rightarrow \psi(x_{j,l}) \equiv \psi_m(x_{j+1,1}) + \left| \tilde{y}_j - \sum_{k=j+1}^{N_T} r_{jk} x_{k,1}^{(m)} - r_{jj} x_{j,l} \right|^2, m=1 \dots E \tag{20}
\end{aligned}$$

第三種情況是開始要估計第 j 根天線的第 l 層假想傳送訊號，其中 $l=2 \dots L-1$ 。假設此時已有 E 組可能的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j,l+1}^{(m)} = [x_{j,l+1}^{(m)} \ x_{j+1,1}^{(m)} \ \dots \ x_{N_T,1}^{(m)}]^T$ 及其對應從第 N_T 根天線累積到第 $j+1$ 根天線的價值函數 $\psi_m(x_{j+1,1})$ ，其中 $m=1 \dots E$ ，且已知更新過後的第一層初步估計門檻 d_1 及初步估計解 $\hat{\mathbf{s}}$ 。估計第 j 根天線的第 l 層假想傳送訊號必須先找到第 $l+1$ 層假想訊號 $x_{j,l+1}^{(m)}$ 的子代訊號，也就是 $x_{j,l} \in A_l^m$ ，而這些子代訊號集合都只有四個元素，所以估測時會根據前幾次已經估計出的 E 組部分天線之傳送訊號組合各別去延伸四條路徑，也就是將不同 m 值對應之訊號集合 A_l^m 中所有的訊號代入(20)式中的 $x_{j,l}$ ，來計算出這 $E \times 4$ 個延伸路徑的價值函數，其中必須特別注意的是式子中雜訊能量並非 σ^2 ，要以 $\text{Var}(\tilde{n}_{j,l})$ 表示。由附錄 4.5.2 的方法得到第 l 層的初步估計門檻 d_l 後，接著從這 $E \times 4$ 個價值函數 $\psi(x_{j,l})$ 中保留小於或等於第 l 層初步估計門檻 d_l 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_{j,l})$ 所對應的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j,l}^{(m)} = [x_{j,l}^{(m)} \ x_{j+1,1}^{(m)} \ \dots \ x_{N_T,1}^{(m)}]^T$ 及其對應從第 N_T 根天線累積到第 $j+1$ 根天線的價值函數 $\psi_m(x_{j+1,1})$ ，其中 $m=1 \dots E$ ，以便演算法執行下一步驟。

第四種情況是開始要估計第 j 根天線的第一層原始傳送訊號，此種情況和第三種情況類似，差別在於更新初步估計門檻的方法。假設此時已有 E 組可能的部分天線之傳送訊號組合 $\boldsymbol{\theta}_{j,2}^{(m)} = [x_{j,2}^{(m)} \ x_{j+1,1}^{(m)} \ \dots \ x_{N_T,1}^{(m)}]^T$ 及其對應從第 N_T 根天線累積到第 $j+1$ 根天線的價值函數 $\psi_m(x_{j+1,1})$ ，其中 $m=1 \dots E$ ，且已知更新過後的第一層初步估計門檻 d_1 及初步估計解 $\hat{\mathbf{s}}$ 。估計第 j 根天線的第一層原始傳送訊號必須先找到第二層假想訊號 $x_{j,2}^{(m)}$ 的子代訊號，也就是 $x_{j,1} \in A_1^m$ ，而這些子代訊號集合都只有四個元素，所以估測時會根據前幾次已經估計出的 E 組部分天線之傳送訊號組合各別去延伸四條路徑，也就是將不同 m 值對應之訊號集合 A_1^m 所有的訊號代入(20)

式中的 $x_{j,l}$ ，並以 $x_{j,1}$ 表示，來計算出這 $E \times 4$ 個延伸路徑的價值函數。接著從這 $E \times 4$ 個價值函數 $\psi(x_{j,1})$ 中取出最小的價值函數（以 $\psi_1(x_{j,1})$ 表示）所對應之部分天線的傳送訊號組合 $\boldsymbol{\theta}_{j,1}^{(1)} = [x_{j,1}^{(1)} \ x_{j+1,1}^{(1)} \ \cdots \ x_{N_T,1}^{(1)}]^T$ ，如果 $\boldsymbol{\theta}_{j,1}^{(1)}$ 矩陣裡的值不等於 $\hat{\mathbf{s}}(j:N_T)$ ，即初步估計解中 j 到 N_T 位置的值，則令一個向量 \mathbf{g} ，其中 1 到 $j-1$ 位置的值為初步估計解 $\hat{\mathbf{s}}$ 中 1 到 $j-1$ 位置的值，而 \mathbf{g} 中 j 到 N_T 位置的值為 $\boldsymbol{\theta}_{j,1}^{(1)}$ 矩陣對應的值，即 $\mathbf{g} = [\hat{s}_1 \ \cdots \ \hat{s}_{j-1} \ \boldsymbol{\theta}_{j,1}^{(1)}]^T$ ，並求得 $\hat{d} = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{g}\|^2$ ；如果 \hat{d} 小於第一層的初步估計門檻 d_1 ，則讓 d_1 等於 \hat{d} ，且更新初步估計解 $\hat{\mathbf{s}} = \mathbf{g}$ 。最後從這 $E \times 4$ 個價值函數 $\psi(x_{j,1})$ 中保留小於或等於第一層初步估計門檻 d_1 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個價值函數 $\psi_m(x_{j,1})$ 及其對應的部分天線之訊號組合 $\boldsymbol{\theta}_{j,1}^{(m)} = [x_{j,1}^{(m)} \ x_{j+1,1}^{(m)} \ \cdots \ x_{N_T,1}^{(m)}]^T$ ，其中 $m = 1 \cdots E$ ，以便演算法執行下一步驟。

此演算法會一直執行到部分天線之傳送訊號組合 $\boldsymbol{\theta}$ 包含所有天線的第一層估計訊號才停止，而此時最小價值函數對應的訊號組合 $\boldsymbol{\theta}^{(1)}$ 即為使用此演算法得到的最佳估計解 $\hat{\mathbf{x}}$ 。執行演算法過程中，初步估計解 $\hat{\mathbf{s}}$ 每經過 L 個階段就會更新一次，且越更新越準確，也代表第一層初步估計門檻 d_1 會越來越小；而第 l 層初步估計門檻 d_l 由附錄 4.5.2 的定義及推導過程可知每次更新 d_l ，即 $d_l = d_1 + \text{bias}_l$ ，因為 d_1 由上述說明會變小，且計算 bias_l 時的偏差值只需附加於尚未進行估測的天線，所以使用 MSBSIDE 演算法估測到越後面的天線時， bias_l 的值會越來越小，因此第 l 層初步估計門檻 d_l 的值也會越來越小。但由第(20)式可知價值函數是不斷累加而得到，所以每估測完一根天線後價值函數都會增加。綜合上述結果得知，當接收端使用 MSBSIDE 演算法估測越後面天線的傳送訊號時，價值函數經過各層初步估計門檻的篩選後被保留下來之個數會越少，反應在複雜度上也會越低。

以下用一個簡單的例子來說明演算法步驟：系統為三根傳送天線及三根接收天線的 V-BLAST 系統，調變使用 16-QAM ($L=2$)，並限制 MSBSIDE 演算法過程中經過各層初步估計門檻篩選後被保留下來之部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 16$ ，且假設排序的 QR 分解所找到之解碼順序是從第三根天線估測回第一根天線。如圖 4.4，虛線是延伸的路徑，實線是經過各層初步估計門檻的篩選後而被保留下來之路徑，圓形是第一層原始傳送訊號，正方形是第二層的假想傳送訊號，節點旁的數字為路徑延伸到此節點所累積的價值函數。估計第三根天線的傳送訊號時，不需使用 N-QAM 的多層架構來搜尋，直接將 16-QAM 的十六個

調變訊號代入(18)式求出十六個價值函數，假設此時更新過後的第一層初步估計門檻 d_1 之值為 3.8，則保留小於或等於此初步估計門檻 d_1 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 2.3 及 3.0。估計第二根天線的傳送訊號時，必須根據上一步驟保留下來的兩條路徑，分別去延伸四個第二層假想傳送訊號，即將第二層假想訊號集合 A_2 的四個假想訊號代入(20)式求出八個價值函數，假設此時更新過後的第二層初步估計門檻 d_2 之值為 15.6，則保留小於或等於此初步估計門檻 d_2 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 11.3 及 14.5；接著根據這兩個第二層假想傳送訊號節點分別去延伸它在第一層原始傳送訊號集合的子代訊號，得到八條延伸路徑，並計算每條延伸路徑的價值函數，假設此時更新過後的第一層初步估計門檻 d_1 之值為 3.5，則保留小於或等於此初步估計門檻 d_1 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 2.4 及 3.4。估計第一根天線的傳送訊號也是一樣的做法，必須根據上一步驟保留下來的兩條路徑，分別去延伸四個第二層假想傳送訊號，即將第二層假想訊號集合 A_2 的四個假想訊號代入(20)式求出八個價值函數，假設此時更新過後的第二層初步估計門檻 d_2 之值為 14.9，則保留小於或等於此初步估計門檻 d_2 的價值函數所對應之節點，此例子被保留下來的節點所對應之價值函數為 8.8；接著再根據此第二層假想傳送訊號節點去延伸它在第一層原始傳送訊號集合的子代訊號，得到四條延伸路徑，並計算每條延伸路徑的價值函數，最後再從這四個價值函數選取最小價值函數所對應的路徑即為使用 MSBSIDE 演算法得到的最佳估計解。

