SYMBOL BY SYMBOL SOFT-INPUT SOFT-OUTPUT MULTIUSER DETECTION FOR FREQUENCY SELECTIVE MIMO CHANNELS

by

Sara Bavarian
Bachelor of Engineering, Sharif University of Technology 2001

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

In the School
of
Engineering Science

SIMON FRASER UNIVERSITY

Fall 2004

APPROVAL

Name: **Sara Bavarian**

Degree: **Master of Applied Science**

Title of Thesis: **Symbol by Symbol Soft-Input Soft-Output Multiuser Detection for Frequency Selective MIMO Channels**

Examining Committee:

Chair: **William A. Gruver**
Professor of School of Engineering Science

____

James K. Cavers
Senior Supervisor
Professor of School of Engineering Science

____

Paul Ho
Supervisor
Professor of School of Engineering Science

____

Rodney Vaughan
Examiner
Professor of School of Engineering Science

Date Defended: ______________________________
2004-12-03

ABSTRACT

We introduce a symbol by symbol, soft-input soft-output (SISO) multiuser detector for frequency selective multiple-input multiple-output (MIMO) channels. The basic principle of this algorithm is to extract a posteriori probabilities (APPs) of all interfering symbols at each symbol interval and then feed these updated APPs as a priori probabilities (apPs) for joint APP extraction in the next symbol interval. Unlike near-optimal block oriented sphere decoding (SD) and soft decision equalization (SDE), the computational complexity of this updating APP (UA) algorithm is linear in the number of symbols but the exponential computational load of optimal joint APP extraction makes the basic UA impractical. To decrease computations we replace the optimal joint APP extractor by a groupwise SISO multiuser detector with a soft sphere decoding core. The resulting reduced complexity updating APP (RCUA) equalizer is flexible in different situations and outperforms the traditional sub-optimal MMSE-DFE without increasing the computational costs substantially.

ACKNOWLEDGEMENTS

I am greatly honoured to thank many people who helped me throughout this research.

I am really grateful to my supervisor, Professor Jim Cavers for bringing me into this program and providing generous support. He has been a great teacher, mentor and advisor in leading me through the field of wireless communication. I wish to thank him for his patience and constant encouragements and giving me the opportunity and freedom to pursue my research. His profound knowledge and scientific curiosity has been a great source of inspiration.

I have to thank my co-supervisor, Professor Paul Ho. His course "mobile and personal communications" gave me a solid base to continue my research. I also have to thank my examiner, Professor Rodney Vaughan for his inspiring course on MIMO channels and for introducing me to the colourful Kiwi dialect during our discussions. I am also grateful that he spent his time on examining of my work with his busy schedule. I feel grateful to all colleagues in Mobile Communication Lab for their support, encouragement and friendship. Special thanks to Shirin, Brad, Mike and Serhat.

I have been fortunate to be in a really friendly environment in Engineering Science Department of Simon Fraser University. For this I am grateful to everybody in the department. Special thank to Raj and Lynn and other staff for their kind, constant help. I also want to thank Penny Simpson, library thesis assistant for her informing workshops and being there to help with writing problems.

Very special thanks to all my family in Iran for all their help and support all my life specially my mom and dad who taught me perseverance. My deepest gratitude to my dearest husband Lawrence and my step son Lawson for their help, support, and encouragement during the long days of work and hot dinners at night.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

apP:	*a priori* Probability
APP:	*A Posteriori* Probability
BCJR:	Bahl, Cocke, Jelinek, Raviv (MAP algorithm)
BER:	Bit Error Rate
CDMA:	Code Division Multiple Access
DDFSE:	Delayed Decision Feedback Sequence Estimation
DFE:	Decision Feedback Equalizer
FDMA:	Frequency Division Multiple Access
FIR:	Finite Impulse Response
FSM:	Finite State Machine
i.i.d.:	independent identically distributed
IMUD:	Iterative Multiuser Detection
ISI:	Intersymbol Interference
JMAP:	Joint Maximum *A Posteriori*
JML:	Joint Maximum Likelihood
LE:	Linear Equalizer
LLR:	Log Likelihood Ratio
MAP:	Maximum *A Posteriori*
MIMO:	Multiple Input Multiple Output
ML:	Maximum Likelihood
MLSE:	Maximum Likelihood Sequence Estimation
MMSE:	Minimum Mean Square Error
MUD:	Multiuser Detection
MUI:	Multiuser Interference
NP-problem:	Nondeterministic Polynomial time
PDA:	Probabilistic Data Association
QoS:	Quality of Service
RCUA:	Reduced Complexity UA

RSSE: Reduced State Sequence Estimation

SD: Sphere Decoding

SDE: Soft Decision Equalization

SER: Symbol Error Rate

SISO: Soft Input Soft Output

SNR: Signal to Noise Ratio

SOVA: Soft Output Viterbi Algorithm

TDMA: Time Division Multiple Access

UA: Updating APP

VA: Viterbi Algorithm

ZF: Zero Forcing

CHAPTER 1 INTRODUCTION

Digital wireless communications systems are now part of our day to day life; we watch satellite TVs, talk on cellular phones and use wireless networking to access the internet. With the ever-increasing demand for wireless communications, service providers need to increase the capacity of their system, while maintaining an acceptable quality of service (QoS). Satisfying this need by allocating more bandwidth is not an economical. Therefore, it is necessary to develop new methods that are efficiently exploit limited bandwidth, and at the same time are simple enough to be practical.

By 1948, Shannon had developed fundamental limits on the efficiency of single-input, single-output communication over a noisy channel. For decades people believed that these limits are the ultimate, but practically unattainable, goals in digital communication. Finally in 1993 the invention of turbo codes by Berrou et al. made it possible for communication systems to work near the Shannon limit [24] but this was not the end of the story. People started to realize that exploiting antenna arrays can increase the capacity of the system. In 1998, Foschini and Gans (also Winters 1987) generalized the Shannon capacity formula for the case of a multiple-input, multiple-output (MIMO) system. The results promised to increase spectral efficiency far beyond single-input single-output Shannon limit [1].

In order to enjoy this huge increase in capacity, we need to develop efficient and reliable receivers for MIMO systems. This design can be challenging, especially in frequency selective channels where the transmitted signal has to be detected in the presence of noise, intersymbol interference (ISI) and multiuser interference (MUI).

Optimal detection in this environment involves solving an integer least squares problem, which is in general NP hard. The BCJR maximum a posteriori (MAP) [2] and Viterbi maximum likelihood (ML) equalizers find the optimal solutions but are not practical because their computational complexity grows exponentially with the product of channel memory size and number of users. Standard suboptimal reduced-complexity methods [3-6] as well as linear prefiltering [7] can be used to decrease the computational load. However, these methods are not so popular owing to the practical difficulties of adapting them to MIMO channels and the trade-off between their performance and computational costs.

For complexity reasons, typical equalizers use heuristic methods that can be linear, such as zero-forcing (ZF) and minimum mean square error (MMSE) equalizers, or nonlinear such as decision feedback equalizers (DFE). Linear equalizers are the simplest, but DFE shows better performance at high SNRs while still having much lower complexity than optimal BCJR and VA. In DFE the effects of past symbols are subtracted using their estimates.

This approach can suffer from error propagation at low SNRs. The other problem with these methods is that they can not handle overloaded situations, where the number of transmit antennas is more than the number of receive antennas, as in a cellular system uplink. Optimal MIMO DFE solutions have been investigated in [8-9], and in [10] low complexity SISO LE and DFE equalizers are developed based on the MMSE criterion.

Recently, some quasi-ML, block detection techniques have been developed that can provide near-optimal performance with low computational complexity under some conditions. Sphere decoding (SD) and soft decision equalization (SDE) are important and will be considered in this thesis because they have already been used in the literature for multiuser detection in a frequency selective MIMO channel.

The SD algorithm proposed by Fincke and Pohst [11] finds all signal points in a sphere of given radius centered on the received vector. Instead of an exhaustive search over all possible data vectors for the least squares solution, we can restrict our search only to the points in the sphere and thereby reduce computation. This technique has many applications in communications. Reference [12] investigates its usage in frequency selective MIMO channels.

The other technique, SDE [13], is based on probabilistic data association (PDA) filtering [14]; it processes the received signal using an iterative posterior probability updating and PDA-type Gaussian forcing. The next chapter covers these methods in more details.

The desirability of these algorithms depends on the situation. SD finds the optimal solution and its computational complexity is shown to be linear in constellation size and polynomial (less than third order) in the number of symbols in the block over a wide range of SNRs, provided that the system is not overloaded [15].

SDE attains near-optimal performance with complexity that is comparable to that of SD. In low SNRs, the complexity of both methods grows; in SD, the sphere radius grows, and in SDE the required number of iterations increases (3-4 iterations at high SNRs and 7-14 iterations at low SNRs). In overloaded conditions where the number of transmit antennas is more than the number of receive antennas, SD complexity climbs exponentially in the number of excess users [15]. SDE complexity is much less sensitive to overload, although performance is somewhat degraded.

SDE provides soft decisions, but it makes no use of a priori information (it does not "take hints"). So SDE equalizers cannot be used in turbo equalization systems where the flow of extrinsic data between the equalizer and decoder achieves higher capacity. The original SD provides only hard decisions but some variations of it have been proposed in [15,18] that provide soft decisions and also utilize a priori probabilities (apPs) in their input, so these equalizers can be used in iterative decoding and equalization systems.

These imperfections are natural to any quasi-ML method and vary from one to another, but the main problem of SD and SDE lies in their block detection processing. Because their computational complexity grows cubically with the number of symbols in the block, we cannot send frames of long or infinite lengths. Further, the algorithms require interblock zero padding. This causes significant losses in capacity when the frames are short and the channel memory is long.

In this thesis, our goal is to develop a sub-optimal symbol by symbol soft-input soft-output (SISO) multiuser detection technique. With complexity that is linear in block size, it avoids the capacity loss of block oriented algorithms. Our updating *a posteriori* probabilities (UA) equalizer processes the signal acquired in each symbol time separately. It is based on calculating joint *a posteriori* probabilities (APP) of all interfering symbols at each symbol time, then updating the APPs at the new symbol time.

So, unlike block oriented techniques, the computational complexity of UA equalizer grows linearly with the number of symbols.

The complexity of optimal joint APP extraction is exponential in the product of number of users and memory length. We therefore use a suboptimal iterative group detection technique inspired by iterative multiuser detection algorithm (IMUD) [19]. We also suggest implementing the core APP extractor in the group detector by a soft SD.

The UA algorithm makes use of the apPs so it can be used in popular iterative decoding systems. Also, because of its symbol-by-symbol nature, it can share information between different base-stations (macrodiversity). This is the subject of our future research. UA performance is far better than sub-optimal DFE methods and is close to optimal except when it is working in an overloaded situation and even then it is also uniquely flexible in that it does not fail or become too computationally complex.

In the rest of this thesis, we first cover the background material in more detail and then we explain UA idea in detail and describe the methods of complexity reduction for basic UA algorithm. As well, we analyse the computational complexity of different algorithms and compare their performances through simulations.

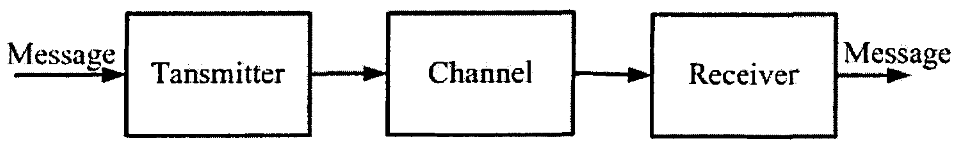
CHAPTER 2 BACKGROUND

In this chapter, we cover the background material and establish the models that will be used throughout this thesis.

2.1 Multiuser Communications

Figure 2.1 shows the basic elements of a digital communication system. In wireless transmission the channel is the electromagnetic spectrum that has to be shared between all different applications and users. If different transmitted signals interfere with each other, it is hard for the receiver to maintain reliable communication. The traditional solution is to keep the signals more or less orthogonal. This way the receiver can easily suppress the undesired signals.

Figure 2.1 Basic elements of a digital communication system



There are different methods to keep the signals orthogonal. One is to send the signals at different frequencies. This method is called FDMA (Frequency Division Multiple Access). TDMA (Time Division Multiple Access) is another technique to share the channel. In this method each time frame is divided into some non-overlapping sub-frames and each user transmits in one of those time slots.

FDMA and TDMA are narrowband multiple access methods compared to wideband CDMA (Code Division Multiple Access) that spreads the signal along code sequences which are separated at the receiver by their relatively small cross-correlation. CDMA signals can share the same bandwidth and time in this way but as the code rate is far more than the actual bit rate, a far wider bandwidth than usual has to be allocated to the CDMA signal.

There are also some other techniques that utilize the physical properties of the signals to increase the spectrum usage. One is sectoring the area using directional antennas as in multi-beam satellite systems [27]. The other technique is implementing cellular concepts that have revolutionized mobile telecommunication technology.

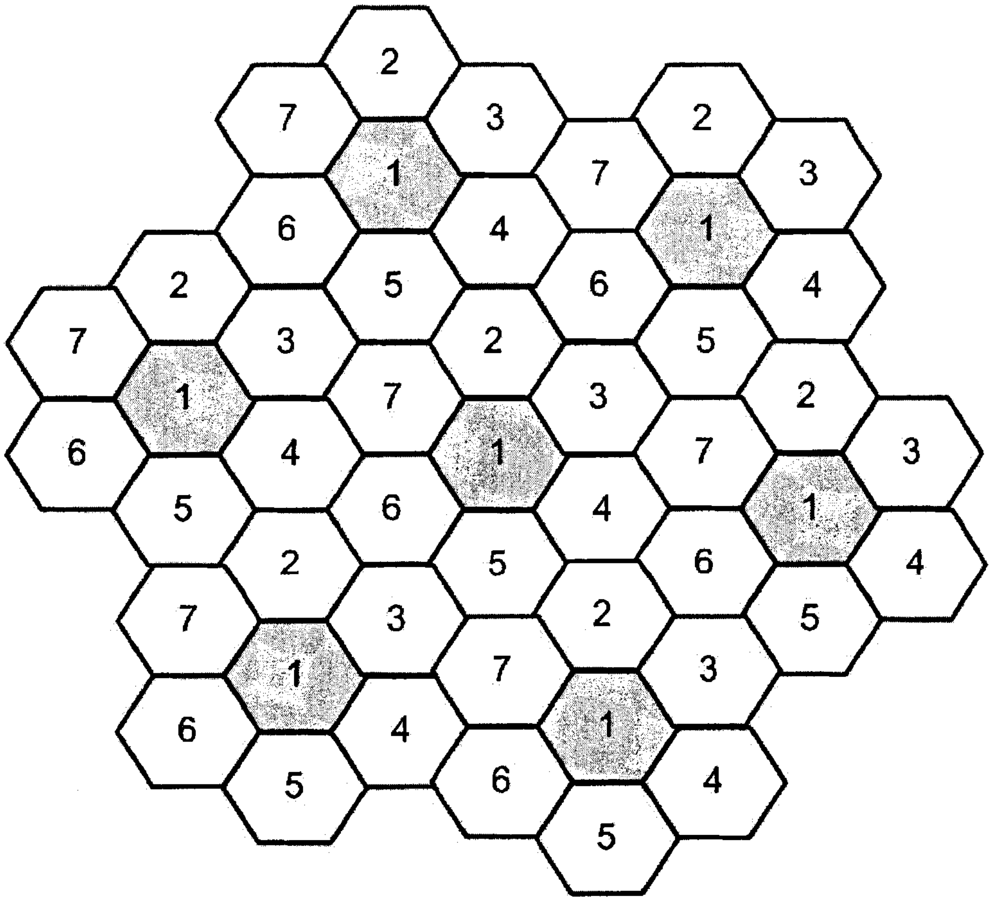
The main idea in cellular systems is to replace one high power base station by many low power base-stations. Each of these low power transmitters would cover a small area that is called a cell. N adjacent cells make a cluster and the whole frequency bandwidth would be divided into N non-overlapping subsets. Each cell in the cluster would use one of these subsets and the whole frequency bandwidth is reused in every cluster. Co-channel cells are separated so path-loss attenuates the co-channel interference to the noise level. Figure 2.2 shows a cellular system with hexagonal cells and cluster size of 7.

All of the orthogonalization methods mentioned above treat interfering signals as noise and ignore their structure or any information about them. As a result, these methods have to sacrifice some of the resources to keep the noise level low enough. For example, in FDMA and TDMA some of the bandwidth has to be used for guard bands between different sub-channels and in CDMA the number of users would be limited by the noise level. Even in cellular systems the whole bandwidth has to be divided between different cells so each cell does not have access to use all the resources.

Unlike the traditional simple single user detection, multiuser detection (MUD) takes explicit account of the structure of interferers. In this way, MUD is able to reduce the damage to the detection of the desired signal. The goal of MUD is to achieve single user performance in the presence of interferers and relax the orthogonalization rules to increase the capacity of the system. To achieve these goals, we have to develop high performance, low complexity MUD algorithms. So this has been an active research area in communications. The key factor in this research is to make appropriate use of mobile channel characteristics.

In the rest of this chapter we first go through the essentials of mobile channels. Following, we will discuss the multiple antenna systems. Finally we will review the different detection techniques and describe the motivations for this thesis.

Figure 2.2 A cellular system with hexagonal cells and a cluster size of N=7



2.2 Mobile Channel Characteristics

Sharing the bandwidth is not the only problem we face in mobile communications systems. Geographical and physical factors like building structures, hills, weather or even an airplane passing in the area, can affect the properties of a wireless channel. Many models have been developed to describe different aspects of mobile channel characteristics. These models help us to design and test different algorithms to improve the performance of mobile communication systems.

2.2.1 Path Loss

As the signal propagates, it gets weaker. In free space, this loss follows inverse square law but in many situations, the signal does not go through the free space. Earth's surface as a large reflecting object changes the model substantially. Equation (2.1) shows the relationship model between the transmitted power P_T and received power P_R and distance d. In free space, the path loss exponent γ would be 2 as the inverse square law predicts but in urban cellular systems, it is usually between 3-4.

$$P_r = \frac{P_t}{d^\gamma} \tag{2.1}$$

This higher exponent path loss model has advantages and disadvantages for wireless communication systems. Transmission power must be higher for a desired coverage area. The power levels of near and far users will also be very different, causing increased interference for low power users. So in many cases a power control system is required. The good news is that the cell coverage is sharper and co-channel interference from other clusters is small.

