

國 立 交 通 大 學

電 信 工 程 學 系

碩 士 論 文

高階調變多輸入多輸出正交分頻多工系統中
基於機率資料聯結演算法之資料檢測方法

Data Detection Methods Based on Probabilistic Data
Association Algorithm for High Order Modulation
MIMO-OFDM Systems

研 究 生：王 冠 群

指 導 教 授：黃 家 齊 博 士

中 華 民 國 九 十 八 年 六 月

高階調變多輸入多輸出正交分頻多工系統中 基於機率資料聯結演算法之資料檢測方法

Data Detection Methods Based on Probabilistic Data Association
Algorithms for High Order Modulation MIMO-OFDM Systems

研究 生：王冠群

Student : Kuan-Chun Wang

指導 教授：黃家齊 博士

Advisor : Dr. Chia-Chi Huang



Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science
in

Communication Engineering

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

高階調變多輸入多輸出正交分頻多工系統中 基於機率資料聯結演算法之資料檢測方法

學生：王冠群

指導教授：黃家齊博士

國立交通大學電信工程學系 碩士班

摘要

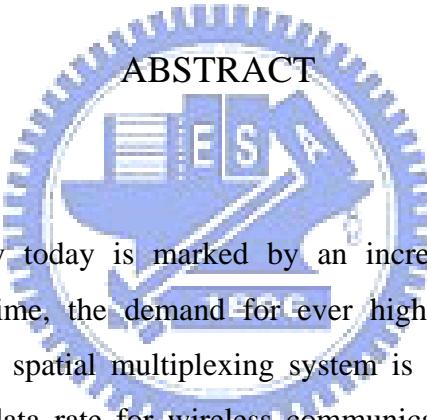
今天我們的資訊社會的特點是變得越來越需要移動性以及無障礙性。同時，對於高傳輸率的需求也大大的增加。多輸入多輸出正交分頻多工(MIMO-OFDM)空間多工系統是一種可行的技術，它可以用來改進無線通訊訊號的品質以及傳輸的速率。在 MIMO-OFDM 空間多工系統的資料檢測中，球狀解碼是一種被廣泛使用的次佳解檢測器。它擁有非常好的效能，但是它的複雜度會隨著通道和訊雜比而改變。在這篇論文中，我們證明了 SSIC 演算法與 PDA 演算法是等效的。此外，我們提出了兩種檢測器，GPDA-MCPDA 與 GPDA-SD，用來降低球狀解碼在接近最佳解的 MIMO-OFDM 空間多工系統下使用高階 QAM 調變(16QAM/ 64QAM)的複雜度。模擬結果顯示，這兩種檢測器都可以達到接近最佳解的效能而且它們的優勢在於擁有比球狀解碼還低的複雜度，特別是在低訊雜比的區域。

Data Detection Methods Based on Probabilistic Data Association Algorithms for High Order Modulation MIMO-OFDM System

Student : Kuan-Chun Wang

Advisors : Dr. Chia-Chi Huang

Department of Communication Engineering
National Chiao Tung University



Our information society today is marked by an increasing need for mobility and accessibility. At the same time, the demand for ever higher data transfer rates is also increasing. A MIMO-OFDM spatial multiplexing system is a proven technology that can improve signal quality and data rate for wireless communications. Sphere Decoding (SD) algorithm is a popular suboptimal data detection method in a MIMO-OFDM spatial multiplexing system. It although performs very well but suffers from the fact that its complexity is a random variable depending on channel and signal to noise ratio (SNR). In this thesis, we establish the equivalence of the SSIC algorithm and the PDA algorithm. Furthermore, we proposed two detectors, GPDA-MCPDA and GPDA-SD, to reduce the complexity of the sphere decoding algorithm for near-optimal detection in a MIMO-OFDM spatial multiplexing system with higher order QAM constellations (16QAM/64QAM). Simulation results demonstrate that both detectors can achieve near-optimal performance with lower complexity as compare with the sphere decoding algorithm, especially at the low SNR regions.

誌 謝

首先感謝指導教授 黃家齊教授對於我研究及生活上的指導與勉勵，以及對於論文內容的建議，使我得以順利完成碩士學位。同時也感謝 吳文榕教授、 陳紹基教授以及 高銘盛教授給予本論文的指教，口試時也提供了我許多的意見，使得論文更加完善。

特別感謝 古孟霖學長與 陸裕威學長，在我研究中所給予的指導，與學長們的討論常使我獲益良多，使我的觀念更加地茁壯。感謝實驗室的同學，奕廷、人豪、曉顥以及王森這兩年來的照顧與陪伴。感謝無線通訊實驗室的所有成員與書舍的陪伴，使我的生活充滿了歡樂與溫馨。

最後，感謝我的家人，給予我精神上的鼓勵與支持。特別要感謝姊姊，在我寫論文的時候幫了我許多忙。



Contents

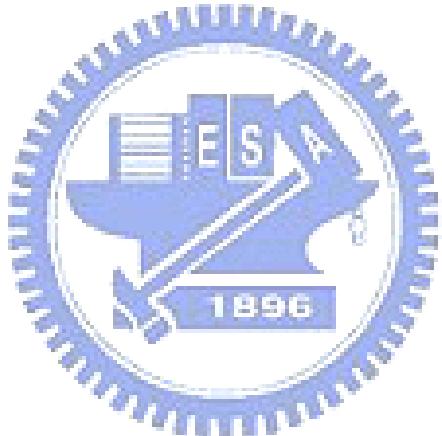
中文摘要	i
ABSTRACT	ii
誌謝	iii
Contents	iv
List of Tables	vi
List of Figures	vii
Chapter 1 Introduction	1
Chapter 2 MIMO-OFDM System	4
2.1 Overview of MIMO System	4
2.2 Overview of OFDM System	5
2.3 MIMO-OFDM System	5
2.4 Detection Schemes on the Receiving Side	7
Chapter 3 Probabilistic Data Association Based Detectors	9
3.1 Probabilistic Data Association Detector	9
3.1.1 Basic Algorithm	10
3.1.2 Computational Refinements	13
3.2 Generalized PDA Detector	14
3.3 Simulation Results	16
Chapter 4 Data Detection in MIMO-OFDM System Based on GPDA Detector	29
4.1 GPDA-MCPDA Detector	29
4.1.1 Markov Chain Monte Carlo Method	29
4.1.1.1 Monte Carlo Integration	30
4.1.1.2 Importance Sampling	31
4.1.1.3 Introduction to Markov Chains	31
4.1.1.4 Gibbs Sampler	35
4.1.2 GPDA-MCPDA Detector	39

4.2	GPDA-SD Detector	41
4.2.1	Sphere Decoding	41
4.2.1.1	Real Sphere Decoding	42
4.2.1.2	Complex Sphere Decoding	46
4.2.2	GPDA-SD Detector	47
4.3	Simulation Results	48
Chapter 5 On the Equivalence between the SSIC Algorithm and the PDA Algorithm		61
5.1	The SSIC Algorithm	61
5.2	The Equivalence of the SSIC Algorithm and the PDA Algorithm	65
5.3	Simulation Results	66
Chapter 6 Conclusion		69
Bibliography		70



List of Tables

Table 3.1 Simulation parameters for comparing PDA with GPDA.	17
Table 4.1 Simulation parameters for MIMO-OFDM system.	48
Table 5.1 Simulation parameters for SSIC algorithm and PDA algorithm.	66



List of Figures

Fig. 2.1 MIMO-OFDM spatial multiplexing system.....	6
Fig. 3.1 Block diagram of the basic PDA procedure.....	12
Fig. 3.2 The BER performance for PDA and GPDA with $q = 4$	20
Fig. 3.3 The BER performance for PDA and GPDA with $q = 16$	21
Fig. 3.4 The effect of sorting for PDA and GPDA with $q = 4$	22
Fig. 3.5 The number of iterations for PDA and GPDA with $q = 4$	23
Fig. 3.6 The number of iterations for PDA and GPDA with $q = 16$	24
Fig. 3.7 The complexity for PDA and GPDA with $q = 4$	25
Fig. 3.8 The complexity for PDA and GPDA with $q = 16$	26
Fig. 3.9 The complexity for PDA and GPDA with different modulation order.....	27
Fig. 3.10 The noise variance estimation error for GPDA and PDA.....	28
Fig. 4.1 Block diagram of the GPDA-MCPDA detector.....	40
Fig. 4.2 The discrepancy between the PDA detector and the MCPDA detector.....	41
Fig. 4.3 A sphere of radius d and centered at \mathbf{r}	42
Fig. 4.4 Search tree of sphere decoding.....	44
Fig. 4.5 Block diagram of the GPDA-SD detector.....	48
Fig. 4.6 The BER performance for MCMC detector with different initial point	52
Fig. 4.7 The BER performance for MCPDA detector with different initial point	53
Fig. 4.8 The BER performance for MCPDA detector with different iterations	54
Fig. 4.9 The BER performance for parallel MCPDA with different combinative iterations ..	55
Fig. 4.10 The BER performance for different detection scheme with $q = 16$	56
Fig. 4.11 The BER performance for different detection scheme with $q = 64$	57
Fig. 4.12 The complexity for different detection scheme with $q = 16$	58
Fig. 4.13 The complexity for different detection scheme with $q = 64$	59
Fig. 4.14 The channel estimation error for different detection scheme.....	60
Fig. 5.1 Block diagram of the basic SSIC procedure.	64
Fig. 5.2 The BER performance for the SSIC algorithm and the PDA algorithm.....	67
Fig. 5.3 The complexity for the SSIC algorithm and the PDA algorithm.....	68

Chapter 1

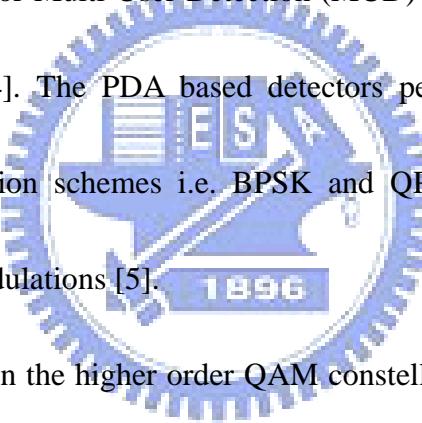
Introduction

The Multiple-input multiple-output orthogonal frequency division multiplexing (MIMO-OFDM) spatial multiplexing system [1] [2] is a proven technology that can improve signal quality and data rate for wireless communications. The global growth of interest in the wireless Internet and digitized audio and video, coupled with a growing wireless high-bandwidth infrastructure leads to a rapidly expanding market for wireless multimedia services. To cope with the growth, wireless services providers today are facing a number of challenges, which include the limited availability of the radio frequency spectrum and a complex time-varying wireless environment. In near future, wireless devices will have the features for always-available connection, higher data rate, longer distance, low-power consumption, better Quality of Service (QoS), fewer dropped packets, and higher network capacity in order to improve service quality for rapid service expansion and configurations.

Combining OFDM with MIMO equipment is the most promising technology for delivering high data rates and robust performance.

In a MIMO-OFDM spatial multiplexing system, the optimum solution is to use Maximum Likelihood (ML) detection. However, ML detection is an exhaustive search; the complexity increases with either the increasing number of transmitting antennas or the

increasing order of modulation. Computational efficient data detection algorithms have been widely explored to achieve the substantial performance gains promised by MIMO-OFDM spatial multiplexing system with QAM constellations. There are two outstanding detectors of those frontrunners, the Sphere Decoding (SD) algorithm [3] and algorithm which based on the Probabilistic Data Association (PDA) [4] [5] principle. The SD algorithm although perform very well but suffers from the fact that its complexity is a random variable depending on channel and signal to noise ratio (SNR). The probabilistic data association is a simpler detection method, originally for Multi-User Detection (MUD) in synchronous Code Division Multiple Access (CDMA) [4]. The PDA based detectors perform well (close to the ML decoder) for simple modulation schemes i.e. BPSK and QPSK, but these results do not emerge from higher order modulations [5].



In this thesis, we focus on the higher order QAM constellations (16QAM/64QAM) data detection in the MIMO-OFDM spatial multiplexing system. We propose two near-optimal performance detectors with the common features which is using the Generalized PDA (GPDA) [5] detector at the low SNR regions. At the high SNR regions, we will combine different detectors to improve the performance. The first proposed detector is combining GPDA and Markov Chain Monte Carlo (MCMC) [6], and it is named MCPDA which incorporates the concept of PDA to calculate the covariance then construct a Markov Chain to make it converge to the target distribution. The second one is based on the SD algorithm using GPDA

solution to be the radius of the sphere. In addition, we prove the Soft Successive Interference Cancellation (SSIC) algorithm and the Probabilistic Data Association (PDA) algorithm are equivalent in the MIMO-OFDM system.

This thesis is organized as follows. In Chapter 2, we describe a MIMO-OFDM system. Chapter 3 introduces the PDA detector and the GPDA detector. In Chapter 4, we propose two modified methods (GPDA-MCPDA, GPDA-SD) of data detection in the MIMO-OFDM spatial multiplexing system. In Chapter 5, we establish the equivalence of the SSIC algorithm and the PDA algorithm in the MIMO-OFDM system. Finally, some conclusions are drawn in Chapter 6.

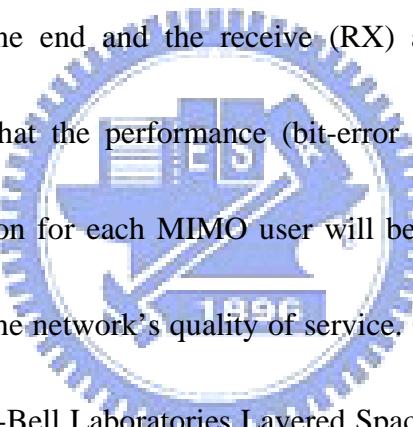


Chapter 2

MIMO-OFDM System

2.1 Overview of MIMO System

A MIMO system can be defined simply. Given an arbitrary wireless communication system, we consider a link for which the transmitting end as well as the receiving end is equipped with multiple antenna elements. The idea behind MIMO is that the signals on the transmit (TX) antennas at one end and the receive (RX) antennas at the other end are “combined” in such a way that the performance (bit-error rate or BER) or the data rate (bits/sec) of the communication for each MIMO user will be improved. Such an advantage can be used to increase both the network’s quality of service. One popular example of such a system is V-BLAST (Vertical-Bell Laboratories Layered Space-Time) suggested by Foschini et al [7].

The logo is circular with a blue and white color scheme. It features a stylized 'B' and 'L' intertwined in the center, with 'BELL' written vertically above them. The year '1898' is at the bottom, and '1998' is at the top. The outer ring contains the text 'BELL' and '100' on the left, and 'YEARS' and '1998' on the right.

MIMO techniques can be basically split into two groups: space time coding (STC) and spatial multiplexing (SM). STC increases the performance of the communication system by coding over the different transmitter branches; whereas SM achieves a higher throughput by transmitting independent data streams on the different transmit branches simultaneously at the same carrier frequency. Since increasing the bit rates is our goal, we will focus on the SM algorithms in this thesis. A potential application of the MIMO principle is the next-generation

wireless local area network (WLAN).

2.2 Overview of OFDM System

OFDM, which was brought up in the mid 60's, is a digital multi-carrier modulation scheme. OFDM is used in numerous wireless transmission standards nowadays (DAB, DVB-T, WiMAX IEEE 802.16, ADSL, WLAN IEEE 802.11a/g), as a result of its capability of high-rate transmission and low-complexity implementation over frequency-selective fading channels.

The basic idea of OFDM is that it divides the available spectrum into several orthogonal subcarriers. Because these subcarriers are narrow-band, they experience flat fading channel and then equalization method of the system becomes very simple. Furthermore, it possesses high spectral efficiency by overlapping these orthogonal subcarriers [8]. Moreover, the insertions of the cyclic prefix (CP), which preserves the periodic extensions of the transmitted signal, can eliminate inter-symbol and inter-carrier interference caused by multipath environments.

2.3 MIMO-OFDM System

The combination of the throughput enhancement of MIMO with the robustness of OFDM against frequency-selective fading caused by severe multipath scattering and

narrowband interference is regarded as a very promising basis for future high data-rate radio communication systems. On the other hand, especially in multiplexing system, interference in MIMO-OFDM is severer than in single input single output (SISO) OFDM system and the complexity of data detection in a MIMO-OFDM system is higher than the complexity in SISO OFDM system.

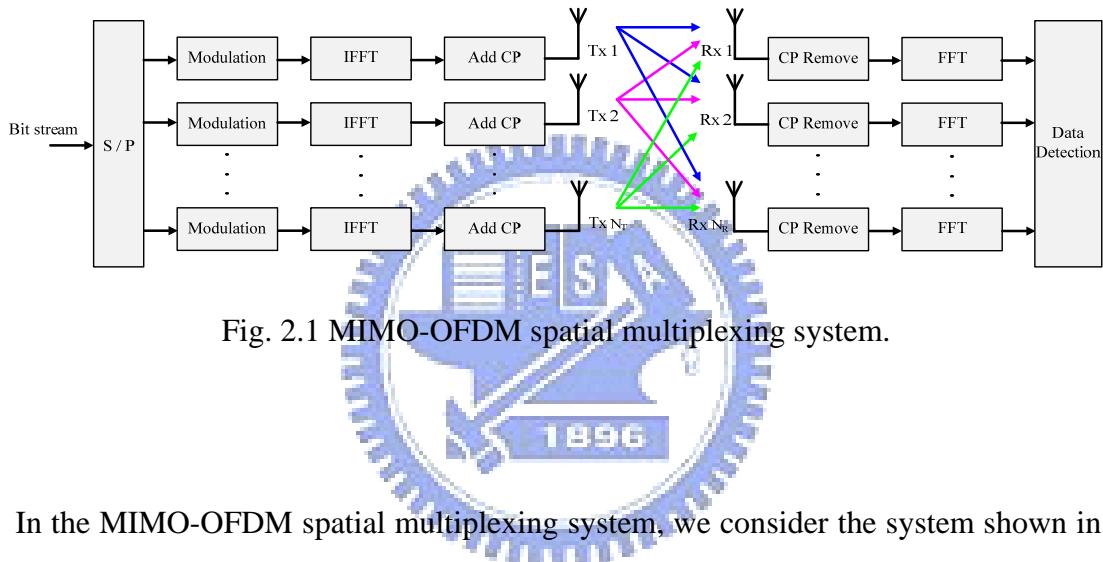


Fig. 2.1 MIMO-OFDM spatial multiplexing system.

In the MIMO-OFDM spatial multiplexing system, we consider the system shown in Fig. 2.1. We assume that there are N_T transmitting antennas and N_R receiving antennas. At the transmitter side, bit stream is divided into N_T data layers and mapped each data layer to be N_T modulated signal streams. N_T modulated signal streams in N_T layer pass through IFFT, add cyclic prefix and then transmit parallel through N_T transmitting antennas. At the receiver side, there are N_R receiving antennas. After cyclic prefix removal and FFT, the received signal vector \mathbf{r} can be expressed as

$$\mathbf{r} = \mathbf{H}\mathbf{a} + \mathbf{v} \quad (2.1)$$

where \mathbf{H} is a N_R by N_T channel matrix, \mathbf{a} is a N_T by 1 transmit vector of symbols, satisfying that each symbol in the constellation is transmitted with equal probability, and \mathbf{v} is an N_R by 1 complex-valued white Gaussian noise vector with zero mean and covariance matrix equal to $\sigma^2 \mathbf{I}$. By assuming a rich scattering model, the elements of the channel matrix \mathbf{H} are independent and identically distributed (i.i.d.) complex Gaussian with zero mean.