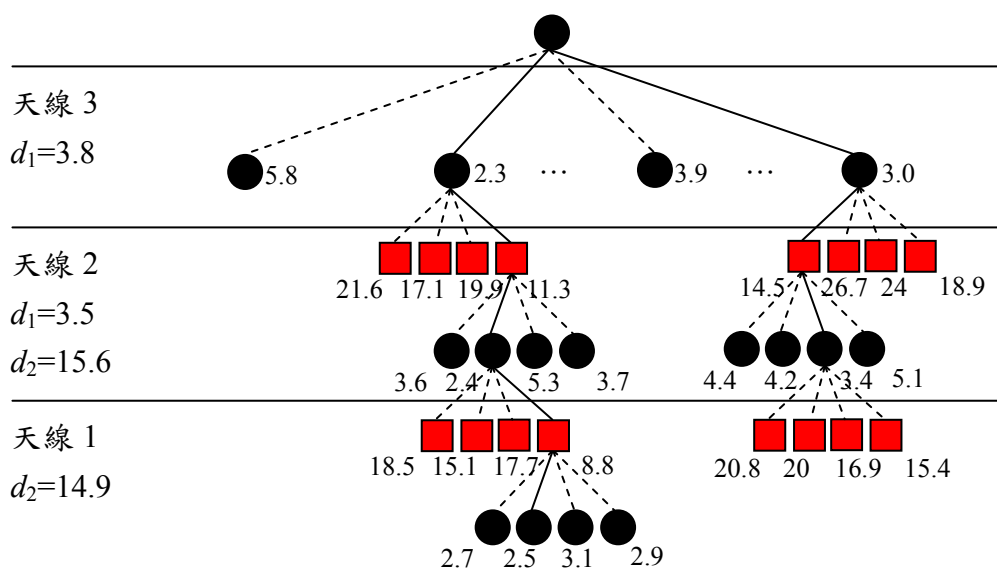


圖 4.4 $N_T = 3$ ， $N_R = 3$ ，運用 MSBSIDE 演算法做估測的樹狀圖示

4.4 演算法總結

初始化.

計算通道矩陣 \mathbf{H} 的排序 QR 分解與 $order$ ，並利用 \mathbf{Q} 矩陣來算出 $\tilde{\mathbf{y}}$ ；接著藉由 $order$ 、 $\tilde{\mathbf{y}}$ 和決定迴授等化估測演算法算出初步的估計解 $\hat{\mathbf{s}}$ 及第一層初步估計門檻 $d_1 = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}\|^2$

步驟一.

$i = N_T, k = order(i)$ ，計算 $\psi(x_{k,1}) = |\tilde{y}_k - r_{kk}x_{k,1}|^2$ ，其中 $x_{k,1} \in A_1$ ；接著從這 N 個 $\psi(x_{k,1})$ 取最小的 $\psi_1(x_{k,1})$ 所對應之 $\theta_{k,1}^{(1)}$ ，如果 $\theta_{k,1}^{(1)}$ 不等於 \hat{s}_k ，就讓 $\mathbf{g} = [\hat{s}_1 \cdots \hat{s}_{k-1} \theta_{k,1}^{(1)} \hat{s}_{k+1} \cdots \hat{s}_{N_T}]^T$ ，並求得 $\hat{d} = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{g}\|^2$ ；如果 $\hat{d} < d_1$ ，則 $d_1 = \hat{d}$ ，且讓 $\hat{\mathbf{s}} = \mathbf{g}$ 。最後從這 N 個 $\psi(x_{k,1})$ 保留小於或等於 d_1 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個 $\psi_m(x_{k,1})$ 及所對應的 $\theta_{k,1}^{(m)}$ ，其中 $m = 1 \cdots E$

步驟二.

$i = i - 1, k = order(i)$

1) $l = L$ ，用下列式子計算延伸第 L 層假想訊號所累積的價值函數：

$$\psi(x_{k,L}) = \psi_m(x_{order(i+1),1}) + \left| \tilde{y}_k - \sum_{j=order(i+1):N_T} r_{kj}x_{j,1}^{(m)} - r_{kk}x_{k,L} \right|^2$$

其中 $m = 1 \cdots E, x_{k,L} \in A_L$ 。接著由附錄 4.5.2 的方法計算 d_L ，並從這 $E \times 4$ 個 $\psi(x_{k,L})$ 中保留小於或等於 d_L 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個 $\psi_m(x_{k,L})$ 所對應的 $\psi_m(x_{order(i+1),1})$ 及 $\theta_{k,L}^{(m)}$ (θ 矩陣 $order(i)$ 位置的值為第 L 層的假想訊號， $order(i+1): N_T$ 位置的值為第一層原始傳送訊號)，其中 $m = 1 \cdots E$

2) $l = L - 1 \cdots 1$ ，根據第 $l+1$ 層的假想訊號 $x_{k,l+1}^{(m)}$ ，用下列式子計算延伸它的子代訊號所累積之價值函數：

$$\psi(x_{k,l}) = \psi_m(x_{order(i+1),1}) + \left| \tilde{y}_k - \sum_{j=order(i+1):N_T} r_{kj}x_{j,1}^{(m)} - r_{kk}x_{k,l} \right|^2$$

其中 $m = 1 \cdots E, x_{k,l} \in A_l^m$

- i) 如果 $l \neq 1$ ，由附錄 4.5.2 的方法計算 d_l ，並從這 $E \times 4$ 個 $\psi(x_{k,l})$ 保留小於或等於 d_l 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個 $\psi_m(x_{k,l})$ 所對應的 $\psi_m(x_{order(i+1),l})$ 及 $\theta_{k,l}^{(m)}$ ，其中 $m=1 \cdots E$
- ii) 如果 $l=1$ 但 $i \neq 1$ ，從這 $E \times 4$ 個 $\psi(x_{k,1})$ 取最小的 $\psi_1(x_{k,1})$ 所對應的 $\theta_{k,1}^{(1)}$ ，若 $\theta_{k,1}^{(1)} \neq \hat{\mathbf{s}}(order(i:N_T))$ ，就讓 $\mathbf{g} = [\hat{s}_1 \cdots \theta_{k,1}^{(1)} \cdots \hat{s}_{N_T}]^T$ (把 $\hat{\mathbf{s}}$ 矩陣 $order(i:N_T)$ 位置的值替換成 $\theta_{k,1}^{(1)}$ 矩陣對應的值)，並求得 $\hat{d} = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{g}\|^2$ ；如果 $\hat{d} < d_1$ ，則 $d_1 = \hat{d}$ ，且讓 $\hat{\mathbf{s}} = \mathbf{g}$ 。接著從這 $E \times 4$ 個 $\psi(x_{k,1})$ 保留小於或等於 d_1 的價值函數，假設保留了 E 個，其餘則刪除，並記錄這 E 個 $\psi_m(x_{k,1})$ 及所對應的 $\theta_{k,1}^{(m)}$ ，其中 $m=1 \cdots E$ 。最後再回到步驟二。
- iii) 如果 $l=1$ 且 $i=1$ ，此時最小的價值函數 $\psi_1(x_{k,1})$ 所對應的 $\theta_{k,1}^{(1)}$ 即為估計出來的 $\hat{\mathbf{x}}$ ，並且結束演算法

4.5 附錄

4.5.1 計算 $\tilde{n}_{j,L}$ 的平均值和變異數

由 4.3 節的推導過程得知 $\tilde{n}_{j,L} = \tilde{n}_{j,1} + r_{jj} \times (x_{j,1} - x_{j,L})$ ，其平均值和變異數的計算如下(21)式及(22)式：

$$\begin{aligned}
 E(\tilde{n}_{j,L}) &= E(\tilde{n}_{j,1} + r_{jj} \times (x_{j,1} - x_{j,L})) \\
 &= E(\tilde{n}_{j,1}) + r_{jj} E(x_{j,1} - x_{j,L}) \\
 &= 0 + r_{jj} (E(x_{j,1}) - E(x_{j,L})) = 0
 \end{aligned} \tag{21}$$

$$\begin{aligned}
 Var(\tilde{n}_{j,L}) &= Var(\tilde{n}_{j,1} + r_{jj} \times (x_{j,1} - x_{j,L})) \\
 &= Var(\tilde{n}_{j,1}) + |r_{jj}|^2 Var(x_{j,1} - x_{j,L}) \\
 &\quad + Covar(\tilde{n}_{j,1}, r_{jj} (x_{j,1} - x_{j,L})) \\
 &= Var(\tilde{n}_{j,1}) + |r_{jj}|^2 Var(x_{j,1} - x_{j,L}) \\
 &= \sigma^2 + |r_{jj}|^2 (E(|x_{j,1}|^2) + E(|x_{j,L}|^2) \\
 &\quad - E(x_{j,1}x_{j,L}^*) - E(x_{j,1}^*x_{j,L}))
 \end{aligned}$$

$$\begin{aligned}
&= \sigma^2 + |r_{jj}|^2 \left(\frac{1}{N} \sum_{a_{1,k} \in A_1} |a_{1,k}|^2 + \frac{1}{N_L} \sum_{a_{L,k} \in A_L} |a_{L,k}|^2 \right. \\
&\quad - \sum_{x_{j,L} \in A_L} x_{j,L}^* \sum_{x_{j,1} \in A_1} x_{j,1} p(x_{j,1}, x_{j,L}) \\
&\quad \left. - \sum_{x_{j,L} \in A_L} x_{j,L} \sum_{x_{j,1} \in A_1} x_{j,1}^* p(x_{j,1}, x_{j,L}) \right) \\
&= \sigma^2 + |r_{jj}|^2 \left(\frac{1}{N} \sum_{a_{1,k} \in A_1} |a_{1,k}|^2 + \frac{1}{N_L} \sum_{a_{L,k} \in A_L} |a_{L,k}|^2 \right. \\
&\quad \left. - \frac{4^{L-1}}{N} \sum_{a_{L,k} \in A_L} |a_{L,k}|^2 - \frac{4^{L-1}}{N} \sum_{a_{L,k} \in A_L} |a_{L,k}|^2 \right) \\
&= \sigma^2 + |r_{jj}|^2 \left(\frac{1}{N} \sum_{a_{1,k} \in A_1} |a_{1,k}|^2 - \frac{1}{N_L} \sum_{a_{L,k} \in A_L} |a_{L,k}|^2 \right) \\
&= \text{Var}(\tilde{n}_{j,1}) + |r_{jj}|^2 (\text{Var}(x_{j,1}) - \text{Var}(x_{j,L})) \tag{22}
\end{aligned}$$