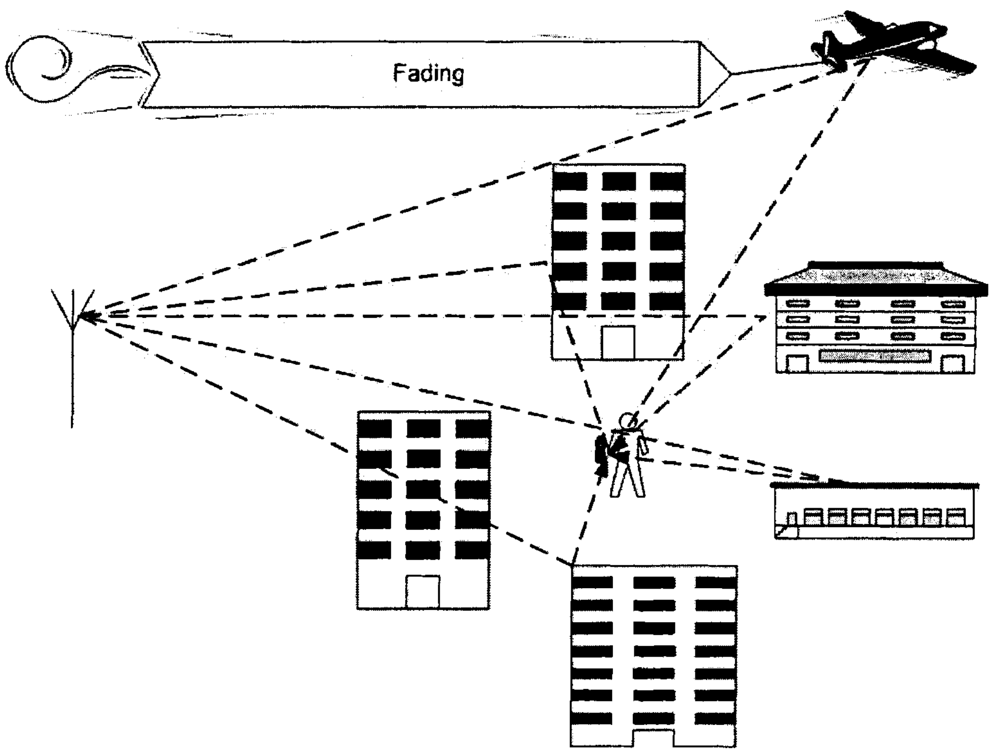
2.2.2 Shadowing

Path loss includes the effect of earth but other large obstacles (hills, buildings, foliage, *etc.*) cause other loss termed shadowing. Shadowing causes variation around the mean power that is modelled by path loss. This variable factor has a lognormal distribution or in other words, power in dB contains a Gaussian term with zero mean. The standard deviation of the Gaussian term is about 6-8 dB. It will be higher in urban settings and indoor systems, but lower in rural areas. Shadowing would cause coverage holes and results in vague cell boundaries instead of the circles that simple path loss calculations predict.

2.2.3 Multipath

Path loss and shadowing models the effect of the earth and large scale obstacles on wireless communication, but these are not the only factors that should be considered in a mobile communication system. Figure 2.3 shows a typical link between a mobile user and a base station. There are a lot of scatterers around the mobile but few around the base station because it is usually situated higher than its surroundings. There are many signal paths between the user and the base station and the receiver in this link picks up the sum of all the reflections. This phenomenon is called multipath and its effects are destructive for reliable communications. Decades of intensive research has been done in order to mitigate these effects.

Figure 2.3 Multipath environment



The impulse response of the channel includes all the paths between the transmitter and the receiver. Each of these paths has its own amplitude, delay, and a phase that cause constructive or destructive interference. This can cause huge changes in signal power over a small distance as short as half a wavelength. The characteristics of the channel are extremely local. A change of only one wavelength in path length causes a phase shift of 2π radians, so moving even a fraction of wavelength in any direction can cause huge changes in the phase and power of the sum of the arriving signals.

This impulse response in turn defines the frequency response of this channel. It means that different carrier frequencies experience different gains and phases. When the mobile moves, the impulse response, and the frequency response change so the channel is a time-varying linear filter. The fastest rate of change which is proportional to the speed of the user is called "Doppler frequency". The time varying gain is called "fading" in the literature. "Delay spread" is another term defining the range of delays in impulse response. Next, we will discuss more about fading and delay spread and their destructive effect on communications.

2.2.3.1 Fading

Fading is one of the major problems in communications and it can cause an irreducible error floor in the link. This means that bit error rate (BER) does not decrease after a certain point, no matter how much the power is increased. Fading can also decrease the capacity of the channel. Therefore, it is paramount to investigate the nature of fading and develop mathematical models to describe this phenomenon.

As mentioned before, fading is the random fluctuation about the mean power that is determined by shadowing and path loss. These fluctuations range typically from 10 dB above or to 40 dB below the mean power. The distance between two deep fades is on average half a wave length. As a result Doppler frequency f_D which shows the fastest rate of change in power is defined by (2.2), v is the speed of the user and λ is the wavelength of the carrier frequency

$$f_D = \frac{v}{\lambda}. \tag{2.2}$$

The Doppler spectrum shows the distribution of power in Doppler domain ν and is the Fourier transform of autocorrelation function of complex gain. These functions are extremely important in analysis of modulation in fading channels, pulse distortions, error floors, effectiveness of interleaving and many other situations.

When the gain varies during a symbol time T, it is called fast fading. This happens if $f_D T \geq 1$. Fast fading makes the communication difficult because we would not be able to track the channel, varying in a symbol time. Fortunately we usually face moderate to slow fading channels where $f_D T \ll 1$ ($f_D < 100$ Hz even when operating in $v = 100$ Km/s and frequencies up to 1 GHz). In this case we can control some of the damage caused by fading by tracking the fading channel using pilot symbols or using differential detection methods.

Many mathematical models have been developed to describe fading. Rayleigh fading happens when the amplitude gain is a zero mean complex Gaussian random variable that is a Rayleigh magnitude and a uniform phase. Physically this happens when a mobile is surrounded by scatterers and there is no line of sight between mobile and base stations.

Measurements show that even with 6 scatterers the gain distribution is Gaussian to a good extent. If there is a line of sight though the distribution is called Rician in which the amplitude gain is a non- zero mean complex Gaussian random variable. These models help us simulate the channels and test different algorithms to improve the performance of the system.

2.2.3.2 Delay Spread

The delay spread τ shows the spread of delays in the multipath channel. If delay spread is small compared to symbol time $\tau \ll T$, then we have a flat fading channel that can be modelled by one hypothetical equivalent path. In other words, the bandwidth of the signal is smaller than the coherence bandwidth of the fading channel so it is almost flat across the signal bandwidth. As a result, there is no interference between consecutive symbols of sequence in this channel.

In the case where the delay spread is significant compared to symbol time, we would have a frequency selective fading channel. It means that the channel is no longer flat because the signal bandwidth is larger than the coherence bandwidth of the channel. As a result, there would be intersymbol interference (ISI) in the system.

The power delay profile is the key to analysis or simulation of delay spread and ISI. It represents the density of power in delay domain that can be treated discrete-time or continuous. Equation (2.3) shows the discrete-time model of the frequency selective channels

$$h(t) = \sum_i g_i \delta(t - T). \tag{2.3}$$

Detection in this channel is made difficult because we have to deal with both fading and ISI. The good news is that if the detection is good enough, then performance of a communication link in a frequency selective channel is better than in a flat fading channel. This sounds odd but can be explained by the diversity we experience in frequency selective channels. Next we discuss diversity in more detail as it one of the most important methods to improve performance in a fading channel.

2.3 Diversity

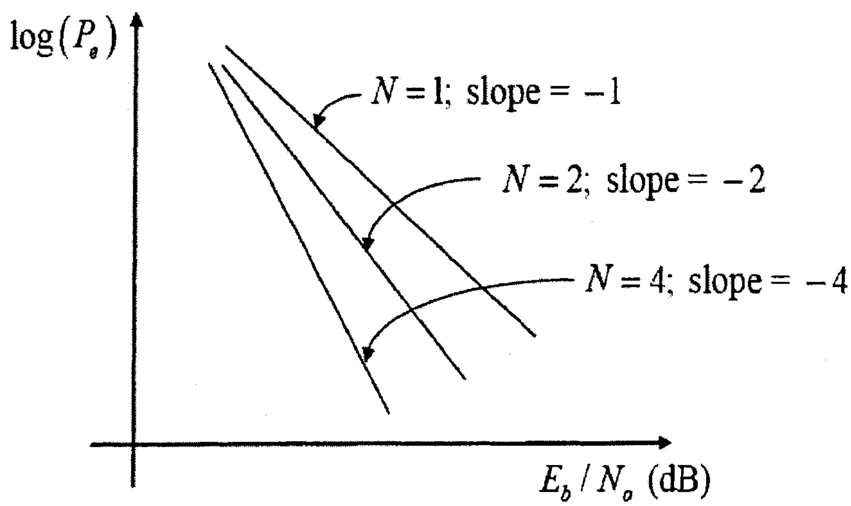
The basic idea in diversity techniques is to use several independently fading channels to transmit the data. Then the receiver would pick up several replicas of the same signal. The probability that all these channels fade simultaneously is very low. In other words, there is higher probability that at least one high quality copy of the signal is present at the receiver. In this way, diversity reduces BER substantially by preventing most of the error bursts that usually happen in deep fades.

Without diversity, the probability of error P_e decreases only as the inverse of SNR $(E_b / N_0)^{-1}$ but if we have N independent channels, the probability of all of them failing would be $(P_e)^N$. Figure 2.4 shows the approximate behaviour of a diversity system. In this case N is called the diversity order of the system and it decreases when the channels are correlated or suboptimal detection methods are used. Several different methods can be used to achieve diversity.

One way is to employ frequency diversity. In this way several copies of the signal will be sent via different uncorrelated channels. In other words, the separation between these channels should be higher than coherence bandwidth to ensure independent fading.

Another way is to use time diversity and send the signal several times over different time frames. Separation between these time frames also has to be long enough to make sure that they fade independently. These methods waste the bandwidth and energy because of the repetitive transmissions.

Figure 2.4 Approximate behaviour of a diversity system



Source: Professor Paul Ho, course note [31], 2003, by permission.

Another more commonly used method is space diversity that employs multiple antennas. Different antennas can obtain independent fading if they have different polarizations or directionality, or if they are far enough spatially. The required separation can be determined by spatial correlation function and is usually a few wavelengths. The extra antennas can be either in the receiver or the transmitter. The advantage of antenna diversity is gaining extra quality or capacity without using extra spectrum. The disadvantage is extra expense, inconvenience, and the extra space.

One more situation to gain diversity is when our signal bandwidth is greater than the coherence bandwidth as in frequency-selective fading channels. In this case we receive several independently faded replicas of the signal through time and we can gain diversity, provided that we use a sophisticated equalizer.

2.4 Multiple-Input Multiple-Output Arrays

Systems with spatial diversity have multiple antenna arrays at one end. In the last section we saw that these systems enjoy increased quality and capacity and also saw that the performance would be degraded by correlation among channel gains experienced by the antennas in an array. In this section we consider a multiple-input multiple-output (MIMO) scenario where multiple antenna arrays are at both ends. This configuration has many degrees of freedom and is expected to provide us with increased capacity and diversity with no increase in required bandwidth.

Pioneering work by Winters [28], Foschini [1] and Telatar [29] ignited much interest in this area. They demonstrate that capacity can be proportional to the smaller of the number of antennas at each end, if a high scattering environment provides independent fading on all the paths between the transmitter and the receiver. This is a huge boost to the single-input single-output capacity.

However, realizing the potential capacities of these systems relies on nature and technology. Natural characteristics of the environment like correlation between paths can degrade the performance of the system substantially. On the other hand, technological issues like antenna patterns, coupling, coding schemes and detection methods can play an important role. Investigating these factors in a MIMO channel is a necessary step towards implementing reliable systems. As the subject is still relatively young, MIMO is an intensive area of research right now.

The focus of this thesis is the problem of finding near optimal detection methods that are low in computational complexity and we are going to introduce a new soft-input soft-output (SISO) detection method for frequency selective MIMO channels. The rest of this chapter first defines the system model we use for our analysis. Then we review different available methods to solve the problem and investigate their properties and at last we express the motivations for this new technique.

2.5 System Model

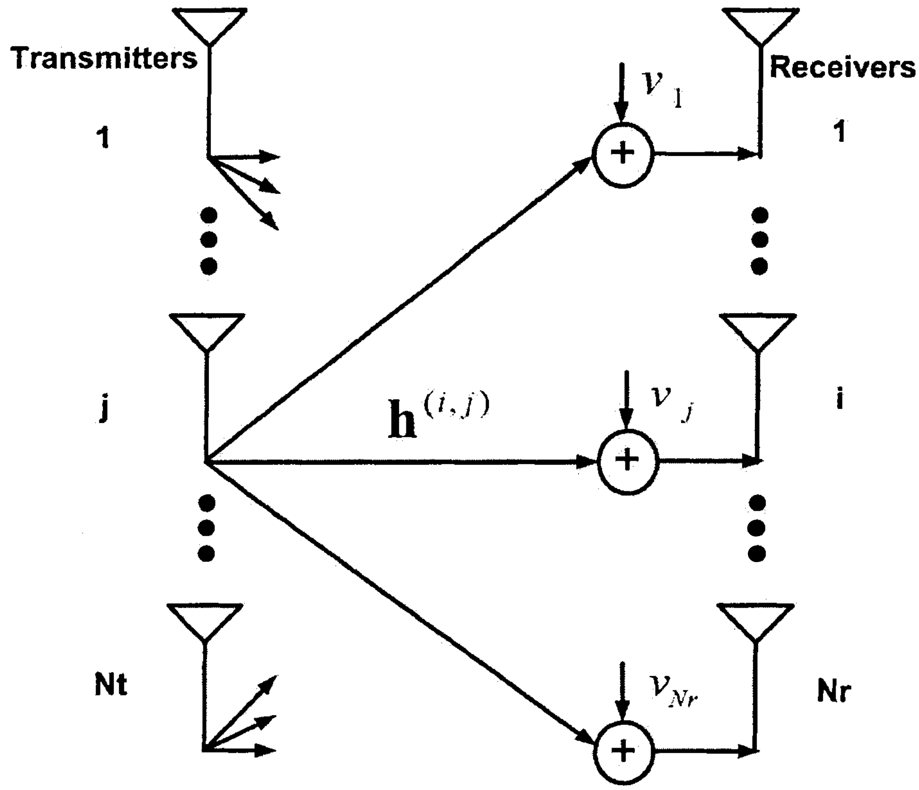
We consider a discrete-time, baseband equivalent of a MIMO system to simplify the formulas and avoid the confusions caused by the complexities of modulations, filtering, sampling, and conversions. For coherent detection methods, we assume that the receiver has perfect channel state information (CSI). The overall channel impulse response is also assumed to be constant in a block of N symbols and changes independently to the next. This block-fading frequency-selective model is frequently used in high data rate systems like EDGE [20] and in literature [13, 15].

Figure 2.5 shows a MIMO system with N_t transmit and N_r receive antennas. Each of the N_tN_r links in this system is modelled as a linear finite impulse response (FIR) dispersive channel with L symbol-spaced taps. Each tap follows an independent Gaussian distribution according to the channel's power delay profile. At the receiver end the signal is perturbed by independent identically distributed (i.i.d) additive complex

Gaussian noise. Single-input single-output channel impulse response from the jth transmitter to the ith receiver is denoted by

$$\mathbf{h}^{(i,j)} = [h_0^{(i,j)}, h_1^{(i,j)}, \ldots, h_{L-1}^{(i,j)}]^T . \tag{2.4}$$

Figure 2.5 MIMO channel model



The symbol sent from the jth user in time interval k is denoted by b_k^j. It typically comes from a M-ary constellation. Here we mostly consider the BPSK case, where b_k^j is $\{\pm 1\}$, in order to simplify the formulas. Generalization of the formulas to M-ary is usually straightforward or can be found in the literature.

We also suppose that each noise sample v_k^i is a complex Gaussian random variable with unit variance, and that they are mutually independent. Unit variance is chosen for simplicity and does not restrict the generality of the results. The received signal at the i[th] antenna in time interval k can then be expressed as

$$y_k^i = \sum_{j=1}^{N_t} \sum_{l=0}^{L-1} h_l^{(i,j)} b_{k-l}^j + v_k^i, \quad k = 1 \ldots (N + L - 1) \tag{2.5}$$

N is the block size but because of the memory of the channel, echoes will be received in the L-1 samples following the last symbol time N. Equation (2.5) can be written in matrix form as

$$\mathbf{y}_k = \sum_{l=0}^{L-1} \mathbf{H}_l \, \mathbf{b}_{k-l} + \mathbf{v}_k \, , \tag{2.6}$$

where

$$\mathbf{y}_k = [y_k^1, y_k^2, \ldots, y_k^{N_r}]^T ,$$

$$\mathbf{v}_k = [v_k^1, v_k^2, \ldots, v_k^{N_r}]^T ,$$

$$\mathbf{b}_k = [b_k^1, b_k^2, \ldots, b_k^{N_t}]^T ,$$

$$\mathbf{H}_l = \begin{pmatrix} h_l^{(1,1)}, h_l^{(1,2)}, & \cdots & , h_l^{(1,N_t)} \\ \vdots & \ddots & \vdots \\ h_l^{(N_r,1)}, h_l^{(N_r,2)}, & \cdots & , h_l^{(N_r,N_t)} \end{pmatrix}.$$

Equivalently,

$$\mathbf{y}_k = \mathbf{H} \tilde{\mathbf{b}}_K + \mathbf{v}_k \, , \tag{2.7}$$

where

$$\mathbf{H} = [\mathbf{H}_{L-1}, \cdots, \mathbf{H}_1, \mathbf{H}_0],$$

$$\tilde{\mathbf{b}}_k = [\mathbf{b}^T_{k-L+1} \quad \cdots \quad \mathbf{b}^T_k]^T.$$

We modify (2.6) to develop the general input-output relation in the MIMO FIR channel

$$\mathbf{y} = \tilde{\mathbf{H}}\mathbf{b} + \mathbf{v}, \tag{2.8}$$

$$\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_{N+L-1}^T]^T,$$

$$\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_{N+L-1}^T]^T,$$

$$\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_{N+L-1}^T]^T,$$

$$\tilde{\mathbf{H}} = \begin{pmatrix} \mathbf{H}_0 & & & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & & \mathbf{0} & \\ \vdots & \vdots & \ddots & & & \\ \mathbf{H}_{L-1} & \cdots & \cdots & \mathbf{H}_0 & & \\ & \ddots & & & \ddots & \\ & & \mathbf{H}_{L-1} & \cdots & \cdots & \mathbf{H}_0 \\ & & & \ddots & \vdots & \vdots \\ & \mathbf{0} & & & \mathbf{H}_{L-1} & \mathbf{H}_{L-2} \\ & & & & & \mathbf{H}_{L-1} \end{pmatrix}.$$

The notations discussed in this section will be used next in describing different detection methods.