2.4 Detection Schemes on the Receiving Side

The complexity of the Maximum Likelihood (ML) detector grows exponentially with the number of transmitting antennas and the size of the signal constellation. This motivates the use of simpler suboptimum detectors in practical applications. Among those are:

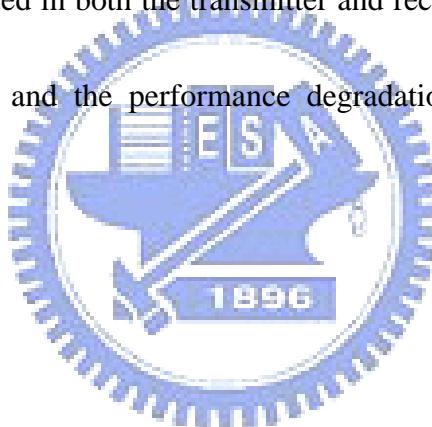
- Zero Forcing (ZF) detectors, which invert the channel matrix. The ZF receiver has a very small complexity that does not depend on the modulation. However, it does not completely exploit the system diversity and suffers from bad performance at low SNR.
- Minimum Mean Square Error (MMSE) detectors, which reduce the combined effect of interference between the parallel channels and the additive noise. The MMSE receiver slightly improves the performance of the ZF receiver, but it requires knowledge of the SNR. Besides, it does not completely exploit the channel diversity either.
- V-BLAST Ordered Successive Interference Cancellation (OSIC) [9], which exploits the timing synchronism inherent in the system model. Furthermore, linear nulling (i.e., ZF) or

MMSE is used to perform the detection. In other words, SIC is based on the subtraction of interference of already detected element of \mathbf{a} from the received signal vector \mathbf{r} . This results in a modified received vector in which effectively fewer interferers are present.

- SD [3] algorithm, which reduces the number of symbol values used in the ML detector.

Note that this type of detectors may preserve optimality while reducing implementation complexity.

Thus, if signal conditions are excellent, the data rate will be more than twice depending on the number of antennas used in both the transmitter and receiver. In that case, the channel matrix is better conditioned and the performance degradation of suboptimal detectors is reduced.

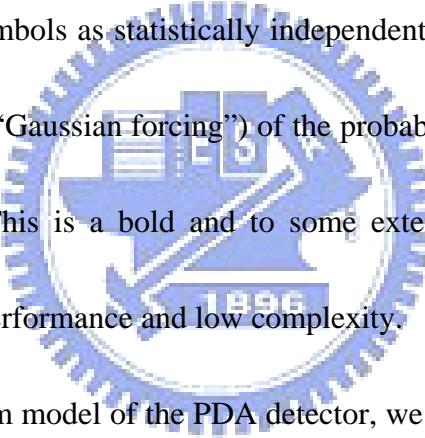


Chapter 3

Probabilistic Data Association Based Detectors

3.1 Probabilistic Data Association Detector

The Probabilistic Data Association (PDA) detector [4] is a highly successful approach to target tracking in the case that measurements are unlabeled and may be spurious. It is based on two approximations. Firstly, the PDA detector only looks at one transmitted symbol at a time, treating the received symbols as statistically independent. The second approximation is the Gaussian approximation (“Gaussian forcing”) of the probability density function (PDF) of the interference and noise. This is a bold and to some extent unjustifiable step, but it is difficult to argue with good performance and low complexity.



Now, to obtain the system model of the PDA detector, we multiply (2.1) from the left by \mathbf{H}^H to obtain

$$\mathbf{y} = \mathbf{G}\mathbf{a} + \mathbf{n} \quad (3.1)$$

where $\mathbf{y} = \mathbf{H}^H \mathbf{r}$, $\mathbf{G} = \mathbf{H}^H \mathbf{H}$, and $\mathbf{n} = \mathbf{H}^H \mathbf{v}$.

The model of the PDA detector is obtained by multiplying (3.1) from the left by \mathbf{G}^{-1} to yield

$$\tilde{\mathbf{y}} = \mathbf{a} + \tilde{\mathbf{n}} = \mathbf{e}_i a_i + \sum_{j \neq i} \mathbf{e}_j a_j + \tilde{\mathbf{n}} \quad (3.2)$$

In (3.2), $\tilde{\mathbf{y}} = \mathbf{G}^{-1} \mathbf{y}$, $\tilde{\mathbf{n}} = \mathbf{G}^{-1} \mathbf{n}$, and \mathbf{e}_i is a column vector whose i th element is 1 and the

other element are 0. In order to obtain computational efficiency, we choose (3.2) as the system model for the PDA algorithm.

3.1.1 Basic Algorithm

In the reformulated MIMO-OFDM spatial multiplexing system model (3.2), we treat the element of \mathbf{a} as independent multivariate random variable where the i th element, a_i , is a member of possible set:

$$a_i \in X_i = \{x_i(m)\}, i \in [1, N_T] \quad (3.3)$$

In (3.3), X_i is the set of distinct values of the QAM symbols. For any element a_i , we associate a vector \mathbf{p}_i whose m th element, $p_i(m)$, is the current estimate of the posterior that $a_i = x_i(m)$. Since direct evaluation of $\Pr(a_i = x_i(m) | \tilde{\mathbf{y}})$ is computationally prohibitive, the PDA algorithm attempts to estimate by using “Gaussian forcing” idea to approximate $\Pr(a_i = x_i(m) | \tilde{\mathbf{y}}, \{\mathbf{p}_j\}_{\forall j \neq i})$, which will serve as the update value for $p_i(m)$.

An important factor in the performance of the PDA algorithm is the order in which the probability vectors $\{\mathbf{p}_i\}_{\forall i}$ are updated. In this case, we use V-BLAST OSIC method [9] to order the update sequence according to SNR in descending order, which can detect the first signal that belongs to the highest order of SNR. It is useful to provide the reliable symbols from the high SNR, and we can use these reliable symbols to detect the other symbols to make the BER performance better.

To estimate the associated probabilities for an element a_i , we treat all other elements $a_j (j \neq i)$ as multivariate random variables, and from (3.2), we define

$$\mathbf{N}_i = \sum_{j \neq i} \mathbf{e}_j a_j + \tilde{\mathbf{n}} \quad (3.4)$$

as the effective noise on a_i , and approximate it as a Gaussian noise with matched mean and covariance:

$$\bar{\mathbf{N}}_i = \sum_{j \neq i} \mathbf{e}_j E[a_j] \quad (3.5)$$

$$\boldsymbol{\Omega}_i = \sum_{j \neq i} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j] + \sigma^2 \mathbf{G}^{-1} \quad (3.6)$$

where $\bar{\mathbf{N}}_i = E[\mathbf{N}_i]$ and $\boldsymbol{\Omega}_i = \text{Cov}[\mathbf{N}_i]$. In (3.5) and (3.6), $E[a_j]$ and $\text{Var}[a_j]$ are given by

$$E[a_j] = \sum_m x_j(m) p_j(m) \quad (3.7)$$

$$\text{Var}[a_j] = \sum_m x_j^2(m) p_j(m) - (E[a_j])^2 \quad (3.8)$$

Therefore, we can calculate $P(\tilde{\mathbf{y}} | a_i = x_i(m))$ as follows:

$$P(\tilde{\mathbf{y}} | a_i = x_i(m)) \propto \exp\left(-(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \bar{\mathbf{N}}_i)^H \boldsymbol{\Omega}_i^{-1} (\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \bar{\mathbf{N}}_i)\right) \quad (3.9)$$

Then, we let

$$\boldsymbol{\theta}_i = \tilde{\mathbf{y}} - \bar{\mathbf{N}}_i \quad (3.10)$$

and

$$\begin{aligned} \alpha_i(m) &= -(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \bar{\mathbf{N}}_i)^H \boldsymbol{\Omega}_i^{-1} (\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \bar{\mathbf{N}}_i) \\ &\propto (2\tilde{\mathbf{y}} - 2\bar{\mathbf{N}}_i - 2\mathbf{e}_i x_i(m))^H \boldsymbol{\Omega}_i^{-1} \mathbf{e}_i x_i(m) \\ &= (2\boldsymbol{\theta}_i - \mathbf{e}_i x_i(m))^H \boldsymbol{\Omega}_i^{-1} \mathbf{e}_i x_i(m) \\ &= \left(\sum_j [2\boldsymbol{\theta}_i^H]_j [\boldsymbol{\Omega}_i^{-1}]_{ji} - x_i^H(m) [\boldsymbol{\Omega}_i^{-1}]_{ii} \right) x_i(m) \end{aligned} \quad (3.11)$$

where $[2\boldsymbol{\theta}_i^H]_j$ is the j th element of $2\boldsymbol{\theta}_i^H$; $[\boldsymbol{\Omega}_i^{-1}]_{ji}$ is the element (j,i) of $\boldsymbol{\Omega}_i^{-1}$. The posterior

probability $P_i(m)$ is then given as

$$P_i(m) = \frac{\exp(\alpha_i(m))}{\sum_l \exp(\alpha_i(l))} \quad (3.12)$$

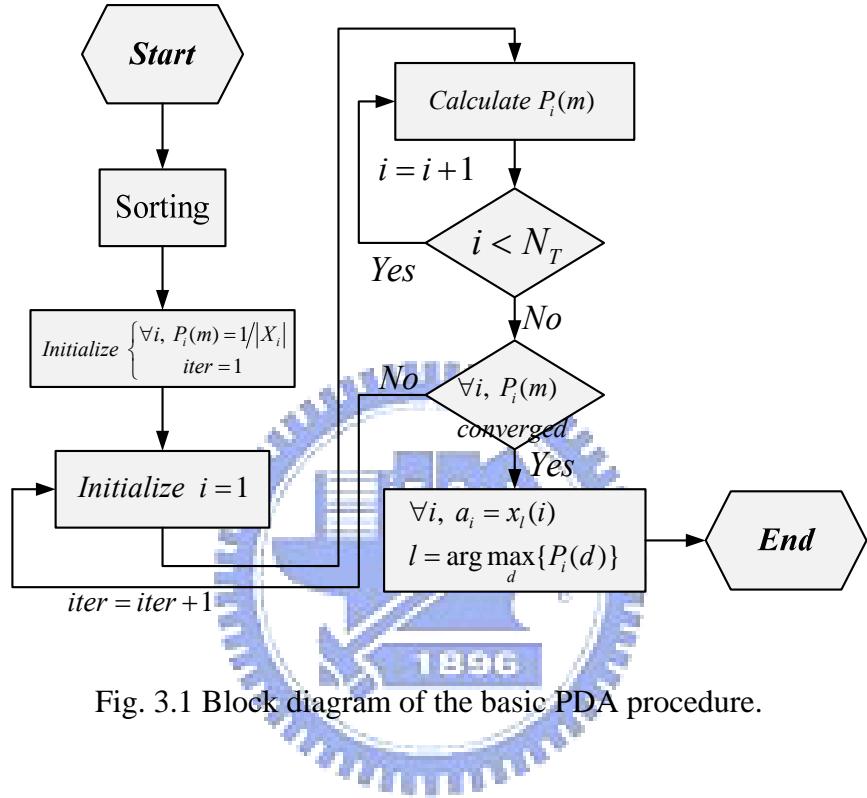


Fig. 3.1 Block diagram of the basic PDA procedure.

The basic procedure for the PDA detector is as follows:

1. Based on the matrix \mathbf{G} in (3.1), we obtain the optimal detection sequence proposed for the V-BLAST OSIC in [9] and denote the sequence as $\{k_i\}_{i=1}^{N_T}$.
2. Initialize the probabilities as $P_i(m) = 1/|X_i| \forall m, \forall i$, and set the iteration counter $iter = 1$.
3. Initialize $i = 1$.
4. Based on the current values of $\{\mathbf{p}_{k_j}\}_{k_j \neq k_i}$, we use the “Gaussian forcing” idea to approximate $\Pr(a_{k_i} = x_{k_i}(m) \mid \tilde{\mathbf{y}}, \{\mathbf{p}_{k_j}\}_{\forall k_j \neq k_i}) \forall m$, and set the results equal to the

corresponding elements of \mathbf{p}_{k_i} .

5. If $i < N_T$, let $i = i + 1$ and back to step 4. Otherwise, carry on step 6.
6. If $\forall i$, \mathbf{p}_i has converged, go to step 7. Otherwise, let $iter = iter + 1$ and return to step 3.
7. For $j = 1, \dots, N_T$, make a decision \hat{a}_j for a_j via

$$\hat{a}_j = x_j(l), \quad l = \arg \max_m \{p_j(m)\} \quad (3.13)$$

The block diagram of the basic PDA procedure is shown in Fig. 3.1.

3.1.2 Computational Refinements

A. Speed-Up–Matrix arithmetic

As noted in [4], although the computation in step 4 is no longer exponential, to calculate the inverse of $\mathbf{\Omega}_i$ for each element directly is still expensive. Further simplifications can be evaluated by applying the Sherman-Morrison-Woodbury formula [10] twice consecutively.

1. Define auxiliary variables $\mathbf{\Omega}$

$$\mathbf{\Omega} = \mathbf{\Omega}_i + \mathbf{e}_i \mathbf{e}_i^T Var[a_i] \quad (3.14)$$

2. Compute $\mathbf{\Omega}_i^{-1}$ via

$$\mathbf{\Omega}_i^{-1} = \mathbf{\Omega}^{-1} + \frac{\mathbf{\Omega}^{-1} \mathbf{e}_i \mathbf{e}_i^T \mathbf{\Omega}^{-1} Var[a_i]}{1 + \mathbf{e}_i^T \mathbf{\Omega}^{-1} \mathbf{e}_i Var[a_i]} \quad (3.15)$$

3. Compute $P_i(m)$ and update $E[a_i]$, $Var[\tilde{a}_i]$.

4. Compute $\mathbf{\Omega}^{-1}$ via

$$\boldsymbol{\Omega}^{-1} = \boldsymbol{\Omega}_i^{-1} - \frac{\boldsymbol{\Omega}_i^{-1} \mathbf{e}_i \mathbf{e}_i^T \boldsymbol{\Omega}_i^{-1} \text{Var}[a_i]}{1 + \mathbf{e}_i^T \boldsymbol{\Omega}_i^{-1} \mathbf{e}_i \text{Var}[a_i]} \quad (3.16)$$

B. Speed-Up–Successive Skip

In our simulations, we have observed that the algorithm generally converges within 2 to 4 iterations for $\text{SNR} < 14$ dB, and within 1 to 2 iterations for $\text{SNR} > 14$ dB. However, the overall complexity can be high if one or two elements of \mathbf{a} exhibit slow convergence. To reduce the complexity in these instances, successive skip is applied each iteration. Note that, the successive skip method is different from the successive cancellation method which mentioned in [4]. The advantage of the successive skip method is implemented easily and lower complexity than the successive cancellation method. The main idea of the successive skip tactic is that the posterior probability of some elements is high enough to let us believe it within the process of converges. So, if we have this element in the next iteration, we can simply skip it.

After z th iteration, we define \mathbf{V} to be the set of elements that satisfy

$$\max\{P_i(m)\}_{\forall m} \geq \varepsilon, \forall i \quad (3.17)$$

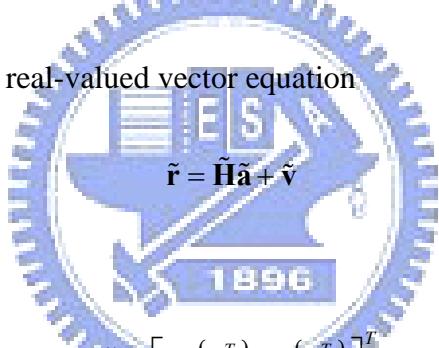
where $\varepsilon = 1 - (0.2\sigma^2)$ is a small positive number. At $z+1$ th iteration, we will skip the elements that belong to \mathbf{V} .

3.2 Generalized PDA Detector

The Generalized PDA (GPDA) detector [5] is present for the special case of

square/rectangular (sqr/rect) QAM. In the case of sqr/rect q -QAM, the GPDA algorithm differs from the PDA approach of [11] by reducing the number of probabilities associated with each transmit symbol. As an apparent consequence of reducing the number of probabilities for sqr/rect QAM, the GPDA shows an improved error probability over the PDA approach used in [11]. A further advantage of GPDA is that it offers a reduced computational cost over that of [11] for the case when the number of receive antennas is greater than the number of transmit antennas.

To obtain the system model for sqr/rect QAM version of the GPDA detector, we begin by transforming (2.1) into the real-valued vector equation



$$\tilde{\mathbf{r}} = \tilde{\mathbf{H}}\tilde{\mathbf{a}} + \tilde{\mathbf{v}} \quad (3.18)$$

where

$$\begin{aligned} \tilde{\mathbf{r}} &= \left[\Re\{\mathbf{r}^T\} \ \Im\{\mathbf{r}^T\} \right]^T \\ \tilde{\mathbf{a}} &= \left[\Re\{\mathbf{a}^T\} \ \Im\{\mathbf{a}^T\} \right]^T \\ \tilde{\mathbf{v}} &= \left[\Re\{\mathbf{v}^T\} \ \Im\{\mathbf{v}^T\} \right]^T \\ \tilde{\mathbf{H}} &= \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \end{aligned} \quad (3.19)$$

Next we multiply (3.18) from the left by $\tilde{\mathbf{H}}^T$ to obtain

$$\mathbf{y} = \mathbf{G}\tilde{\mathbf{a}} + \mathbf{n} \quad (3.20)$$

where $\mathbf{y} = \tilde{\mathbf{H}}^T\tilde{\mathbf{r}}$, $\mathbf{G} = \tilde{\mathbf{H}}^T\tilde{\mathbf{H}}$, and $\mathbf{n} = \tilde{\mathbf{H}}^T\tilde{\mathbf{v}}$. Note that, because the elements of \mathbf{H} are modeled as i.i.d. complex Gaussian, $\tilde{\mathbf{H}}$ will almost always have full rank and consequently the symmetric matrix \mathbf{G} will be positive definite with probability nearly one.

The model for sqrt/rect QAM version of the GPDA detector is obtained by multiplying (3.20) from the left by \mathbf{G}^{-1} to yield

$$\tilde{\mathbf{y}} = \tilde{\mathbf{a}} + \tilde{\mathbf{n}} = \mathbf{e}_i \tilde{a}_i + \sum_{j \neq i} \mathbf{e}_j \tilde{a}_j + \tilde{\mathbf{n}} \quad (3.21)$$

In (3.21), $\tilde{\mathbf{y}} = \mathbf{G}^{-1}\mathbf{y}$, $\tilde{\mathbf{n}} = \mathbf{G}^{-1}\mathbf{n}$, and \mathbf{e}_i is a column vector whose i th element is 1 and whose other elements are 0. \tilde{a}_i is a member of one of two possible sets:

$$\tilde{a}_i \in \begin{cases} S_{\Re} = X_i = \{x_i(m)\}, i \in [1, N_T] \\ S_{\Im} = X_i = \{x_i(m)\}, i \in [N_T + 1, 2N_T] \end{cases} \quad (3.22)$$

In (3.22), S_{\Re} and S_{\Im} are the sets of distinct values that can be assumed by the real and imaginary parts of the QAM symbols respectively. Thus equation (3.21) can be solved via

PDA which we introduced in Section 3.1. Note that, when we calculate equation (3.9) in the GPDA algorithm, it should be modeling by the real Gaussian distribution rather than the complex Gaussian distribution and the noise variance should be $\sigma^2/2$ rather than σ^2 .