其中機率值 $p(x_{j,1}, x_{j,L})$ 可以從(16)式得到。從 4.1 節的定義可知各層訊號值 $x_{j,l}$ 的機率分佈都是均勻分佈，所以其平均值 $E(x_{j,l})$ 等於零，變異數 $\text{Var}(x_{j,l})$ 為：

$$\frac{1}{N_l} \sum_{a_{l,k} \in A_l} |a_{l,k}|^2, \quad l=1 \cdots L \tag{23}$$

4.5.2 兩種計算第 l 層初步估計門檻 d_l 的方法，其中 $l=2 \cdots L$

為了方便說明，與 4.3 節相同，將排序的 QR 分解所找到之解碼順序設成從第 N_T 根天線估測回第一根天線，即 $\text{order} = \{1, 2, 3 \cdots N_T\}$ ， \mathbf{R} 是一個標準倒三角形矩陣，且假設此時已經估測到第 j 根天線的第 l 層假想傳送訊號，第 $j+1$ 根天線到第 N_T 根天線的第一層原始傳訊訊號已估測出來。估測第 j 根天線的第 l 層假想傳送訊號必須要根據第 l 層初步估計門檻 d_l 來篩選價值函數，但此時我們只有更新過後的第一層初步估計門檻 d_1 及初步估計解 $\hat{\mathbf{s}}$ 的資訊，其中 $d_1 = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}\|^2$ ，可以用下列式子表示：

$$d_1 = \left\| \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N_T} \\ 0 & r_{22} & \cdots & r_{2N_T} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \hat{s}_{1,1} \\ \hat{s}_{2,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} \right\|^2 \quad (24)$$

其中 $\hat{s}_{k,l}$ 為第 k 根天線的第 l 層初步估計解。而第 l 層初步估計門檻 d_l 與第一層初步估計門檻 d_1 間並非獨立關係，根據此關係的計算方式及表示方法不同，分成以最大值為基礎（maximum based）和以平均值為基礎（mean based）兩種方法，詳細如下。

4.5.2.1 以最大值為基礎（maximum based）計算 d_l

因為估測到第 j 根天線，且此時是以第 l 層的假想訊號來做估測，要得到第 l 層初步估計門檻 d_l 必須將 $\hat{\mathbf{s}}$ 矩陣中尚未進行估測的天線之值都以第 l 層初步估計解表示、已進行估測的天線之值不需改變，且此新矩陣以 $\hat{\mathbf{s}}_l^{\max}$ 表示，即 $\hat{\mathbf{s}}_l^{\max} = [\hat{s}_{1,l}, \hat{s}_{2,l}, \dots, \hat{s}_{j,l}, \hat{s}_{j+1,l}, \dots, \hat{s}_{N_T,l}]^T$ ，此時計算出的第 l 層初步估計門檻 d_l 稱為 d_l^{\max} ，且可以用 $\|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}_l^{\max}\|^2$ 表示。除此之外，第 l 層初步估計解 $\hat{s}_{k,l}$ 會等於第一層初步估計解 $\hat{s}_{k,1}$ 再加上一個訊號偏差值，所以將(24)式 $\hat{\mathbf{s}}$ 矩陣的 $order(1:j)$ 位置之值都加上一個訊號偏差值即可得到以最大值為基礎的第 l 層初步估計門檻 d_l^{\max} ，且 d_l^{\max} 可以進一步化簡成(25)式，如下：

$$d_l^{\max} = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}_l^{\max}\|^2$$

$$\Rightarrow d_l^{\max} = \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \begin{bmatrix} r_{11} & \cdots & r_{1j} & r_{1j+1} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & \cdots & 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \hat{s}_{1,l} \\ \vdots \\ \hat{s}_{j,l} \\ \hat{s}_{j+1,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} \right\|^2$$

$$\Rightarrow d_l^{\max} = \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \begin{bmatrix} r_{11} & \cdots & r_{1j} & r_{1j+1} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & \cdots & 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \hat{s}_{1,1} + \lambda_{1,l} \\ \vdots \\ \hat{s}_{j,1} + \lambda_{j,l} \\ \hat{s}_{j+1,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} \right\|^2$$

$$\Rightarrow d_l^{\max} = \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \hat{s}_{1,1} \\ \vdots \\ \hat{s}_{j,1} \\ \hat{s}_{j+1,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2$$

$$\Rightarrow d_l^{\max} \leq \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_j \\ \tilde{y}_{j+1} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \hat{s}_{1,1} \\ \vdots \\ \hat{s}_{j,1} \\ \hat{s}_{j+1,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} + \mathbf{R} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2$$

$$= d_1 + \left\| \begin{bmatrix} r_{11} & \cdots & r_{1j} & r_{1j+1} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & r_{jj} & r_{jj+1} & \cdots & r_{jN_T} \\ 0 & \cdots & 0 & r_{j+1j+1} & \cdots & r_{j+1N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2$$

$$= d_1 + \left\| \begin{bmatrix} r_{11}\lambda_{1,l} + r_{12}\lambda_{2,l} + \cdots + r_{1j}\lambda_{j,l} \\ r_{22}\lambda_{2,l} + \cdots + r_{2j}\lambda_{j,l} \\ \vdots \\ r_{jj}\lambda_{j,l} \end{bmatrix} \right\|^2$$

$$= d_1 + |r_{11}\lambda_{1,l} + r_{12}\lambda_{2,l} + \cdots + r_{1j}\lambda_{j,l}|^2 + |r_{22}\lambda_{2,l} + \cdots + r_{2j}\lambda_{j,l}|^2 + \cdots + |r_{jj}\lambda_{j,l}|^2$$

$$\begin{aligned}
&\leq d_1 + |r_{11}\lambda_{1,l}|^2 + |r_{12}\lambda_{2,l}|^2 + \cdots + |r_{1j}\lambda_{j,l}|^2 \\
&\quad + |r_{22}\lambda_{2,l}|^2 + \cdots + |r_{2j}\lambda_{j,l}|^2 + \cdots + |r_{jj}\lambda_{j,l}|^2 \\
&= d_1 + |r_{11}|^2|\lambda_{1,l}|^2 + |r_{12}|^2|\lambda_{2,l}|^2 + \cdots + |r_{1j}|^2|\lambda_{j,l}|^2 \\
&\quad + |r_{22}|^2|\lambda_{2,l}|^2 + \cdots + |r_{2j}|^2|\lambda_{j,l}|^2 + \cdots + |r_{jj}|^2|\lambda_{j,l}|^2 \quad (25) \\
&= d_1 + |r_{11}|^2 \times |\lambda_{1,l}|^2 + (|r_{12}|^2 + |r_{22}|^2) \times |\lambda_{2,l}|^2 + \\
&\quad \cdots + (|r_{1j}|^2 + |r_{2j}|^2 + \cdots + |r_{jj}|^2) \times |\lambda_{j,l}|^2
\end{aligned}$$