2.6 Multiuser Detection in Frequency Selective MIMO Channels

Different algorithms can be adapted to be used for multiuser detection in the system described above. In this section, we review these techniques and explain their qualities and characteristics. This analysis reveals the shortcomings in current methods, motivating the work in this thesis. Here are a few issues and definitions that should be noted before the beginning of the discussion:

Detection can be performed according to different criteria such as minimum bit error rate (BER), symbol error rate (SER) or sequence error rate. In multiuser detection, we are usually interested in minimum sequence error rate criteria because symbols are often overlapping in our measurements. In such cases it is often easier to formulate a sequence detector.

In digital communications whenever the data is in the form of a finely graded multilevel confidence measure concerning the probabilities of the symbols, it is called soft information. If the data carries the decisions about the symbols, it is called hard information. Hard information is easier to handle and transfer but ignores some of the data and degrades the performance of the system as a result.

Different situations may occur in our MIMO system. If the number of transmitters N_t is more than the number of receivers N_r, this is called an overloaded situation. $N_t = N_r$ is critically loaded and $N_t < N_r$ is under loaded. It is important to investigate the performance of different detection methods under these situations.

Most of the techniques discussed here have originally been used in other systems, such as in decoding convolutional codes or in equalizers used to suppress ISI. They can be adapted to our more difficult situation where they have to suppress both ISI and MUI but some of the references discuss other or general instances of their usage.

2.6.1 Optimal Detection Methods

Optimal detection shows the limits of performance in a system. One of its attractive features is that it provides the many users with almost the same performance and the order of diversity as a single user operating without interference. As we explain next, the barrier is the huge computational burden of optimal methods.

In order to minimise the sequence error rate, the constraint in (2.9) has to be satisfied. The set $\{\hat{\mathbf{b}}_i\}$ includes all possible transmitted vectors and $\mathbf{y}$ is the received vector. The estimated vector $\hat{\mathbf{b}}(\mathbf{y})$ is defined as

$$\begin{aligned}\hat{\mathbf{b}}(\mathbf{y}) &= \arg\max_i p(\hat{\mathbf{b}}_i \mid \mathbf{y}) \\ &= \arg\max_i p(\hat{\mathbf{b}}_i, \mathbf{y}) \\ &= \arg\max_i p(\mathbf{y} \mid \hat{\mathbf{b}}_i) p(\hat{\mathbf{b}}_i).\end{aligned} \tag{2.9}$$

This is the maximum a posteriori (MAP) sequence estimator.

If all sequences are equiprobable or if a priori probabilities (apP) are unknown then

$$\hat{\mathbf{b}}(\mathbf{y}) = \arg\max_i p(\mathbf{y} \mid \hat{\mathbf{b}}_i). \tag{2.10}$$

This is maximum likelihood sequence estimation (MLSE). Optimal estimation is performed according to either MAP or ML definitions.

As we saw in last section, $\mathbf{y} = \tilde{\mathbf{H}}\mathbf{b} + \mathbf{v}$ and $\mathbf{V}$ is a vector of independent complex Gaussian random variables so

$$p(\mathbf{y} \mid \hat{\mathbf{b}}_i) \propto \exp(-c\left\|\mathbf{y} - \tilde{\mathbf{H}}\hat{\mathbf{b}}_i\right\|^2), \tag{2.11}$$

where c is a real positive constant.

The maximization problem in (2.10) is equivalent to a least squares problem

$$\hat{\mathbf{b}}(\mathbf{y}) = \arg\min_i \left\|\mathbf{y} - \tilde{\mathbf{H}}\hat{\mathbf{b}}_i\right\|^2. \tag{2.12}$$

Similarly for MAP

$$\hat{\mathbf{b}}(\mathbf{y}) = \arg\min_i \left\{c\left\|\mathbf{y} - \tilde{\mathbf{H}}\hat{\mathbf{b}}_i\right\|^2 - \ln p(\hat{\mathbf{b}}_i)\right\}. \tag{2.13}$$

It is clear from these formulas that MAP requires more computation and ML ignores the apPs.

2.6.1.1 Joint MAP and Joint ML

Joint detection relies on the knowledge of gain values $\tilde{\mathbf{H}}$ and on the finite alphabet (FA) properties of the signal. It involves calculating the metric in MAP or ML formulas for all instances of the transmitted vector $\hat{\mathbf{b}}_i$ and finding the minimum metric. If we use the metric in (2.12), the detection method is called Joint ML (JML) but if we use the metric in (2.13), it is Joint MAP (JMAP) detection. JMAP employs the apPs but requires more computations.

Both techniques provide us with hard decisions and will have the same answer if the sequences are equiprobable or if we have no prior information. Their performance is optimal but their computational complexity is proportional with the number of possible vectors $\{\hat{\mathbf{b}}_i\}$ and it grows exponentially with the number of symbols in the block. So if we use BPSK signals, block size of N and N_t users the computational complexity would be in the order $O(2^{N_t N})$.

2.6.1.2 Joint APP Extraction

JMAP and JML both provide hard decisions but the joint APP extractor uses as inputs the a priori probabilities and the received signals to generate the a posteriori probabilities, usually in the form of log likelihood ratios (LLR)

$$L(b_k^j|\mathbf{y}) = \ln\frac{p(b_k^j=+1|\mathbf{y})}{p(b_k^j=-1|\mathbf{y})} = \ln\frac{\sum_{\hat{\mathbf{b}}_i\in D:\hat{b}_k^j=+1}\exp(-\left\|\mathbf{y}-\tilde{\mathbf{H}}\hat{\mathbf{b}}_i\right\|^2+\ln p(\hat{\mathbf{b}}_i)-\ln p(\hat{b}_k^j))}{\sum_{\hat{\mathbf{b}}_i\in D:\hat{b}_k^j=-1}\exp(-\left\|\mathbf{y}-\tilde{\mathbf{H}}\hat{\mathbf{b}}_i\right\|^2+\ln p(\hat{\mathbf{b}}_i)-\ln p(\hat{b}_k^j))}, \quad (2.14)$$

$$D=\{\pm 1\}^{N_t N}.$$

Its performance and complexity is the same as JMAP and JML but because it is a soft-input soft-output (SISO) algorithm, it can be used in an iterative decoding system.

2.6.1.3 Trees and Trellises

Bayes' theorem states that

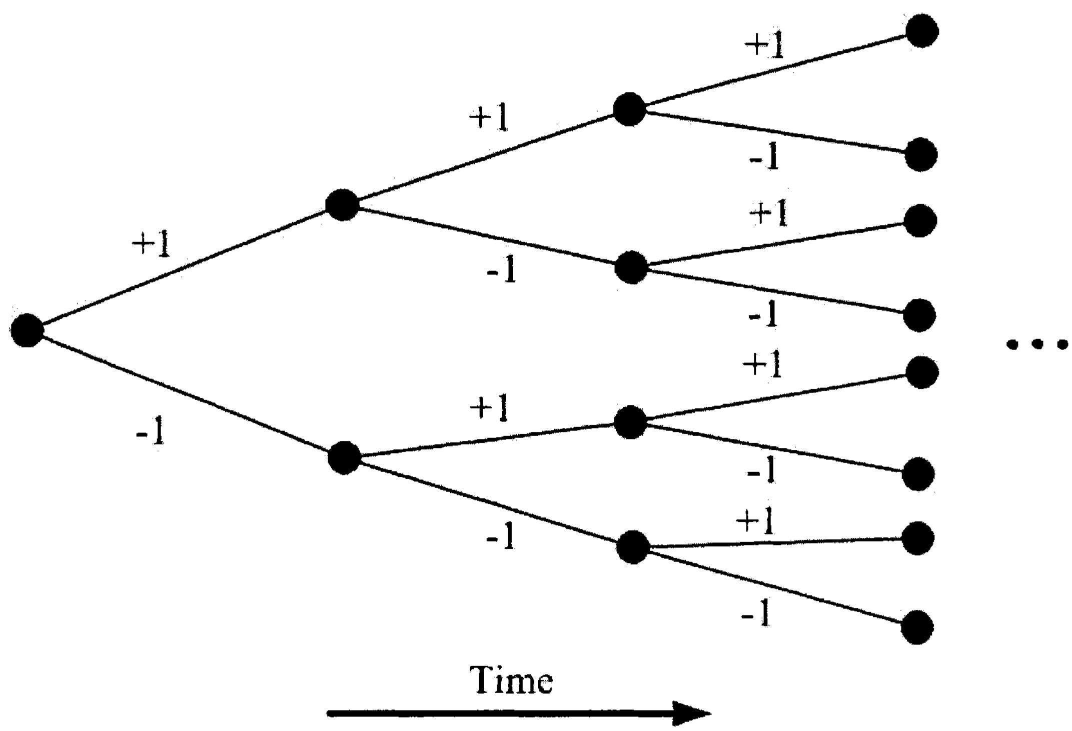
$$p(A,B)=p(A\,|\,B)p(B). \quad (2.15)$$

By applying this theorem, we can formulate our probability metric recursively

$$\underbrace{p(\mathbf{y}(1:n) \mid \hat{\mathbf{b}}_i)}_{\text{Metric at step } n} = p(\mathbf{y}(n) \mid \mathbf{y}(1:n-1), \hat{\mathbf{b}}_i) \underbrace{p(\mathbf{y}(1:n-1) \mid \hat{\mathbf{b}}_i)}_{\text{Metric at step } n-1} . \tag{2.16}$$

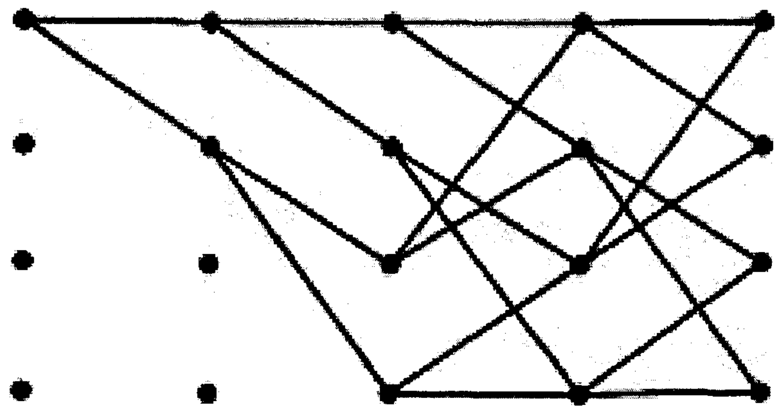
We can convert the recursion formula (2.16) to an additive metric by using LLRs instead of probabilities. Then we can form all instances of $\hat{\mathbf{b}}_i$ by a tree that spreads branches of symbols over time. Figure 2.6 shows a tree of candidate binary sequences. This way, we can calculate the metrics of all vectors efficiently over the tree. We start from the beginning and calculate the metric of each node recursively by adding the proper update to the metric of its predecessor node on the branch. When we get to a leaf, the minimum metric belongs to the optimal estimation branch. This method is called a tree search.

Figure 2.6 A tree of candidate binary sequences



If the memory is finite, our system is a finite state machine (FSM). In other words, the locality in time changes the tree to a trellis. Figure 2.7 shows how the tree changes to a trellis in a system with one user and memory of 2. Trellis structure is very important and is necessary when we use efficient recursive algorithms, such as the Viterbi algorithm (VA) and BCJR. Next we review these algorithms.

Figure 2.7 Trellis for a system with one user and memory of 2



2.6.1.4 Viterbi Algorithm

The invention of maximum likelihood sequence estimation by Viterbi [21] in 1967 is a major milestone in digital communication. A classical interpretation of the Viterbi algorithm can be found in Forney's often quoted paper [22]. VA is based on calculating path metrics along a trellis. Then at each node, the path with the least metric survives and the other ones are discarded. This way, the VA finds the path with the least metric which is the maximum likelihood path.

VA performs optimal detection with less computational complexity than joint detection. The number of computations in VA grows linearly with block size N. The number of branches in the trellis is $2^{N,L}$ if binary symbols are used. So the computational complexity is in the order of $O(N2^{N_I L})$ which is a huge saving compared to $O(2^{N_I N})$ of joint detection if the memory size L is relatively small.

The original version of VA provides only hard decisions about the ML path through the trellis. Some modified versions of VA can also supply soft-outputs, most notably soft-output Viterbi algorithm (SOVA) proposed by Hagenauer and Hoeher [23] but the performance of these algorithms is not as good as MAP algorithms in iterative systems.

2.6.1.5 BCJR

The VA finds the most likely sequence of transmitted symbols and does not necessarily result in minimum SER but its performance is close to it. Minimum SER detection algorithm was proposed by Bahl et al. [2] in 1974. This maximum a posteriori (MAP) algorithm is known by the initials of its inventors Bahl, Cocke, Jelinek, and Raviv (BCJR). The BCJR algorithm provides not only the estimated sequence, but also the probabilities of the symbols. These soft output decisions are essential for iterative detection. BCJR slightly outperforms VA but because of its significantly higher complexity it was rarely used in practice until the introduction of turbo codes in 1993.

Reference [24] presents a good explanation of BCJR algorithm. BCJR is also based on the trellis structure, but unlike VA, it performs both forward and backwards recursions through the trellis to calculate probabilities of the branches and probabilities of the states knowing the future and the past. $\gamma_k(s,s')$ is the probability of the branch from state s' to state s at time k. $\alpha_{k-1}(s')$ is the probability that trellis is at state s' at time $k-1$ knowing the past channel outputs. $\beta_k(s)$ is the probability that trellis is at state s at time k knowing the future channel outputs. The a posteriori LLRs are then calculated using these values:

$$L(b_k^j|\mathbf{y}) = \ln \frac{\sum_{(s,s') \Rightarrow b_k^j = +1} \alpha_{k-1}(s')\gamma_k(s,s')\beta_k(s)}{\sum_{(s,s') \Rightarrow b_k^j = -1} \alpha_{k-1}(s')\gamma_k(s,s')\beta_k(s)} \tag{2.17}$$

These calculations seem to be very complicated but they can be simplified by using Jacobian logarithm formula:

$$\begin{aligned} \ln(e^{x_1} + e^{x_2}) &= \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|}) \\ &= \max(x_1, x_2) + f_c(|x_1 - x_2|) \\ &= g(x_1, x_2) \end{aligned} \tag{2.18}$$

So we can use a look-up table, instead of doing the calculations every time. This is called Log-MAP algorithm. We can also simplify the algorithm more by this approximation:

$$\ln(e^{x_1} + e^{x_2}) \approx \max(x_1, x_2) \tag{2.19}$$

This called Max-Log-MAP algorithm and causes a slight degradation in performance. It is shown in [25] that this setting is inherently equivalent to a combination of forward and backward VA processors, coupled by a dual-maxima computation and that it can be implemented by processing load no more than four times of VA with some moderate memory requirements. As a result, the order of computational complexity in BCJR would be the same as VA $O(N2^{N_t L})$.

2.6.1.6 Complexity Reduction Methods

Even by using VA and BCJR, optimal detection can require a huge computational load especially when the memory length of the channel L is long. One way to decrease the computations is to settle for a reduced number of states as in delayed decision-feedback sequence estimation (DDFSE) or reduced-state sequence estimation (RSSE). DDFSE [3] consists of a VA processor with a reduced memory size and a decision feedback filter to cancel the effect of the symbols that are not considered in VA. RSSE, on the other hand, reduces the number of states using set partitioning [4].

An alternative approach to complexity reduction is to keep fewer survivor paths or nodes through the tree or trellis. The M algorithm [5] keeps only the best M choices but, in the T algorithm [6], paths or nodes that are better than a certain threshold survive. Another standard suboptimal method is to use linear prefiltering [7] to modify the impulse response before the actual processing.

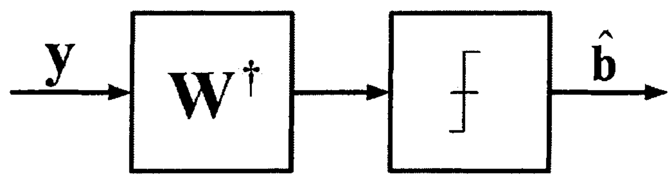
The practical difficulties of adapting these techniques to MIMO channels and the trade-off between their performance and computational costs, make them less popular. So in practice, most often a multi-channel equalizer is used to suppress the interference caused by ISI or MUI.