3.3 Simulation Results

Perfect Channel State Information (CSI)	
Perfect noise variance estimation	
Number of subcarrier	64
Length of cyclic prefix	16
Channel	Rayleigh Fading
Path	2

Relative power (dB)	(0,0)
Modulation	QPSK, 16QAM, 64QAM

Table 3.1 Simulation parameters for comparing PDA with GPDA.

In this Section, we use several computer simulation examples to show the performance and the computational cost of the PDA detector and the GPDA detector. We also compare the PDA detector and the GPDA detector with the V-BLAST ZF OSIC [9] and the optimal ML detector in the examples. The simulation parameters are shown in Table 3.1.

In Fig. 3.2, we compare the BER performance for the GPDA detector and the PDA detector with $q = 4$. The result shows that the BER performance of the GPDA detector has about 1 dB improvements as compared with the PDA detector. Since the dimension of the GPDA detector is two times as the dimension of the PDA detector making the PDF of the interference and noise closer to Gaussian distribution. In Fig. 3.3, it shows the performance of the aforementioned detectors for the case $q = 16$.

In Fig. 3.4, we compare the effect of sorting for the GPDA detector and the PDA detector with $q = 4$. The result shows that the BER performance of the PDA and the GPDA detectors has approximately more 2 dB than that of the unsorted PDA and the unsorted GPDA detectors. Thus, we can identify that sorting is an important factor for the GPDA and the PDA detectors.

In Fig. 3.5, we compare the number of iterations for the GPDA detector and the PDA

detector with $q = 4$. The result shows that both the GPDA detector and the PDA detector just need 2-4 iterations to converge. In Fig. 3.6, it shows the BER performance of the aforementioned detectors for the case $q = 16$.

In Fig. 3.7, we compare the complexity for the GPDA detector and the PDA detector with $q = 4$. For the case of complexity of the system, the measurement was calculated using *FLOPS* function in *MATLAB* [14], which counts the approximated floating point operations that the algorithm needs to complete decoding in one block of transmitted symbols. The result shows that there is a great gap of the original complexity between the GPDA detector and the PDA detector. After using the matrix speed-up (speed-up I) tactic, the gap between the GPDA detector and the PDA detector has reduced. If we use matrix speed-up tactic and successive skip tactic simultaneously (speed-up II), the complexity may reduce once more. Overall, the complexity of the GPDA detector is slight more than that of the PDA detector for $q = 4$.

In Fig. 3.8, we compare the complexity for the GPDA detector and the PDA detector with $q = 16$. As the figure suggests that, after using speed-up tactic, the complexity of the GPDA detector will be significantly less than that of the PDA detector.

In Fig. 3.9, we compare the complexity for the GPDA detector and the PDA detector with different modulation order, at SNR=0 dB. We can observe that with greater modulation order, the gap between the GPDA detector and the PDA detector gets wider.

Fig. 3.10 shows the noise variance estimation error for the GPDA detector and the PDA

detector with $q = 16$, at SNR=25 dB. The result shows that the GPDA detector and the PDA detector are almost free from the impact of the noise variance estimated error. Note that, $E[\tilde{\sigma}^2] = E[\sigma^2] + E[\Delta\sigma^2]$ where $E[\tilde{\sigma}^2]$ is estimated noise variance composed of real noise variance $E[\sigma^2]$ and estimation error $E[\Delta\sigma^2]$.

After above comparisons, the GPDA detector shows improved BER performance over the PDA detector by reducing the number of probabilities associated with each transmit symbol. Moreover, the complexity of the GPDA detector is much less than that of the PDA detector, especially in high order modulation. Hence, that is the reason why we choose the GPDA detector rather than the PDA detector.



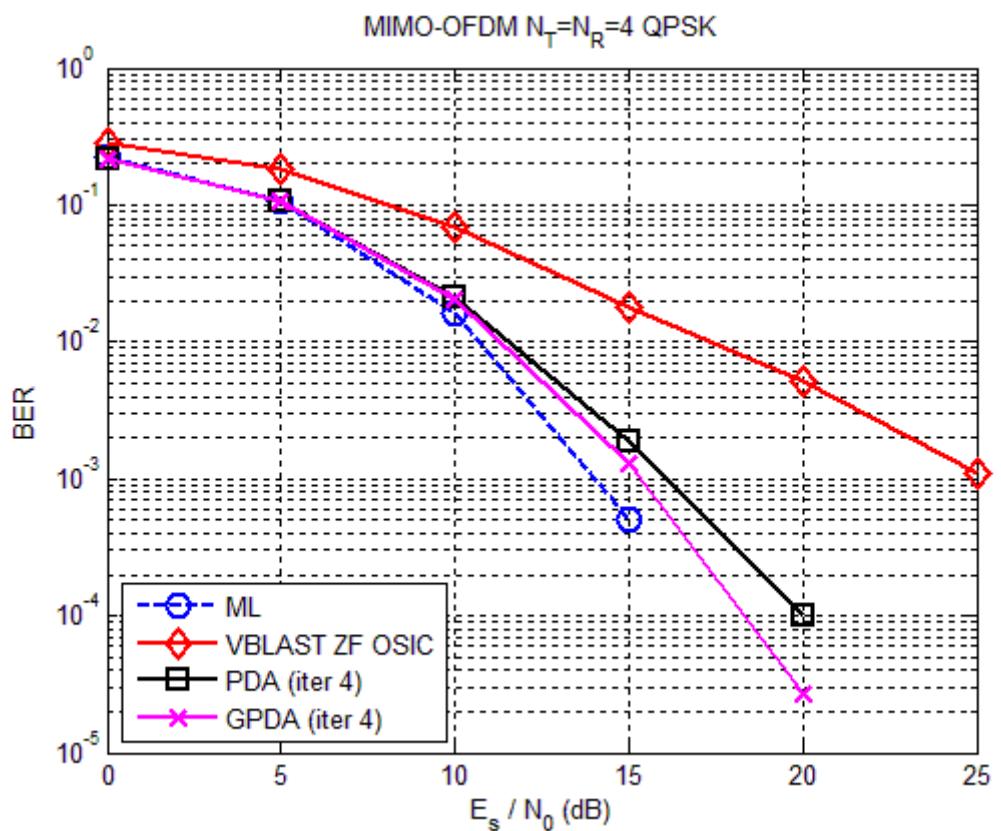


Fig. 3.2 The BER performance for PDA and GPDA with $q = 4$.

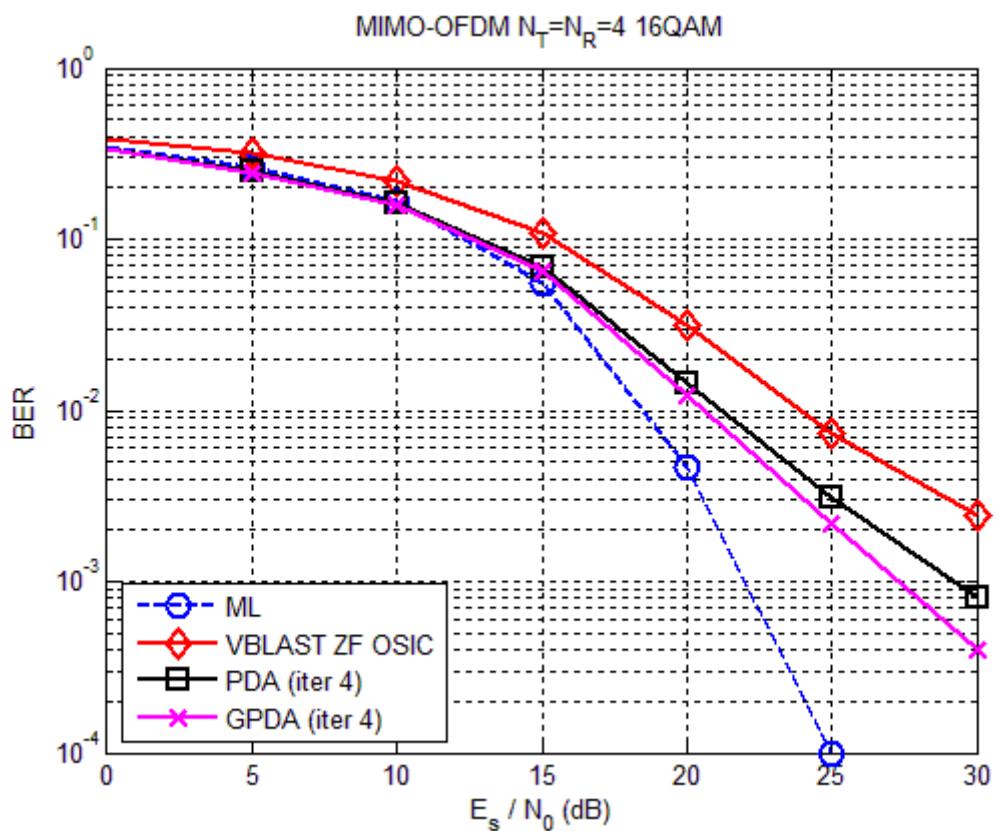


Fig. 3.3 The BER performance for PDA and GPDA with $q = 16$.

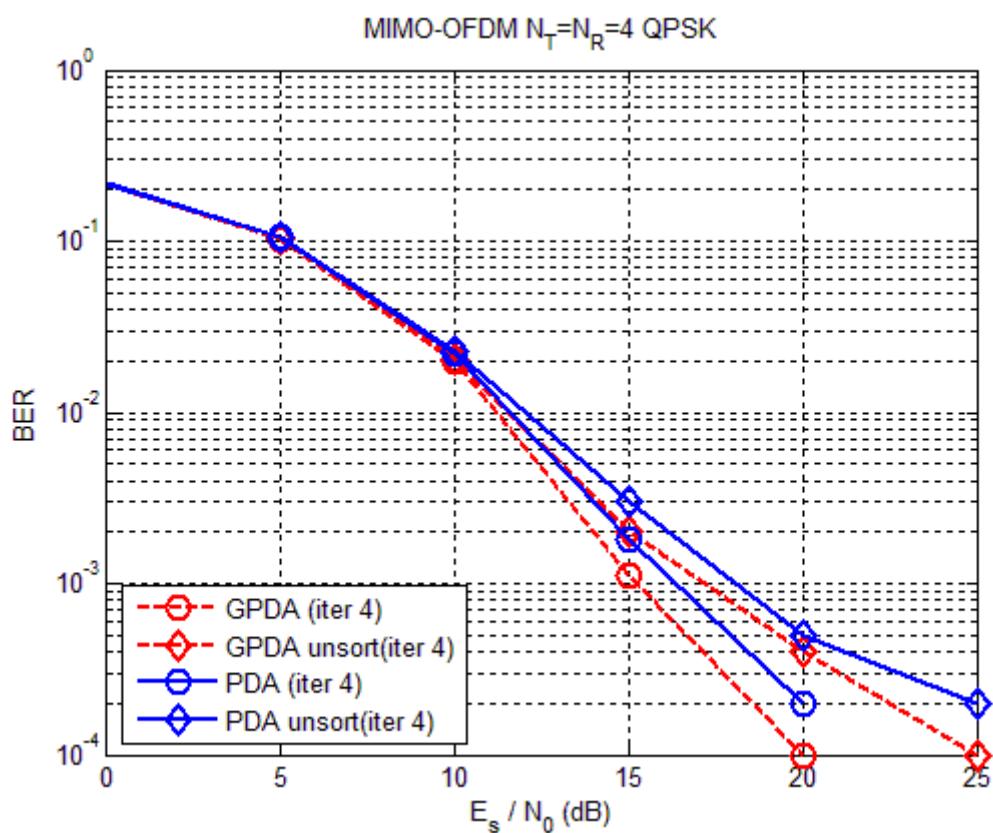


Fig. 3.4 The effect of sorting for PDA and GPDA with $q = 4$.

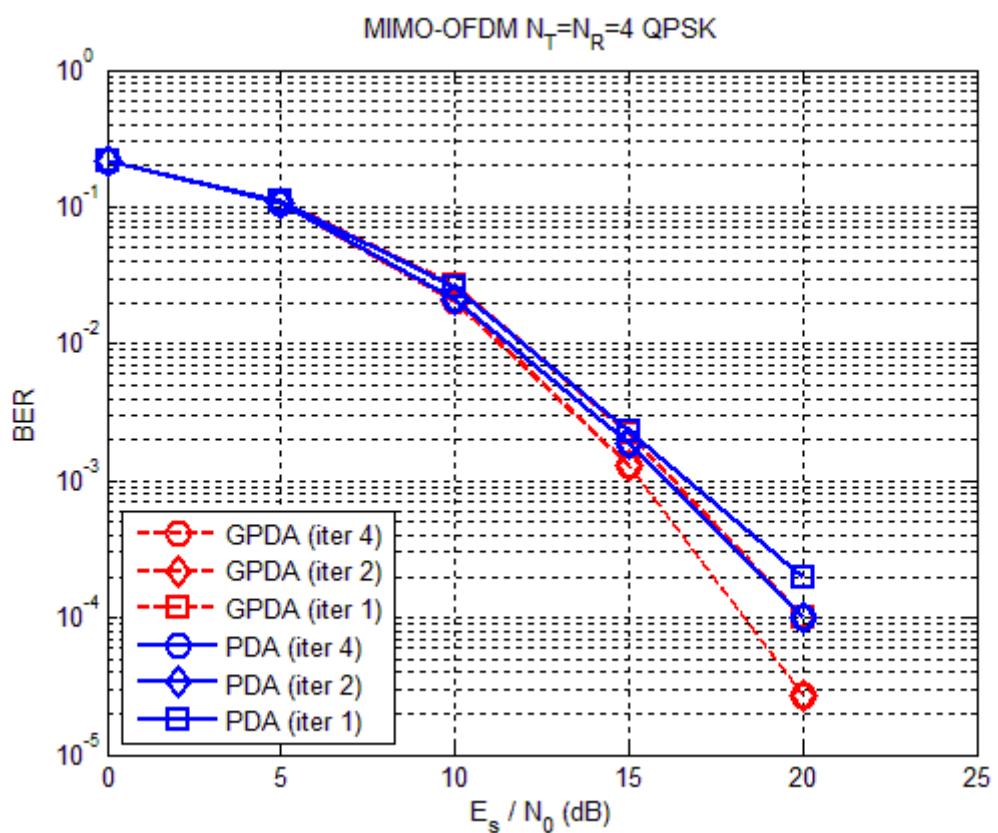


Fig. 3.5 The number of iterations for PDA and GPDA with $q = 4$.

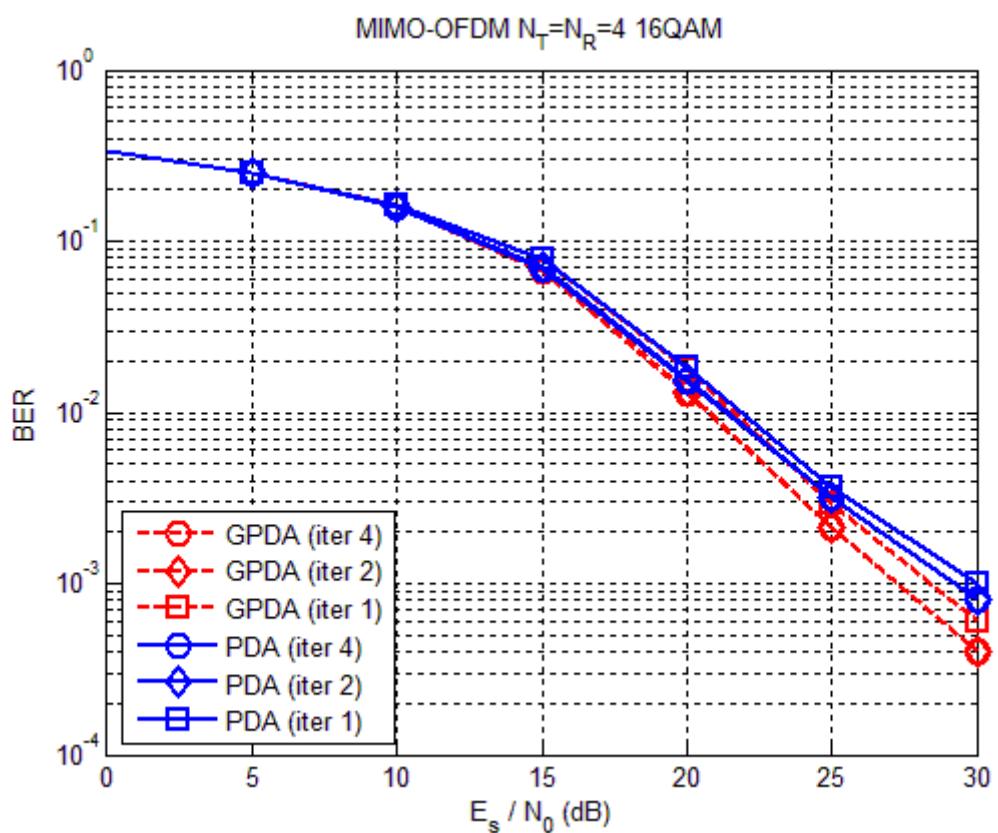


Fig. 3.6 The number of iterations for PDA and GPDA with $q = 16$.

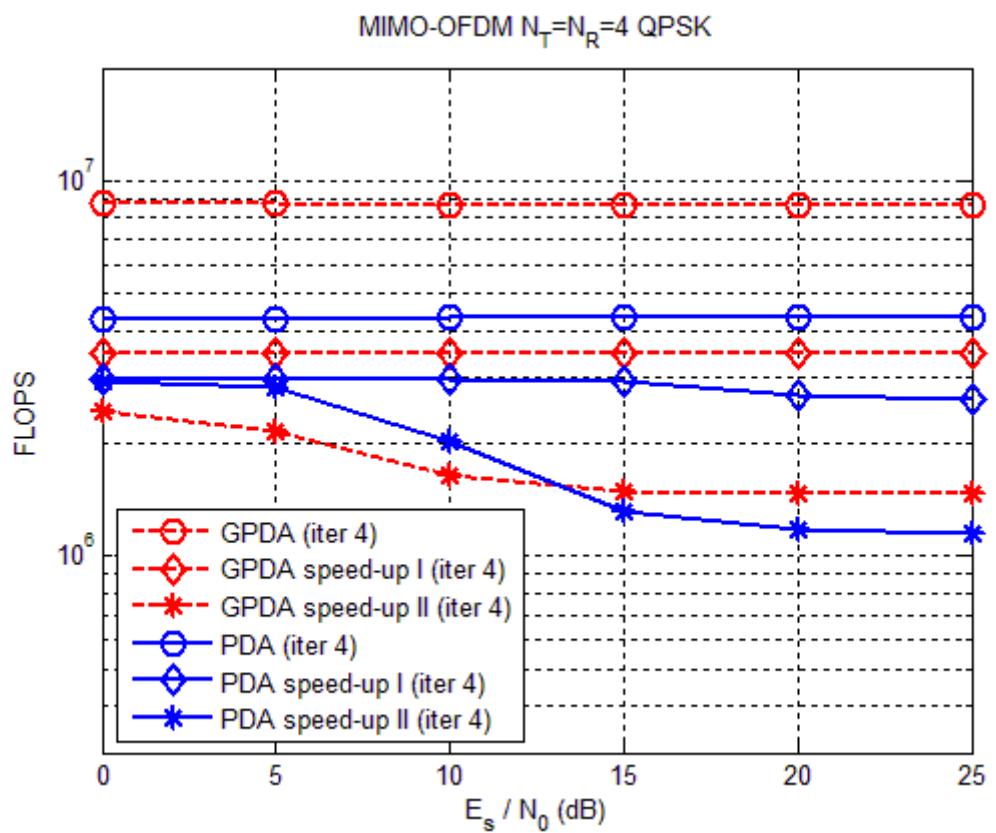


Fig. 3.7 The complexity for PDA and GPDA with $q = 4$.

