

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Iterative Detection, Decoding and Channel Parameter
Estimation for Orthogonally Modulated DS-CDMA Systems

PEI XIAO



CHALMERS

Communication Systems Group
Department of Signals and Systems
School of Electrical Engineering
Chalmers University of Technology

Göteborg 2004

Iterative Detection, Decoding and Channel Parameter Estimation
for Orthogonally Modulated DS-CDMA Systems
PEI XIAO

Copyright ©PEI XIAO, 2004.

ISBN 91-7291-420-3

Doktorsavhandlingar vid
Chalmers tekniska högskola
Ny serie nr 2102
ISSN 0346-718X

Technical Report No. 473.
School of Electrical Engineering,
Chalmers University of Technology.
ISSN 1651-498X

Communication Systems Group,
Department of Signals and Systems,
School of Electrical Engineering,
Chalmers University of Technology.
SE-412 96 Göteborg, Sweden
Telephone + 46 (0)31-772 1000

This thesis has been prepared using L^AT_EX.

Printed by Chalmers Reproservice,
Göteborg, Sweden, January 2004.

In this thesis, we study iterative detection, decoding and channel parameter estimation algorithms for asynchronous direct-sequence code division multiple access (DS-CDMA) systems employing orthogonal signalling formats and long scrambling codes.

Multiuser detection techniques are widely used to combat the detrimental effects of multipath fading and multiple access interference (MAI), which are the major impairments in CDMA communication systems. Although the emphasis is placed on nonlinear interference cancellation schemes, several linear interference filtering techniques are also discussed in the first part of the thesis. The multi-stage parallel interference canceler (PIC) is evaluated analytically and compared with simulation results. To prevent performance degradation of PIC due to error propagation, some soft cancellation schemes using soft decision feedback are presented.

Most multiuser detectors rely on accurate channel information, which needs to be estimated in practice. For the purpose of channel estimation, both classic and Bayesian methods are considered in this thesis, depending on whether prior knowledge about the parameters to be estimated is available or not. We focus on the decision directed approach when estimating the fading channel gains. That is, the receiver estimates the channel parameters based on the detected data, no training sequences are needed. The estimated channel coefficients are also used to regenerate the signal of each user for the purpose of interference cancellation.

Another essential channel parameter to be estimated is the propagation delay. Many studies show that multiuser detectors need very accurate delay estimates to perform well. We propose some suboptimal synchronization algorithms that achieve good acquisition performance in presence of MAI and have reduced complexity compared to the optimum maximum likelihood estimator.

In the second part of the thesis, we employ the turbo processing principle and study iterative demodulation and decoding of a convolutionally coded and orthogonally modulated asynchronous DS-CDMA system. Several iterative schemes based on soft demodulation and decoding algorithms are presented. The performance of different strategies are evaluated and proved to achieve substantial performance gains compared to the conventional hard decision based scheme,

especially when the soft demodulator is assisted by decision directed channel estimation and interference cancellation techniques, and also when demodulation and decoding are performed jointly in an iterative manner.

It is also shown that iterative decoding with properly corrected extrinsic information or with non-extrinsic/extrinsic adaptation enables the system to operate reliably in the presence of severe multiuser interference. Additional gain is noticed when soft information rather than hard decision feedback is used for channel estimation and interference cancellation.

Keywords: direct sequence code division multiple access (DS-CDMA), data detection, multiuser detection, channel estimation, synchronization, acquisition, delay estimation, long scrambling codes, orthogonal modulation, convolutional codes, soft demodulation, iterative decoding, maximum likelihood (ML), maximum a posteriori (MAP), soft-input soft-output (SISO), extrinsic information, additive white Gaussian noise (AWGN), multiple access interference (MAI), intersymbol interference (ISI), multipath Rayleigh-fading channels, performance analysis.