2.6.2 Linear Methods

Linear filtering is a conventional suboptimal channel equalization technique. Purely linear methods employ the knowledge of channel $\tilde{\mathbf{H}}$ to solve the least square problem (2.12) and to simplify the problem; they ignore the finite alphabet (FA) properties of the signal. Figure 2.8 shows a linear equalization system. The linear filtering is in the form of a matrix $\mathbf{W}^{\dagger}$ (transpose conjugate of $\mathbf{W}$) which provides the estimated symbols by $\mathbf{W}^{\dagger}\mathbf{y}$. Each component is then mapped to the nearest point in the signal constellation

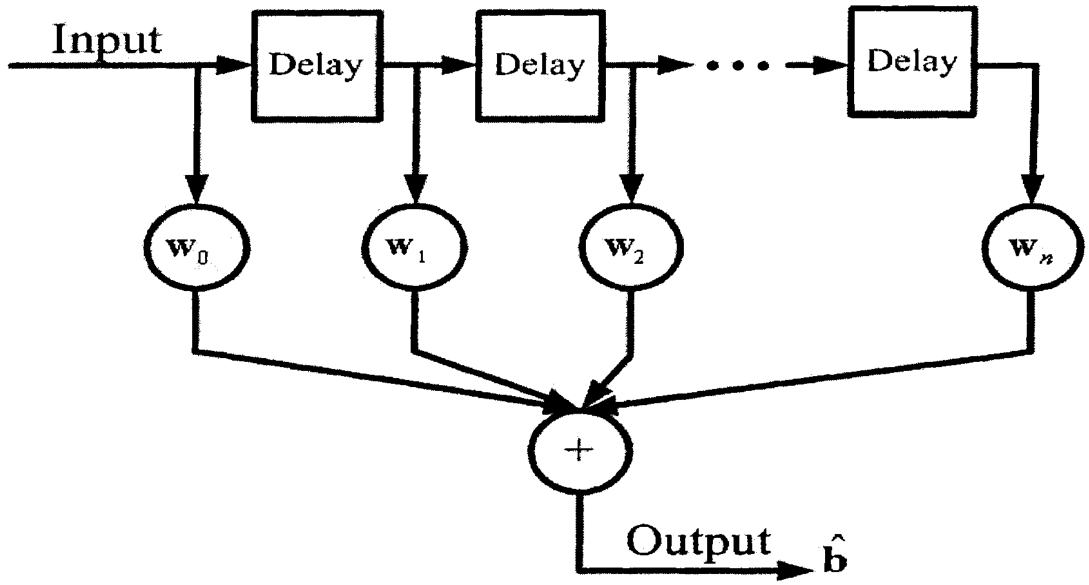
$$\hat{\mathbf{b}} = \lceil \mathbf{W}^{\dagger}\mathbf{y} \rfloor . \tag{2.20}$$

Figure 2.8 Linear equalizer



This matrix format is simple and makes the analysis easier, but in practise we usually consider the locality in time and implement linear techniques in the form of multiuser transversal filters (Figure 2.9). The implementation is generally the simplest in linear schemes and the computational costs are low, however, these methods produce a noticeable performance degradation in the sense of both BER and diversity order. This class of detectors includes zero forcing (ZF) and minimum mean square error (MMSE) equalizers.

Figure 2.9 Linear transversal filter



2.6.2.1 ZF

ZF is based on nulling all other users by using the pseudo-inverse in order to make a decision about one user. This algorithm is thoroughly discussed in [17]. $\mathbf{W}$ is derived as follows and is equal to the pseudo-inverse of $\tilde{\mathbf{H}}$

$$\begin{aligned}
\mathbf{y} &= \tilde{\mathbf{H}}\mathbf{b} + \mathbf{v}, \\
\tilde{\mathbf{H}}^{\dagger}\mathbf{y} &= \tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}}\mathbf{b} + \tilde{\mathbf{H}}^{\dagger}\mathbf{v}, \\
(\tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^{\dagger}\mathbf{y} &= (\tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}})^{-1}(\tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}})\mathbf{b} + (\tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^{\dagger}\mathbf{v}, \\
\mathbf{W}^{\dagger}\mathbf{y} &= \mathbf{b} + \mathbf{W}^{\dagger}\mathbf{v}.
\end{aligned}$$

So the transmitted vector $\mathbf{b}$ can be estimated by

$$\hat{\mathbf{b}} = \mathbf{W}^{\dagger}\mathbf{y}, \tag{2.21}$$

$$\mathbf{W} = \tilde{\mathbf{H}}(\tilde{\mathbf{H}}^{\dagger}\tilde{\mathbf{H}})^{-1}. \tag{2.22}$$

The computational complexity of this process is $O((NN_t)^3)$, but it would be $O(NN_t^3L^2)$ for detection and $O((N_tL)^2)$ for initialization if we use the transversal filters instead of block detection. Unfortunately, ZF loses an order of diversity with each additional user, so if the number of transmit antennas is more than the number of receive antennas, the diversity order would be less than one and we would experience an irreducible error floor. This means that ZF can not handle an overloaded situation. The other problem with ZF is that the term $\mathbf{W}^\dagger\mathbf{v}$ might cause noise enhancement and degrade the performance, if $\tilde{\mathbf{H}}$ is poorly conditioned.

2.6.2.2 MMSE

The minimum mean square error (MMSE) technique employs a more practical criterion for linear detection to achieve an improved performance compared to ZF. MMSE includes the noise power in filter calculations and instead of nulling all the users, it attenuate them to noise level and in this way controls the noise enhancement problem encountered in ZF.

The filter matrix $\mathbf{W}$ in MMSE is defined as

$$\mathbf{W} = \tilde{\mathbf{H}}(\tilde{\mathbf{H}}^\dagger\tilde{\mathbf{H}} + N_0\mathbf{I})^{-1}, \tag{2.23}$$

where N_0 is the noise power.

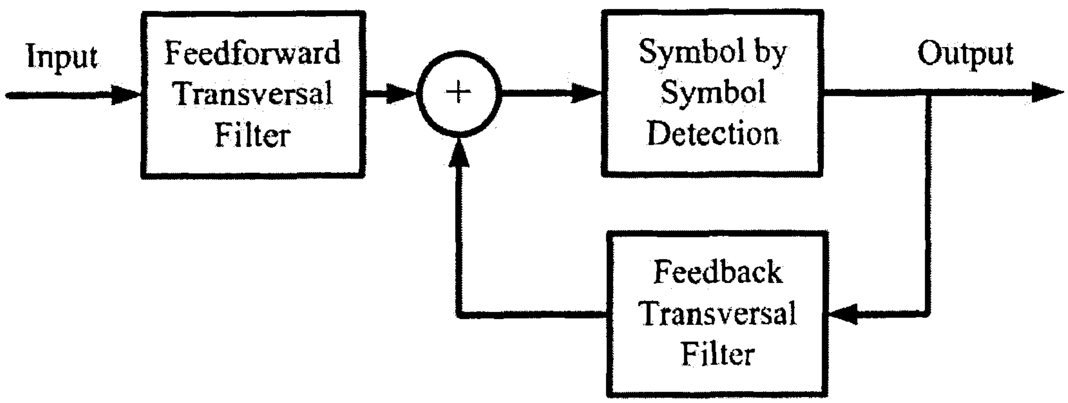
Reference [16] explains MMSE and its characteristics and presents the derivation of the above formula. MMSE does not sacrifice diversity order for the user weaker than noise level but when all the users are strong, MMSE has the same diversity order as ZF: $N_r - N_t + 1$, so it can not handle overloaded situations either. Performance achieved by

MMSE is better than ZF and is the best linear filtering is capable of. Unfortunately, it is still far from optimal.

2.6.3 Decision Feedback Methods

Limited performance of linear filtering has motivated a considerable amount of research in nonlinear detectors with low computational complexity. The decision feedback equalizer (DFE) is an effective solution to this problem. It employs the finite alphabet property of the signal in a non-linear feedback loop to the system (Figure 2.10). DFE subtracts the effects of past symbols using their estimates and improves the performance at high enough SNR while still having the same order of computational complexity as the linear equalizers (LE).

Figure 2.10 Decision feedback equalizer



The problem with DFE is that it can suffer from catastrophic error propagation at low SNRs. Like LE, DFE can not handle overloaded situations and it does not experience the enhanced diversity order caused by delay spread in the channel, because of subtracting the effects of past symbols. Both LE and DFE originally provide hard decisions but with some modifications, low complexity SISO LE and DFE equalizers are developed [10] that can be used in turbo equalization systems. Optimal MIMO DFE solutions have been investigated in [8, 9] but their performance is still far from optimal.

2.6.4 Quasi ML Techniques

Some quasi-ML methods have recently been introduced in literature that achieve near optimal performance with low computational cost. Some restrictions apply to their performance and complexity.

2.6.4.1 Soft Decision Equalization

Soft decision equalization (SDE) is introduced by Shoumin Liu and Zhi Tian [13] in the quest for a low complexity, near-optimum equalization technique. It is built on probabilistic data association (PDA) multiuser detector [14] that was originally developed for near-ML detection in CDMA systems. SDE extends this method to a narrowband, frequency selective MIMO channel described in 2.5. It also eliminates the need for ZF pre-processing performed in PDA that imposed an invertibility constraint on the channel matrix.

SDE is a block detection algorithm rather than a sequence detection method like VA. So at the end of each block it is necessary to send a sequence of zeros to eliminate the interblock interference. This is called Zero padding and causes some redundancy especially when the memory of channel is long and the block is short.

SDE is based on generating tentative soft decisions in the presence interference and simplified decision updating by forcing the composite effect of noise and interference to be Gaussian. It considers the general MIMO block matrix model $\mathbf{y} = \tilde{\mathbf{H}}\mathbf{b} + \mathbf{v}$. If $b(i)$

is the ith element of $\mathbf{b}$ and $\mathbf{h}(i)$ is the ith column of $\tilde{\mathbf{H}}$, we can rewrite the formula in this way

$$\begin{aligned} \mathbf{y} &= \mathbf{h}(i)b(i) + \sum_{j=1, j\neq i}^{N_t N} \mathbf{h}(j)b(j) + \mathbf{v}, \\ \mathbf{h}(i)b(i) &= \mathbf{y} - \sum_{j=1, j\neq i}^{N_t N} \mathbf{h}(j)b(j) + \mathbf{v}. \end{aligned} \tag{2.24}$$

Adopting the PDA filtering idea, SDE treats the transmitted bits $\{b(i)\}_1^{NN_t}$ as Gaussian random variables. So they are fully characterised by their means and variances that can be easily calculated using the apPs:

$$\mu_i = \Pr(b(i) = +1) - \Pr(b(i) = -1) = 2\Pr(b(i) = +1) - 1 \tag{2.25}$$

$$\sigma_i^2 = 4\Pr(b(i) = +1)(1 - \Pr(b(i) = +1)) \tag{2.26}$$

Because the channel is linear, $\mathbf{h}(i)b(i) \mid \mathbf{y}$ is approximated as a Gaussian vector and its pdf can be described by its mean and covariance as follows.

$$p_{b(i)|\mathbf{y}}(\mathbf{h}(i)b(i) \mid \mathbf{y}) \propto \exp\{-\frac{1}{2}(\mathbf{h}(i)b(i) - \mathbf{g}_i)^\dagger \mathbf{R}_i^{-1}(\mathbf{h}(i)b(i) - \mathbf{g}_i)\}, \tag{2.27}$$

$$\mathbf{g}_i = E\{(\mathbf{h}(i)b(i) \mid \mathbf{y})\} = \mathbf{y} - \sum_{j=1, j\neq i}^{NN_t} \mu_j \mathbf{h}(j), \tag{2.28}$$

$$\mathbf{R}_i = \operatorname{var}\{(\mathbf{h}(i)b(i) \mid \mathbf{y})\} = \sum_{j=1, j\neq i}^{NN_t} \sigma_j \mathbf{h}_j \mathbf{h}_j^\dagger + N_0 \mathbf{I}. \tag{2.29}$$

SDE updates the APPs using the pdf formula (2.27). Then the updated APPS are used to calculate new mean and variances using (2.28) and(2.29). SDE performs this process for the whole block iteratively until all the APPs converge which usually happens very quickly at 3-5 iterations for high SNR and 7-14 iterations for low SNR.

The computational complexity of this technique is polynomial in the number of inputs $O((NN_t)^3)$. Its performance is better than suboptimal MIMO-DFE. SDE experiences some performance loss in overloaded situations but unlike most other techniques, it does not fail or become too computationally complex.

A major problem with SDE is that although it is a soft decision technique, it can not be used in an iterative decoding system. Because SDE does not make use of the a priori probabilities, no matter what the apPs are, outputs would be the same. Another problem is caused by the fact that SDE is a block detection technique with computational complexity that is almost cubic with the block size. So if the blocks are long, the computational costs will be high. On the other hand if the blocks are short and the memory is long, the redundancies caused by the zero padding structure, will degrade the efficiency of the system.

2.6.4.2 Sphere Decoding

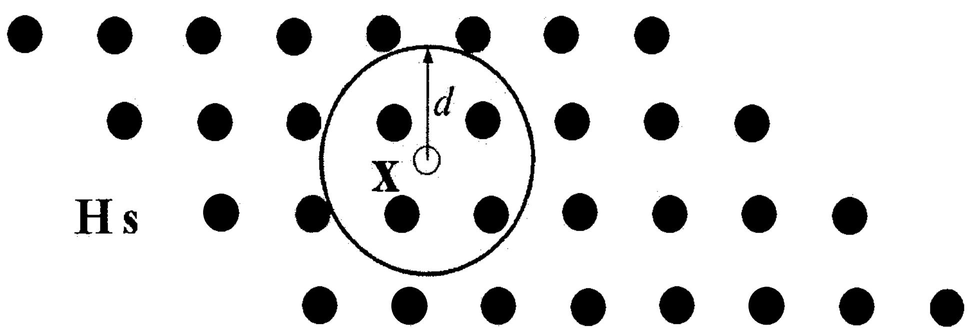
The sphere decoding algorithm is also a quasi-ML, block detection technique. It promises to find the optimal solution with low computational costs under some conditions. SD was first introduced by Finke and Pohst [11] in the context of the closest point search in lattices but it has become very popular in digital communication literature. Its various applications include lattice codes, CDMA systems, MIMO systems, global positioning system (GPS), *etc.*

As noted above, optimal ML detection leads to solving a minimum least square problem (2.12) and finding an exact solution for it is, in general, NP hard. By using the FA property of the signal, we can change this complex valued model to an integer least squares problem of the form

$$\begin{aligned} &\min_{\mathbf{s}} \|\mathbf{x}-\mathbf{H}\mathbf{s}\|^2, \\ &\mathbf{s}\in D^m \subset \mathbb{Z}^m, \\ &\mathbf{H}\in \mathbb{R}^{n\times m}, \\ &\mathbf{x}\in \mathbb{R}^{n\times 1}. \end{aligned} \tag{2.30}$$

As entries of $\mathbf{s}$ are integer, $\mathbf{s}$ spans a rectangular m-dimensional lattice and for any real matrix $\mathbf{H}$, $\mathbf{Hs}$ spans a skewed lattice. Therefore, given the real vector $\mathbf{x}$ and the skewed lattice $\mathbf{Hs}$, the integer least square problem would be equivalent to finding the closest lattice point to $\mathbf{x}$ in Euclidean sense (Figure 2.11) and SD algorithm can then be employed on this problem.

Figure 2.11 Geometric representation of the integer least-squares problem



The basic idea of SD is limit our search only to the lattice points $\mathbf{s}$ that lie in a sphere of radius d around the given vector $\mathbf{x}$ and in this way save on computations. It is clear that the closest point inside the sphere is also the closest point in the lattice. The two important questions are how to find the points inside the sphere and how to choose the radius d. SD proposes an efficient way to solve the first question but does not really address the first question. There are different methods in the literature that use the qualities of the noise and the channel to choose the radius efficiently [15].

As the integer least-square problem is NP hard, the worst case complexity of SD is still exponential. However the complexity will be a random variable because both $\mathbf{x}$ and $\mathbf{H}$ are random. So it is meaningful to investigate the average complexity of the algorithm. Reference [15] shows that the average complexity of SD is polynomial (often sub-cubic) in the number of inputs over a wide range of SNRs, rates and dimensions.

SD is originally a hard decision algorithm but with some modification it can provide soft decisions [15, 18]. This research promises a low complexity MAP detector that can be used in implementing practical, high performance iterative decoding systems. Exact Max-Log based MAP decisions can be acquired by performing a set of hard SD processes [26] so the computational complexity of the soft SD has the same sub-cubic order as the hard SD.

Reference [12] covers the application of SD in our frequency selective MIMO channel model. SD, like SDE, is a block detection algorithm so it will cause redundancy when the block is short and the channel memory is long but the blocks can not be chosen too long because the computational complexity increases cubically. The other problem is that the computational complexity increases exponentially in low SNRs and in overloaded situations.

2.7 Motivation

As mentioned above, both SD and SDE are block detection methods with and their computational complexity grows almost cubically with the number of inputs. They are not efficient for detection in long sequences and each of them have some restrictions. SD complexity grows exponentially in overloaded situations and SDE does not take hints. The other option is to perform optimal detection which has a huge computational cost or to use the conventional DFE and suffer in performance. Our goal here is to develop a SISO MUD algorithm to address some of these problems.

CHAPTER 3 UPDATING APP ALGORITHM

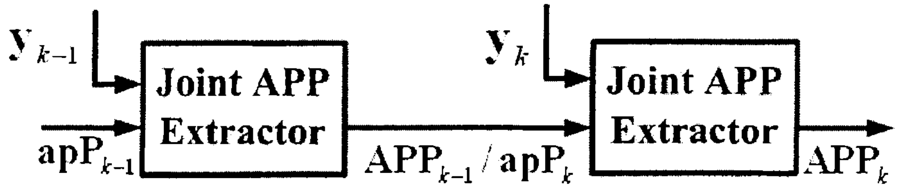
In this chapter, we introduce a sub-optimal symbol by symbol SISO multiuser detection technique. The basic idea of this updating APP (UA) algorithm is to process the received data at each time interval separately, extract the information and then pass only the partially accumulated APPs to the next time. We will show that this new algorithm achieves better BER than the conventional sub-optimal MIMO DFE and its performance is close to optimal in many cases.

Unlike the block oriented algorithms, UA complexity is linear in block size but exponential in the product of number of users and memory length. This exponential growth makes the basic UA algorithm impractical unless its complexity is reduced. We will discuss the key ideas, analyse the process and simulate the performance of the basic UA algorithm in this chapter. Then we will develop reduced complexity UA algorithm in chapter 4.

3.1 Basic UA Algorithm

Our UA equalizer is based on processing the signal received in each symbol time separately. It performs joint APP extraction at each symbol time. A joint APP extractor block, explained in 2.6.1.2, receives the apPs and the received signal in the input, and generates the updated APPs at the output. The UA algorithm then uses these soft outputs as apPs for joint APP extraction in the next symbol time. Figure 3.1 shows the basic structure of UA algorithm.