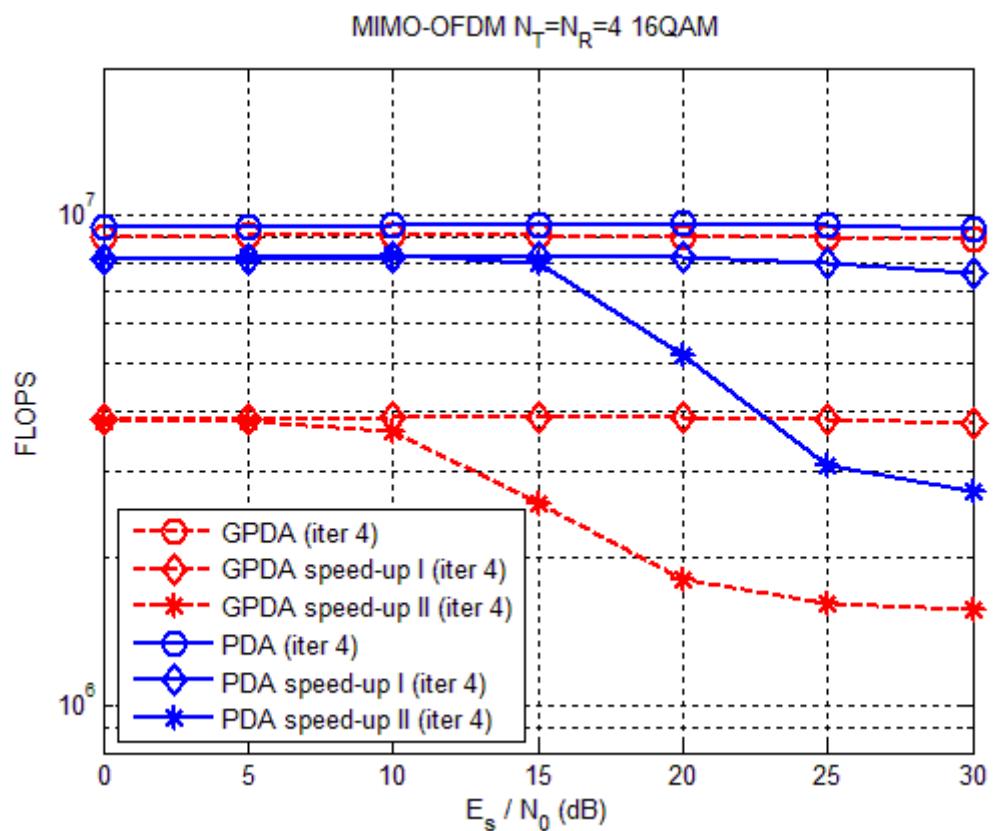


Fig. 3.8 The complexity for PDA and GPDA with $q = 16$.

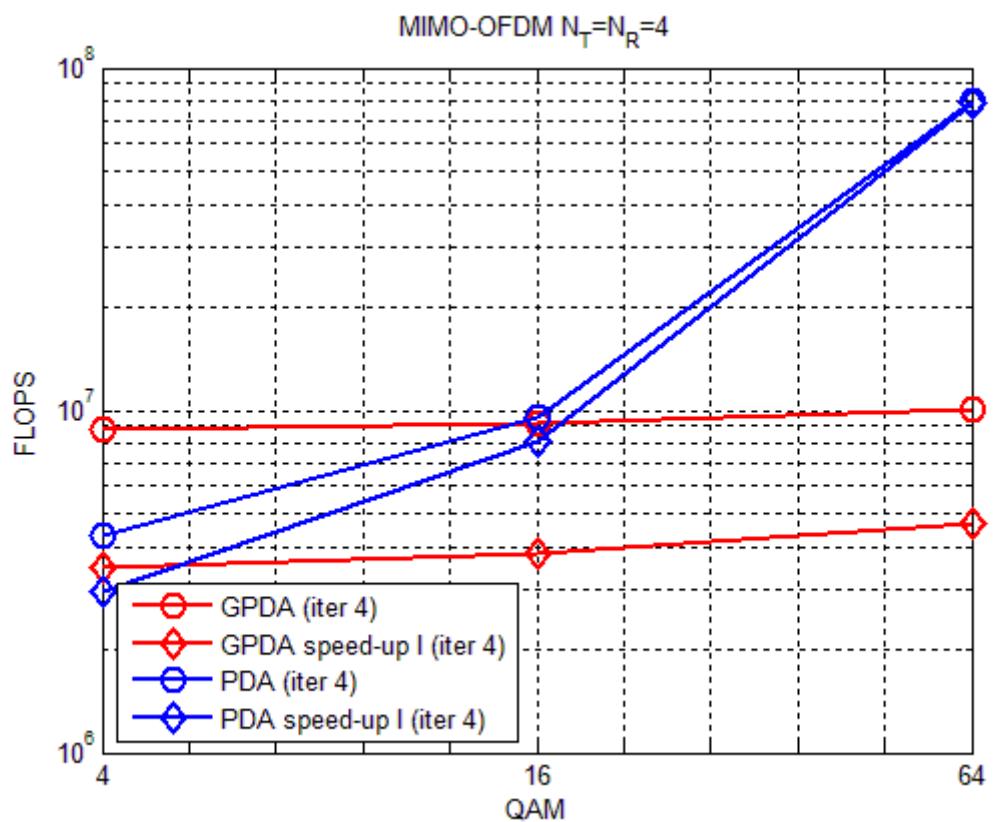


Fig. 3.9 The complexity for PDA and GPDA with different modulation order.

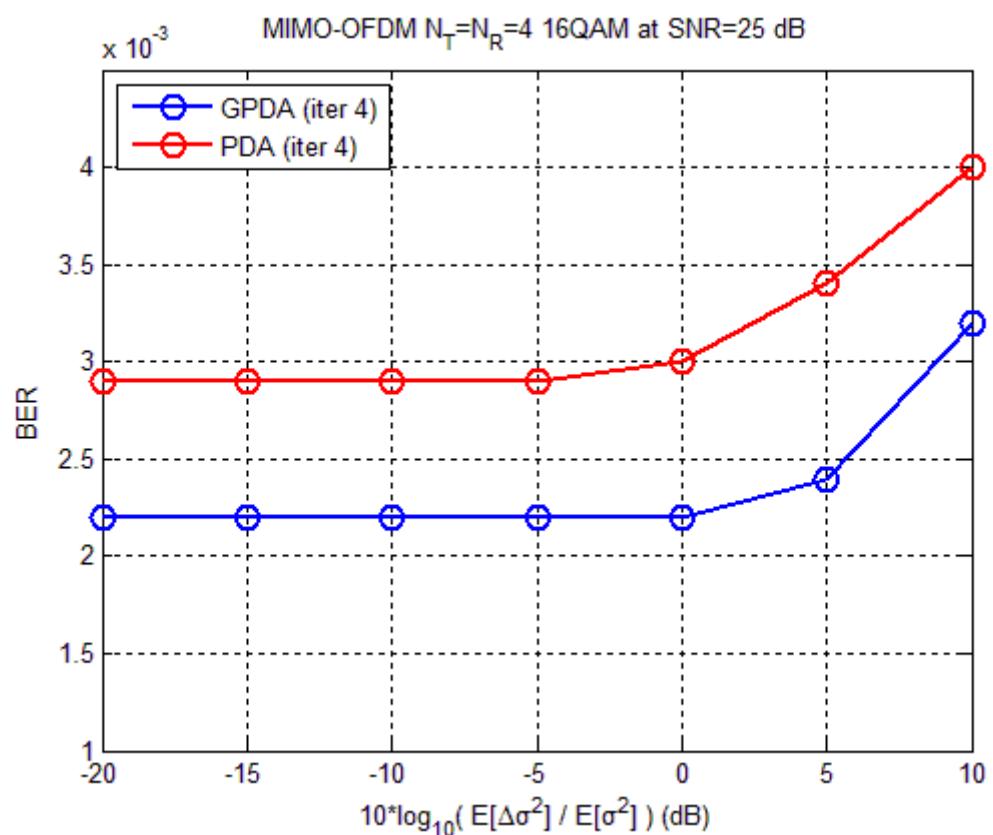


Fig. 3.10 The noise variance estimation error for GPDA and PDA.

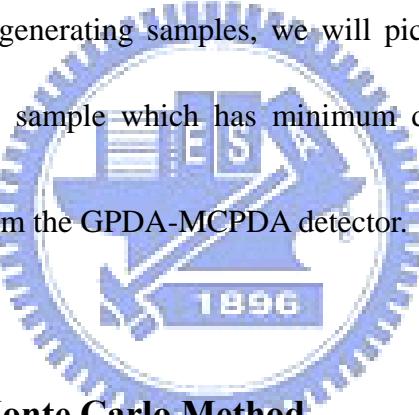
Chapter 4

Data Detection in MIMO-OFDM System Based on GPDA Detector

4.1 GPDA-MCPDA Detector

The basic idea of the GPDA-MCPDA detector is using the GPDA detector at the low SNR regions, and using parallel MCPDA method to generate numbers of random samples at the high SNR regions. After generating samples, we will pick up a sample from the final iteration of parallels, and the sample which has minimum distance (i.e. $\arg \min_{a \in X_i} \|\mathbf{r} - \mathbf{H}a\|^2$).

Thus, we can get a solution from the GPDA-MCPDA detector.



4.1.1 Markov Chain Monte Carlo Method

A major limitation towards more widespread implementation of Bayesian approaches is that obtaining the posterior distribution often requires the integration of high-dimensional functions. This can be computationally very difficult, but several approaches short of direct integration have been proposed. The MCMC methods [6], which attempt to simulate direct draws from some complex distribution of interest. MCMC approaches are so-named because one uses the previous to randomly generate the next sample value, generating a Markov Chain (as the transition probabilities between sample values are only a function of the most recent

sample value).

The realization in the early 1990's that one particular MCMC method, the Gibbs sampler, is widely applied to a broad class of Bayesian problems has sparked a major increase in the application of Bayesian analysis.

4.1.1.1 Monte Carlo Integration

The original Monte Carlo approach was a method developed by physicists to use random number generation to compute integrals. Suppose we wish to compute a complex integral

$$\int_a^b h(x)dx \quad (4.1)$$

If we can decompose $h(x)$ into the product of a function $f(x)$ and a probability density function $p(x)$ defined over the interval (a,b) , then note that

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_{p(x)}[f(x)] \quad (4.2)$$

so that the integral can be expressed as an expectation of $f(x)$ over the density $p(x)$. Thus, if we draw a large number x_1, \dots, x_n of random variables from the density $p(x)$, then

$$\int_a^b h(x)dx = E_{p(x)}[f(x)] \approx \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (4.3)$$

This is referred to as Monte Carlo integration.

Monte Carlo integration can be used to approximate posterior (or marginal posterior) distributions required for a Bayesian analysis. Consider the integral $I(y) = \int f(y|x)p(x)dx$, which we approximate by

$$\hat{I}(y) = \frac{1}{n} \sum_{i=1}^n f(y | x_i) \quad (4.4)$$

where x_i are draws from the density $p(x)$.

4.1.1.2 Importance Sampling

It was observed in the preceding Section 4.1.1.1 that the integral can be approximate by Monte Carlo integration. However, not every density $p(x)$ can be drawn directly. Now, we suppose the density $q(x)$ roughly approximates the density $p(x)$, then

$$\int f(x) p(x) dx = \int f(x) \left(\frac{p(x)}{q(x)} \right) q(x) dx = E_{q(x)} [f(x) \left(\frac{p(x)}{q(x)} \right)] \quad (4.5)$$

This forms the basis for the method of importance sampling, with

$$\int f(x) p(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i) \frac{p(x_i)}{q(x_i)} \quad (4.6)$$

where x_i are draws from the density $q(x)$.

4.1.1.3 Introduction to Markov Chains

Before introducing the Gibbs sampler, a few introductory comments on Markov Chains are in order. Let X_t denote the value of a random variable at time t , and let the state space refer to the range of possible X values. The random variable is a Markov process if the transition probabilities between different values in the state space depend only on the random variable's current state, i.e.,

$$\Pr(X_{t+1} = s_j \mid X_0 = s_k, \dots, X_t = s_i) = \Pr(X_{t+1} = s_j \mid X_t = s_i) \quad (4.7)$$

Thus for a Markov random variable the only information about the past needed to predict the future is the current stage of the random variable, knowledge of the values of earlier states do not change the transition probability. A Markov chain refers to a sequence of random variables (X_0, \dots, X_n) generated by a Markov process. A particular chain is defined most critically by its transition probabilities, $p(i, j) = p(i \rightarrow j)$, which is the probability that a process at stage s_i moves to state s_j in a single step,

$$p(i, j) = p(i \rightarrow j) = \Pr(X_{t+1} = s_j \mid X_t = s_i) \quad (4.8)$$

Let

$$\pi_j(t) = \Pr(X_t = s_j) \quad (4.9)$$

denote the probability that the chain is in state s_j at time t , and let $\pi(t)$ denote the row vector of the state space probabilities at step t . We start the chain by specifying a starting vector $\pi(0)$. Often all the elements of $\pi(0)$ are zero except for a single element of 1, corresponding to the process starting in that particular state. As the chain progresses, the probability values get spread out over the possible state space.

The probability that the chain has state value s_i at time (or step) $t+1$ is given by the Chapman-Kolomogrov equation, which sum over the probability of being in a particular state at the current step and the transition probability from that state into state s_i ,

$$\begin{aligned}
\pi_i(t+1) &= \Pr(X_{t+1} = s_i) \\
&= \sum_k \Pr(X_{t+1} = s_i \mid X_t = s_k) \Pr(X_t = s_k) \\
&= \sum_k \Pr(k \rightarrow i) \pi_k(t) = \sum_k \Pr(k, i) \pi_k(t)
\end{aligned} \tag{4.10}$$

Successive iteration of the Chapman-Kolomogrov equation describes the evolution of the chain.

We can more compactly write the Chapman-Kolomogrov equation in matrix form as follows. Define the probability transition matrix \mathbf{P} as the matrix whose i, j th element is $P(i, j)$, the probability of moving from state i to state j , $P(i \rightarrow j)$. The Chapman-Kolomogrov equation becomes



$$\pi(t+1) = \pi(t)\mathbf{P} \tag{4.11}$$

Using the matrix form, we immediately see how to quickly iterate the Chapman-Kolomogrov equation, as

$$\pi(t) = \pi(t-1)\mathbf{P} = (\pi(t-2)\mathbf{P})\mathbf{P} = \pi(t-2)\mathbf{P}^2 \tag{4.12}$$

Continuing in this fashion show that

$$\pi(t) = \pi(0)\mathbf{P}^t \tag{4.13}$$

Defining the n -step transition probability $p_{i,j}^{(n)}$ as the probability that the process is in state j given that it started in state i , n step ago, i.e.,

$$p_{i,j}^{(n)} = \Pr(X_{t+n} = s_j \mid X_t = s_i) \tag{4.14}$$

it immediately follows that $p_{i,j}^n$ is just the i, j -th element of \mathbf{P}^n .

Finally, a Markov chain is said to be irreducible if there exists a positive integer such that

$p_{i,j}^{(n)} > 0$ for all i, j . That is, all states communicate with each other, as one can always go from any state to any other state (although it may take more than one step). Likewise, a chain is said to be aperiodic when the number of steps required to move between two states (say x and y) is not required to be multiple of some integer. Put another way, the chain is not forced into some cycle of fixed length between certain states.

A Markov chain may reach a stationary distribution π^* , where the vector of probabilities of being in any particular given state is independent of the initial condition. The stationary distribution satisfies

$$\pi^* = \pi^* \mathbf{P} \quad (4.15)$$

The condition for a stationary distribution is that the chain is irreducible and aperiodic. When a chain is periodic, it can cycle in a deterministic fashion between states and hence never settles down to a stationary distribution.

A sufficient condition for a unique stationary distribution is that the detailed balance equation holds,

$$P(j \rightarrow k)\pi_j^* = P(k \rightarrow j)\pi_k^* \quad (4.16)$$

If equation (4.16) holds for all i, k , the Markov chain is said to be reversible, and hence equation (4.16) is also called the reversibility condition. Note that this condition implies $\pi = \pi \mathbf{P}$, as the j th element of $\pi \mathbf{P}$ is

$$(\pi \mathbf{P})_j = \sum_i \pi_i P(i \rightarrow j) = \sum_i \pi_j P(j \rightarrow i) = \pi_j \sum_i P(j \rightarrow i) = \pi_j \quad (4.17)$$

with the last step following since rows sum to one.

4.1.1.4 Gibbs Sampler

One problem with applying Monte Carlo integration is in obtaining samples from some complex probability distribution. Attempts to solve this problem are the roots of MCMC methods. In particular, they trace to attempts by mathematical physicists to integrate very complex functions by random sampling, and resulting Metropolis-Hastings sampling [6]. The Gibbs sampler [6] [15] (introduced in the context of image processing by Geman 1984), is a special case of Metropolis-Hastings sampling wherein the random value is always accepted (i.e. $\alpha = 1$). The task remains to specify how to construct a Markov Chain with values converged to the target distribution. The key to the Gibbs sampler is that we only consider the univariate conditional distributions (the distribution when all of the random variables but one is assigned fixed value). Such conditional distributions are far easier to simulate than complex joint distributions and usually have simpler forms. Thus, we simulate n random variables sequentially from the n univariate conditions rather than generating a single n -dimensional vector in a single pass using the full joint distribution.

To introduce the Gibbs sampler, consider a bivariate random variable (x, y) , and suppose we want to compute one or both marginal, $p(x)$ and $p(y)$. The idea behind the sampler is that it is far easier to consider a sequence of conditional distributions, $p(x|y)$ and

$p(y|x)$, than to obtain the marginal by integral of the joint density $p(x,y)$, e.g., $p(x) = \int p(x,y)dy$. The sampler starts with some initial value y_0 for y and obtains x_0 by generating a random variable from the conditional distribution, $p(x|y=y_0)$. We use x_0 to generate a new value of y_1 from the sampler. To do so, we draw from the conditional distribution based on the value x_0 , $p(y|x=x_0)$. The sampler proceeds as follows:

$$x_i \sim p(x|y=y_{i-1}) \quad (4.18)$$

$$y_i \sim p(y|x=x_i) \quad (4.19)$$

Repeating this process for k times, then it generates a Gibbs sequence of length k , where a subset of points (x_j, y_j) for $1 \leq j \leq m < k$ is taken as our simulated draws from the full joint distribution. The first m times of the process, called burn-in period, can make the Markov Chain converge to the distribution that near its stationary one.