$$\Rightarrow d_i^{\max} \equiv d_1 + bias_i^{\max}$$

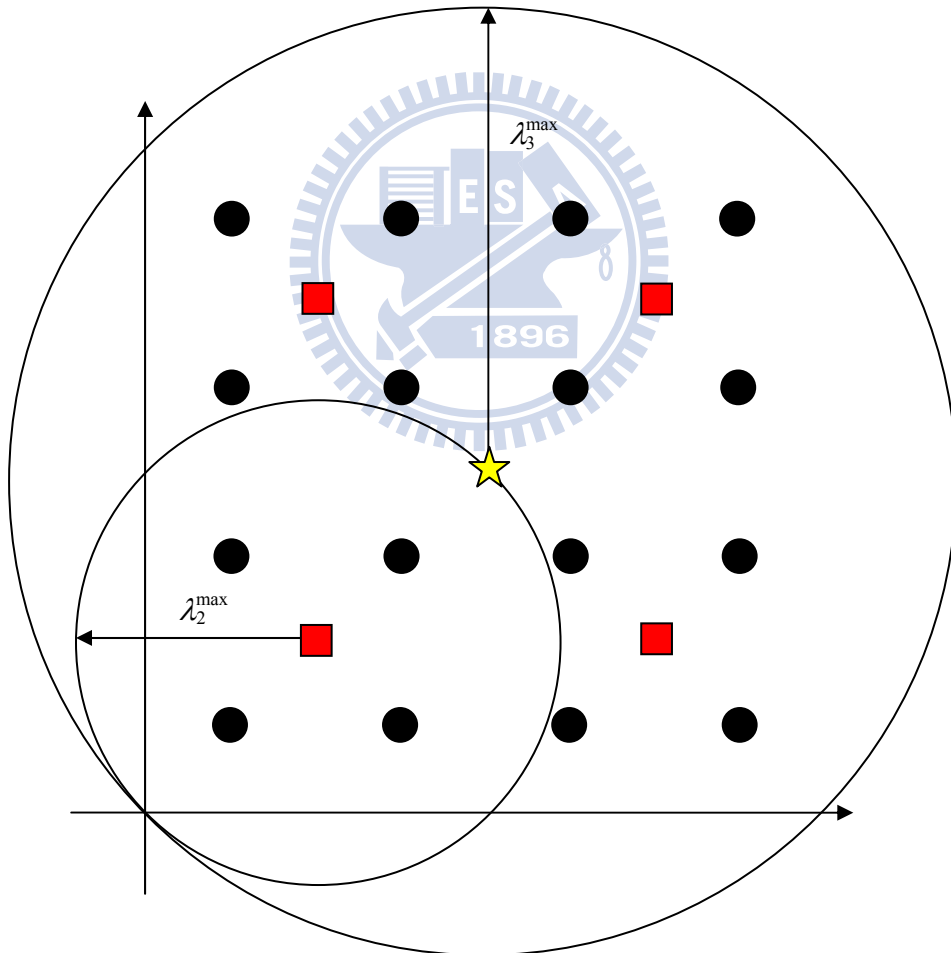


圖 4.5 在 64-QAM 訊號座標圖下以最大值為基礎計算估計門檻時，各層的訊號偏差值大小及訊號值涵蓋的範圍

其中 $\lambda_{k,l}$ 是指第一層初步估計解 $\hat{s}_{k,1}$ 變成第 l 層初步估計解 $\hat{s}_{k,l}$ 所要加的訊號偏差值。以一個 64-QAM 調變訊號座標圖第一象限的所有原始訊號及假想訊號為例子，如圖 4.5，星形訊號為第三層初步估計解 $\hat{s}_{k,3}$ ，圓形訊號為第一層初步估計解 $\hat{s}_{k,1}$ ，當第一層初步估計解 $\hat{s}_{k,1}$ 要變成第三層初步估計解 $\hat{s}_{k,3}$ 時必須加上一個訊號偏差值 $\lambda_{k,3}$ ，但是根據 $\hat{s}_{k,1}$ 訊號點的不同會有不同的訊號偏差值 $\lambda_{k,3}$ ，在某些情況下如果訊號偏差值取太小，可能會將最佳路徑排除，所以考慮到估測的正確性，在以第三層假想訊號做估測時，所有第一象限可能出現的訊號值都要以此象限的星形假想訊號表示，此時最大的偏差值即為原點到星形假想訊號的距離，以 λ_3^{\max} 表示。如圖 4.5，以 λ_3^{\max} 為半徑、星形假想訊號為圓心畫圓，最外圈圓內區域的訊號值為使用此方法於以第三層假想訊號做估測時必須列入考慮的訊號值範圍；而以 λ_2^{\max} 為半徑、正方形假想訊號為圓心畫圓，內圈圓內區域的訊號值為使用此方法於以第二層假想訊號做估測時必須列入考慮的訊號值範圍。因此，以最大值為基礎計算第 l 層初步估計門檻 d_l 時，(25)式中的 $|\lambda_{1,l}|^2$ 、 $|\lambda_{2,l}|^2 \cdots |\lambda_{j,l}|^2$ 都可以用 $|\lambda_l^{\max}|^2$ 來表示，指的是第一層原始訊號變成第 l 層假想訊號必須加的最大訊號偏差值能量，所以(25)式的 $bias_l^{\max}$ 可以修改成(26)式：

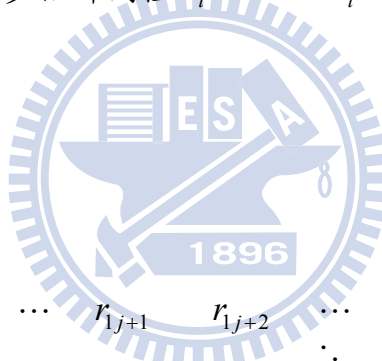
$$\begin{aligned}
 bias_l^{\max} &= |r_{11}|^2 \times |\lambda_{1,l}|^2 + (|r_{12}|^2 + |r_{22}|^2) \times |\lambda_{2,l}|^2 + \\
 &\quad \cdots + (|r_{1j}|^2 + |r_{2j}|^2 + \cdots + |r_{jj}|^2) \times |\lambda_{j,l}|^2 \\
 &= (|r_{11}|^2 + |r_{12}|^2 + |r_{22}|^2 + \cdots + |r_{1j}|^2 \\
 &\quad + |r_{2j}|^2 + \cdots + |r_{jj}|^2) \times |\lambda_l^{\max}|^2
 \end{aligned} \tag{26}$$

4.5.2.2 以平均值為基礎 (mean based) 計算 d_l

以平均值為基礎計算第 l 層初步估計門檻 d_l 與以最大值為基礎的計算方法類似，不同的地方在於各層的訊號偏差值並非取最大值 λ_l^{\max} ，而是取訊號偏差值的平均值，以 λ_l^{mean} 表示。以一個 64-QAM 調變訊號座標圖第一象限的所有原始訊號及假想訊號為例子，如圖 4.5，第三層的訊號偏差值如果以平均值 λ_3^{mean} 表示，因為 λ_3^{mean} 小於 λ_3^{\max} ，所以以 λ_3^{mean} 為半徑、星形假想訊號為圓心畫圓，並不能包含所有第一象限中可能出現的訊號值，此時計算出的第三層初步估計門檻 d_3^{mean} 會因為

值太小而可能將最佳路徑排除，經模擬驗證得知的結果也是如此，估測的錯誤率會較高。

為了解決此問題，以平均值為基礎來計算第 l 層初步估計門檻 d_l 時必須多考慮前一根已經估測出來的天線之訊號偏差值，讓 d_l^{mean} 的值變大，做法如下：因為估測到第 j 根天線，且此時是以第 l 層的假想訊號來做估測，要得到第 l 層初步估計門檻 d_l 必須將 $\hat{\mathbf{s}}$ 矩陣中未進行估測的天線之值及前一次估測的天線之值都以第 l 層初步估計解表示，而除了前一次估測的天線之值外，其餘已進行估測的天線之值不需改變，且此新矩陣以 $\hat{\mathbf{s}}_l^{mean}$ 表示，即 $\hat{\mathbf{s}}_l^{mean} = [\hat{s}_{1,l} \ \hat{s}_{2,l} \ \cdots \ \hat{s}_{j+1,l} \ \hat{s}_{j+2,l} \ \cdots \ \hat{s}_{N_T,l}]^T$ ；此時計算出的第 l 層初步估計門檻 d_l 稱為 d_l^{mean} ，且可以用 $\|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}_l^{mean}\|^2$ 表示。除此之外，第 l 層初步估計解 $\hat{s}_{k,l}$ 會等於第一層初步估計解 $\hat{s}_{k,1}$ 再加上一個訊號偏差值，所以將(24)式 $\hat{\mathbf{s}}$ 矩陣的 $order(1:j+1)$ 位置之值都加上一個訊號偏差值即可得到以平均值為基礎的第 l 層初步估計門檻 d_l^{mean} ，且 d_l^{mean} 可以進一步化簡成(27)式，如下：



$$d_l^{mean} = \|\tilde{\mathbf{y}} - \mathbf{R}\hat{\mathbf{s}}_l^{mean}\|^2$$

$$\Rightarrow d_l^{mean} = \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{j+1} \\ \tilde{y}_{j+2} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \begin{bmatrix} r_{11} & \cdots & r_{1j+1} & r_{1j+2} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & r_{j+1j+1} & r_{j+1j+2} & \cdots & r_{j+1N_T} \\ 0 & \cdots & 0 & r_{j+2j+2} & \cdots & r_{j+2N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \hat{s}_{1,l} \\ \vdots \\ \hat{s}_{j+1,l} \\ \hat{s}_{j+2,l} \\ \vdots \\ \hat{s}_{N_T,l} \end{bmatrix} \right\|^2$$

$$\Rightarrow d_l^{mean} = \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{j+1} \\ \tilde{y}_{j+2} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \begin{bmatrix} r_{11} & \cdots & r_{1j+1} & r_{1j+2} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & r_{j+1j+1} & r_{j+1j+2} & \cdots & r_{j+1N_T} \\ 0 & \cdots & 0 & r_{j+2j+2} & \cdots & r_{j+2N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \hat{s}_{1,1} + \lambda_{1,l} \\ \vdots \\ \hat{s}_{j+1,1} + \lambda_{j+1,l} \\ \hat{s}_{j+2,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} \right\|^2$$