Contents	v
1 Introduction	1
1.1 A CDMA Spread Spectrum Communication System	1
1.1.1 Channel coding	1
1.1.2 Modulation	2
1.1.2.1 M -ary orthogonal modulation	2
1.1.2.2 Baseband modulation	4
1.1.3 Spreading	5
1.1.4 Carrier modulation	7
1.1.5 Channel model	7
1.1.6 Downmixing and despreading	10
1.1.7 Demodulation	12
1.1.8 Channel decoding	12
1.2 Thesis Outline	13
1.3 Contributions	16
I Uncoded System	19
2 System Model for Uncoded System	21
3 Multiuser Detection	27
3.1 Parallel Interference Cancellation (PIC)	28
3.2 Approximate Maximum Likelihood (AML) Detection	29
3.3 Iterative Interference Suppression (IIS)	31
3.4 Whitened Matched Filter (WMF)	32
3.5 Soft Interference Cancellation	33
3.5.1 MAP based soft PIC	34

3.5.2	Nonlinear MMSE based soft PIC	36
3.6	Numerical results	39
3.7	Summary	42
4	Theoretical Evaluation of PIC Performance	45
4.1	Introduction	45
4.2	Theoretical Analysis For Equal Power Diversity Branches	46
4.2.1	Analysis for non-coherent first stage	46
4.2.2	Analysis for multistage PIC	49
4.3	Theoretical Analysis For Unequal Power Diversity Branches	56
4.3.1	Analysis for non-coherent first stage	56
4.3.2	Analysis for multistage PIC	58
4.4	Analytical Results and Performance Comparison	60
4.5	Conclusions	71
5	Estimation of Fading Channels	73
5.1	Introduction	73
5.2	Maximum Likelihood Channel Estimator	75
5.3	Linear MMSE Channel Estimator	76
5.4	First Order Kalman Filter	78
5.5	Second Order Kalman Filter	80
5.6	Soft Channel Estimator	83
5.7	Numerical Results	83
5.8	Conclusions	93
5.9	Appendix: Theoretical Analysis for Channel Estimation	94
5.9.1	Derivation of CRLB for ML channel estimator	94
5.9.2	Derivation of error covariance for LMMSE channel estimator	99
6	Delay Estimation	101
6.1	Introduction	101
6.2	ML Approach to Delay Estimation	104
6.3	Conventional Sliding Correlator	105
6.4	ML Single User Delay Estimator	107
6.5	Whitened Sliding Correlator	109
6.6	Linear MMSE Delay Estimator	111
6.7	Subspace-Based Delay Estimator	111
6.8	AML(1): Successive ML Delay Estimator	115
6.9	AML(2): Parallel ML Delay Estimator	116
6.10	AML(3): Combined SML and PML	117
6.11	Summary	118

7	Pulse Shaping Considerations	121
7.1	Simulation Models	122
7.2	Numerical Results	123
7.3	Summary	131
II	Coded System	135
8	System Model for Coded System	137
9	Soft Demodulation and Decoding	143
9.1	Introduction	144
9.2	Demodulation and Decoding Schemes	145
9.2.1	Scheme 1: partitioned hard demodulation, HIVA decoding	145
9.2.2	Scheme 2: partitioned soft demodulation, SIVA decoding	146
9.2.3	Scheme 3: integrated soft demodulation and SIVA decoding	150
9.2.4	Scheme 4: integrated soft demodulation and Log-MAP decoding	150
9.3	Soft demodulation algorithms	154
9.3.1	Soft demodulation with matched filter	154
9.3.2	Soft demodulation with hard/soft decision interference cancellation	156
9.3.3	Soft demodulation with hard/soft decision interference suppression	158
9.4	Numerical Results	161
9.5	Conclusions	165
10	Iterative Decoding with Extrinsic Information	169
10.1	Introduction	170
10.2	SISO Outer Convolutional Decoder	172
10.3	SISO Inner Block Decoder	175
10.3.1	Conventional single user approach	176
10.3.2	MUD approach	179
10.3.2.1	Hard decision interference cancellation	179
10.3.2.2	Soft decision interference cancellation	180
10.4	Numerical Results	181
10.5	Correction/Adaptation of Extrinsic Information	183
10.6	Conclusions	189
	Appendix	191
	Bibliography	191

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Prof. Erik Ström for his great input to this thesis, which would not have been completed without his patient guidance, continuous inspiration and in-depth expertise in the field of CDMA. I am indebted to him also for his caring attitude, his concern for my academic pursuit and personal development. In addition, his enthusiasm and aptitude for pedagogy make him an excellent teacher, we have learned a lot from him after being his teaching assistants in the undergraduate courses.