When more than two variables are involved, the sampler is extended in the obvious fashion. In particular, the value of the k th variable is drawn from the distribution

$p(\theta^{(k)} | \theta^{(-k)})$ where $\theta^{(-k)}$ denotes a vector containing all of the variables but k . Thus, during the i th iteration of the sample, to obtain the value of $\theta_i^{(k)}$, we draw from the distribution

$$\theta_i^{(k)} \sim p(\theta^{(k)} | \theta^{(1)} = \theta_i^{(1)}, \dots, \theta^{(k-1)} = \theta_i^{(k-1)}, \theta^{(k+1)} = \theta_{i-1}^{(k+1)}, \dots, \theta^{(n)} = \theta_{i-1}^{(n)}) \quad (4.20)$$

For example, if there are four variables, (w, x, y, z) , the sampler becomes

$$\begin{aligned} w_i &\sim p(w|x=x_{i-1}, y=y_{i-1}, z=z_{i-1}) \\ x_i &\sim p(x|w=w_i, y=y_{i-1}, z=z_{i-1}) \\ y_i &\sim p(y|w=w_i, x=x_i, z=z_{i-1}) \\ z_i &\sim p(z|w=w_i, x=x_i, y=y_i) \end{aligned} \quad (4.21)$$

Now, we consider the equation (3.2) and we use Gibbs sampler to generate samples of \mathbf{a} from the distribution $P(\mathbf{a} | \tilde{\mathbf{y}})$. The procedure as follows:

- generate initial samples $\bar{\mathbf{a}}^{-N_b}$ randomly
- for $n = -N_b + 1$ to N_s
 - sample \bar{a}_1^n from $P(\bar{a}_1 | \bar{a}_2^{n-1}, \bar{a}_3^{n-1}, \dots, \bar{a}_{N_T}^{n-1}, \tilde{\mathbf{y}})$
 - sample \bar{a}_2^n from $P(\bar{a}_2 | \bar{a}_1^n, \bar{a}_3^{n-1}, \dots, \bar{a}_{N_T}^{n-1}, \tilde{\mathbf{y}})$
 - \vdots
 - sample $\bar{a}_{N_T}^n$ from $P(\bar{a}_{N_T} | \bar{a}_1^n, \bar{a}_2^n, \dots, \bar{a}_{N_T-1}^n, \tilde{\mathbf{y}})$
- end

then, we consider $P(\bar{a}_i | \bar{\mathbf{a}}_{-i}^n, \tilde{\mathbf{y}})$, and it can be factorize as

$$\begin{aligned}
 P(\bar{a}_i | \bar{\mathbf{a}}_{-i}^n, \tilde{\mathbf{y}}) &= \frac{P(\tilde{\mathbf{y}}, \bar{\mathbf{a}}_{-i}^n | \bar{a}_i) P(\bar{a}_i)}{P(\tilde{\mathbf{y}}, \bar{\mathbf{a}}_{-i}^n)} \\
 &= \frac{P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n) P(\bar{\mathbf{a}}_{-i}^n | \bar{a}_i) P(\bar{a}_i)}{P(\tilde{\mathbf{y}} | \bar{\mathbf{a}}_{-i}^n) P(\bar{\mathbf{a}}_{-i}^n)} \\
 &= \frac{P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n) P(\bar{a}_i | \bar{\mathbf{a}}_{-i}^n)}{P(\tilde{\mathbf{y}} | \bar{\mathbf{a}}_{-i}^n)} \\
 &= \frac{P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n) P(\bar{a}_i | \bar{\mathbf{a}}_{-i}^n)}{\sum_{\bar{a}_i} P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n) P(\bar{a}_i | \bar{\mathbf{a}}_{-i}^n)} \\
 &= \frac{P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n)}{\sum_{\bar{a}_i} P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{\mathbf{a}}_{-i}^n)}
 \end{aligned} \tag{4.22}$$

where $\bar{\mathbf{a}}_{-i}^n = [\bar{a}_1^n, \dots, \bar{a}_{i-1}^n, \bar{a}_{i+1}^{n-1}, \dots, \bar{a}_{N_T}^{n-1}]$ and $P(\tilde{\mathbf{y}} | \bar{a}_i = x_i(m), \bar{\mathbf{a}}_{-i}^n)$ can be calculated as follows:

$$\begin{aligned}
 P(\tilde{\mathbf{y}} | \bar{a}_i = x_i(m), \bar{\mathbf{a}}_{-i}^n) &\propto \exp(-(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j=1}^{i-1} \mathbf{e}_j \bar{a}_j^n - \sum_{j=i+1}^{N_T} \mathbf{e}_j \bar{a}_j^{n-1})^H \Phi^{-1} \\
 &\quad (\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j=1}^{i-1} \mathbf{e}_j \bar{a}_j^n - \sum_{j=i+1}^{N_T} \mathbf{e}_j \bar{a}_j^{n-1}))
 \end{aligned} \tag{4.23}$$

thus, we define

$$\begin{aligned}
\gamma_i(m) &= -(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j=1}^{i-1} \mathbf{e}_j \bar{a}_j^n - \sum_{j=i+1}^{N_T} \mathbf{e}_j \bar{a}_j^{n-1})^H \Phi^{-1} (\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j=1}^{i-1} \mathbf{e}_j \bar{a}_j^n - \sum_{j=i+1}^{N_T} \mathbf{e}_j \bar{a}_j^{n-1}) \\
&\propto (2\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - 2\Lambda_i)^H \Phi^{-1} \mathbf{e}_i x_i(m) \quad \left(\text{Let } \Lambda_i = \sum_{j=1}^{i-1} \mathbf{e}_j \bar{a}_j^n - \sum_{j=i+1}^{N_T} \mathbf{e}_j \bar{a}_j^{n-1} \right) \quad (4.24) \\
&= (2\mathbf{k}_i - \mathbf{e}_i x_i(m))^H \Phi^{-1} \mathbf{e}_i x_i(m) \quad \left(\text{Let } \mathbf{k}_i = \tilde{\mathbf{y}} - \Lambda_i \right) \\
&= \left(\sum_j [2\mathbf{k}_i^H]_j [\Phi^{-1}]_{ji} - x_i^H(m) [\Phi^{-1}]_{ii} \right) x_i(m)
\end{aligned}$$

where $[2\mathbf{k}_i^H]_j$ is the j th element of $2\mathbf{k}_i^H$, $[\Phi^{-1}]_{ji}$ is the element (j,i) of Φ^{-1} and

$$\Phi = \sigma^2 \mathbf{G}^{-1} \quad (4.25)$$

finally, we can obtain

$$P(\bar{a}_i | \bar{a}_{-i}^n, \tilde{\mathbf{y}}) = \frac{P(\tilde{\mathbf{y}} | \bar{a}_i, \bar{a}_{-i}^n)}{\sum_l \exp(\gamma_i(l))} \quad (4.26)$$

In this routine, “for” loop examines the state variables \bar{a}_k in order, $N_b + N_s$ times.

The first N_b iterations of the loop, called burn-in period, are to let the Markov Chain converge to near its stationary distribution. During the next N_s iterations, the Gibbs sampler generate the N_s samples, i.e., $\bar{\mathbf{a}}^n = [\bar{a}_1^n, \bar{a}_2^n, \dots, \bar{a}_{N_T}^n]^T$, for $n = 1, 2, \dots, N_s$. In general, the last sample can be the solution of the distribution. Since the distribution should be converged after $N_b + N_s$ times iteration.

There are two problems of Gibbs sampler:

- 1) How do you choose an initial point? A poor choice of initial point will greatly increase the required burn-in time, and an area of much current research is whether an optimal initial point can be found.

2) How many iterates are needed? This question does not have exact answer, the majority answers are obtained from the experience.

4.1.2 GPDA-MCPDA Detector

In [12], the author mentioned that at higher values of SNR, some of the transition probabilities in the underlying Markov Chain may become very small. As a result, the Markov Chain may be effectively divided into a number of nearly disjoint chains. The term nearly disjoint here means the transition probabilities that allow movement between the disjoint chains are very low. Therefore, a Gibbs sampler that starts from a random point will remain within the set of points surrounding the initial point and thus may not get a chance of visiting sufficient points to find the global solution. In [13] two solutions for solving this problem were proposed: (i) run a number of parallel Gibbs sampler with different initial points; (ii) while running the Gibbs sampler, we assume a noise variance which is higher than it actually is. These two methods turned out to be effective for low and medium SNRs.

In the parallel MCPDA detector, we will focus on these two methods which mentioned above to improve the performance of MCMC method. First, we use parallel Gibbs samplers with the initial point generated randomly. Second, we compute covariance according to (3.6) rather than (4.25), so named MCPDA. Since we take the variance of residual interference caused by the random samples into account in the equation (3.6), the covariance will increase.

Furthermore, with few times of iteration, the covariance will be gradually narrowing. This may be regarded as automatic Simulated Annealing method [6]. Finally, we will pick up a sample from the final iteration of parallels, and the sample which has minimum distance (i.e. $\arg \min_{a \in X_i} \|\mathbf{r} - \mathbf{H}a\|^2$). Thus, we can get a solution from the parallel MCPDA detector.

In Chapter 3, we have mentioned that the GPDA detector performs well at the low SNR regions, so we can only use the GPDA detector at the low SNR regions; however, with the SNR increasing (exceed M dB), the performance of the GPDA detector will get worse gradually. Thus, we need to use parallel MCPDA method to assist the GPDA detector in order to reach better performance, so named GPDA-MCPDA detector. Moreover, MCPDA is similar to GPDA, we only need to add few blocks, and then the GPDA detector will become the MCPDA detector. Therefore, it may be quite simple to implement. The block diagram of the GPDA-MCPDA detector is shown in Fig. 4.1 and the discrepancy between the GPDA detector and the MCPDA detector is shown in Fig. 4.2.

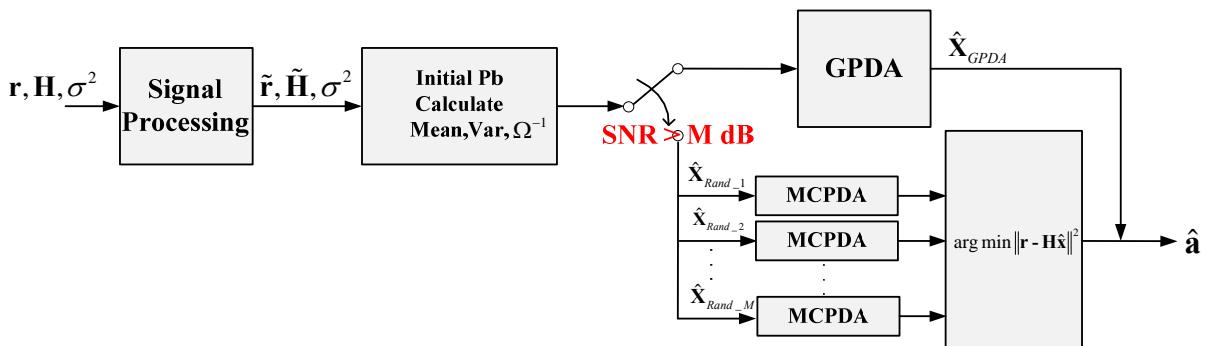


Fig. 4.1 Block diagram of the GPDA-MCPDA detector.

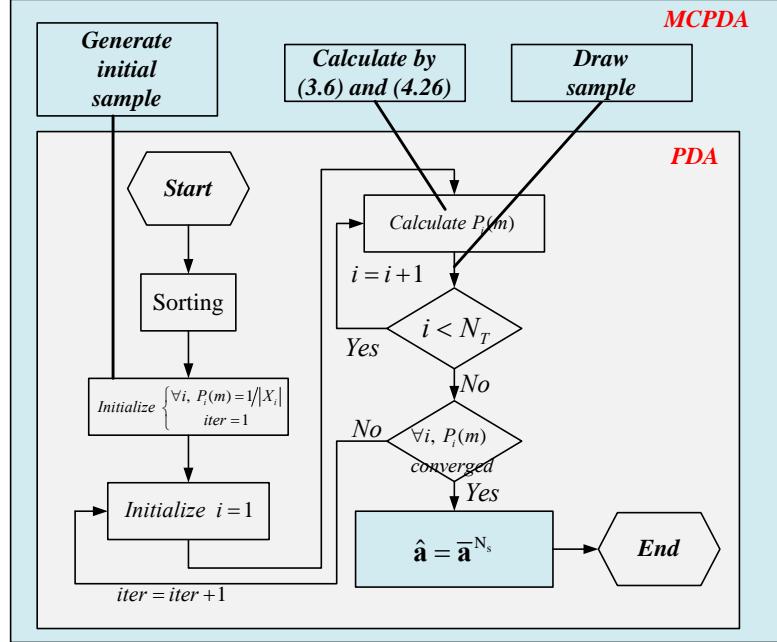


Fig. 4.2 The discrepancy between the PDA detector and the MCPDA detector.

4.2 GPDA-SD Detector

As we know, the GPDA detector does not have good performance at the high SNR regions. In order to solve this problem, we try to find a solution which is better than the GPDA solution. Therefore, we use the Sphere Decoding (SD) algorithm to do this work, so named GPDA-SD, which can attain a better performance and lower complexity in the MIMO-OFDM spatial multiplexing system.

4.2.1 Sphere Decoding

The sphere decoding [3] algorithm is a quasi-ML detection technique. It promises to find the optimal solution with low computational costs under some conditions. The SD algorithm

was first introduced by Finke and Pohst [16] in the context of the closest point search in lattices but it has become very popular in digital communication literature. Its various applications include lattice codes, CDMA systems, MIMO systems, global positioning system (GPS), *etc.*

4.2.1.1 Real Sphere Decoding

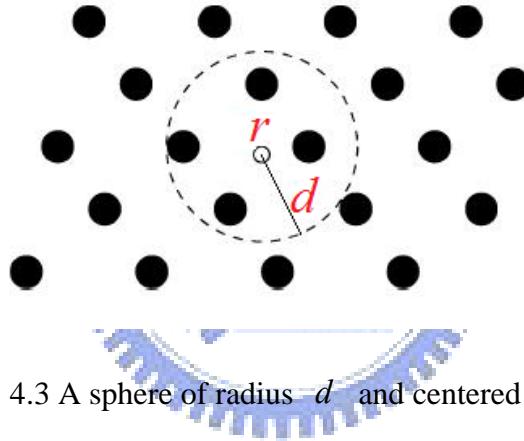


Fig. 4.3 A sphere of radius d and centered at \mathbf{r} .

In communication system, the SD algorithm is used to solve the ML problem as follows:

$$\hat{\mathbf{a}}_{\text{ML}} = \arg \min_{\mathbf{a} \in X_i} \|\mathbf{r} - \mathbf{H}\mathbf{a}\|^2 \quad (4.27)$$

The computational complexity of above exhaustive search method is really high. Therefore, the SD algorithm is brought up to avoid the exhaustive search and search only over the possible \mathbf{a} which lie in a certain sphere of radius d around the given vector \mathbf{r} , thereby reducing the search space and, hence, the required computations (see Fig. 4.3).

It is clear that the closest point inside the sphere will also be the closest point for the whole point. However, close scrutiny of this basic ideal leads to two key questions.

- 1) How do you choose radius d ? Clearly, if radius is too large, we obtain too many points, and the search remains exponential in size, whereas if radius is too small, we obtain no points inside the sphere.
- 2) How can we tell which points are inside the sphere? If this requires testing the distance of each point from \mathbf{r} , then there is no point in SD, as we will still need an exhaustive search.

The SD algorithm does not really address the first question, but usually uses ZF solution to be the radius of the sphere. However it does propose an efficient way to answer the second. The

basic observation is the following. Although it is difficult to determine the points inside a general N_T -dimensional sphere, it is trivial to do so in the one-dimensional case. The reason is that a one dimensional sphere reduces to the endpoints of an interval, and so the desired points will be the integer values that lie in this interval. We can use this observation to go from dimension k to dimension $k+1$. Suppose that we have determined all k -dimensional points that lie in a sphere of radius d . Then, for any such k -dimensional point, the set of admissible values of the $(k+1)$ th dimensional coordinate that lie in the higher dimensional sphere of the same radius d forms an interval.

The above means that we can determine all points in a sphere of dimension N_T and

radius d by successively determining all points in spheres of lower dimensions $1, 2, \dots, N_T$ and the same radius d . Such an algorithm for determining the points in an N_T -dimensional sphere essentially constructs a tree where the branches in the k th level of the tree correspond to the points inside the sphere of radius d and dimension k (see Fig. 4.4). Moreover, the complexity of such an algorithm will depend on the size of the tree, i.e., on the number of points visited by the algorithm in different dimensions.

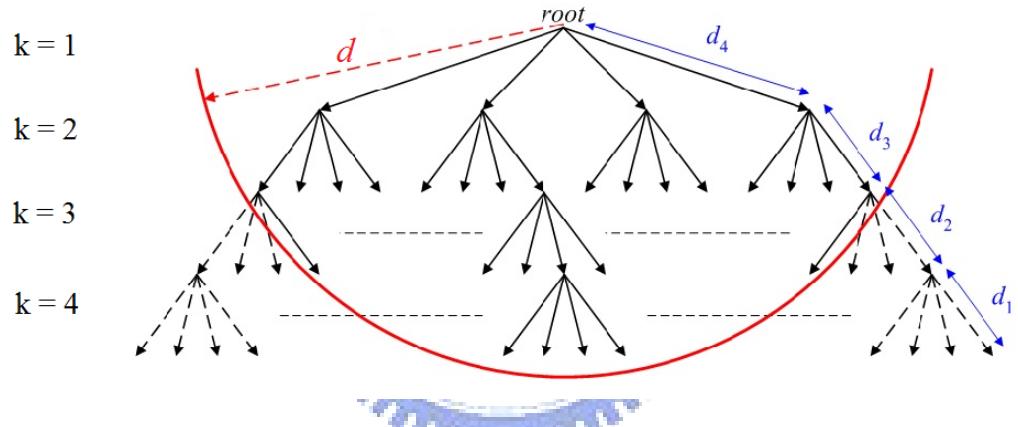


Fig. 4.4 Search tree of sphere decoding.