$$\begin{aligned}
\Rightarrow d_l^{mean} &= \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{j+1} \\ \tilde{y}_{j+2} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \hat{s}_{1,1} \\ \vdots \\ \hat{s}_{j+1,1} \\ \hat{s}_{j+2,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j+1,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2 \\
\Rightarrow d_l^{mean} &\leq \left\| \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{j+1} \\ \tilde{y}_{j+2} \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} - \mathbf{R} \times \begin{bmatrix} \hat{s}_{1,1} \\ \vdots \\ \hat{s}_{j+1,1} \\ \hat{s}_{j+2,1} \\ \vdots \\ \hat{s}_{N_T,1} \end{bmatrix} \right\|^2 + \left\| \mathbf{R} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j+1,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2 \\
&= d_1 + \left\| \begin{bmatrix} r_{11} & \cdots & r_{1j+1} & r_{1j+2} & \cdots & r_{1N_T} \\ 0 & \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & \cdots & r_{j+1j+1} & r_{j+1j+2} & \cdots & r_{j+1N_T} \\ 0 & \cdots & 0 & r_{j+2j+2} & \cdots & r_{j+2N_T} \\ 0 & 0 & \cdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & r_{N_T N_T} \end{bmatrix} \times \begin{bmatrix} \lambda_{1,l} \\ \vdots \\ \lambda_{j+1,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2 \\
&= d_1 + \left\| \begin{bmatrix} r_{11}\lambda_{1,l} + r_{12}\lambda_{2,l} + \cdots + r_{1j+1}\lambda_{j+1,l} \\ r_{22}\lambda_{2,l} + \cdots + r_{2j+1}\lambda_{j+1,l} \\ \vdots \\ r_{j+1j+1}\lambda_{j+1,l} \end{bmatrix} \right\|^2 \\
&= d_1 + |r_{11}\lambda_{1,l} + r_{12}\lambda_{2,l} + \cdots + r_{1j+1}\lambda_{j+1,l}|^2 \\
&\quad + |r_{22}\lambda_{2,l} + \cdots + r_{2j+1}\lambda_{j+1,l}|^2 + \cdots + |r_{j+1j+1}\lambda_{j+1,l}|^2 \\
&\leq d_1 + |r_{11}\lambda_{1,l}|^2 + |r_{12}\lambda_{2,l}|^2 + \cdots + |r_{1j+1}\lambda_{j+1,l}|^2 \\
&\quad + |r_{22}\lambda_{2,l}|^2 + \cdots + |r_{2j+1}\lambda_{j+1,l}|^2 + \cdots + |r_{j+1j+1}\lambda_{j+1,l}|^2 \\
&= d_1 + |r_{11}|^2 |\lambda_{1,l}|^2 + |r_{12}|^2 |\lambda_{2,l}|^2 + \cdots + |r_{1j+1}|^2 |\lambda_{j+1,l}|^2 \\
&\quad + |r_{22}|^2 |\lambda_{2,l}|^2 + \cdots + |r_{2j+1}|^2 |\lambda_{j+1,l}|^2 + \cdots + |r_{j+1j+1}|^2 |\lambda_{j+1,l}|^2 \\
&= d_1 + |r_{11}|^2 \times |\lambda_{1,l}|^2 + (|r_{12}|^2 + |r_{22}|^2) \times |\lambda_{2,l}|^2 + \\
&\quad \cdots + (|r_{1j+1}|^2 + |r_{2j+1}|^2 + \cdots + |r_{j+1j+1}|^2) \times |\lambda_{j+1,l}|^2
\end{aligned}$$

$$\Rightarrow d_l^{mean} \equiv d_1 + bias_l^{mean} \quad (27)$$

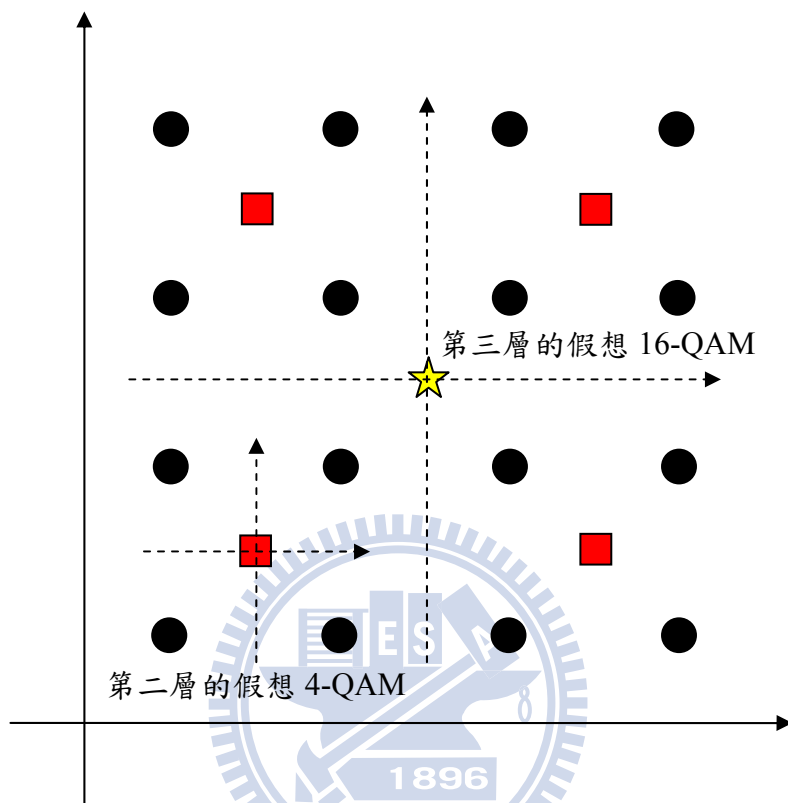
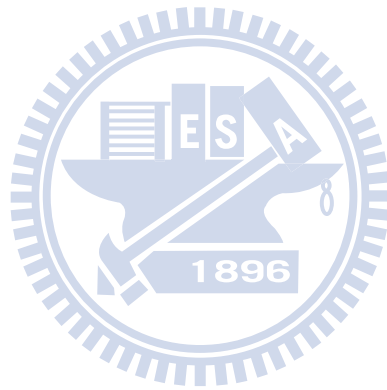


圖 4.6 在 64-QAM 訊號座標圖下以平均值為基礎計算估計門檻時，各層的假想 N-QAM 調變訊號座標圖

其中 $\lambda_{k,l}$ 是指第一層初步估計解 $\hat{s}_{k,1}$ 變成第 l 層初步估計解 $\hat{s}_{k,l}$ 所要加的訊號偏差值。以一個 64-QAM 調變訊號座標圖第一象限的所有原始訊號及假想訊號為例子，如圖 4.6，星形訊號為第三層初步估計解 $\hat{s}_{k,3}$ ，圓形訊號為第一層初步估計解 $\hat{s}_{k,1}$ ，當第一層初步估計解 $\hat{s}_{k,1}$ 要變成第三層初步估計解 $\hat{s}_{k,3}$ 時必須加上一個訊號偏差值 $\lambda_{k,3}$ ，但是根據 $\hat{s}_{k,1}$ 訊號點的不同會有不同的訊號偏差值 $\lambda_{k,3}$ ，這裡並非取最大的訊號偏差值，而是以平均概念來代表這些不同大小的訊號偏差值，以 λ_3^{mean} 表示。如圖 4.6，此時 $|\lambda_3^{mean}|^2$ 就相當於把星形位置當成原點，其餘十六個圓形訊號點所組成的假想 16-QAM 調變訊號之平均能量； $|\lambda_2^{mean}|^2$ 就相當於把正方形位置當成原點，周圍四個圓形訊號點所組成的假想 4-QAM 調變訊號之平均能量，其中 N-QAM 調變訊號的平均能量為 $2 \times (N-1)/3$ 。因此，以平均值為基礎計算第 l

層初步估計門檻 d_l 時，(27)式中的 $|\lambda_{1,l}|^2$ 、 $|\lambda_{2,l}|^2 \cdots |\lambda_{j+1,l}|^2$ 都可以用 $|\lambda_l^{mean}|^2$ 來表示，指的是第一層原始訊號變成第 l 層假想訊號必須加的平均訊號偏差值能量，所以(27)式的 $bias_l^{mean}$ 可以修改成(28)式：

$$\begin{aligned}
 bias_l^{mean} &= |r_{11}|^2 \times |\lambda_{1,l}|^2 + (|r_{12}|^2 + |r_{22}|^2) \times |\lambda_{2,l}|^2 + \\
 &\quad \cdots + (|r_{1j+1}|^2 + |r_{2j+1}|^2 + \cdots + |r_{j+1j+1}|^2) \times |\lambda_{j+1,l}|^2 \\
 &= (|r_{11}|^2 + |r_{12}|^2 + |r_{22}|^2 + \cdots + |r_{1j+1}|^2 \\
 &\quad + |r_{2j+1}|^2 + \cdots + |r_{j+1j+1}|^2) \times |\lambda_l^{mean}|^2
 \end{aligned} \tag{28}$$