There are many other people who have made their impact on my work. I would like address my acknowledgement to Florent Munier for proofreading this thesis; Zihuai Lin and Bartosz Mielczarek for sharing their knowledge on iterative decoding; Anders Persson for our discussions on channel coding, among his help on other matters; Aijun Cao from Ateliertelecom AB for providing me the insight into practical aspects of the CDMA systems; Pål Orten and Prof. Arne Svensson for their contribution to Chapter 7 of this thesis, Pål's previous work has always been an enlightenment for my research; Eva Axelsson and Arne for helping me with various administrative tasks; my teaching fellows Ali Behravan and Florent for the nice cooperation in our undergraduate teaching.

I appreciate very much the IT++ library developed by Prof. Tony Ottosson, former colleague Pål Frenger, and Associate Prof. Thomas Eriksson. It has been a great help for the computer simulations in my work, although it is not complete and needs further development.

Special thanks are due to my old roommate Tommy Svensson and new roommate Mats Rydström. I appreciate their friendship and enjoyed sharing room with them. It was a pleasure to have discussions with Hongxia Zhao, Wei Wang and Ming Chen, on technical and social issues. I also enjoyed working with Associate Prof. Erik Agrell on the IMP evaluation, regret very much for missing the opportunity to do research with him. Other

present and former colleagues have also contributed to the nice working environment in our group: Sorour Falahati, Maxime Flament, Johan Lassing, Fredrik Malmsten, Matts-ola Wessman, Krister Norlund, Aleksander Väljamäe, etc..

In the meantime, I wish to thank my former supervisor Prof. Jarmo Harju from Tampere University of Technology, Finland, for leading me to the road of graduate studies in the field of Telecommunications, for the time and effort he spent on my education.

Finally, I would like to express my gratitude to my wife, my son and my parents. All my fulfillments would not have been possible without the ever-lasting love and support from my family.

The work presented in this thesis has been sponsored by Swedish Research Council (VETENSKAPSRÅDET).

Pei Xiao
Göteborg, January, 2004

AWGN:	Additive White Gaussian Noise
AML:	Approximate Maximum Likelihood
BER:	Bit Error Rate
CE:	Channel Estimation
CMF:	Chip Matched Filter
CRLB:	Cramer-Rao Lower Bound
FEC:	Forward Error Correction
GA:	Gaussian Approximation
HDIC:	Hard Decision Interference Cancellation
HDIS:	Hard Decision Interference Suppression
HIVA:	Hard Input Viterbi Algorithm
IC:	Interference Cancellation
IIS:	Iterative Interference Suppression
IS:	Interference Suppression
ISI:	Intersymbol Interference
LLR:	Log-Likelihood Ratio
MAI:	Multiple Access Interference
MAP:	Maximum A Posteriori
MF:	Matched Filter

MFD:	Maximum Free Distance
ML:	Maximum Likelihood
MMSE:	Minimum Mean Square Error
MRC:	Maximum Ratio Combining
MSE:	Mean Square Error
MUD:	Multiuser Detection
NFR:	Near Far Ratio
NMIC:	Nonlinear MMSE Interference Cancellation
PAM:	Pulse Amplitude Modulation
PIC:	Parallel Interference Cancellation
PML:	Parallel ML delay estimator
PN:	Pseudo Noise
RC:	Raised Cosine
RRC:	Root Raised Cosine
SC:	Sliding Correlator
SDIC:	Soft Decision Interference Cancellation
SDIS:	Soft Decision Interference Suppression
SIC:	Successive Interference Cancellation
SINR:	Signal-to-Interference-plus-Noise Ratio
SISO:	Soft Input, Soft-Output
SIVA:	Soft Input Viterbi Algorithm
SML:	Successive ML delay estimator
SNR:	Signal-to-Noise Ratio
SOVA:	Soft Output Viterbi Algorithm
UMTS:	Universal Mobile Telecommunications System
VA:	Viterbi Algorithm
WMF:	Whitened Matched Filter
WSC:	Whitened Sliding Correlator

Some background information related to this thesis is provided in this chapter. The emphasis is placed on the introduction of the communication system model shown in Fig. 1.1 and some fundamental issues inside each functional block of the system model. At the end of this introductory chapter, the outline of the thesis and the included papers are briefly introduced.

Please note that this introduction is tailored to the system considered in this thesis and is by no mean generic nor all inclusive. For a more detailed and broader overview of digital communication systems in general and CDMA spread spectrum communication systems in particular, the readers are recommended to consult [1, 2, 3, 4, 5, 6, 7, 8].