With this brief discussion, we can now be more specific about the problem at hand. To this end, we shall assume that $N_R \geq N_T$. Note that, the point \mathbf{Ha} lies inside a sphere of radius d centered at \mathbf{r} if and only if

$$d^2 \geq \|\mathbf{r} - \mathbf{Ha}\|^2 \quad (4.28)$$

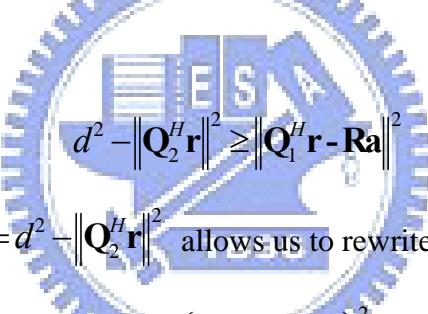
In order to break the problem into the sub-problems described above, it is useful to consider the QR factorization of the matrix \mathbf{H}

$$\mathbf{H} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(N_R - N_T) \times N_T} \end{bmatrix} \quad (4.29)$$

where \mathbf{R} is N_T by N_T upper triangular matrix, and $\mathbf{Q} = [\mathbf{Q}_1 \ \mathbf{Q}_2]$ is an N_R by N_R orthogonal matrix. The matrices \mathbf{Q}_1 and \mathbf{Q}_2 represent the first N_T and last $N_R - N_T$ orthonormal columns of \mathbf{Q} , respectively. The condition (4.28) can be written as

$$\begin{aligned} d^2 &\geq \left\| \mathbf{r} - [\mathbf{Q}_1 \ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{a} \right\|^2 \\ &= \left\| \begin{bmatrix} \mathbf{Q}_1^H \\ \mathbf{Q}_2^H \end{bmatrix} \mathbf{r} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{a} \right\|^2 \\ &= \left\| \mathbf{Q}_1^H \mathbf{r} - \mathbf{R} \mathbf{a} \right\|^2 + \left\| \mathbf{Q}_2^H \mathbf{r} \right\|^2 \end{aligned}$$

In other words



$$d^2 - \left\| \mathbf{Q}_2^H \mathbf{r} \right\|^2 \geq \left\| \mathbf{Q}_1^H \mathbf{r} - \mathbf{R} \mathbf{a} \right\|^2 \quad (4.30)$$

Defining $\mathbf{y} = \mathbf{Q}_1^H \mathbf{r}$ and $d'^2 = d^2 - \left\| \mathbf{Q}_2^H \mathbf{r} \right\|^2$ allows us to rewrite this as

$$d'^2 \geq \sum_{i=1}^{N_T} \left(y_i - \sum_{j=i}^{N_T} r_{i,j} a_j \right)^2 \quad (4.31)$$

where $r_{i,j}$ denotes an (i,j) entry of \mathbf{R} . Here is where the upper triangular property of \mathbf{R} comes in handy. The right-hand side of the above inequality can be expanded as

$$\begin{aligned} d'^2 &\geq (y_{N_T} - r_{N_T, N_T} a_{N_T})^2 \\ &\quad + (y_{N_T-1} - r_{N_T-1, N_T} a_{N_T} - r_{N_T-1, N_T-1} a_{N_T-1})^2 + \dots \end{aligned} \quad (4.32)$$

where the first term only on a_{N_T} , the second term on $\{a_{N_T}, a_{N_T-1}\}$, and so on. Therefore, a necessary condition for $\mathbf{H} \mathbf{a}$ to lie inside the sphere is that $d'^2 \geq (y_{N_T} - r_{N_T, N_T} a_{N_T})^2$. This condition is equivalent to a_{N_T} belonging to the interval

$$\left\lceil \frac{-d' + y_{N_T}}{r_{N_T, N_T}} \right\rceil \leq a_{N_T} \leq \left\lfloor \frac{d' + y_{N_T}}{r_{N_T, N_T}} \right\rfloor \quad (4.33)$$

where $\lceil \cdot \rceil$ denotes rounding to the nearest larger element in the set of numbers that spans the point. Similarly, $\lfloor \cdot \rfloor$ denotes rounding to the nearest smaller element in the set of numbers that spans the point.

Of Course, (4.33) is by no means sufficient. For every a_{N_T} satisfying (4.33), defining

$d'^2 = d'^2 - (y_{N_T} - r_{N_T, N_T} a_{N_T})^2$ and $y_{N_T-1|N_T} = y_{N_T-1} - r_{N_T-1, N_T} a_{N_T}$, a stronger necessary condition can be found by looking at the first two terms in (4.32), which leads to a_{N_T-1} belonging to the interval

$$\left\lceil \frac{-d'_{N_T-1} + y_{N_T-1|N_T}}{r_{N_T-1, N_T-1}} \right\rceil \leq a_{N_T-1} \leq \left\lfloor \frac{d'_{N_T-1} + y_{N_T-1|N_T}}{r_{N_T-1, N_T-1}} \right\rfloor \quad (4.34)$$

One can continue in a similar fashion for a_{N_T-2} and so on until a_1 , thereby obtaining all points belonging to (4.28).

4.2.1.2 Complex Sphere Decoding

The SD algorithm described above applies on a real system where \mathbf{a} is chosen from a real point, but in communication systems we face to deal with complex systems, because of the modulation scheme we used is QPSK. In this case, equation (4.28) becomes as follows:

$$d^2 \geq \|\tilde{\mathbf{r}} - \tilde{\mathbf{H}}\tilde{\mathbf{a}}\|^2 \quad (4.35)$$

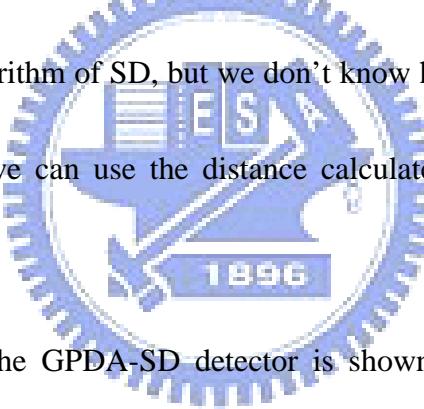
where

$$\begin{aligned}
\tilde{\mathbf{r}} &= \left[\Re\{\mathbf{r}^T\} \ \Im\{\mathbf{r}^T\} \right]^T \\
\tilde{\mathbf{a}} &= \left[\Re\{\mathbf{a}^T\} \ \Im\{\mathbf{a}^T\} \right]^T \\
\tilde{\mathbf{H}} &= \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix}
\end{aligned} \tag{4.36}$$

If \mathbf{a} belongs to QPSK then each entry of $\tilde{\mathbf{a}}$ belongs to BPSK. Thus equation (4.35) can be solved via SD which we introduced in Section 4.2.1.1.

4.2.2 GPDA-SD Detector

In the Section 4.2.1.1, two key problems of the SD algorithm are mentioned. The second one can be solved by the algorithm of SD, but we don't know how to choose the radius in the first problem. In this case, we can use the distance calculated by GPDA as the radius of sphere.



The block diagram of the GPDA-SD detector is shown in Fig. 4.5. As the diagram indicates, it can be divided into three types. First, when SNR is below M dB, which SNR is quite low, we can use the GPDA detector directly to achieve near-optimal solution. Second, when SNR is between M dB and N dB, we can use the GPDA-SD detector to assist the GPDA detector to reach better performance. Finally, when SNR is over N dB, the complexity of the SD algorithm (using the ZF solution to be the radius of the sphere) may be lower than that of the GPDA-SD detector, hence using the SD algorithm directly will be a better choice.

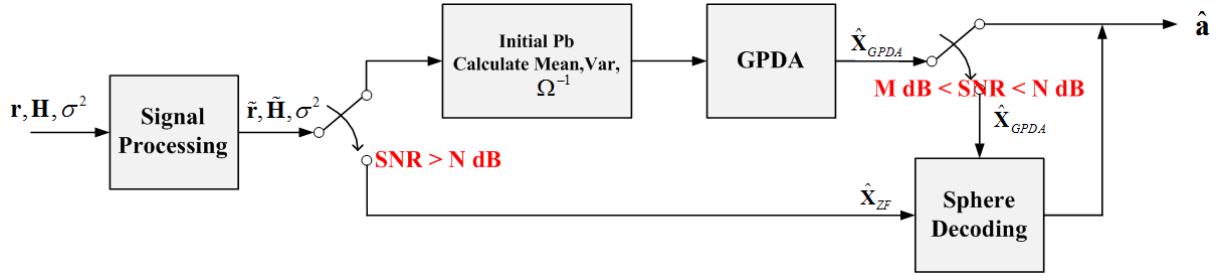


Fig. 4.5 Block diagram of the GPDA-SD detector.

4.3 Simulation Results

Perfect channel information	
Perfect noise variance estimation	
Number of subcarrier	64
Length of cyclic prefix	16
Channel	Rayleigh Fading
Path	2
Relative power (dB)	(0,0)
Number of iterations (GPDA)	2 (if no mention)
Modulation	16QAM (M=16, N=30) 64QAM (M=21, N=35)

Table 4.1 Simulation parameters for MIMO-OFDM system.

In this Section, we compare the BER performance between MCMC and MCPDA, and then we show the influence on the number of iterations and the number of parallel for the MCPDA detector. After that, we compare the BER performance of proposed two detectors

with the optimal ML detector, the SD algorithm, and the GPDA detector. We also compare the complexity of proposed two detectors with that of the SD algorithm. Finally, we will show the effect of channel estimation error for different detection scheme. Note that, our simulations are based on real-valued signal model (i.e. equation (3.21)).

Fig. 4.6 and Fig. 4.7 show the BER performance for the MCMC detector and the MCPDA detector with different initial point, respectively, where obtained by running 5 times iteration. As we can observe in Fig. 4.6, the initial point dominates the BER performance seriously, but in the Fig. 4.7, it does not absolutely dominate the BER performance. Thus, the MCPDA detector solved the initial point problem which in the MCMC detector. Moreover, if the initial point does not affect BER performance too much, why do not we choose an initial point which has lower complexity?

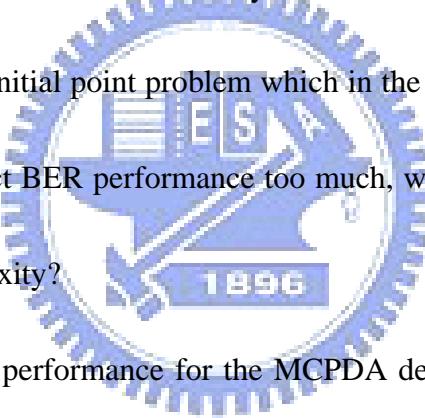


Fig. 4.8 shows the BER performance for the MCPDA detector with different iterations and $q = 16$. The result shows that the MCPDA detector only needs few times iteration to be converged. In other words, with the number of iteration increasing, the BER performance of the MCPDA will get little better. Thus, we must increase number of parallel to reach better BER performance.

Fig. 4.9 shows the BER performance for parallel MCPDA detector with different combinative iterations and $q = 16$. The result shows that when the parallel MCPDA detector with the number of parallel increasing, the BER performance of the parallel MCPDA can be

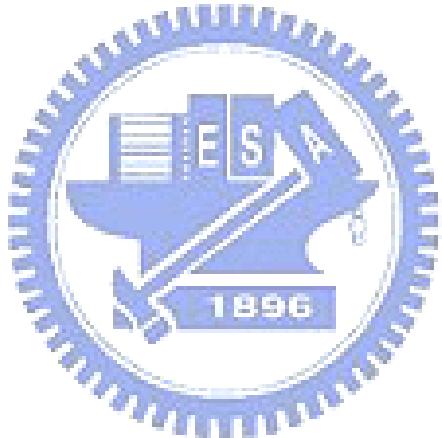
better. But, how many parallels we need? That is another question which is worth us pondering.

Fig. 4.10 shows the BER performance for different detection schemes with $q = 16$. As we can observe from Fig. 4.10, at $\text{BER}=10^{-3}$, the GPDA detector and the ML detection have about 5 dB disparity. The ML detector and the SD algorithm have the same performance; the GPDA-SD detector and the GPDA-MCPDA detector are really close to the ML detector. In this simulation, the SD algorithm uses the distance calculated by ZF as the radius of sphere and GPDA-MCPDA is obtained by running 20 parallel randomly initialized MCPDA; each MCPDA has depth of 2. In Fig. 4.11, it shows the BER performance of the aforementioned detectors for the case $q = 64$.

Fig. 4.12 shows the complexity for different detection schemes with $q = 16$. The result shows that the complexity of the GPDA-SD detector and that of the GPDA-MCPDA are both lower than that of the SD algorithm, especially at low SNR regions. Nevertheless, at the high SNR regions, the complexity of the GPDA-SD detector is significantly less than that of the GPDA-MCPDA. In addition, when the SNR exceeds the N dB, the complexity of the SD algorithm may continue to decrease and less than that of the GPDA-SD detector, thus we will switch the GPDA-SD detector to the SD algorithm. In Fig. 4.13, it shows the complexity of the aforementioned detectors for the case $q = 64$.

Fig. 4.14 shows the channel estimation error for different detection schemes with

$q = 16$, at SNR=25 dB. The result shows that the GPDA detector is not too sensitive to channel, whereas the ML detector, the GPDA-MCPDA detector and the GPDA-SD detector are sensitive to channel. Note that, $E[|\tilde{\mathbf{H}}|^2] = E[|\mathbf{H}|^2] + E[|\Delta\mathbf{H}|^2]$ where $E[|\tilde{\mathbf{H}}|^2]$ is estimated channel composed of real channel $E[|\mathbf{H}|^2]$ and estimation error $E[|\Delta\mathbf{H}|^2]$.



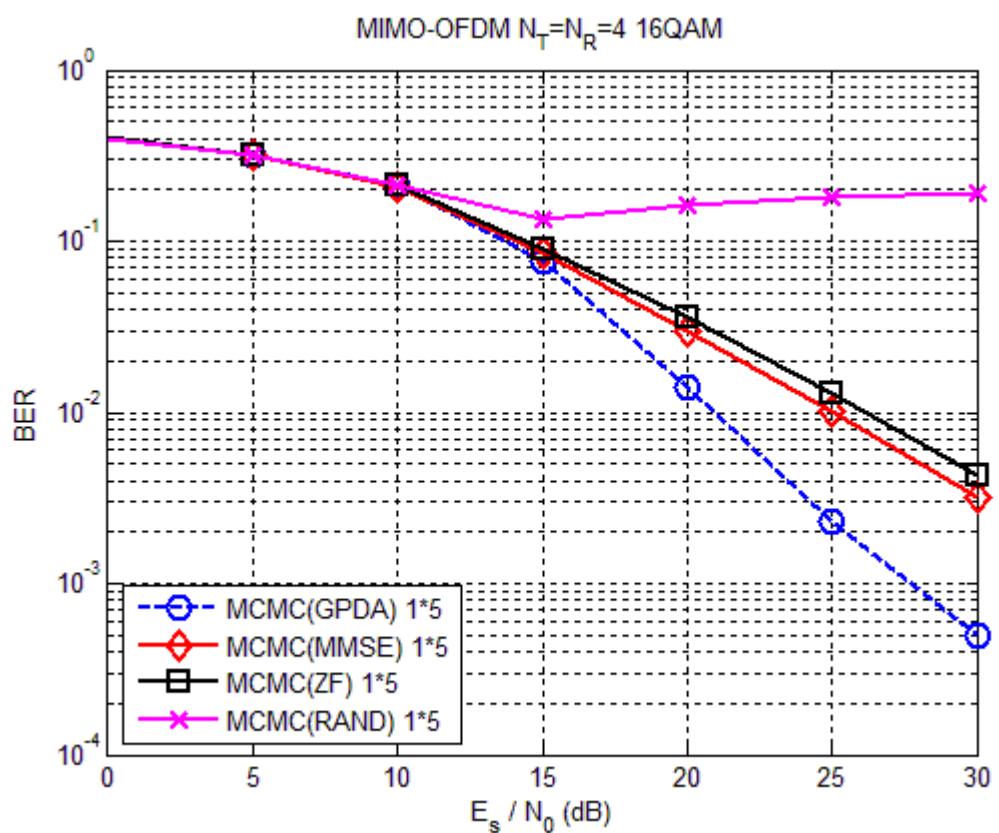


Fig. 4.6 The BER performance for MCMC detector with different initial point.

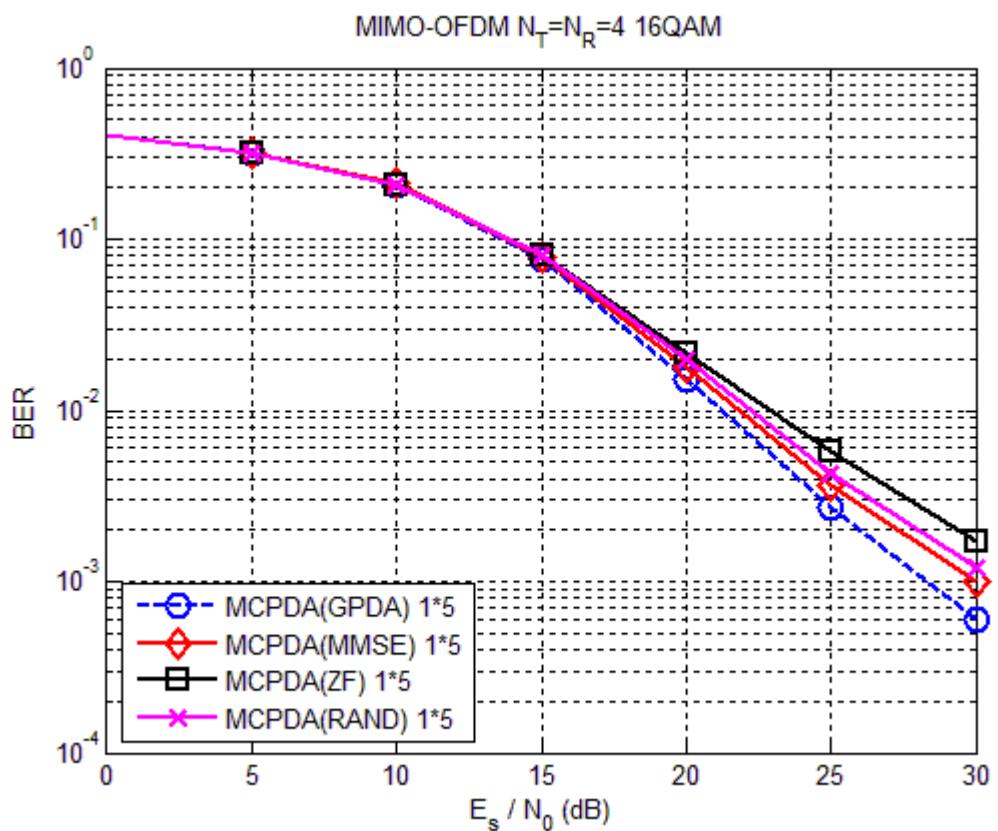


Fig. 4.7 The BER performance for MCPDA detector with different initial point.

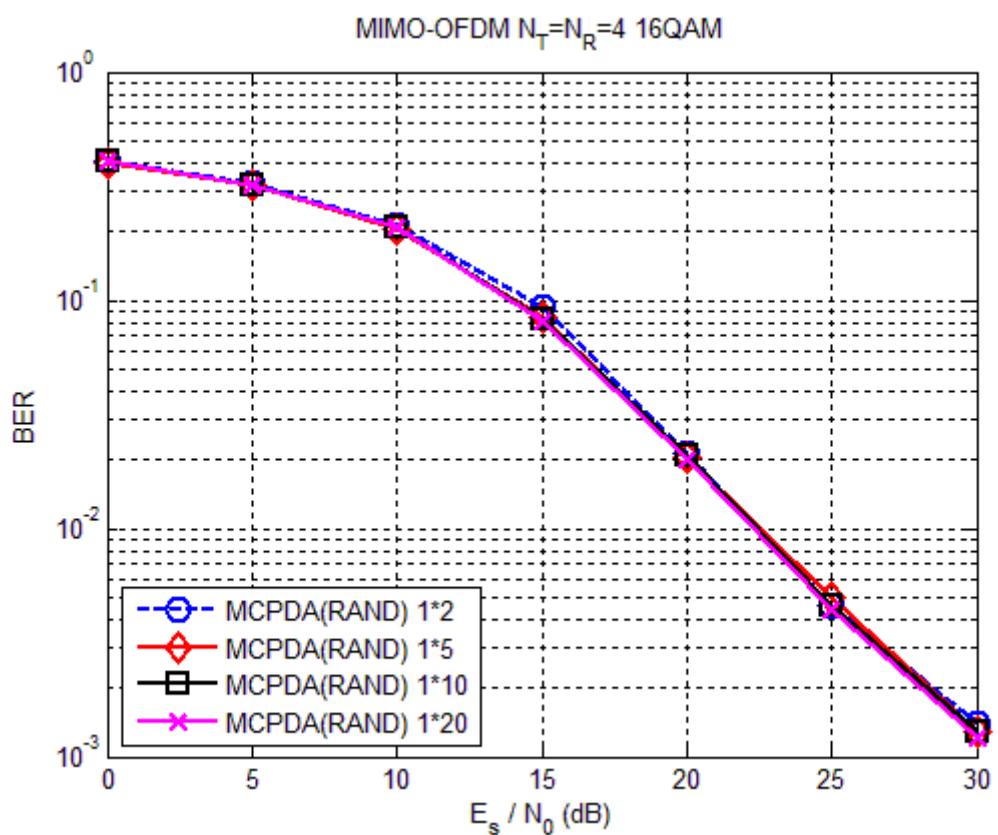


Fig. 4.8 The BER performance for MCPDA detector with different iterations.