五、模擬結果

於第三章及第四章詳細介紹幾種以廣度為主做搜尋的演算法後，本章節將利用電腦來模擬並分析第四章所提出的 MSBSIDE 演算法應用於 V-BLAST 系統之接收端估測的解碼錯誤率與複雜度，並與 SQRD-M 演算法、SBSIDE 演算法及 MSQRD-M 演算法(即將 N-QAM 的多層架構搜尋特性應用於 SQRD-M 演算法中的演算法)做比較。使用的系統參數如下：模擬軟體為 Matlab，模擬次數為 10^7 次，在四根傳送天線 $N_T=4$ 及四根接收天線 $N_R=4$ 的 V-BLAST 系統架構下，調變分別使用 4-QAM、16-QAM 及 64-QAM 來做比較。接收端與傳送端間的通道使用 Rayleigh fading channel，且為 slow fading，即在一個訊框大小 (frame size) 的時間內，通道不會改變，但相鄰兩訊框的通道彼此是獨立的；因為 V-BLAST 的系統架構並未包含編碼器及解碼器，所以此模擬中的訊框大小設為一個調變訊號的位元數大小，例如 4-QAM 的訊框大小為二位元、16-QAM 的訊框大小為四位元。此章節各張模擬結果圖的橫坐標為訊雜比 (SNR)，其定義為傳送訊號的總能量與雜訊能量的比值，如(29)式，其中 P 為每根天線傳送訊號的平均能量， σ^2 為雜訊能量。

$$SNR(dB) \equiv 10 \times \log_{10} \left(\frac{N_T \times P}{2\sigma^2} \right) \quad (29)$$

圖 5.1 為第一章提到為了解決 ML 估測因全域搜尋所帶來的高複雜度而提出的幾種低複雜度之演算法，並且在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較圖。由圖可知，OSIC-ZF 演算法與 OSIC-MMSE 演算法雖然在演過法過程中的運算量非常低，但是在錯誤率的表現上並不理想，與 ML 估測曲線的差距仍然很大；以 OSIC-MMSE 演算法作估測的錯誤率雖然比 OSIC-ZF 演算法低，但其演算法過程中除了與 OSIC-ZF 演算法相同需要計算通道矩陣的反矩陣外，還必須把雜訊能量也列入考慮，額外的估測雜訊能量也讓複雜度更高，所以 OSIC-MMSE 演算法之複雜度相對的也比 OSIC-ZF 演算法高。在複雜度與估測錯誤率間取得較好的平衡為 SGA 演算法、SQRD-M 演算法與 SBSIDE 演算法，但 SGA 演算法必須算出通道矩陣的反矩陣，而反矩陣的運算量比 QR 分解的運算量高很多，且演算法過程中會讓每根天線間彼此獨立的白色高斯雜訊變成非獨立的雜

訊，所以接收端必須付出額外的複雜度來做估測；雖然在錯誤率逼近 ML 估測之前，如 M 值等於二時，SGA 演算法的錯誤率會比 SQRD-M 演算法低很多，但其計算通道矩陣的反矩陣之運算量卻高很多，且 SQRD-M 演算法在 M 值等於四時與 SGA 演算法相同，仍然可以讓錯誤率逼近 ML 估測，所以這也是本篇論文所提出的演算法皆以 QR 分解的低複雜度為基礎，而不使用計算反矩陣的高複雜度當成演算法的基礎。

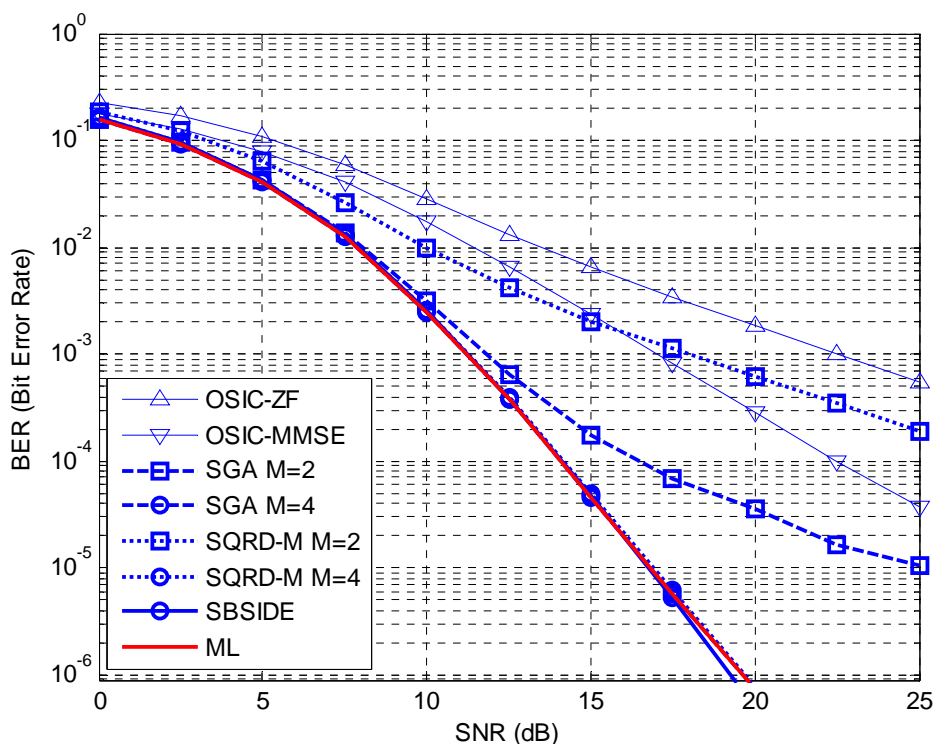


圖 5.1 OSIC-ZF 演算法、OSIC-MMSE 演算法、SGA 演算法、SQRD-M 演算法與 SBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較

圖 5.2 為 SQRD-M 演算法與 SBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較圖。由第三章的 SBSIDE 演算法得到之結論，我們限制演算法過程中被保留下來的部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 4$ 。從圖 5.2 可知，SQRD-M 演算法在 M 值等於四時及 SBSIDE 演算法的錯誤率都可以逼近使用 ML 估測的最佳錯誤率，且複雜度比 ML 估測低很多。SQRD-M 演算法的 M 值大概等於調變訊號集合的大小時，錯誤率即可逼近 ML 估測曲線，此圖是使用 4-QAM 調變的模擬結果，所以 M 值等於四的錯誤率即可逼近 ML 估測曲線，

也驗證了 SBSIDE 演算法過程中被保留下來的部分天線之傳送訊號組合個數 E 最多不超過四個時，即可讓錯誤率逼近 ML 估測。另外，SQRD-M 演算法的 M 值越高，複雜度也會越高，所以由此圖可以明顯的看出複雜度與錯誤率間彼此的平衡關係，並且可以知道 SBSIDE 演算法在此關係中取得較好的平衡。

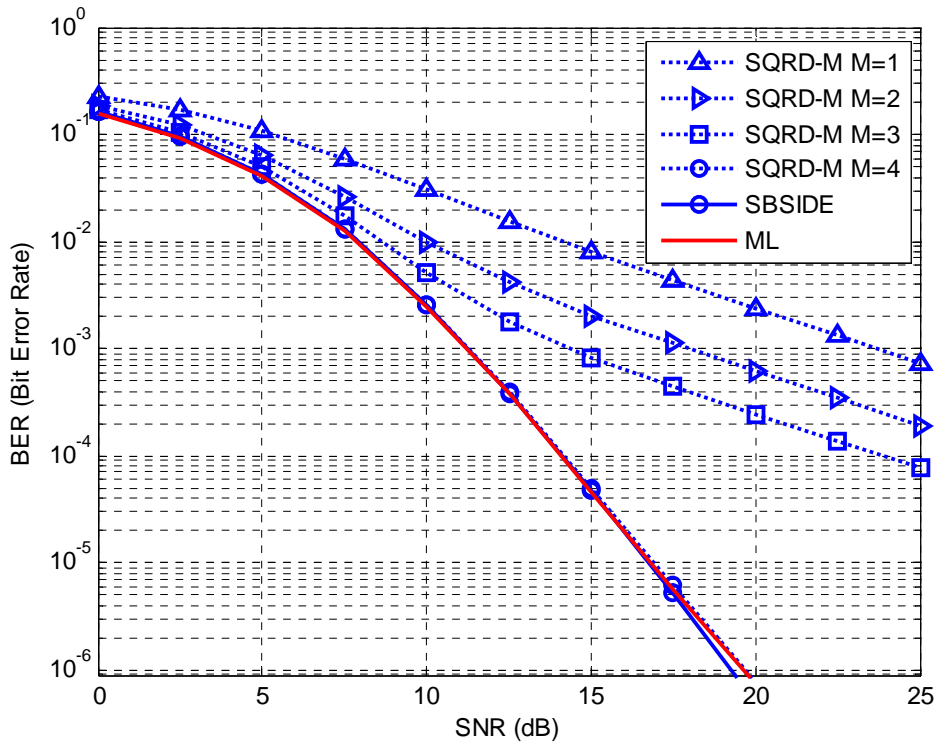


圖 5.2 SQRD-M 演算法與 SBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 4-QAM 下的錯誤率比較

圖 5.3 為 SQRD-M 演算法、SBSIDE 演算法、MSQRD-M 演算法和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的錯誤率比較圖。其中，我們限制 SBSIDE 演算法及 MSBSIDE 演算法過程中被保留下來的部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 16$ 。與圖 5.2 得到相同的結果，SQRD-M 演算法的 M 值大概等於調變訊號集合的大小及 SBSIDE 演算法的 E 值在我們的限制下時，錯誤率都可逼近 ML 估測曲線；並且由圖可知，SQRD-M 演算法及 SBSIDE 演算法加入 N-QAM 的多層架構搜尋特性後，錯誤率並不會變差，與未使用 N-QAM 的多層架構搜尋時的錯誤率相同，主要的差別在於複雜度，如圖 5.5。除此之外，此圖中的 MSBSIDE 演算法根據第 l 層的初步估計門檻 d_l 之計算方法不同，其中