1.1 A CDMA Spread Spectrum Communication System

The system depicted in Fig. 1.1 consists of three major components: transmitter, channel, and receiver. First, we introduce the basic elements of the transmitter which converts the electrical signal into a form that is suitable for transmission through the physical channel. The functional blocks in the transmitter include channel coding, modulation, spreading and carrier modulation.

1.1.1 Channel coding

The purpose of the channel coding is to introduce in a controlled manner some redundancy in the binary information sequence which can be used at

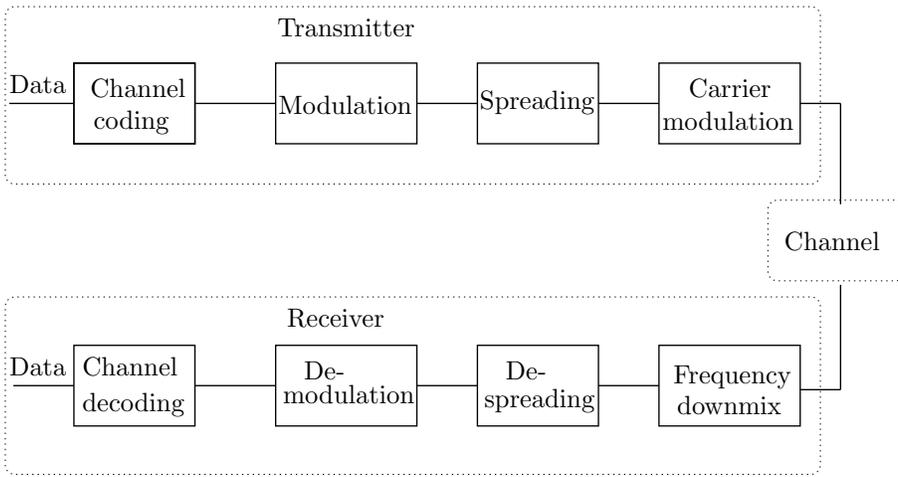


Figure 1.1: A CDMA spread spectrum communication system model.

the receiver to overcome the effects of noise and interference encountered in the transmission of the signal through the channel.

In this thesis, we consider the use of convolutional codes. Convolutional codes are commonly specified by 3 parameters: the number of output bits n , the number of input bits k and the constraint length L . A convolutional encoder with parameters $(n, k, L) = (2, 1, 3)$, and code rate $R_c = \frac{k}{n} = \frac{1}{2}$, is illustrated in Fig. 1.2. It consists of a $L = 3$ -stage shift register that holds the information bits. The shift register stages are connected to modulo-2 adders. The connections are determined by the generator polynomial, which is $(111, 101)$ in this case. The encoder operates on the incoming message sequence, one bit at a time. For each input bit b_1 , the encoder outputs two coded bits $u_1 = b_1 \oplus b_0 \oplus b_{-1}$, $u_2 = b_1 \oplus b_{-1}$. It is obvious that the outputs not only depend on the incoming information bit b_1 , but also on the previous two information bits stored in the shift register (the bits in the shaded stages of the shift register).

The encoder may be regarded as a finite state machine. The final two stages of the shift register hold past inputs and function as the memory of the machine. In this example, there are $L - 1 = 2$ memory stages and hence 2^{L-1} possible states.

1.1.2 Modulation

1.1.2.1 M -ary orthogonal modulation

In this thesis, we are mainly concerned with M -ary orthogonal modulation, which is accomplished by a family of block codes, called Walsh (Hadamard) code, which is used by CDMA users as orthogonal spreading sequences and

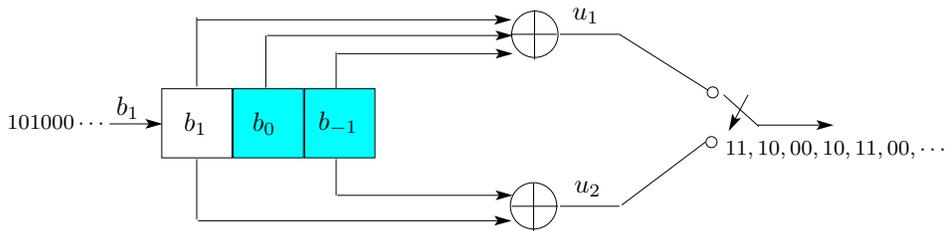


Figure 1.2: A rate 1/2 convolutional encoder with generator polynomial (111, 101).

by the coding community as an error correcting code. The Walsh code is a set of $M = 2^l$ mutually orthogonal vectors. The convolutionally encoded sequences (or information sequences in uncoded case) in binary form are partitioned into groups of length l . There are exactly $M = 2^l$ different patterns of l -bit sequence, and each block of l -bit is mapped into one of the M patterns.