Figure 3.1 Basic structure of UA

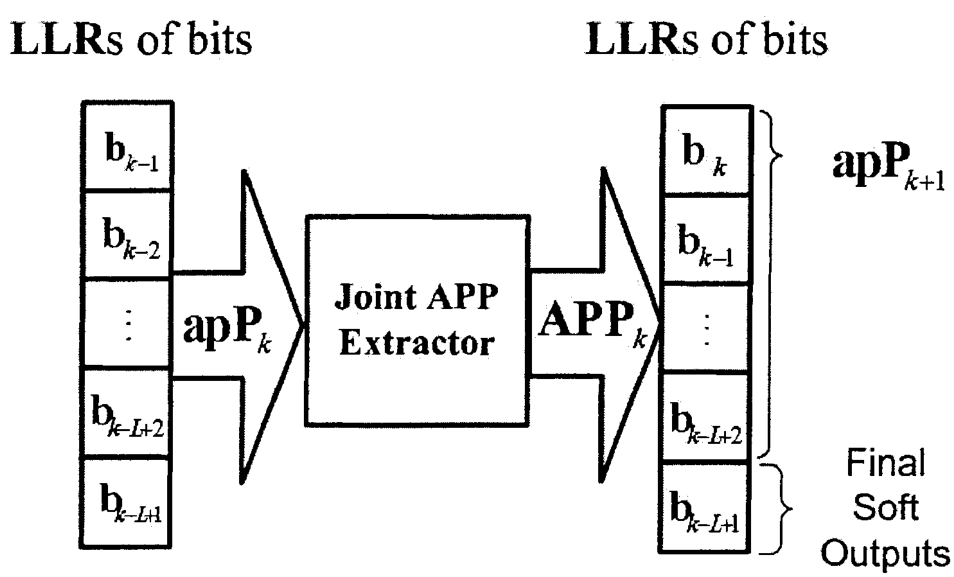


The received vector at time k, $\mathbf{y}_k$, is influenced by the symbols in the L most recent symbol times as we saw in section 2.5

$$\mathbf{y}_k = \sum_{l=0}^{L-1} \mathbf{H}_l \mathbf{b}_{k-l} + \mathbf{v}_k .$$

The probability of each symbol is updated in the L time intervals that the symbol contributes to the received signal. After that the symbol's soft decision will be final. Figure 3.2 shows the process to make these soft decisions.

Figure 3.2 The process of making soft decisions in UA



In this way, UA can provide soft decisions with a very short delay that is equal to the memory length of the channel L. In contrast, most of the other techniques like BCJR, SDE and SD have to wait for the whole block to arrive before making any decisions. Only linear algorithms and decision feedback methods can have shorter delays than UA. Even VA delays are usually 5-10 times more. This quality enables UA to process infinitely long sequences at the cost of sub-optimal performance.

Processing the signal in separate times also saves a great deal of memory. VA and BCJR need a lot of memory for the metrics of the paths, states and branches; UA just carries APPs of the symbols from time to time. Low memory requirements will simplify the detection process and make it more practical.

In this section we introduced the basic UA idea and observed that it causes short delay and reduces the memory requirements but there is still much more to explore to get a better understanding of this algorithm. Next we will analyse UA and justify the assumptions associated with it. We will assess its performance and compare it with other algorithms by performing simulations in different situations, and in the last section we estimate the complexity order of this algorithm, which is its most important restriction.

3.2 UA Analysis

As we mentioned before, UA is a sub-optimal detection algorithm and like all sub-optimal methods, it is based on some heuristics and approximations. In this section, we will analyse the formulas for joint APP extraction and explain the approximations leading to the development of UA algorithm.

In order to simplify the analysis, we consider the case where only one symbol is sent from each user at the same time. The vector of the transmitted symbols is denoted $\mathbf{b} = [b^1, b^2, ..., b^{N_t}]^T$ and $\mathbf{y} = [\mathbf{y}_0^T, \mathbf{y}_2^T, ..., \mathbf{y}_{L-1}^T]^T$ is a vector of the received vectors from time 0 to L-1. The APP of the bit b^i is $p(b^i|\mathbf{y})$ and if $\mathbf{y}_0^{L-2} = [\mathbf{y}_0^T, \mathbf{y}_2^T, ..., \mathbf{y}_{L-2}^T]^T$, we can write it as

$$
\begin{aligned}
p(b^i \mid \mathbf{y}) &= \frac{p(\mathbf{y}, b^i)}{p(\mathbf{y})} \\
&= \frac{p(\mathbf{y}_{L-1}, \mathbf{y}_0^{L-2} \mid b^i) p(b^i)}{p(\mathbf{y}_{L-1}, \mathbf{y}_0^{L-2})}
\end{aligned}
\tag{3.1}
$$

Now we'll use two approximations on (3.1) to obtain the recursive UA formulation, and examine the approximations further below. The first approximation is that the measurements are conditionally independent in order to factor the numerator. The other approximation is that the measurements are unconditionally independent, to factor the denominator. Then

$$
\begin{aligned}
p(b^i \mid \mathbf{y}) &\approx \frac{p(\mathbf{y}_{L-1} \mid b^i) p(\mathbf{y}_0^{L-2} \mid b^i) p(b^i)}{p(\mathbf{y}_{L-1}) p(\mathbf{y}_0^{L-2})} \\
&= \frac{p(\mathbf{y}_{L-1} \mid b^i) \tilde{p}_0^{L-2}(b^i)}{p(\mathbf{y}_{L-1})}.
\end{aligned}
\tag{3.2}
$$

Equation (3.2) suggests that instead of extracting APP on all the received signals we can perform APP extraction on the received signals at time L, $\mathbf{y}_{L-1}$ using the information obtained from the other received signals $\tilde{p}_0^{L-2}(b^i)$. By applying this formula recursively for the other steps, we will obtain the basic UA algorithm.

However, the UA algorithm is sub-optimal because the approximations used for developing it are not exact in most situations. The first approximation (3.3) is that measurements are conditionally independent and it is only exact when the only transmitted bit is b^i.

$$p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}\,|\,b^i) \approx p(\mathbf{y}_{L-1}\,|\,b^i)\,p(\mathbf{y}_0^{L-2}\,|\,b^i)\,. \tag{3.3}$$

The exact formula is

$$\begin{aligned} p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}\,|\,b^i) &= \sum_{\overline{\mathbf{b}}^i} p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}\,|\,\overline{\mathbf{b}}^i,b^i)\,p(\overline{\mathbf{b}}^i) \\ &= \sum_{\overline{\mathbf{b}}_i} p(\mathbf{y}_{L-1}\,|\,\overline{\mathbf{b}}^i,b^i)\,p(\mathbf{y}_0^{L-2}\,|\,\overline{\mathbf{b}}^i,b^i)\,p(\overline{\mathbf{b}}^i). \end{aligned} \tag{3.4}$$

where $\overline{\mathbf{b}}^i$ is the vector of bits from users other than i.

The second approximation (3.5) is that the measurements are unconditionally independent. This one is not accurate even when there is one bit in the system.

$$p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}) \approx p(\mathbf{y}_{L-1})\,p(\mathbf{y}_0^{L-2}) \tag{3.5}$$

The exact formula is

$$\begin{aligned} p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}) &= \sum_{\mathbf{b}} p(\mathbf{y}_{L-1},\mathbf{y}_0^{L-2}\,|\,\mathbf{b})\,p(\mathbf{b}) \\ &= \sum_{\mathbf{b}} p(\mathbf{y}_{L-1}\,|\,\mathbf{b})\,p(\mathbf{y}_0^{L-2}\,|\,\mathbf{b})\,p(\mathbf{b}). \end{aligned} \tag{3.6}$$

Although these approximations are not exactly true, it is an approximation that intuitively justifies the steps of UA algorithm. That is why UA is a sub-optimal algorithm rather than an optimal method and the only way we can estimate the amount of loss is through performing simulations in different situations.

3.3 UA Performance

To evaluate the UA algorithm, we study the BER performance versus SNR of various detectors in a simulated block fading system model explained in 2.5. BCJR represents the optimal performance and MMSE-DFE shows the traditional sub-optimal results. Recently introduced near-optimal SDE is also illustrated in the figures to allow better judgments.

In these simulations we use an exponential power delay profile which is common to observe in an urban setting [30]. So the mean power of channel taps decreases exponentially and is truncated at L,

$$\begin{gathered} P_i = ke^{-\tau i},\ i = 0 \ldots L-1 \\ k \text{ constant} \\ \tau \text{ decay constant} \end{gathered} \tag{3.7}$$

The exact value of the decay constant for is different for different channels. Throughout this thesis τ is chosen to be 1 so only the first 2-3 taps will be strong enough in SNRs of interest to contribute to diversity order. SNR is defined as sum of the mean power of all taps divided by noise power

$$SNR = \left(\sum_{i=0}^{L-1} P_i\right) / N_0 = \frac{k}{N_0}\left(\sum_{i=0}^{L-1} e^{-i}\right) \tag{3.8}$$

All the simulations are performed for BPSK signals over 10,000 random channels for the block length of 10 bits. Although UA can handle blocks of any length, the blocks are kept short to facilitate comparison with block oriented SDE, since its complexity grows cubically with sequence length.

We compare the results under three different conditions; critically loaded where the number of transmit antennas is equal to the receive antennas, underloaded where the transmitters are less than receivers and overloaded where it is the reverse. These simulations will give us an idea about the UA performance and characteristics.

3.3.1 Critically Loaded

We examine the case where the number of transmit antennas is equal to the number of receive antennas $N_t = N_r = 2$, memory length $L = 3$ and block length is $N = 10$ over 10,000 runs. Figure 3.3 shows the results; UA outperforms MMSE-DFE and is within 1 dB from the optimal BCJR over most of the SNR range.

3.3.2 Under Loaded

Figure 3.4 shows an under loaded condition where receivers are more than transmitters $N_t = 2$, $N_r = 4$, $L = 3$ and $N = 10$ over 10,000 turns. In this situation the loss of the UA algorithm is negligible and in the order of fraction of a dB. Unlike MMSE-DFE, UA also enjoys full diversity order.

3.3.3 Overloaded

Studying the overloaded situation is important because most of the detection techniques experience difficulties in handling this case Figure 3.5 illustrates the performance of different algorithms where number of receivers is less than the number of transmitters $N_t = 2$, $N_r = 1$, $L = 3$ and $N = 10$ over 10,000 random runs. As we see, MMSE-DFE experiences an irreducible error floor. SDE and UA are close together and experience some loss compared to optimal BCJR. SDE converges to a floor BER in high

SNRs while basic UA seems to just suffer from 1.5 dB loss in SNR. The estimates that SDE uses for soft interference cancellation are not accurate enough in this case and cause the error floor.

Figure 3.3 BER performance of critically loaded system $N_t = N_r = 2$, $L = 3$ and $N = 10$

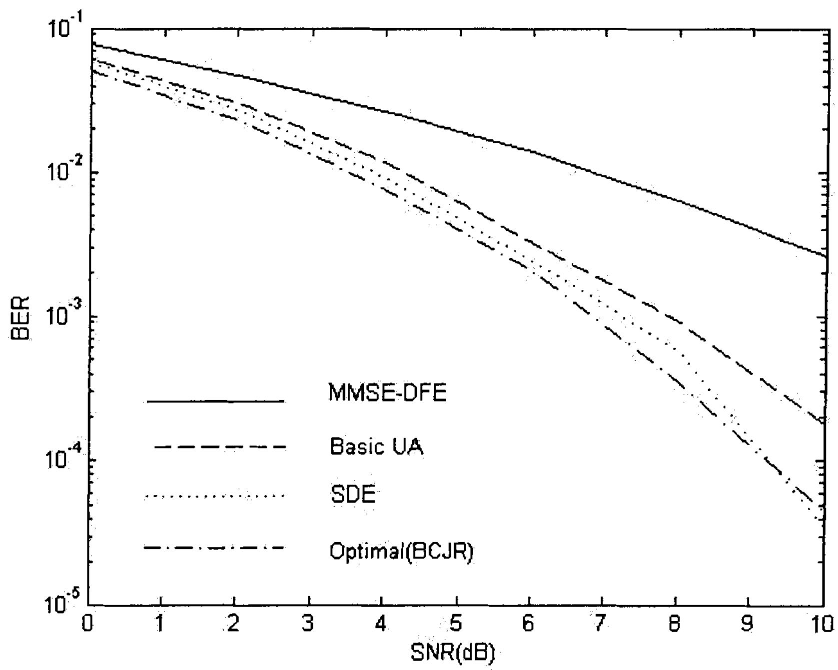


Figure 3.4 BER performance of under loaded system $N_t = 2$, $N_r = 4$, $L = 3$ and $N = 10$

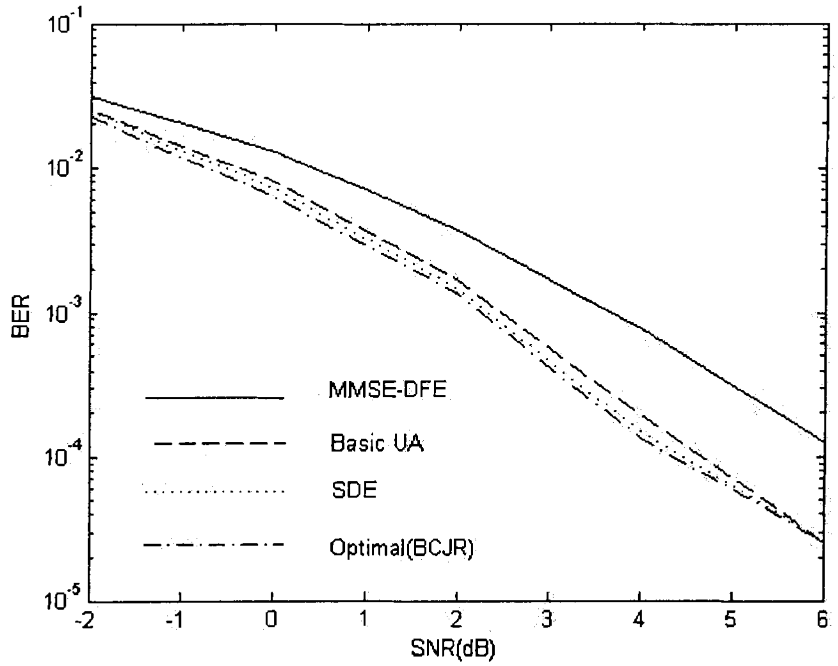
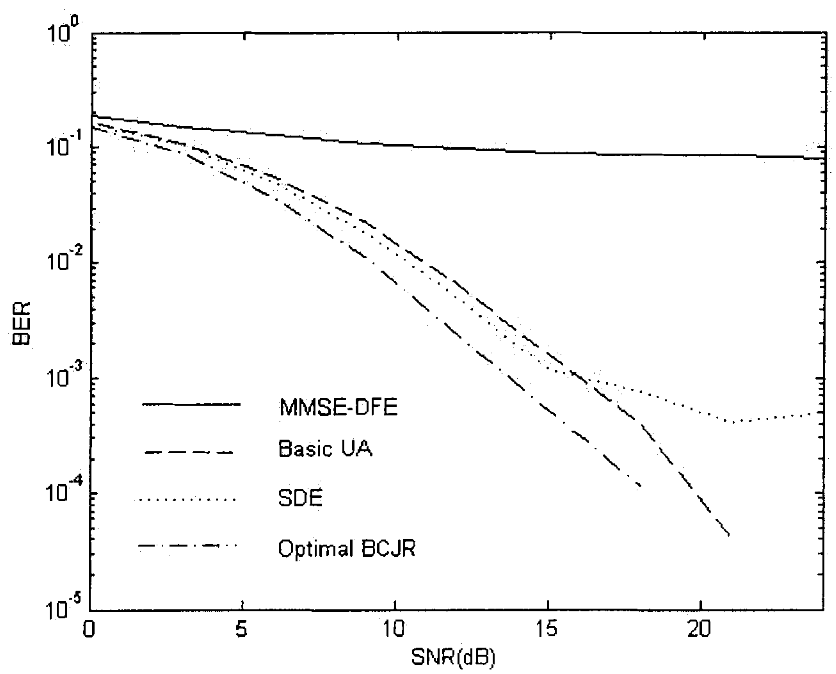


Figure 3.5 BER performance of overloaded system $N_t = 2$, $N_r = 1$, $L = 3$ and $N = 10$



3.4 UA Complexity

UA processes each received vector separately from the others so its computational complexity is linear in the number of symbols, but each of these processes in the basic algorithm includes joint APP extraction, and the complexity of optimal joint APP extraction is exponential in the product of number of users and memory length, as we explained in 2.6.1.2.

This exponential growth can cause an impracticable computational cost when the users are numerous or the channel memory length is large. To implement this system we need to modify the optimal joint APP extractor in order to decrease its computational load without sacrificing much of the performance. That modification is the topic of chapter 4.

3.5 Summary

In this chapter we introduced the basic UA algorithms and explored its characteristics. UA performance is far better than sub-optimal DFE methods and is close to optimal except when it is working in an overloaded situation. Even then its BER is acceptable, and does not exhibit a floor. UA needs less memory than most other detection algorithms, and its complexity grows linearly with block size. However, it is not practical in the basic form presented in this chapter, because the number of computations at each step grows exponentially with the product of the number of users and the memory length.

CHAPTER 4 COMPLEXITY REDUCTION IN UA ALGORITHM

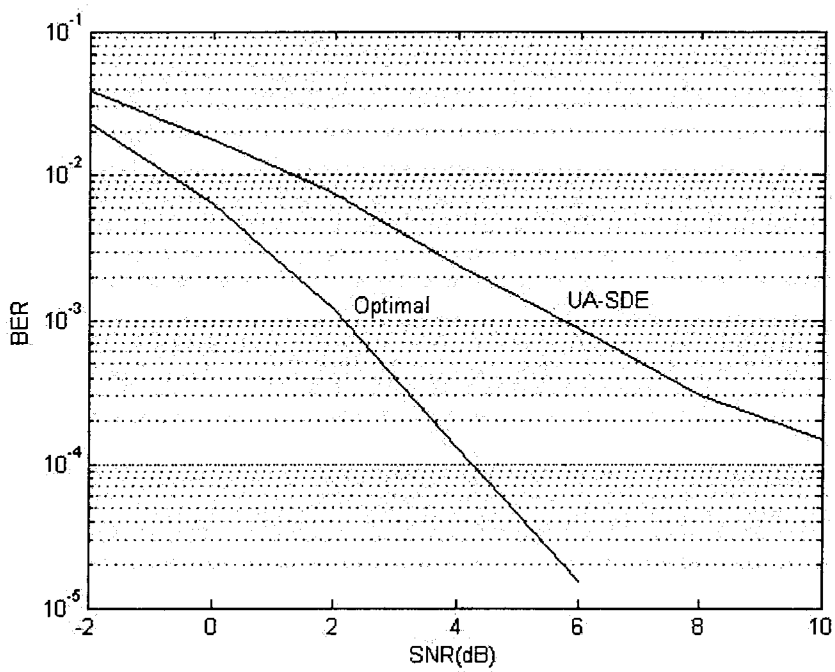
As explained above, the duty of the joint APP extractor block in the UA structure is to perform SISO MUD. However, the optimal joint APP extraction, with its exponential computational complexity, makes the basic UA algorithm impractical. Our goal in this chapter is to replace the optimal joint APP extraction with another SISO MUD algorithm in order to enjoy the good qualities of UA with reduced complexity and minimal performance degradation.