$l=2 \cdots L$ ，如本篇論文附錄 4.5.2 提供的方法，分別命名為 MSBSIDE-MAX 演算法及 MSBSIDE-MEAN 演算法，這兩種演算法在錯誤率的表現上是相同的，皆可逼近 ML 估測曲線，且在調變使用 64-QAM 下也會得到相同的結果，如圖 5.4。

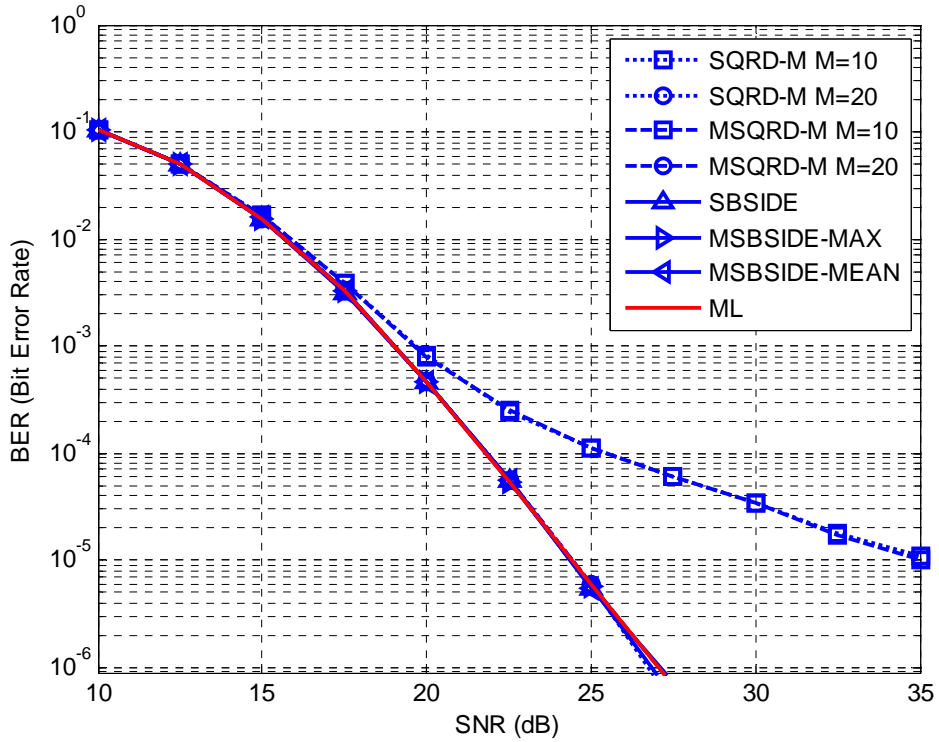


圖 5.3 SQRD-M 演算法、SBSIDE 演算法、MSQRD-M 演算法和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的錯誤率比較

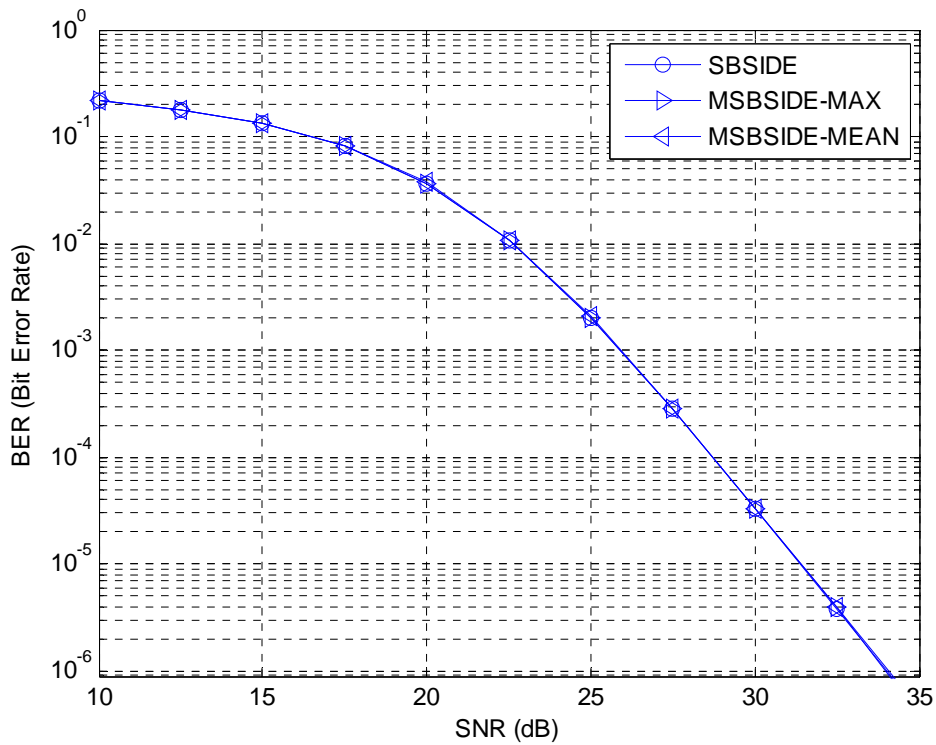


圖 5.4 SBSIDE 演算法、MSBSIDE-MAX 演算法和 MSBSIDE-MEAN 演算法在 4×4 的 V-BLAST 系統、調變使用 64-QAM 下的錯誤率比較

圖 5.5 為 SQRD-M 演算法、SBSIDE 演算法、MSQRD-M 演算法和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的複雜度比較圖。為了公平比較，讓這些演算法都在錯誤率逼近 ML 估測的情況下比較複雜度，也就是讓 SQRD-M 演算法及 MSQRD-M 演算法的 M 值等於十六，並限制 SBSIDE 演算法及 MSBSIDE 演算法過程中被保留下來的部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 16$ 。此圖中的 Y 坐標軸為估測出四根傳送天線的傳送訊號後，總共計算或比較價值函數的平均個數，因為此類演算法的運算量及複雜度主要是耗費在計算與比較價值函數上，所以這裡討論的複雜度並不包含 SBSIDE 演算法及 MSBSIDE 演算法過程中計算及更新初步估計解與各層初步估計門檻的複雜度、MSQRD-M 演算法及 MSBSIDE 演算法過程中計算這些假想訊號集合及找到對應之子代訊號的複雜度。由此圖可知，MSQRD-M 演算法總共計算或比較價值函數的平均個數大約是 SQRD-M 演算法的一半，而對越高階的調變所節省的複雜度會越多，且此兩種演算法的複雜度都是固定值，不隨訊雜比改變。MSBSIDE-MAX 演算法及 MSBSIDE-MEAN 演算法在總共計算或比較價值函數的平均個數上與 SBSIDE 演

算法相比較後，複雜度的下降幅度並沒有像 MSQRD-M 演算法與 SQRD-M 演算法相比較那麼多，但整體複雜度還是比 MSQRD-M 演算法低很多。SBSIDE 演算法和 MSBSIDE 演算法的複雜度並非固定值，當訊雜比低時，以決定迴授等化估測演算法估算出的初步估計解並不準確，導致各層初步估計門檻的值較大，經過篩選後會有較多的價值函數被保留下來，因此，在估測下一根天線時就必須計算與比較較多的價值函數，所以複雜度會較高；相反的，當訊雜比高時，估測出的初步估計解較準確，各層初步估計門檻的值會較小，經過篩選後只有些許價值函數被保留下來，所以複雜度較低。除此之外，由圖 5.5 也可以得知 MSBSIDE-MAX 演算法與 MSBSIDE-MEAN 演算法在 E 值為 $1 \leq E \leq 16$ 的限制下，雖然錯誤率都可以逼近使用 ML 估測的最佳曲線，但在複雜度的表現上，MSBSIDE-MEAN 演算法卻比 MSBSIDE-MAX 演算法低，這也表示 MSBSIDE-MEAN 演算法在複雜度與錯誤率的關係中取得較好的平衡。

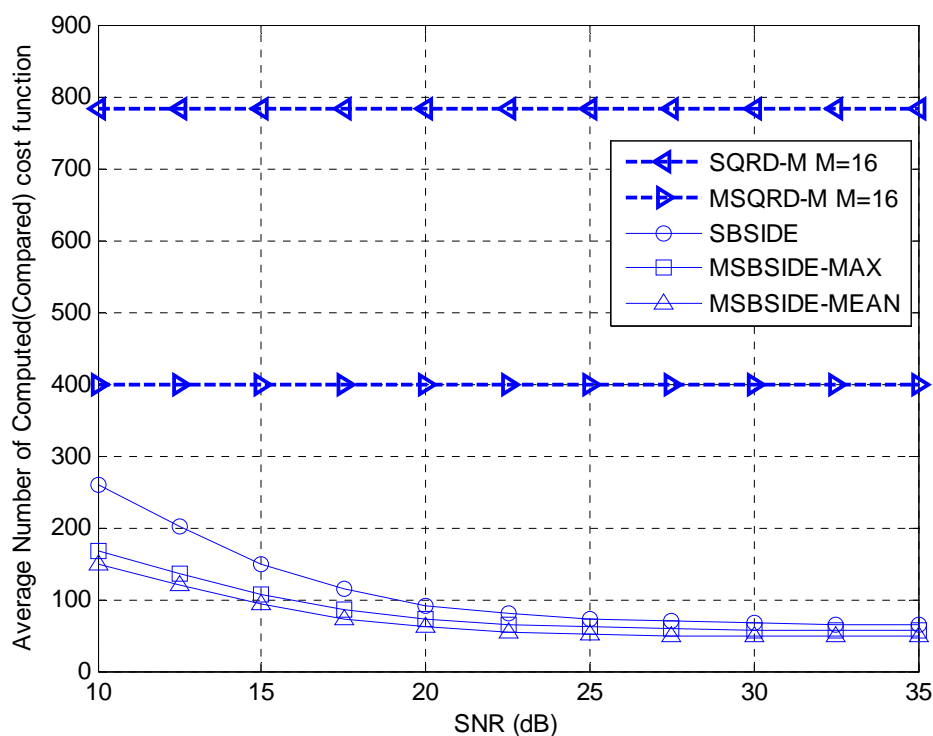


圖 5.5 SQRD-M 演算法($M=16$)、SBSIDE 演算法、MSQRD-M 演算法($M=16$)和 MSBSIDE 演算法在 4×4 的 V-BLAST 系統、調變使用 16-QAM 下的複雜度比較