The Walsh codewords are produced in a simple iterative scheme illustrated in Fig. 1.3. The rule is to produce the next matrix of Walsh codes by repeating the entire present matrix to the right and to the bottom and to repeat with all signs reversed to the diagonal. The resulting matrices contain one row of all +1s, and the remaining rows each has equal numbers of +1s and -1s.

The use of Walsh code is widespread in practical CDMA systems. For example, it is used in IS-95 system for orthogonal modulation in the uplink and user separation in the downlink; in 3G systems, it is used for spreading and channelization.

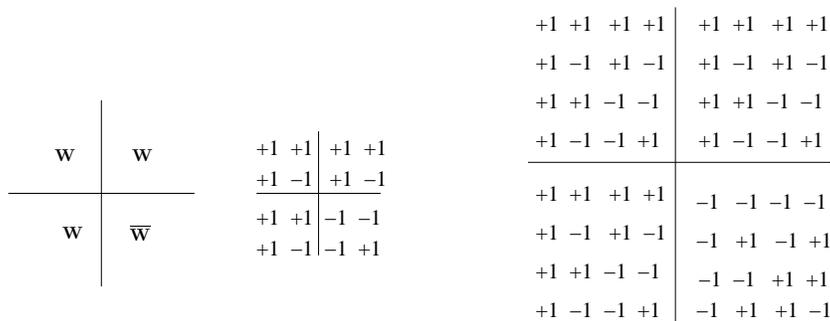


Figure 1.3: Structure of the Walsh codes (overline denotes the binary complement of the bits in the matrix).

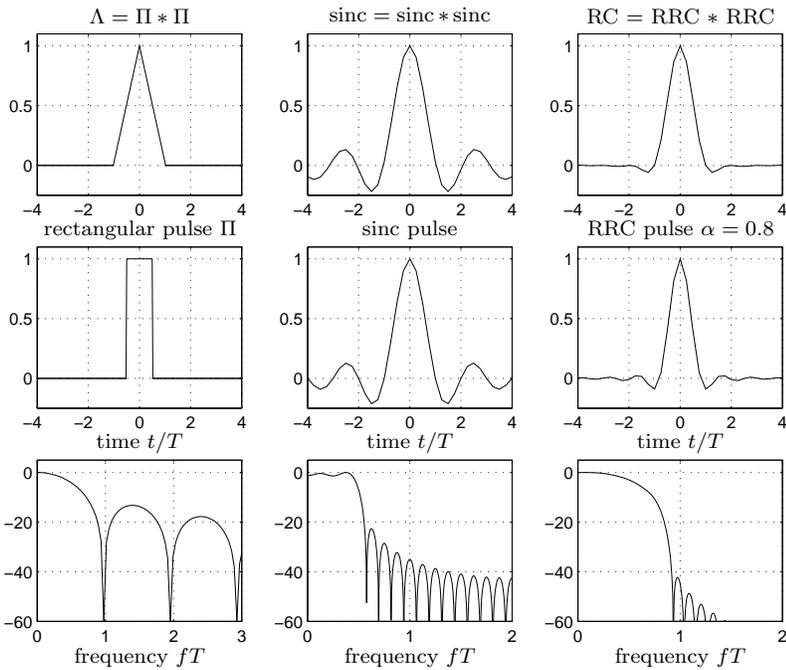


Figure 1.4: Comparison of different pulses and their spectra. The x-axis for pulses is normalized to the symbol interval T , the frequency axis for spectra is normalized to the symbol rate $1/T$, the magnitude of the spectra is plotted in dB scale. All pulses are normalized to a peak value of unity, and truncated between $-4T \sim 4T$.

1.1.2.2 Baseband modulation

The baseband modulation is performed to map the binary information sequence into signal waveforms. The digital data 1s and 0s (or equivalently +1s and -1s) cannot be transmitted through any channel without first transforming the digits to waveforms that are compatible with the channel, e.g., satisfy the bandwidth limitation imposed by the channel. This can be accomplished by representing these information bits or symbols using pulses, and a number of such pulses form a pulse train, which carries the information across the channel. Next, we will examine different pulse shapes that can be used to represent the bits and symbols.