There are a number of ways in the literature to perform MUD with less computational complexity, such as SD, SDE defined in chapter 2 and iterative MUD (IMUD) that will be explained later. SD is not a proper option for this application because it has polynomial complexity only when we have a full rank channel matrix, and with rank deficiency the computations grow exponentially with number of deficiencies. So in our case, SD is not desirable because the MUD in the UA algorithm is usually overloaded; that is the product of the number of users and memory length usually exceeds the number of receiver antennas.

Using an SDE-based MUD technique for joint APP extraction will have polynomial computational complexity but SDE experiences huge performance loss in an overloaded situation. Moreover, it does not "take hints" so it can not use the apPs from the previous time intervals.

As a result, SDE is not a good candidate to replace the optimal joint APP extractor. This fact can also be confirmed by performing simulations. Figure 4.1 compares the performance of optimal detection and UA-SDE algorithm and shows a loss of more than 3 dB in most cases. The simulations comprise of 10,000 runs in a system with 2 transmit and 4 receive antennas and block length of 100 bits.

Figure 4.1 Performance comparison $N_t = 2$, $N_r = 4$, $L = 5$ **and** $N = 100$



On the other hand, the IMUD algorithm recently introduced in [19] is an iterative groupwise SISO MUD method that reduces the computations especially in overloaded cases where the number of transmit antennas is more than receivers. So applying its principles to our problem seems to be beneficial. Therefore, we propose replacing the optimal joint APP extractor with a suboptimal iterative group detection technique inspired by IMUD to reduce the computational complexity of UA algorithm without causing much degradation in performance.

In the rest of this chapter we will introduce IMUD and explain how we can adapt it according to the nature of our problem. We also suggest implementing the core APP extractor in the group detector by a soft SD to further reduce the computations. Then we will explore the performance of the resulting reduced complexity UA algorithm and estimate its performance and characteristics.

4.1 Iterative Multiuser Detection

The IMUD algorithm divides the interfering symbols into non-overlapping groups that are detected separately and in succession. To detect a group, IMUD first removes the effect of all other groups by soft cancellation. Then the results go through a joint APP extractor which provides soft decisions for the symbols in the selected group. The soft decisions in turn allow soft cancellation of that group's symbols in the detection of subsequent groups.

After all the groups are detected, we do the whole process again and iterate until all the APPs converge, which usually happens in 2-3 iterations. So the computational complexity of IMUD is exponential in the group size N_g, instead of the whole number of symbols. Because N_g is smaller than the number of all symbols, we gain a huge reduction in computation at the cost of relatively small performance degradation.

In the original IMUD, users are assigned to a group in MMSE-VBLAST order, but our problem has some natural grouping strategies according to the transmission time.

$$\mathbf{y}_k = \mathbf{H}\tilde{\mathbf{b}}_K + \mathbf{v}_k \tag{4.1}$$

$$\mathbf{H} = [\mathbf{H}_{L-1}, \cdots, \mathbf{H}_1, \mathbf{H}_0]$$

$$\tilde{\mathbf{b}}_k = [\mathbf{b}_{k-L+1}^T \quad \cdots \quad \mathbf{b}_k^T]^T$$

The whole $N_t L$ interfering symbols in $\tilde{\mathbf{b}}_k$ can be divided into L groups $\mathcal{G}_i = \{\mathbf{b}_{k+1-i}\}$, $i = 1...L$ of the N_t symbols comprising $\mathbf{b}_{k+1-i}$. We define $\mathbf{h}_i$ as the i$^{\text{th}}$ column of $\mathbf{H}$ and b_i as the bit associated with it. We can rewrite (4.1), symbol-wise and group-wise, respectively

$$\mathbf{y}_k = \sum_{i=1}^{N_t L} \mathbf{h}_i b_i + \mathbf{v}_k, \tag{4.2}$$

$$\mathbf{y}_k = \sum_{b_i \in \mathcal{G}_j} \mathbf{h}_i b_i + \sum_{i=1, b_i \notin \mathcal{G}_j}^{N_t L} \mathbf{h}_i b_i + \mathbf{v}_k, \qquad j = 1...L. \tag{4.3}$$

Using the a priori probabilities, we can calculate the mean and variance of each bit

$$\mu_i = \Pr(b_i = +1) - \Pr(b_i = -1) = 2\Pr(b_i = +1) - 1, \tag{4.4}$$

$$\sigma_i^2 = 4\Pr(b_i = +1)(1 - \Pr(b_i = +1)). \tag{4.5}$$

What we know about b_i is μ_i and σ_i^2 shows the amount of uncertainty. For soft cancellation of the effects of this bit on the measurements we should subtract the mean and account for its variance in correlation matrix of noise. In other words, we suppose that what we do not know about b_i is a complex, zero mean Gaussian random noise with variance σ_i^2. Equations below show the results when all the groups except $\mathcal{G}_j$ (the group being detected) are cancelled.

$$\mathbf{y}_k^j = \mathbf{y}_k - \sum_{i=1, b_i \notin \mathcal{G}_j}^{N_t L} \mathbf{h}_i \mu_i, \qquad j = 1 \ldots L \tag{4.6}$$

$$\mathbf{y}_k^j = \sum_{b_i \in \mathcal{G}_j} \mathbf{h}_i b_i + \mathbf{n}_k^j \tag{4.7}$$

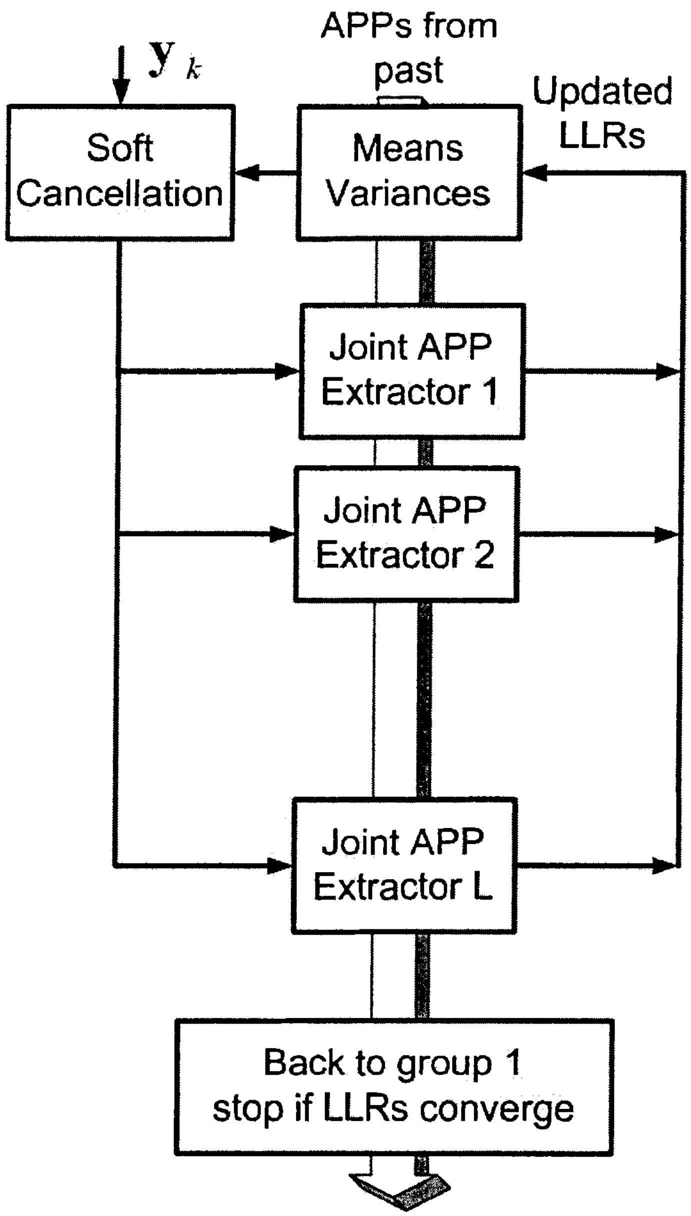
$$\mathbf{R}_k^j = \sum_{i=1, b_i \notin \mathcal{G}_j}^{N_t L} \mathbf{h}_i \mathbf{h}_i^\dagger \sigma_i^2 + \mathbf{I} \tag{4.8}$$

We then feed $\mathbf{y}_k^j$, $\mathbf{R}_k^j$ and a priori probabilities of the symbols in $\mathcal{G}_j$ to a joint APP extractor and use the resulting LLRs to update means and variances of the symbols in $\mathcal{G}_j$.

Symbols in $\mathcal{G}_k$ are the most recent symbols, so we have no a priori information about them. The best strategy is to start the detection from this group because we have apP of all other groups so we are able to cancel them. Then $\mathcal{G}_k$ will be detected with the best quality. Next, we detect the other groups in descending order. After all the groups are finished we go back to $\mathcal{G}_k$ and start the process again. LLRs typically converge in two (sometimes three) iterations.

It should be noted that the joint APP extractor here is a little bit different from the one in section 2.6.1.2 because of the colored noise (or we can still use that after first whitening the noise using Cholesky decomposition). Figure 4.2 shows the structure of the resulting UA-IMUD system.

Figure 4.2 UA-IMUD structure



4.2 Soft Sphere Decoder

The core of the UA-IMUD algorithm (Figure 4.2) still employs the optimal joint APP extractor. We can further reduce the computational costs of UA algorithm, if we replace the optimal joint APP extractors in the group detector by soft sphere decoders. Soft SD finds the signal points in a sphere of predetermined radius around $\mathbf{y}_k^j$. Then instead of the optimal exhaustive search on all possible points, it restricts the calculations only to the points in the sphere. LLRs are then calculated as

$$L(b_i|\mathbf{y}_k) = \log \frac{\sum_{\mathbf{b}\in S:b_i=+1} e^{(-\|\mathbf{y}_k-\mathbf{Hb}\|^2+\sum_j \log p(b_j))}}{\sum_{\mathbf{b}\in S:b_i=-1} e^{(-\|\mathbf{y}_k-\mathbf{Hb}\|^2+\sum_j \log p(b_j))}} \quad (4.9)$$

$$S = \{\text{points in the sphere}\}$$

This way, the computational complexity of this block decreases from 2^{N_t} to $O(N_t^3)$ if $N_t \le N_r$ and to $O(2^{N_t-N_r} N_r^3)$ if $N_t > N_r$. In the case where $L=1$, there is no memory in the system and UA would be symbol by symbol multiuser detection via sphere decoding.

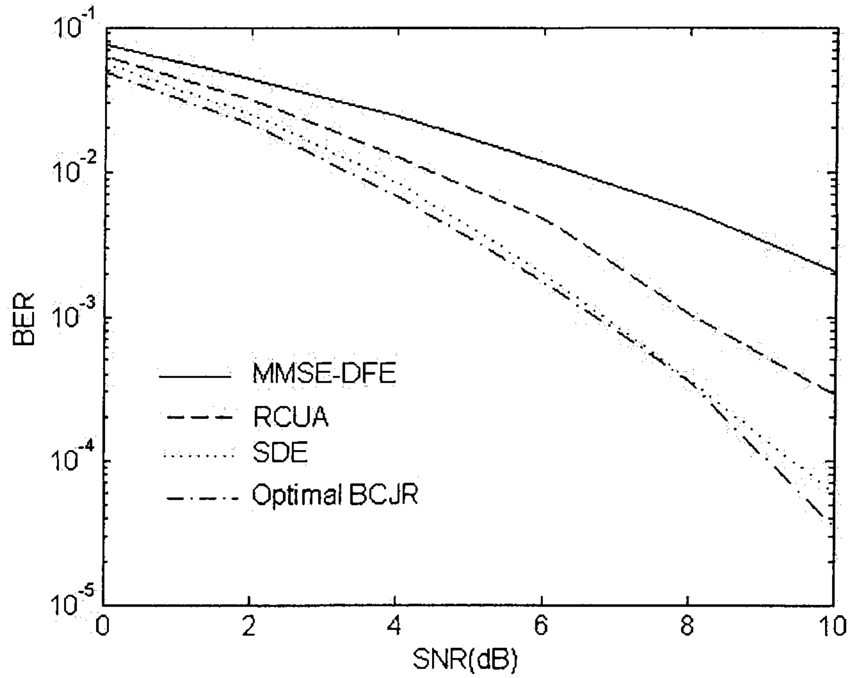
4.3 BER Performance

Reduced complexity UA (RCUA) algorithm solves the problem of excessive computational load of basic UA by using a combination of IMUD and soft SD but because both IMUD and soft SD are suboptimal algorithms, RCUA will suffer from some performance loss. In this section we will evaluate the BER performance versus SNR of RCUA, compared with various detectors by performing simulations similar to those in section 3.3.

4.3.1 Critically Loaded System $N_t = N_r$

Figure 4.3 shows the case where $N_t = N_r = 2$, $L = 4$ and block size $N = 10$. As we see, RCUA algorithm still shows a better result than MMSE-DFE not only in BER but also in the diversity that it enjoys, shown from the slope of the curve. It is within 1.5 dB from the optimal BCJR over most of the SNR range. The loss of RCUA compared to the basic UA of chapter 3 seems negligible in this case (see Figure 3.3 for comparison).

Figure 4.3 BER performance of critically loaded system $N_t = N_r = 2$, $L = 4$ **and** $N = 10$



4.3.2 Under Loaded System $N_t < N_r$

Figure 4.4 shows the case where $N_r = 4$, $N_t = 2$, $L = 4$ and block size $N = 10$. RCUA performance is very good. Again it does not show a noticeable degradation from basic UA. As it seems the extra information coming from more antennas helps the algorithm to feed back better estimates and improve the performance.

4.3.3 Overloaded System $N_t > N_r$

Most techniques handle this case poorly. SD can keep the optimal performance only by computational complexity that grows exponentially with the product of number of deficiencies and block size. SDE manages to keep a sub-optimal performance, though with an error floor. However, it cannot be used in an iterative system, so it can not improve performance through turbo equalization and decoding.

As we see in Figure 4.6 where $N_r = 1$, $N_t = 2$, $L = 4$ and $N = 10$, RCUA keeps a suboptimal performance but still no irreducible error floor. It seems that in this case RCUA suffers the most loss in performance and diversity order (compare with Figure 3.5). The reason is that the system is overloaded and we gain less information from each step. So the soft cancellations in next steps will be less accurate and the overall loss will be more than other cases.

In comparison SDE offers better performance and diversity order at low SNRs but saturates at a floor BER value. As the degree of overload $N_t - N_r$ grows, both SDE and RCUA experience substantially increased performance loss. Figure 4.6 and Figure 4.7 illustrate this situation. As we see SDE performance deteriorates more than RCUA with increased degree of overload.

Figure 4.4 BER performance of under loaded system $N_r = 4$, $N_t = 2$, $L = 4$ and $N = 10$

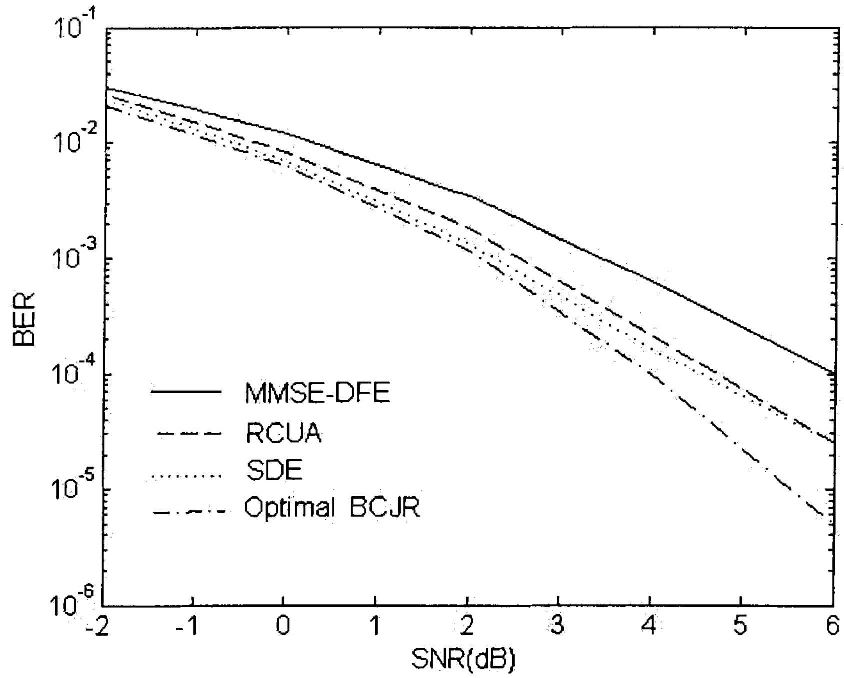


Figure 4.5 BER performance of overloaded system $N_t = 2$, $N_r = 1$, $L = 3$ and $N = 10$

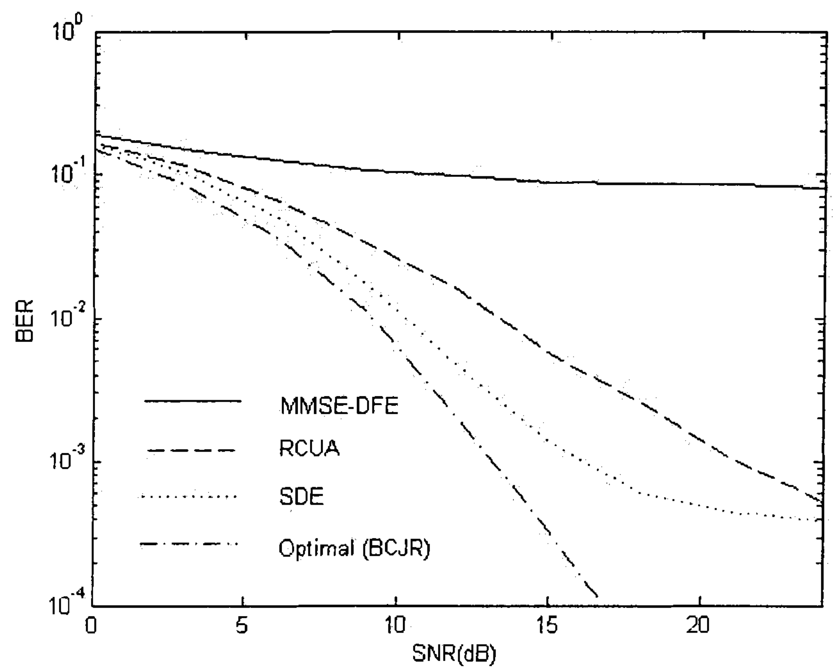


Figure 4.6 BER performance of overloaded system $N_r = 1$, $N_t = 3$, $L = 4$ and $N = 10$