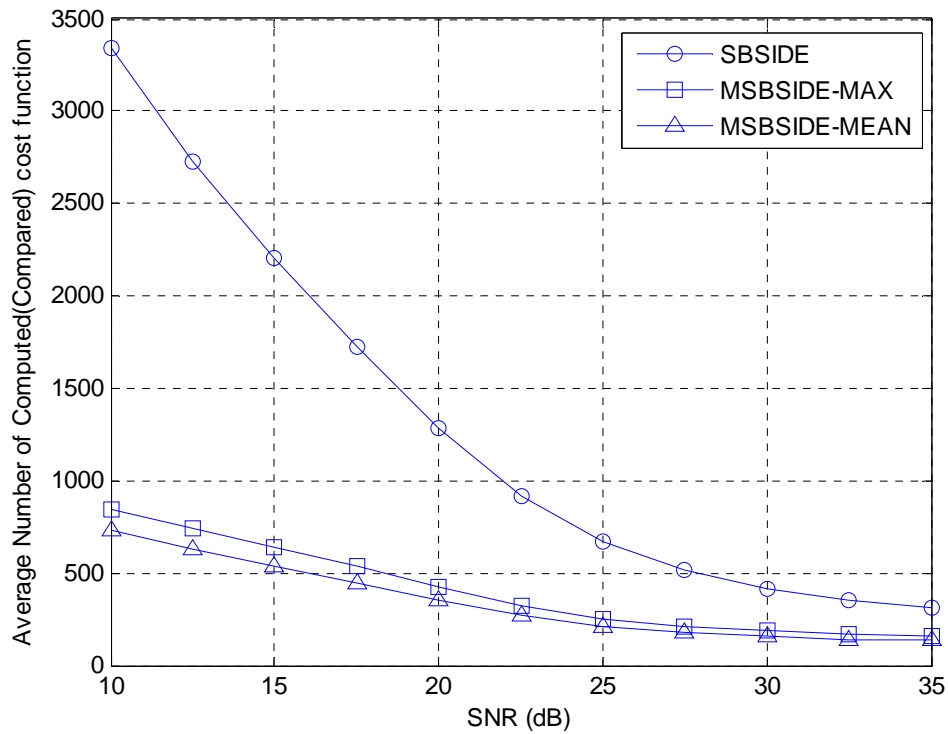


圖 5.6 SBSIDE 演算法、MSBSIDE-MAX 演算法和 MSBSIDE-MEAN 演算法在 4×4 的 V-BLAST 系統、調變使用 64-QAM 下的複雜度比較

圖 5.6 為 SBSIDE 演算法、MSBSIDE-MAX 演算法和 MSBSIDE-MEAN 演算法在 4×4 的 V-BLAST 系統、調變使用 64-QAM 下的複雜度比較圖。此圖中限制 SBSIDE 演算法與 MSBSIDE 演算法過程中被保留下來的部分天線之傳送訊號組合個數 E 為 $1 \leq E \leq 64$ 。與圖 5.5 比較可知，使用 N-QAM 的多層架構搜尋特性於越高階的調變所節省的複雜度會越明顯。除此之外，由圖 5.5 及圖 5.6 可以得知，雖然 SBSIDE 演算法的複雜度並非固定值，且高訊雜比與低訊雜比間複雜度的差距很大，但使用 N-QAM 的多層架構搜尋特性於 SBSIDE 演算法上卻能減少高訊雜比與低訊雜比間複雜度的差距，此種現象在越高階的調變會越明顯。

六、結論

隨著科技日新月異，我們對無線通訊的傳輸品質與傳輸速度越來越講究，除了傳輸品質與速度外，演算法與電路設計的複雜度也必須在可接受的範圍內。而本篇論文結合了 B-SIDE 演算法的適應性調整門檻來限制搜尋路徑個數並降低複雜度的特性、排序的 QR 分解之特定解碼順序來增加估測訊號之正確性與 N-QAM 的多層架構搜尋特性而衍生出 MSB-SIDE 演算法，來達到降低複雜度的目的，且讓估測錯誤率逼近 ML 估測。除此之外，本篇論文根據更新各層初步估計門檻方法的不同，將 MSB-SIDE 演算法分為 MSB-SIDE-MAX 演算法與 MSB-SIDE-MEAN 演算法，這兩種演算法與 B-SIDE 演算法、SQRD-M 演算法及 MSQRD-M 演算法相同，錯誤率都可以逼近 ML 估測的最佳曲線，而且此兩種演算法的低複雜度特性更適合用於高階的調變，如 16-QAM、64-QAM，來增加無線通訊的傳輸品質與傳輸速度。

經由電腦模擬驗證，此兩種演算法確實可以用很低的複雜度讓解碼錯誤率逼近於 ML 估測，且當訊雜比越高或使用越高階的調變時，越能顯現出此兩種演算法帶來的好處。MSB-SIDE 演算法仍然有改進的地方，因為此演算法過程中所算出的各層初步估計門檻都是動態改變值，導致被保留下來的部分天線之傳送訊號組合個數 E 並非固定值，讓複雜度隨著訊雜比的改變而變動；雖然加入 N-QAM 的多層架構搜尋特性後可以緩和 high SNR 與 low SNR 間複雜度的差距，但複雜度與訊雜比間並非獨立關係，變動的複雜度對電路設計會較困難；所以，如何保留 MSB-SIDE 演算法降低複雜度與錯誤率逼近 ML 估測的優點，且設定一個複雜度門檻來讓整體演算法的複雜度保持固定值是此演算法可以改進的地方。除此之外，本論文討論的演算法都是操作在 V-BLAST 系統架構下，主要著重於系統的多工程度 (multiplexing) 上；而在分層時空碼的眾多架構中，有些架構可以用來增加系統的分集程度 (diversity)，可以幫助接收端的估測與解碼；所以，如何在系統的多工程度與分集程度上取得最佳的平衡後，再配合編碼器與解碼器得到最佳的系統架構，並探討 MSB-SIDE 演算法應用於此種系統架構下的效能增益也是未來的研究方向。

參考文獻

- [1] G.J. Foschini, "Layered space-time architectures for wireless communications in a fading environment when using multi-element antennas," *Bell-Labs Tech. Journal*, vol. 1, no. 2, pp. 41-59, 1996.
- [2] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, pp. 311-335, Mar. 1998.
- [3] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel." in *Proc. URSI Int. Symp. Signals, Systems, Electronics*, Pisa, Italy, Sept.-Oct. 1998, pp. 295-300.
- [4] K.L. Lo, S. Marinkovic, Z. Chen, and B. Vucetic, "Performance Comparison of Layered Space Time Codes", in *Proc. IEEE Int. Conf. Commun.*, New York, USA, pp. 1382-1387, 2002.
- [5] Chin, W.H. "QRD based tree search data detection for MIMO communication systems." *Vehicular Technology Conference*, 2005. VTC 2005-Spring. 2005 IEEE 61st.
- [6] D. Wubben, R. Bohnke, V. Kuhn, K.D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition." in *Proc. IEEE VTC*, vol. 1, pp. 508-512, Orlando, USA, Oct. 2003.
- [7] S. Rho, J. Cha, J. Kang "Stack-based tree searching algorithm for V-BLAST detection with implementation friendly reduction techniques." *IEEE Conference*, PIMRC.2007.
- [8] Y. Jia, C. Andrieu, R. J. Piechocki, and M. Sandell, "Gaussian approximation based mixture reduction for near optimal detection for MIMO system," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 997-999, Nov 2005.
- [9] Y. Su, Xian-Da Zhang, X. Wang, "A Successive Interference Cancellation Algorithm

in MIMO Systems via Breadth-First Search.” *IEEE International Conference*, March 31 2008-April 4 2008.

- [10] Kim, B.-S. and K. Choi, “A very low complexity QRD-M algorithm based on limited tree search for MIMO systems,” *Vehicular Technology Conference*, 1246–1250, 2008.
- [11] Y. Jia, C. Andrieu, R. J. Piechocki, and M. Sandell, “Depth-first and Breadth-first search based multilevel SGA algorithms for near optimal symbol detection in MIMO systems.” *IEEE Transactions on Wireless Comm.*, vol. 7, no. 3, March 2008.
- [12] H. G. Kang, I. Song, J. Oh, J. Lee, S. Yoon, “Breadth-First Signal Decoder: A Novel Maximum-Likelihood Scheme for Multi-Input–Multi-Output Systems.” *IEEE Trans on Vehicular Technology*, Volume 57, Issue 3, 2008 pp. 1576–1584.
- [13] Branka Vucetic, Jinhong Yuan, *Space time coding*, WILEY, 2003.