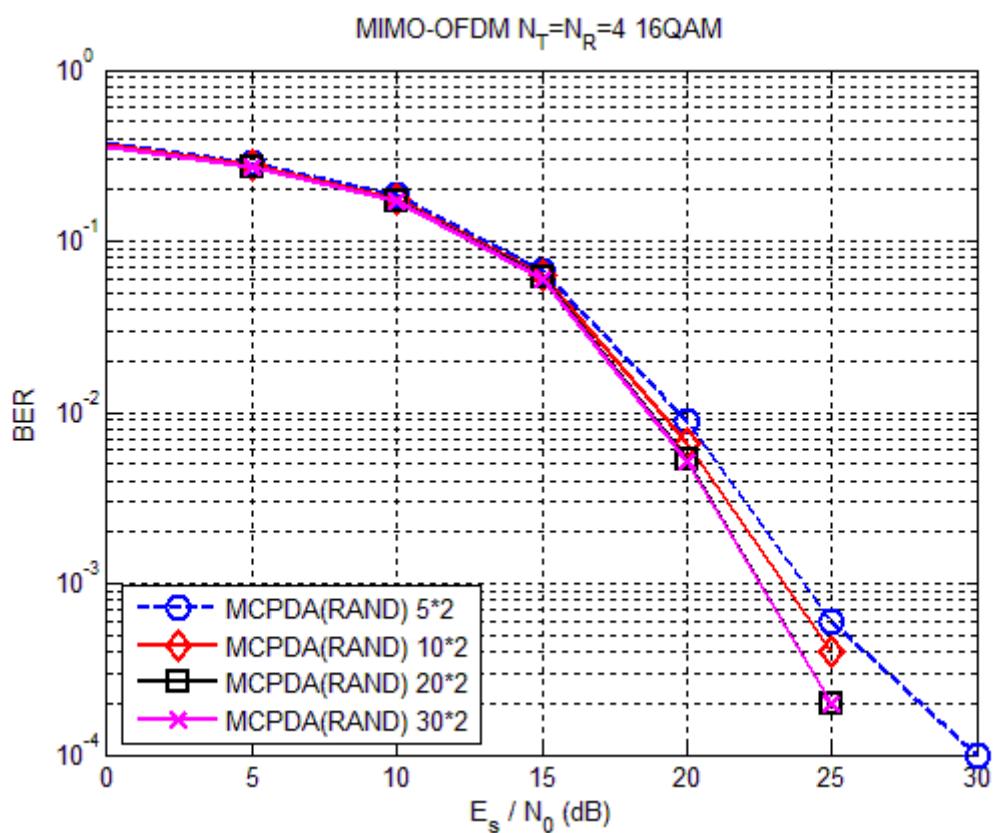


Fig. 4.9 The BER performance for parallel MCPDA with different combinative iterations

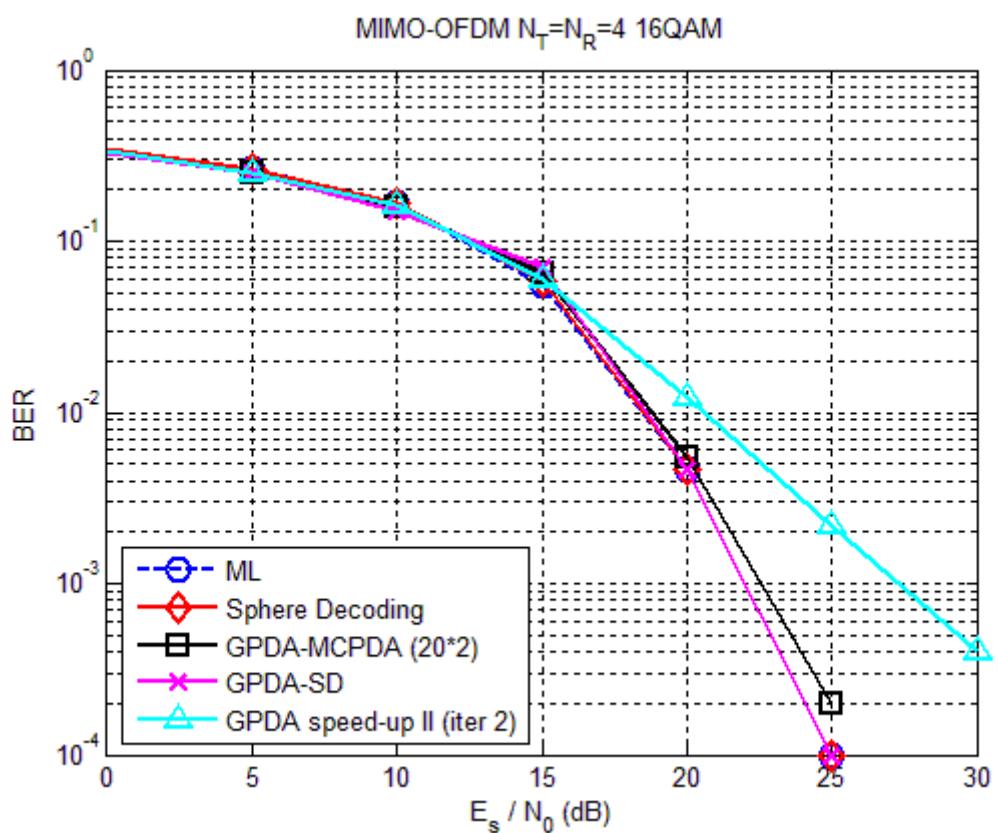


Fig. 4.10 The BER performance for different detection scheme with $q = 16$.

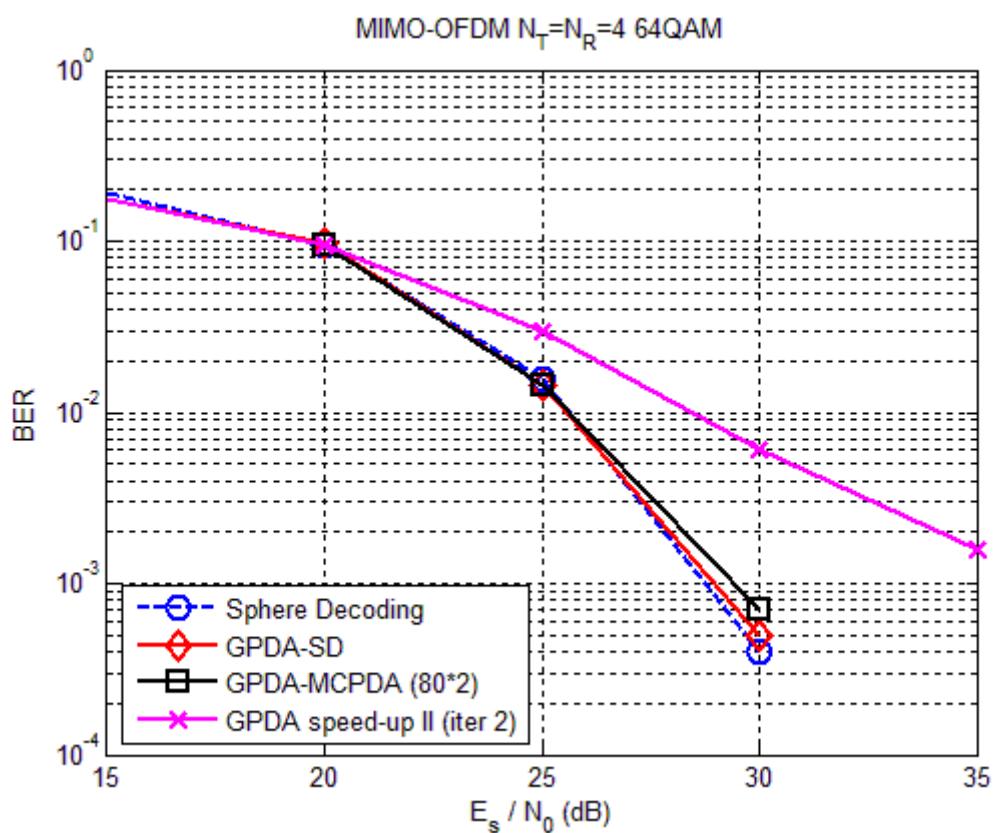


Fig. 4.11 The BER performance for different detection scheme with $q = 64$.

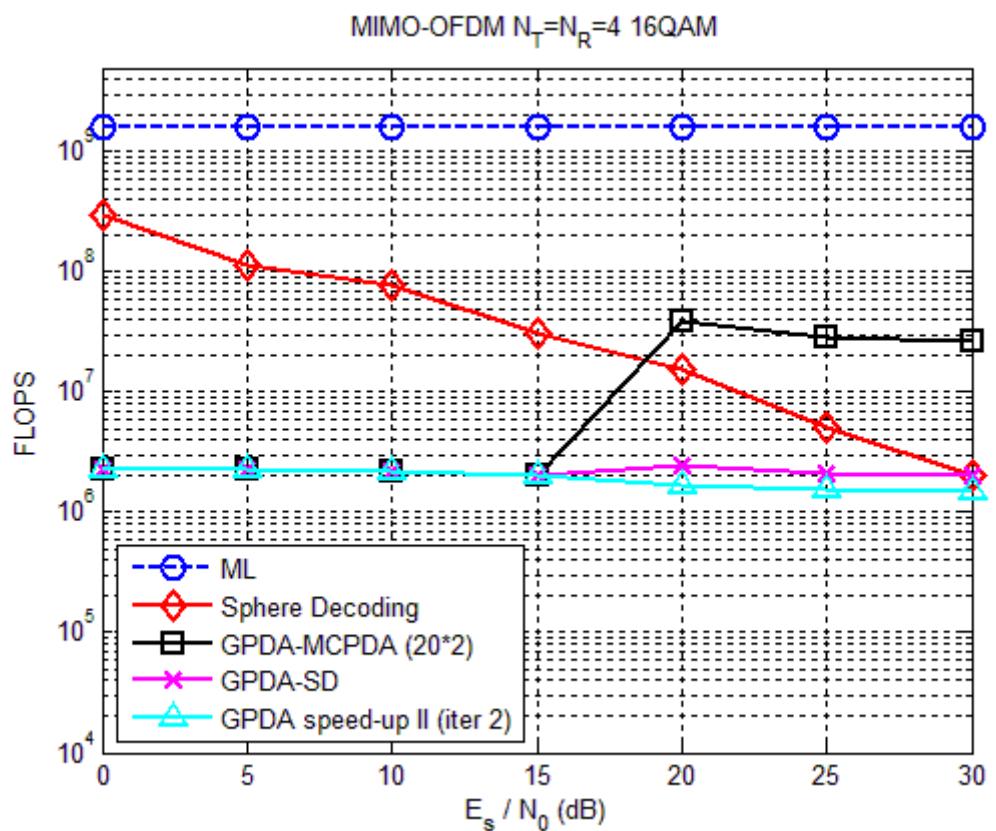


Fig. 4.12 The complexity for different detection scheme with $q = 16$.

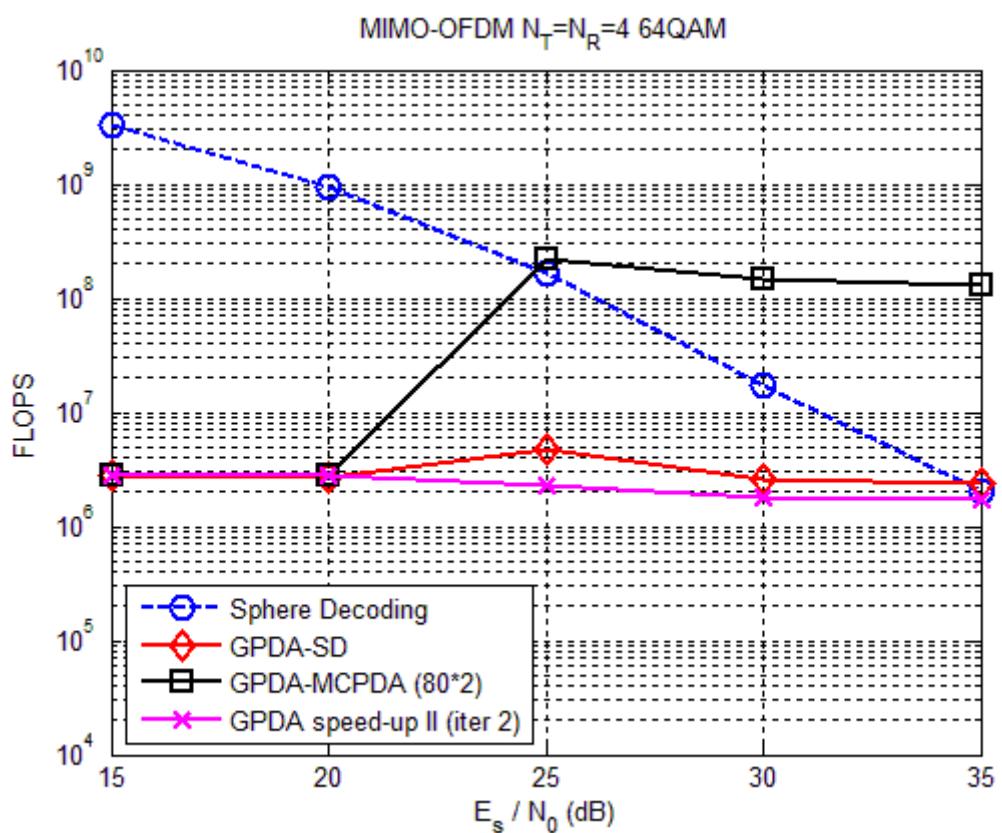


Fig. 4.13 The complexity for different detection scheme with $q = 64$.

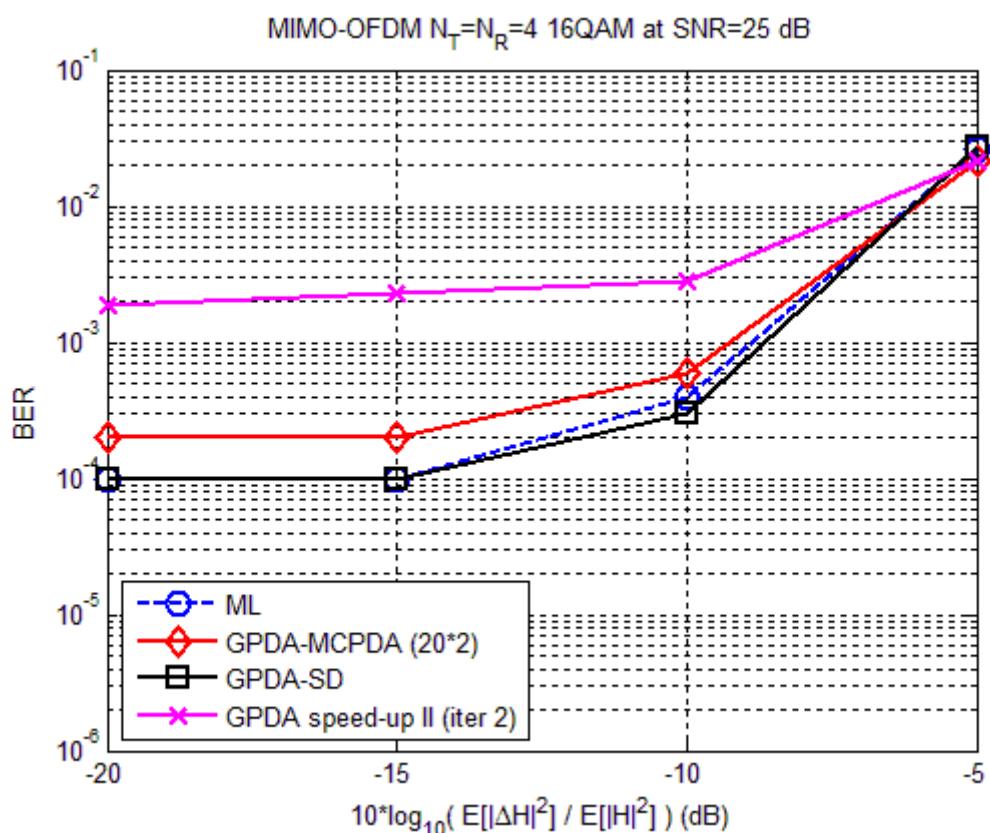


Fig. 4.14 The channel estimation error for different detection scheme.

Chapter 5

On the Equivalence between the SSIC Algorithm and the PDA Algorithm

In this Chapter, we prove the Soft Successive Interference Cancellation (SSIC) algorithm and the PDA algorithm are equivalent. The simulation results demonstrate that the SSIC algorithm and the PDA algorithm have exactly the same BER performance.

5.1 The SSIC Algorithm

In communication systems, the successive interference cancellation algorithm is a simple idea, which has been widely used in many fields. Unfortunately, if some symbols were not reliably detected, it would make the BER performance worse. Therefore, instead of “cancelling by signal subtraction” (which would cause error propagation in case of incorrect decoding), we resort to a “soft” *mean value*, which can deal with reliability information for symbols rather than with hard decisions only, i.e. SSIC [17] algorithm.

To obtain the system model of the SSIC algorithm, we multiply (2.1) from the left by \mathbf{H}^H to obtain

$$\mathbf{y} = \mathbf{G}\mathbf{a} + \mathbf{n} \quad (5.1)$$

where $\mathbf{y} = \mathbf{H}^H \mathbf{r}$, $\mathbf{G} = \mathbf{H}^H \mathbf{H}$, and $\mathbf{n} = \mathbf{H}^H \mathbf{v}$.

The model of the SSIC algorithm is obtained by multiplying (5.1) from the left by \mathbf{G}^{-1} to yield

$$\tilde{\mathbf{y}} = \mathbf{a} + \tilde{\mathbf{n}} = \mathbf{e}_i a_i + \sum_{j \neq i} \mathbf{e}_j a_j + \tilde{\mathbf{n}} \quad (5.2)$$

In (5.2), $\tilde{\mathbf{y}} = \mathbf{G}^{-1}\mathbf{y}$, $\tilde{\mathbf{n}} = \mathbf{G}^{-1}\mathbf{n}$, $a_i \in X_i = \{x_i(m)\}$, $i \in [1, N_T]$ and \mathbf{e}_i is a column vector whose i th element is 1 and the other element are 0. In order to obtain computational efficiency and to compare with the PDA algorithm under the fair condition, we choose (5.2) as the system model for the SSIC algorithm.