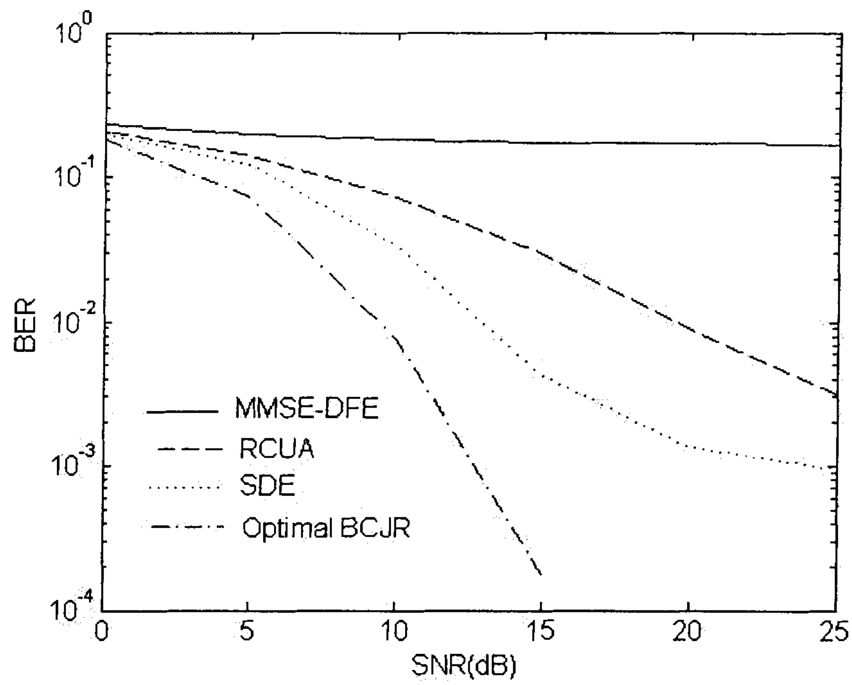
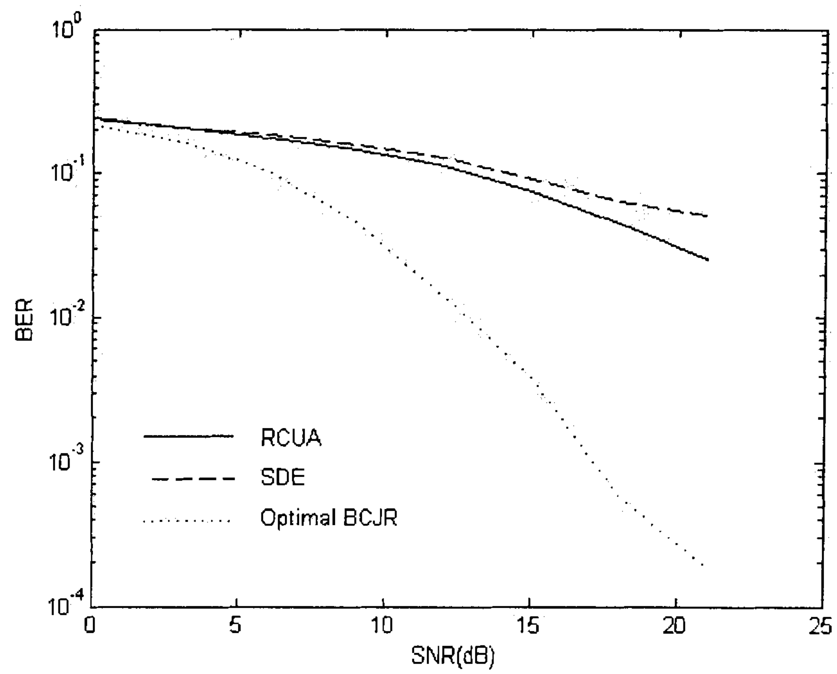


Figure 4.7 BER performance of overloaded system $N_r = 1$, $N_t = 4$, $L = 2$ and $N = 10$



4.4 Complexity of RCUA

In last section we saw that reducing the complexity of UA algorithm causes small performance loss in most cases and we compared the performance with other algorithms. Now, we have to evaluate the overall computational cost of RCUA. As we mentioned in Chapter 3, the order of computational complexity in basic UA is $O(N2^{N_tL})$. By using IMUD we will reduce $O(N2^{N_tL})$ to $O(NL2^{N_t})$ because the groups will each have N_t users. When we use SD as the core, $O(2^{N_t})$ will be reduced to $O(N_r^3 2^{N_t-N_r})$ in an overloaded system and to $O(N_t^3)$ in a non-overloaded case as we explained in chapter 2.

Table 4.1 below compares the asymptotic computational complexity and performance of different algorithms in overloaded and non-overloaded situations provided that the SNRs are high enough. It shows that RCUA is very flexible, and does not fail or become too computationally complex in any situation.

However, the table does not show how low SNRs affect the computational complexity increases of different algorithms. SDE needs more iterations to converge and the number of points in the sphere increases as the radius of the sphere grows in SD [15]. The same thing happens for the core SD in RCUA but the effect in not huge because the worst case for RCUA is $O(NL2^{N_t})$ which far less than the worst case for SD $O(2^{N_tN})$.

Table 4.1 Comparison of asymptotic (high SNR) computational complexities and performance

	Overloaded	Non-overloaded
BCJR	$O(N2^{N_t L})$ Optimal	$O(N2^{N_t L})$ Optimal
SDE	$O((NN_t)^3)$ Sub-optimal	$O((NN_t)^3)$ Near-optimal
SD	$O((NN_r)^3 2^{N(Nt-Nr)})$ Optimal but too complex	$O((NN_t)^3)$ Optimal
MMSE-DFE	$O(N((N_t+N_r)L)^3)$ Fails	$O(N((N_t+N_r)L)^3)$ Sub-optimal
RCUA	$O(NLN_r^3 2^{(N_t-N_r)})$ Sub-optimal	$O(NLN_t^3)$ Sub-optimal

4.5 Characteristics

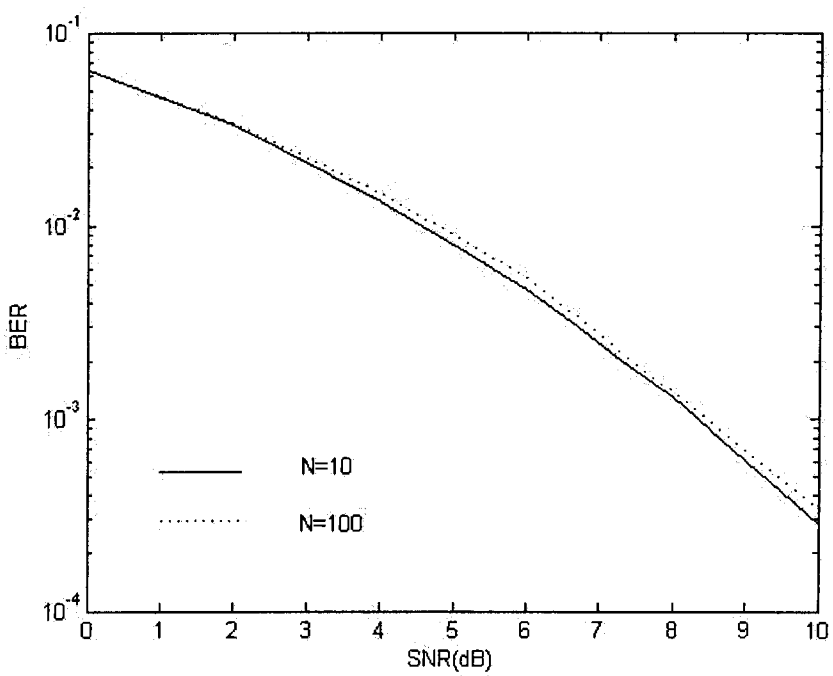
In this section we investigate various characteristics of UA algorithm.

4.5.1 Different Block Lengths

We have been using a block fading system model in this thesis to develop UA algorithm and compare its performance with the block oriented techniques. However, this algorithm is not block oriented and it can process long or infinite streams of data. It can also be used in adaptive systems where the channel impulse response changes over a block.

As an experiment we changed the block length in simulations and we expected that there would be no change in the performance of the system. Figure 4.8 shows the BER performance does not change within the accuracy of the results. The simulations have run 10,000 times on a system with $N_t = N_r = 2$ and $L = 3$.

Figure 4.8 Comparison of performance with different block lengths $N_t = N_r = 2$ **and** $L = 3$

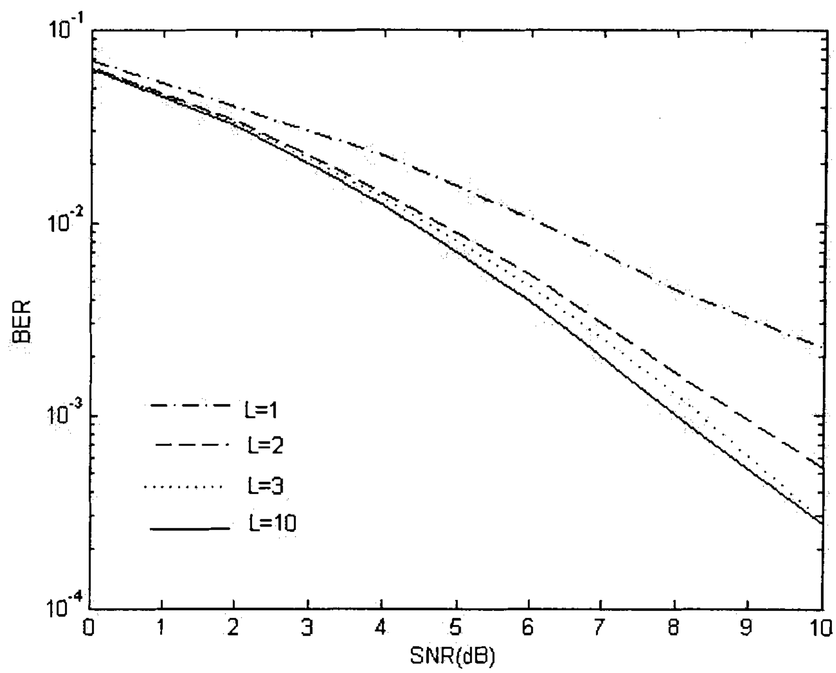


4.5.2 Different Channel Memory Lengths

Within our exponential power delay profile we changed the memory length of the channel to observe the performance of RCUA detector under different channel models.

Figure 4.9 shows the results of simulations performed 10,000 times on a system with $N_t = N_r = 2$ and $N = 10$. There is a noticeable improvement in performance when L increases from 1 to 2. The reason is if $L = 1$, channel has no memory and we receive the signal only once and this lack of diversity causes the performance to be worse than the case where $L = 2$. There is a small further improvement for $L = 3$ and almost no further improvement for $L = 10$. This observation can be justified by the fact that only the first few delayed versions of the signal are strong enough to increase the diversity order or improve the performance of the system.

Figure 4.9 Comparison of performance with different channel memory lengths $N_t = N_r = 2$ and $N = 10$



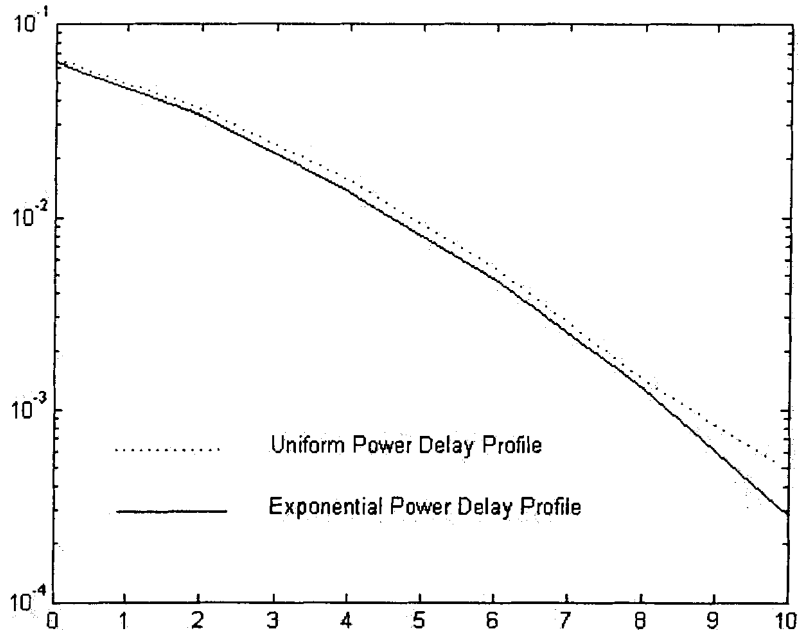
4.5.3 Different Power Delay Profiles

We have done all our previous simulations on systems with exponential power delay profile. As we mentioned before, this power delay profile is commonly observed in an urban setting. It is interesting though to investigate the effects of changing the power delay profile on the performance of our RCUA detector. For this comparison we have chosen a uniform power delay profile where all the taps of the channel have equal average power. Again we choose $N_t = N_r = 2$, $N = 10$, $L = 3$ and perform the experiment 10,000 times.

The results are illustrated in Figure 4.10 and show a small degradation in performance in the uniform case compared to exponential. The reason has to do with the RCUA algorithm's important first estimates of the signal, which are taken from the lowest delay arrivals (Section 4.1). In an exponential profile these have strength significantly above the average of the taps. In a uniform setting the first estimates of the signal are less accurate as less power is the first few taps compared to the exponential setting. These less accurate estimates will then be used as apPs for soft cancellation in the next steps of detection and cause the small performance degradation.

The fact that the difference between two different cases is very small shows that RCUA performs almost the same under different power delay profiles so we can generalize the results obtained in other sections under exponential power delay profile.

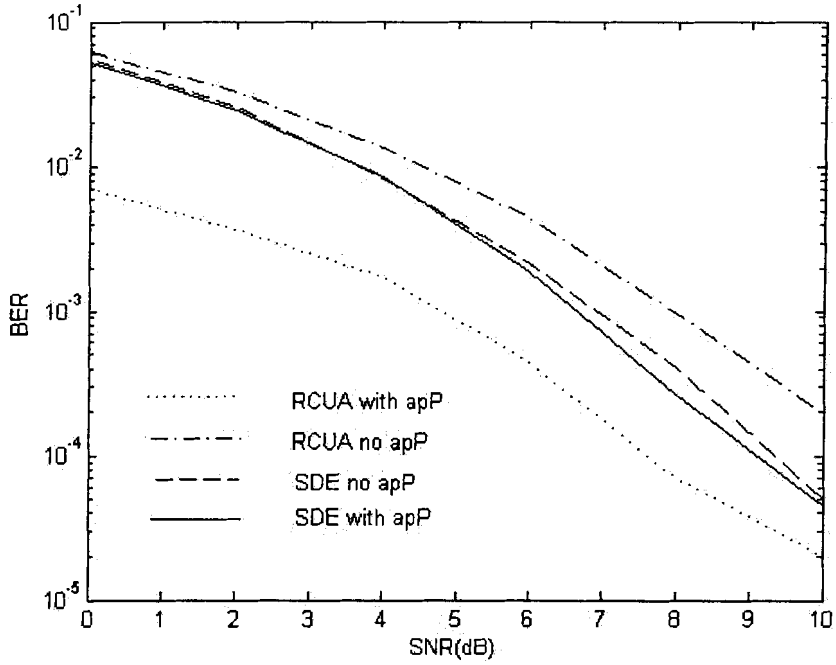
Figure 4.10 Performance comparison with exponential and uniform power delay profiles $N_t = N_r = 2$, $N = 10$ and $L = 3$



4.5.4 Effect of *a priori* Information

We have mentioned before that unlike UA, SDE does not take hints. To prove this fact, we have done some simulations to compare the performance of SDE and RCUA in the cases where the detectors have some or no a priori information about the signal. Figure 4.11 shows the results of a simulation performed in the case. $N_t = N_r = 2$, $L = 4$, $N = 10$ and a priori LLRs are +1 for +1 and -1 for -1 which means that the probability of correct bit is 0.7311. As we see SDE does not use these hints but RCUA uses the information and improves the BER performance. It is obvious from these results that SDE can not be used in an iterative decoding system.

Figure 4.11 Effect of *a priori* information
$N_t = N_r = 2$, $L = 4$ **and** $N = 10$



4.5.5 Sharing Information

In the last section, we saw that UA has the ability to utilize a priori information. We also know that UA can provide us soft information with small delays and relatively small computational costs. Now if we use UA in a cellular system, different base stations can share their information. This is called macrodiversity, and it is a solution for shadowing problems traditionally because if a user is in the shadow region of one base station it might have acceptable reception for the link to another base station.

Existing literature about macrodiversity [32,33] show that it can increase the capacity of the system by increasing the number of co-channel signals but these investigations have used optimal detection techniques. It is interesting and promising to consider the performance of cellular macrodiversity system using RCUA. This is the topic of our future research.

4.5.6 Near-Far Effect

In all the cases we have investigated before the users had equal powers but in a real situation users will have different mean power. This difference in power level is usually caused by path loss as the users nearer to base stations will be stronger than the far users and might cause relatively high interference for the weak users. To avoid this near-far effect, power control systems are usually necessary.

Here we have investigated the near far effect caused by different detection algorithms. The simulations are performed in a system with 2 users in which one of them is 5 dB stronger than the other $N_t = N_r = 2$, $L = 3$, $N = 10$. Figures below illustrate the BER performance of each user against its own SNR for different algorithms. As we see in MMSE-DFE the weak user experiences some loss but in BCJR, SDE and RCUA the difference in performance is small and helps the weak user.