Nyquist pulses that result in no intersymbol interference (ISI) are usually used to represent the information sequence. Examples of Nyquist pulses, e.g., the triangular, sinc, Raised Cosine¹ (RC) pulses are shown in the upper plots of Fig. 1.4. The pulse at transmitter (called transmit-

¹so called because the transition band (the zone between passband and stopband) is shaped like part of a cosine waveform.

ting filter) is employed mainly to constraint the modulation to the regulated bandwidth, while in the receiver, the pulse (called receiving filter) is necessary to remove the interference and minimize the noise entering the demodulator. Usually, a Nyquist pulse is evenly split between the transmitting filter and the receiving filter so that the convolution denoted by '*' of these two is a Nyquist pulse. Therefore, the rectangular, sinc and Root Raised Cosine (RRC) pulses shown by the middle plots of Fig. 1.4 should be ones that are used at the transmitter and receiver². Their corresponding frequency spectra are given in the lower plots of Fig. 1.4.

The rectangular pulse has no intrusion into the next symbol slot, it is an ideal shape for eliminating ISI. Although having the deficiency of infinite bandwidth occupancy, the rectangular pulse is, however, employed in our work to simplify the computer simulations.

Among the sinc and RRC pulses, the latter one is commonly used because of the ease of implementation and better performance for ISI elimination at the cost of some extra bandwidth determined by the rolloff factor α . One of the research topics in this thesis is to study the behavior of data detectors when RRC pulses are employed.

The sinc and RRC pulses are noncausal and have infinite length, they are thereby not physically realizable. To make them realizable, we have to use delayed and truncated versions of these pulses.

To form M -ary orthogonal signaling formats, the Walsh codewords are baseband modulated by Nyquist pulses to construct M orthogonal waveforms $s_i(t), i = 0, 1, \dots, M - 1$. The decimal equivalent of each l -bit sequence denoted by i is then determined and one of the M distinct waveforms corresponding to that decimal number i is transmitted to convey the l -bit pattern to the receiver in each symbol interval. This is in contrast to the binary modulation, in which the binary digit 0 is mapped into a waveform $s_0(t)$ and the the binary digit 1 into a waveform $s_1(t)$.

Fig. 1.5 is an example of 4-ary orthogonal modulation ($l = 2$). In this case, $2^2 = 4$ different binary-bit patterns (00), (01), (10), (11) with representations $i = 0, 1, 2, 3$, are mapped respectively to the orthogonal waveforms corresponding to the Walsh codewords:

$$(+1, +1, +1, +1), (+1, -1, +1, -1), (+1, +1, -1, -1), (+1, -1, -1, +1).$$

1.1.3 Spreading

Spreading and despreading are the main features that distinguish the spread spectrum systems from general digital communication systems. The traditional approach to digital communications is based on the idea of transmitting as much information as possible in as narrow a signal bandwidth as possible. Spread spectrum is a technology used to combat the

²Strictly speaking, $\text{sinc} = \text{sinc} * \text{sinc}$, $\text{RC} = \text{RRC} * \text{RRC}$ only hold when the waveforms are not truncated.

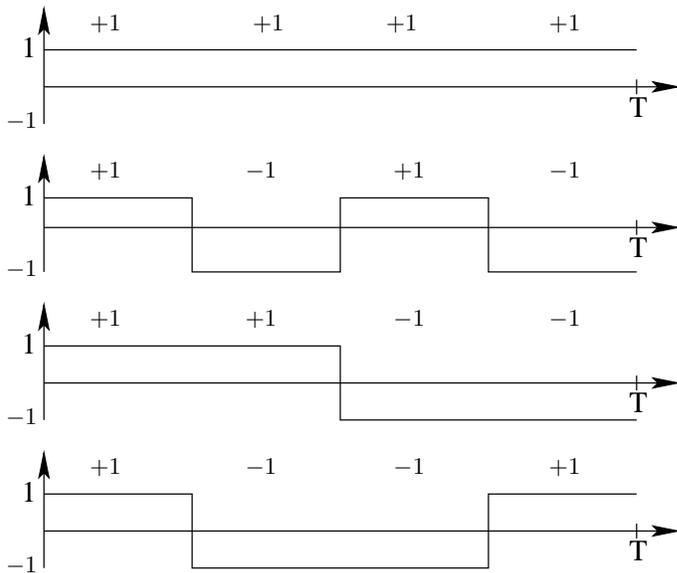


Figure 1.5: Walsh codes of length 4 represented by a rectangular pulse.