After interference-cancelled received signal vector \mathbf{z} can be expressed as follows:

$$\mathbf{z} = \tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] = \mathbf{e}_i a_i + \underbrace{\sum_{j \neq i, j \in D} \mathbf{e}_j (a_j - E[a_j])}_{\mathbf{I}_D} + \underbrace{\sum_{j \neq i, j \notin D} \mathbf{e}_j a_j + \tilde{\mathbf{n}}}_{\mathbf{I}_U} \quad (5.3)$$

where \mathbf{I}_U is the interferences from undetected symbols, \mathbf{I}_D is the interference due to the decision error of previously detected symbols, and we define D to be a set of symbols which have been detected. Thus, to estimate the associated probabilities for an element a_i , we treat all other elements $a_j (j \neq i)$ as multivariate random variables, and from equation (5.3), we define

$$\mathbf{B}_i = \sum_{j \neq i, j \in D} \mathbf{e}_j (a_j - E[a_j]) + \sum_{j \neq i, j \notin D} \mathbf{e}_j a_j + \tilde{\mathbf{n}} \quad (5.4)$$

as the effective noise on a_i , and approximate it as a Gaussian noise with matched mean and covariance:

$$\bar{\mathbf{B}}_i = \sum_{j \neq i, j \notin D} \mathbf{e}_j E[a_j] \quad (5.5)$$

$$\begin{aligned}
\Sigma_i &= \sum_{j \neq i, j \in D} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j - E[a_j]] + \sum_{j \neq i, j \notin D} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j] + \sigma^2 \mathbf{G}^{-1} \\
&= \sum_{j \neq i, j \in D} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j] + \sum_{j \neq i, j \notin D} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j] + \sigma^2 \mathbf{G}^{-1} \\
&= \sum_{j \neq i} \mathbf{e}_j \mathbf{e}_j^T \text{Var}[a_j] + \sigma^2 \mathbf{G}^{-1}
\end{aligned} \tag{5.6}$$

where $\bar{\mathbf{B}}_i = E[\mathbf{B}_i]$ and $\Sigma_i = \text{Cov}[\mathbf{B}_i]$. In (5.5) and (5.6), $E[a_j]$ and $\text{Var}[a_j]$ are given by

$$E[a_j] = \sum_m x_j(m) p_j(m) \tag{5.7}$$

$$\text{Var}[a_j] = \sum_m x_j^2(m) p_j(m) - (E[a_j])^2 \tag{5.8}$$

Therefore, we can calculate $P(\mathbf{z} | a_i = x_i(m))$ as follows:

$$P(\mathbf{z} | a_i = x_i(m)) \propto \exp\left(-(\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i)^H \Sigma_i^{-1} (\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i)\right) \tag{5.9}$$

Then, we let



$$\eta_i = \mathbf{z} - \bar{\mathbf{B}}_i \tag{5.10}$$

and

$$\begin{aligned}
\beta_i(m) &= -(\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i)^H \Sigma_i^{-1} (\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i) \\
&\propto (2\mathbf{z} - \mathbf{e}_i x_i(m) - 2\bar{\mathbf{B}}_i)^H \Sigma_i^{-1} \mathbf{e}_i x_i(m) \\
&= (2\eta_i - \mathbf{e}_i x_i(m))^H \Sigma_i^{-1} \mathbf{e}_i x_i(m) \\
&= \left(\sum_j [2\eta_i^H]_j [\Sigma_i^{-1}]_{ji} - x_i^H(m) [\Sigma_i^{-1}]_{ii} \right) x_i(m)
\end{aligned} \tag{5.11}$$

where $[2\eta_i^H]_j$ is the j th element of $2\eta_i^H$ and $[\Sigma_i^{-1}]_{ji}$ is the element (j,i) of Σ_i^{-1} . The posterior probability $P_i(m)$ is then given as

$$P_i(m) = \frac{\exp(\beta_i(m))}{\sum_l \exp(\beta_i(l))} \tag{5.12}$$

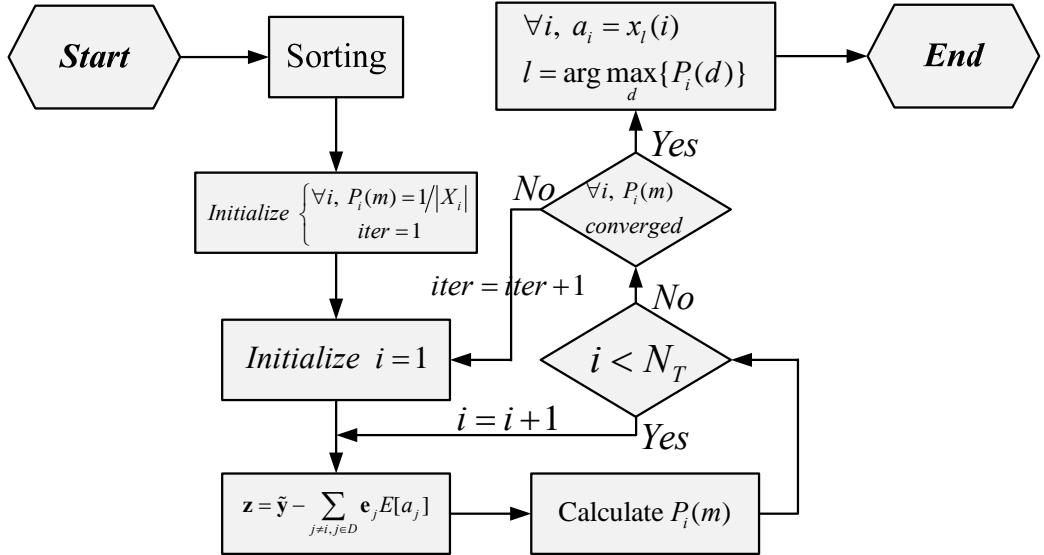


Fig. 5.1 Block diagram of the basic SSIC procedure.

The basic procedure for the SSIC detector is as follows:

1. Based on the matrix \mathbf{G} in (5.1), we obtain the optimal detection sequence proposed for the V-BLAST OSIC in [9] and denote the sequence as $\{k_i\}_{i=1}^{N_T}$.
2. Initialize the probabilities as $P_i(m) = 1/|X_i| \forall m, \forall i$, and set the iteration counter $iter = 1$.
3. Initialize $i = 1$.
4. Let $\mathbf{z} = \tilde{\mathbf{y}} - \sum_{k_j \neq k_i, k_j \in D} \mathbf{e}_{k_j} E[a_{k_j}]$.
5. Calculate $P(\mathbf{z} | a_{k_i} = x_{k_i}(m))$ by approximate the \mathbf{z} in (5.3) as Gaussian distributed, and set the results equal to the corresponding elements of \mathbf{p}_{k_i} .
6. If $i < N_T$, let $i = i + 1$ and back to step 4. Otherwise, carry on step 7.
7. If $\forall i$, \mathbf{p}_i has converged, go to step 8. Otherwise, let $iter = iter + 1$ and return to step 3.
8. For $j = 1, \dots, N_T$, make a decision \hat{a}_j for a_j via

$$\hat{a}_j = x_j(l), \quad l = \arg \max_m \{ p_j(m) \} \quad (5.13)$$

The block diagram of the basic SSIC procedure is shown in Fig. 5.1.

5.2 The Equivalence of the SSIC Algorithm and the PDA Algorithm

In Section 5.1, we can calculate $P(\mathbf{z}|a_i = x_i(m))$ by equation (5.9) as follows:

$$P(\mathbf{z}|a_i = x_i(m)) \propto \exp(-(\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i)^H \Sigma_i^{-1} (\mathbf{z} - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i))$$

Substituting equation (5.3) in to equation (5.9), we can get

$$P\left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] \middle| a_i = x_i(m)\right) \propto \exp\left(-\left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i\right)^H \Sigma_i^{-1} \left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m) - \bar{\mathbf{B}}_i\right)\right) \quad (5.14)$$

Substituting equation (5.5) in to equation (5.14), we can obtain

$$\begin{aligned} P\left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] \middle| a_i = x_i(m)\right) &\propto \exp\left(-\left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m) - \sum_{j \neq i, j \notin D} \mathbf{e}_j E[a_j]\right)^H \Sigma_i^{-1} \left(\tilde{\mathbf{y}} - \sum_{j \neq i, j \in D} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m) - \sum_{j \neq i, j \notin D} \mathbf{e}_j E[a_j]\right)\right) \\ &\propto \exp\left(-\left(\tilde{\mathbf{y}} - \sum_{j \neq i} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m)\right)^H \Sigma_i^{-1} \left(\tilde{\mathbf{y}} - \sum_{j \neq i} \mathbf{e}_j E[a_j] - \mathbf{e}_i x_i(m)\right)\right) \\ &= P(\tilde{\mathbf{y}}|a_i = x_i(m)) \quad (\text{Equation (3.9)}) \\ &\propto \exp\left(-\left(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j \neq i} \mathbf{e}_j E[a_j]\right)^H \Omega_i^{-1} \left(\tilde{\mathbf{y}} - \mathbf{e}_i x_i(m) - \sum_{j \neq i} \mathbf{e}_j E[a_j]\right)\right) \end{aligned} \quad (5.15)$$

where $\Sigma_i = \text{Cov}[\mathbf{B}_i] = \text{Cov}[\mathbf{N}_i] = \Omega_i$ can be observed in equation (3.6) and equation (5.6).

Therefore, using equation (3.9) and equation (5.9) can obtain the same posterior probability

$P_l(m)$. In other words, we confirm that the SSIC algorithm and the PDA algorithm are

equivalent.

5.3 Simulation Results

Perfect channel information	
Perfect noise variance estimation	
Number of subcarrier	64
Length of cyclic prefix	16
Channel	Rayleigh Fading
Path	2
Relative power (dB)	(0,0)
Modulation	16QAM

Table 5.1 Simulation parameters for SSIC algorithm and PDA algorithm.

In this Section, we show the BER performance for the SSIC algorithm and the PDA algorithm. Furthermore, we will compare the complexity for the SSIC algorithm and the PDA algorithm. The simulation parameters are shown in Table 5.1.

Fig. 5.2 shows the BER performance for the SSIC algorithm and the PDA algorithm. As we can observe from Fig. 5.2, the SSIC algorithm has same the BER performance with the PDA algorithm.

Fig. 5.3 shows the complexity for the SSIC algorithm and the PDA algorithm. The result

shows that the complexity of the SSIC algorithm is slightly more than that of the PDA algorithm. Since the SSIC algorithm needs to subtract the soft information form $\tilde{\mathbf{y}}$.

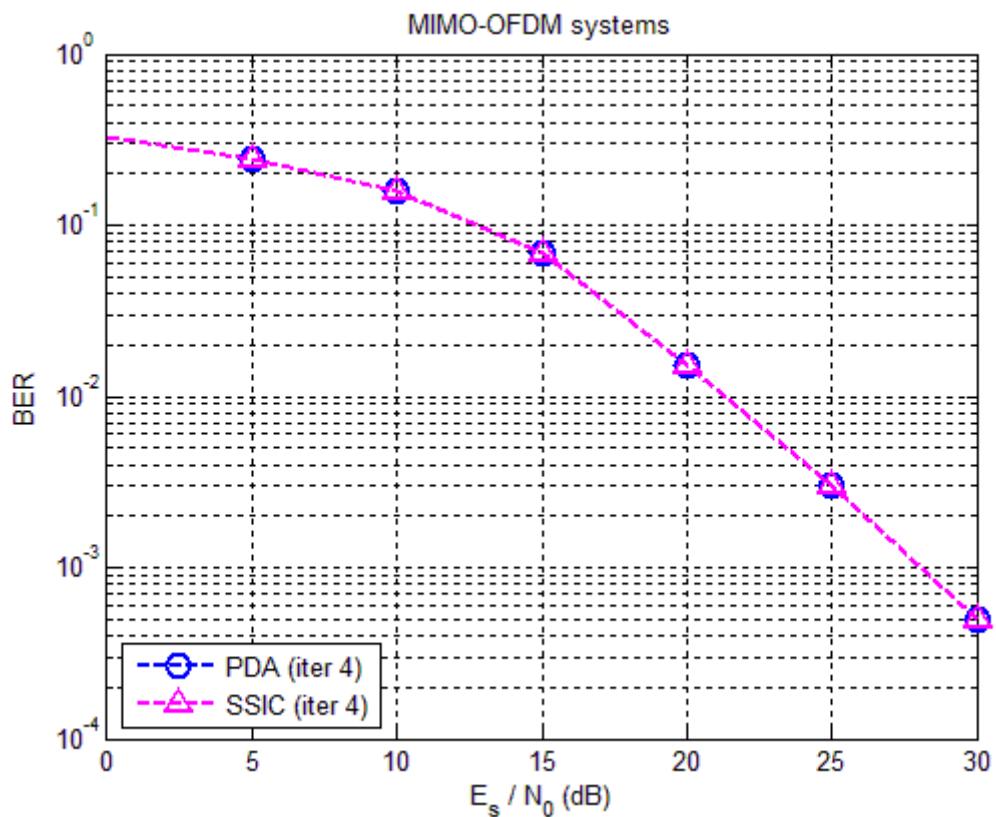


Fig. 5.2 The BER performance for the SSIC algorithm and the PDA algorithm.

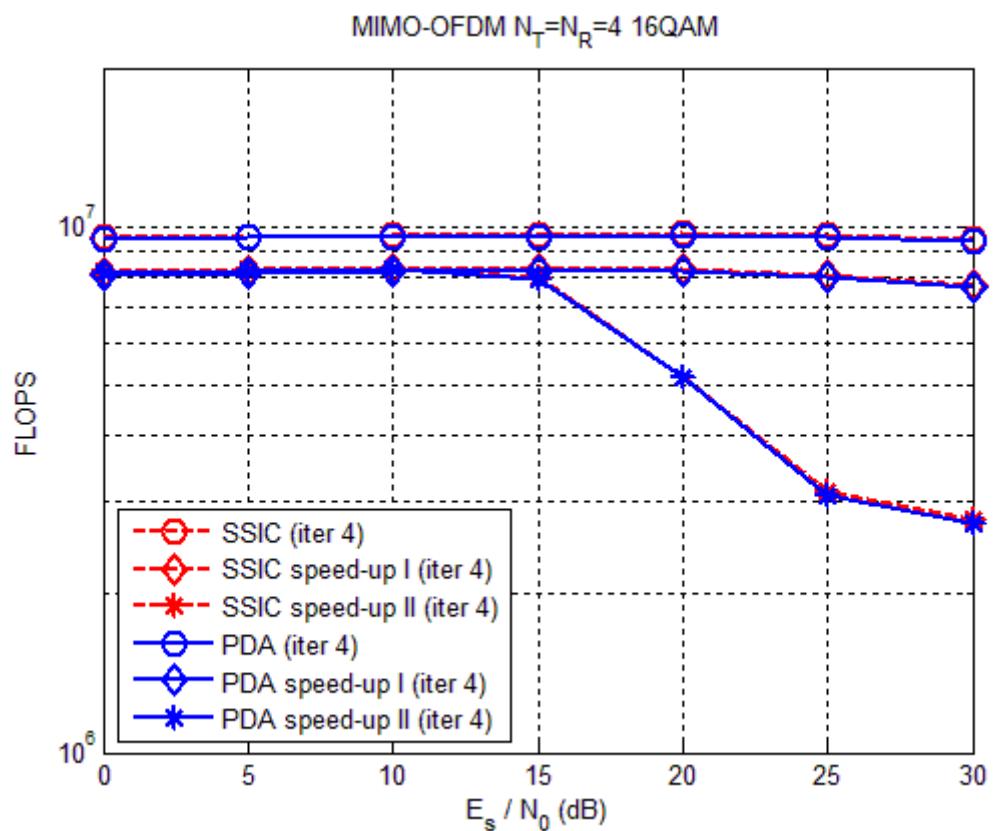
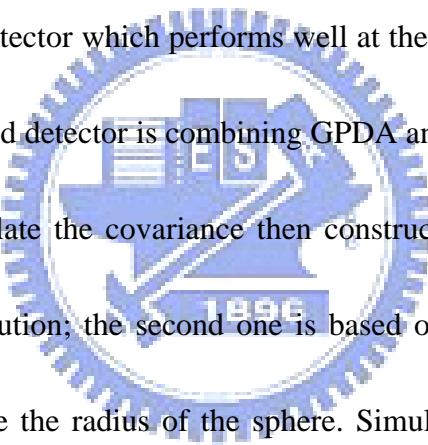


Fig. 5.3 The complexity for the SSIC algorithm and the PDA algorithm.

Chapter 6

Conclusion

In this thesis, we establish the equivalence of the SSIC algorithm and the PDA algorithm. Furthermore, we proposed two detectors, GPDA-MCPDA and GPDA-SD, to reduce the complexity of the SD algorithm for near-optimal detection in a MIMO-OFDM spatial multiplexing system with higher order QAM constellations (16QAM/64QAM). These two methods exploit the GPDA detector which performs well at the low SNR regions. At the high SNR regions, the first proposed detector is combining GPDA and MCMC, which incorporates the concept of PDA to calculate the covariance then construct a Markov Chain to make it converge to the target distribution; the second one is based on the SD algorithm using the GPDA detector solution to be the radius of the sphere. Simulation results demonstrate that both detectors can achieve near-optimal performance with lower complexity as compare with the SD algorithm, especially at the low SNR regions.



Bibliography

- [1] G. Stuber et al., "Broadband MIMO-OFDM wireless communications," Proceedings of the IEEE, vol. 92, no. 2, pp. 271-294, Feb. 2004.
- [2] A. v. Zelst and T. Schenk, "Implementation of a MIMO OFDM-based wireless LAN system," IEEE Trans. on Sign. Proc., vol. 52, no. 2, pp. 483-494, Feb. 2004.
- [3] B. Hassibi and H. Vikalo, "On sphere decoding algorithm. I. Expected complexity," IEEE Trans. on Sign. Proc., vol. 53, no. 8, pp. 2806-2818, Aug. 2005.
- [4] J. Luo, K. et al., "Near-optimal multiuser detection in synchronous CDMA using probabilistic data association," IEEE Commun. Lett., vol. 5, pp. 361-363, Sept. 2001.
- [5] D. Pham et al., "A generalized probabilistic data association detector for multiple antenna systems," IEEE Commun. Lett., vol. 8, pp. 205-207, Apr. 2004.
- [6] C.P. Robert and G. Casella, Monte Carlo statistical Methods, Springer-Verlag, New York, 1999.
- [7] P. W. Wolniansky et al., "V-Blast: An architecture for realizing very high data rates over the rich-scattering channel," in Proc. Int. Symp. Signals, Systems and Electronics, pp. 295-300, Sept. 1998.
- [8] O. Edfors et al., "An introduction to orthogonal frequency-division multiplexing," Division of Signal Processing, Lulea University of Technol., Sweden, Research Report 1996:16, Sept. 1996.
- [9] G. D. Golden et al., "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," Electron. Lett., vol. 35, no. 1, pp. 14-16, Jan. 1999.
- [10] W. W. Hager, "Updating the Inverse of a Matrix," SIAM Review, vol. 31, no. 2, pp. 221-239, June 1989.
- [11] S. Liu and Z. Tian, "Near-optimal soft decision equalization for frequency selective

MIMO channels,” IEEE Trans. on Sign. Proc., vol. 52, no. 3, pp. 721-733, Mar. 2004.

[12] X. Mao et al., “Markov Chain Monte Carlo MIMO Detection Methods for High Signal-to-Noise Ratio Regimes,” IEEE Globecom 2007 Wireless Communications , Washington DC, Nov. 26-30, 2007.

[13] B. Farhang-Boroujeny et al., “Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems,” IEEE Trans. on Sign. Proc., vol. 54, no. 5, pp. 1896-1909, May 2006.

[14] <http://icl.cs.utk.edu/papi/>; <http://www.mathworks.com/>.

[15] G. Casella and E. I. George, “Explaining the Gibbs sampler,” The American Statistician, vol. 46, no. 3, pp. 167–174, Aug. 1992.

[16] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” Mathematics of Computation, vol. 44, pp. 463-471, Apr. 1985

[17] L. Song et al., “Successive Interference Cancellation Schemes for Time-Reversal Space-Time Block Codes,” IEEE Trans. on Veh. Technol., vol. 57, no. 1, pp. 642-648, Jan. 2008.