Figure 4.12 Near-far effect in MMSE-DFE
$N_t = N_r = 2$, $L = 3$ and $N = 10$

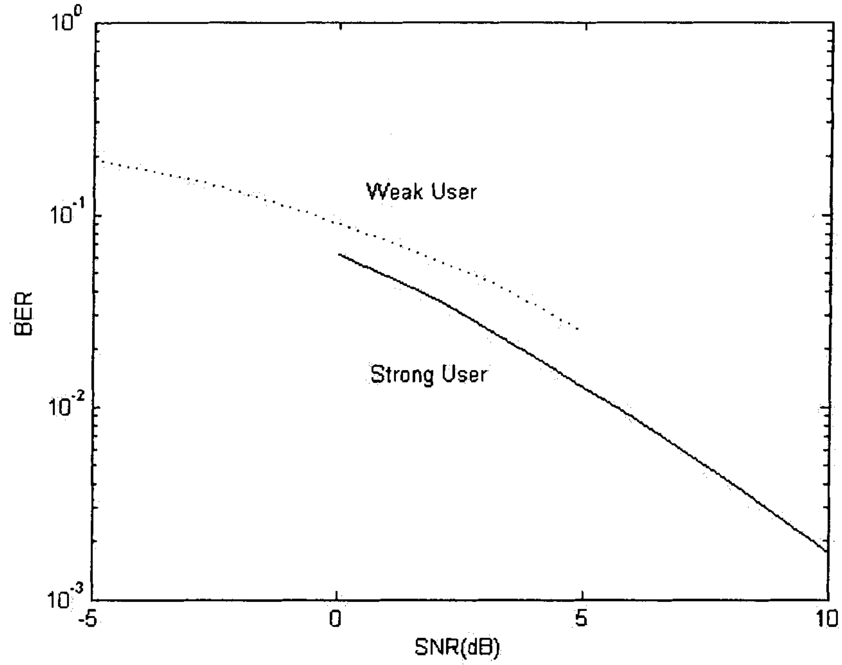


Figure 4.13 Near-far effect in BCJR
$N_t = N_r = 2$, $L = 3$ and $N = 10$

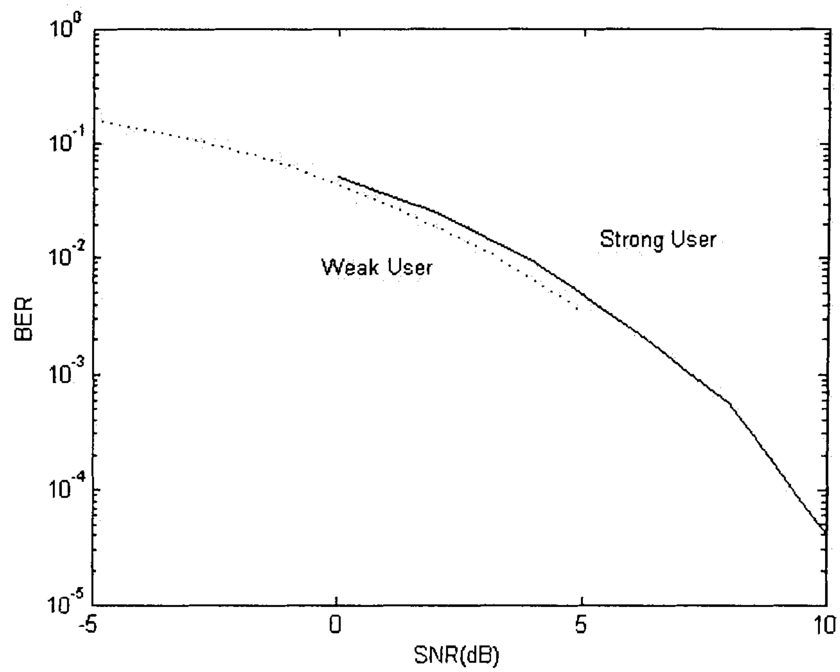


Figure 4.14 Near-far effect in RCUA
$N_t = N_r = 2$, $L = 3$ and $N = 10$

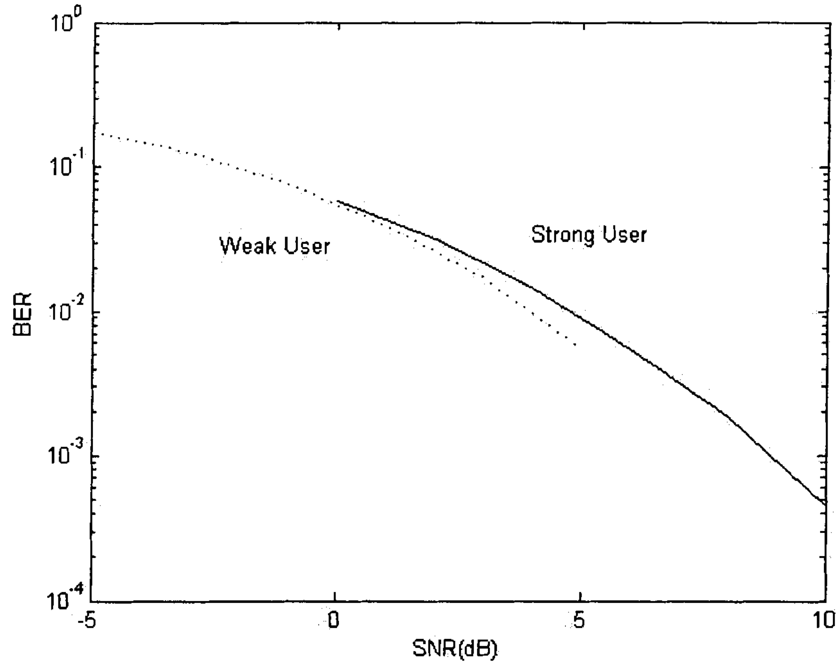
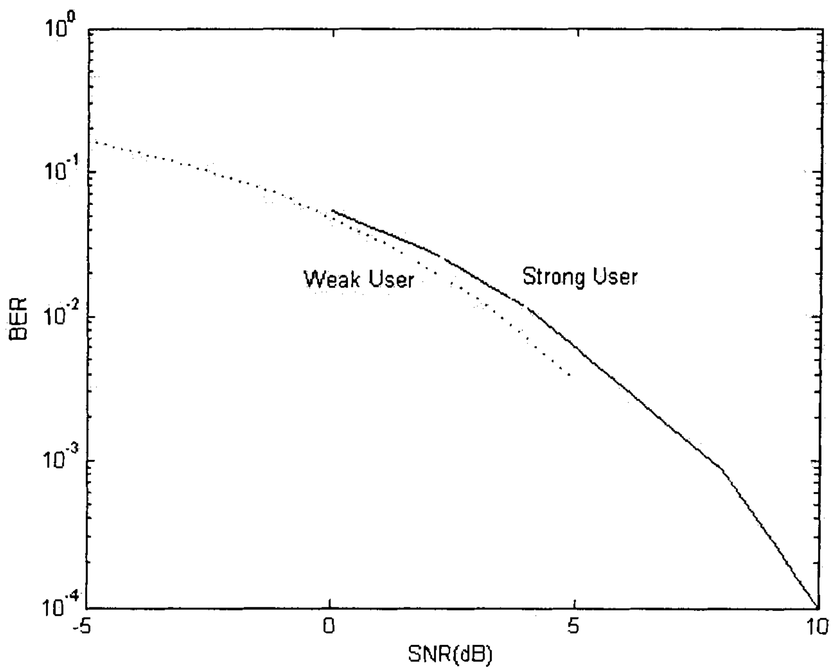


Figure 4.15 Near-far effect in SDE
$N_t = N_r = 2$, $L = 3$ and $N = 10$



4.6 Different Variations in RCUA

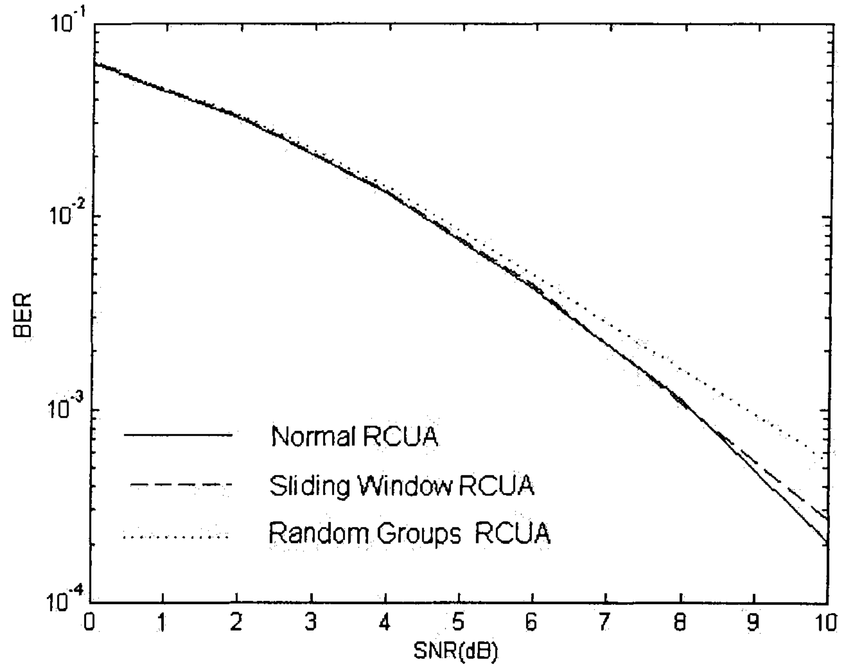
As we mentioned before we adapt IMUD according to the nature of our system model to develop RCUA. So each group will include the symbols sent at a time interval and group detection starts with the most recent group because we have the least amount of information about its symbols. Iterations continue successively on the same groups. These steps describe the normal RCUA algorithm but many other variations to ordering and grouping is also possible.

One way is to form the groups by a sliding window of length N_t. Detection still starts from the most recent bits sent but the next group will be only one bit different from the previous one. This sliding window version of RCUA increases computations in one iteration by a factor N_t, but we can not predict how it influences the performance or if it needs fewer iterations to converge.

As the performance of the normal IMUD algorithm [19] improves with random group selection, another suggestion is to do the first iteration in normal RCUA format and for the next iterations choose random groups. Figure 4.16 compares the performance of these two variations with the normal RCUA through simulation in the case where $N_t = N_r = 2$, $L = 4$ and block size $N = 10$ over 10,000 runs. As we see, normal RCUA shows the best performance of all and this will justify our choices in developing this algorithm.

Figure 4.16 Performance comparison of variations in RCUA

$N_t = N_r = 2$, $L = 4$ **and** $N = 10$



CHAPTER 5 CONCLUSIONS

We introduced the new SISO updating APP (UA) equalizer for symbol by symbol multiuser detection in frequency selective MIMO channel. One of UA principal advantages is that its computational load is linear in the block or packet length. Also its flexible structure permits any measurements at different times, polarizations, microdivesity antennas, macrodiversity antennas, etc. to be incorporated into a common decision structure.

The basic UA algorithm extracts APPs from the received signals at one time interval and then passes only these probabilities as apPs to the next time interval. Because of the exponential complexity of optimal joint APP extraction, basic UA algorithm is not practical. We then reduced its complexity using the powerful tools of soft SD, soft cancellation and group detection in a hybrid way.

The resulting RCUA achieves performance gain over the traditional sub-optimal MMSE-DFE with comparable order of computational complexity. UA performance is near optimal and comparable to BCJR, SDE and SD in non-overloaded situations. Even in an overloaded situation, its BER performance is acceptable and does not saturate at a floor like SDE. Moreover, the number of computations in RCUA does not increase as much as SD in an overloaded system so RCUA is the most flexible algorithm to variation in the number of users.

UA computational complexity is linear in the number of symbols because of the symbol by symbol process. So unlike SD and SDE, it is able to process long or infinite streams of data. Unlike SDE, UA can also be used in iterative turbo equalization systems. The overall characteristics and flexibility of this algorithm make it worthwhile to be considered in future wireless communication systems.

5.1 Future Research

In this thesis we developed RCUA and evaluated its complexity, performance and characteristics. There is still a lot more to be explored about different aspects of this algorithm. It is a SISO algorithm so its performance can be evaluated in turbo-equalization systems, using different codes and iterating between the decoder and UA equalizer. If we can find codes that can be decoded using RCUA without much loss then we will be able to implement turbo-equalization systems with less complexity and shorter delay.

RCUA is flexible and does not cause much delay relative to most other algorithms in providing soft decisions, so if we use it in a cellular system, it can share information between different base-stations in real-time. This macrodiversity system has the potential to accommodate co-channel users in neighbouring cells or to increase the number of co-channel users in a cell. This can cause a huge increase in capacity without using more bandwidth or increasing the computation load substantially. Investigating such a system can be interesting and fruitful.

Another interesting topic is to develop sliding window techniques for soft SD. Shorter windows will prevent the increased computational burden caused by long block lengths and will decrease the delays. Evaluating the performance and comparing it with UA is one of our suggested areas of future research.

REFERENCE LIST

[1] G.J. Foschini and M.J. Gans, "On the limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Commun.*, no. 6, pp. 315-335, 1998.

[2] L.R. Bahl, J.Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284-287, March 1974.

[3] A. Duel- Hallen and C. Heegard, "Delayed decision feedback sequence estimation,'' *IEEE Trans. Commun.*, vol. 37, no. 5, pp.428-436, May 1989.

[4] M. V. Eyuboglu and S.U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback,'' *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 12-20, Jan. 1988.

[5] C. Fragouli, N. Al-Dhahir, S. N. Diggavi, and W. Turin, " Prefiltered space-time M-BCJR equalizer for frequency selective channels," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 742-752, May 2002

[6] V. Franz and J. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 186-195, Feb. 1998

[7] G. Bauch and N. Al-Dhahir, "Reduced-complexity space-time turbo equalization for frequency selective MIMO channels," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, October 2002

[8] A. Lozano and C. Papadias, "Layered space-time receivers for frequency-selective wireless channels,'' *IEEE Trans. Commun.*, vol. 50, no.1, pp. 65-73, Jan. 2002

[9] N. Al-Dhahir and A. H. Sayed, " The finite-length multi-input multi-output MMSE-DFE," *IEEE Trans. Signal Proc.*, vol. 48, no. 10, pp. 2921-2936, October 2000

[10] M. Tuchler, R. Koetter, and A.C. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.* , vol. 50, no. 5, pp. 754-767, May 2002

[11] 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, April 1985

[12] H. Vikalo and B. Hassibi, "Maximum-likelihood sequence detection of multiple antenna systems over dispersive channels,'' *EURASIP Journal on Applied Signal Processing, Special Issue on Space-Time Coding*, May 2002, pp. 525-531

[13] S. Liu and Z. Tian, "Near-optimum soft decision equalization for frequency selective MIMO channels,'' *IEEE Trans. Signal Proc.*, vol.52, no. 3, pp. 721-733, March 2004

[14] J. Luo, K. R. Pattipati, P. K. Willett, and F. Hasegawa, "Near-optimal multi-user detection in synchronous CDMA using probabilistic data association," *IEEE Commun. Lett.*, vol. 5, no. 9, pp. 361-363, 2001.

[15] H. Vikalo, *Sphere Decoding Algorithm for Digital Communications*, PhD Dissertation, Stanford University , 2003

[16] J. G. Proakis, *Digital Communications*, 4th Ed., McGraw Hill, NY.

[17] S. Haykin, *Adaptive Filters*, 4th Ed., McGraw Hill, NY.

[18] Seung Young Park, Soo Ki choi, Chung Gu Kang, and Ki Ho Kim, "Sphere decoding-based iterative MAP received for spatial Multiplexing," *Vehicular Technology Conference*, vol. 4, pp. 2324-2328, April 2003

[19] B. Zarikoff, Investigation of an Iterative Groupwise Soft Input/Soft Output Multiuser Detection Algorithm, MASc Thesis, Simon Fraser University, June 2004.

[20] A. Furuskar, S. Mazur, F. Muller, and H. Olofsson, "EDGE: enhanced data rates for GSM and TDMA/136 evolution," *IEEE Personal Commun.*, vol. 6, pp. 56-66, June 1999

[21] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. on Info. Theory*, vol. 13, no. 2, pp. 260-269, April 1967

[22] G. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973

[23] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proc. IEEE GLOBECOM*, Dallas, TX, pp. 47.1.1-47.1.7 , 1989

[24] L. Hanzo, T. H. Liew, and B. L. Yeap, Turbo Coding, *Turbo Equalization and Space-Time Coding for Transmission over Fading Channels*, John Wiley & Sons, 2002

[25] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE Jour. of selected areas in commun.*, vol.16, no. 2, pp.260-264, Feb.1998

[26] R.W. Wang and G.B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," *Wireless Communications and Networking Conference WCNC 2004*, vol.3, pp.1620-1625, March 2004

[27] M.L. Moher, "Multiuser decoding for multibeam systems," *IEEE Trans. on Veh. Tech.*, vol. 49, no.4, pp.1226-1233, July 2000

[28] J. Winters, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment," *IEEE J. Select. Areas Commun.*, vol.5, no. 5, pp. 871-878, June 1987

[29] E. Teletar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecomm. ETT*, vol. 10, no. 6, pp. 585-596, Nov. 1999

[30] R.J.C. Bultitude and G.K. Bedal, "Propagation characteristics on microcellular urban mobile radio channels at 910 MHz", *IEEE J Selected Areas Commun*, vol 7, no 1, pp 31-39, Jan. 1989

[31] P. Ho, Course notes for ENSC 832: Mobile Personal Communication, Chapter 7, 2003. Downloaded [November 2004] from *http://www.ensc.sfu.ca/people/faculty/ho/ENSC832/Diversity%20Techniques.pdf*

[32] L. Welburn, Macrodiversity and Multiuser Detection: High Performance Tools in DS-CDMA System, PHD Thesis, Simon Fraser University, 2001

[33] M.C. Valenti and B.D. Woerner, "Iterative Multiuser Detection, Macrodiversity Combining, and Decoding for the TDMA Cellular Uplink," *IEEE Jour. of Selected Areas in Commun.*, vol. 19, no. 8, Aug. 2001

OTHER WORKS CONSULTED

J. K. Cavers, *Mobile Channel Characteristics*, second edition, Shady Island Press, 2003

R. Vaughan and J. B. Andersen, Channels, *Propagation and Antennas for Mobile Communications*, IEE, 2003