MAI and intentional jamming by spreading the transmitted signal over a wide frequency band, much wider than the minimum bandwidth required by the message signal.

In the system discussed in this thesis, all the users use the same orthogonal functions, the signal of one user cannot be distinguished from the others. Therefore, each user's signal needs to be spread by a distinct pseudonoise (PN) sequence after the orthogonal modulation.

Spreading can be used for the purpose of data scrambling. The Walsh codes do not generally have low cross-correlation at arbitrary time shifts, some of them are even shifted versions of the others. Therefore, the PN codes are used to provide low correlation between shifted versions of the transmitted signals in order to facilitate synchronization and reduce inter-symbol interference.

Fig. 1.6 shows the functional diagram and the basic elements of a spread spectrum system. The code generator output $c(t)$ in the transmitter is a chip sequence of rate R_c which is typically much higher than the data rate R_b (the ratio R_c/R_b is called spreading ratio or spreading factor). In short-code CDMA systems, $c(t)$ is periodic with the period equal to the symbol duration. While in long-code CDMA systems, $c(t)$ is different from symbol to symbol, its period is much longer than the symbol duration.

The code synchronizer must precisely synchronize the receiver code generator to the code of the received signal (the required timing accuracy is well below half a chip). The code synchronization is a challenging and one

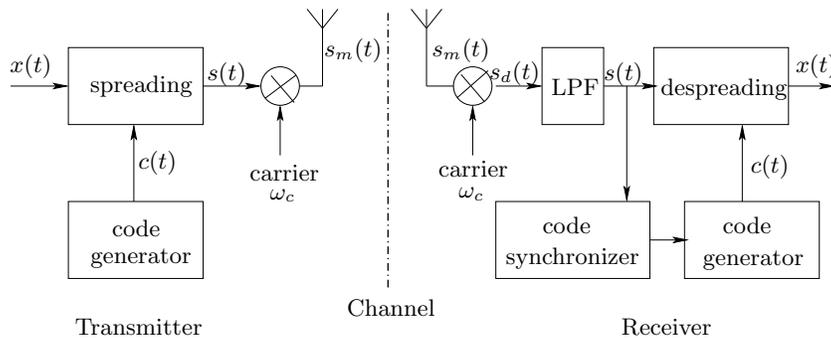


Figure 1.6: A typical spread spectrum system

of the most critical tasks in mobile communications and is also one of the main themes in this thesis.

The spreading process is demonstrated in Fig. 1.7, it consists of multiplying rectangular pulses modulated data $x(t)$ and a PN spreading sequence $c(t)$. The product signal $s(t) = x(t)c(t)$ is the signal after spreading. In our example, we assume information bits (10) are transmitted. After 4-ary orthogonal modulation, the information bearing signal $x(t)$ is therefore the third waveform in Fig. 1.5.

1.1.4 Carrier modulation

Carrier modulation is the process of shifting the frequency components of baseband pulse shapes to a suitable frequency band in order to efficiently pass the signals through the channel. This process usually involves modulating the amplitude, frequency, and/or phase of a carrier.

The nature of the radio transmission necessitates the use of carrier modulation. For example, a 2400 symbols/sec pulse train is an audio-bandwidth signal and is ideal for transmission over a telephone line. But sending the same signal by radio would require an antenna hundreds of kilometers long. On the other hand, if the baseband pulses are first modulated on a higher frequency carrier, e.g., a 900 MHz carrier, the antenna length would be about a few centimeters. For this reason, carrier or bandpass modulation is an essential step for most systems involving radio transmission.

As shown in Fig. 1.7, the carrier modulation is accomplished by multiplying the spread baseband signal $s(t)$ with a sinusoidal carrier of the form $\cos(\omega_c t)$, i.e., $s_m(t) = s(t) \cos(\omega_c t) = x(t)c(t) \cos(\omega_c t)$.

1.1.5 Channel model

The channel is the physical medium used to convey the signal from the transmitter to the receiver. In radio transmission, channel is usually the

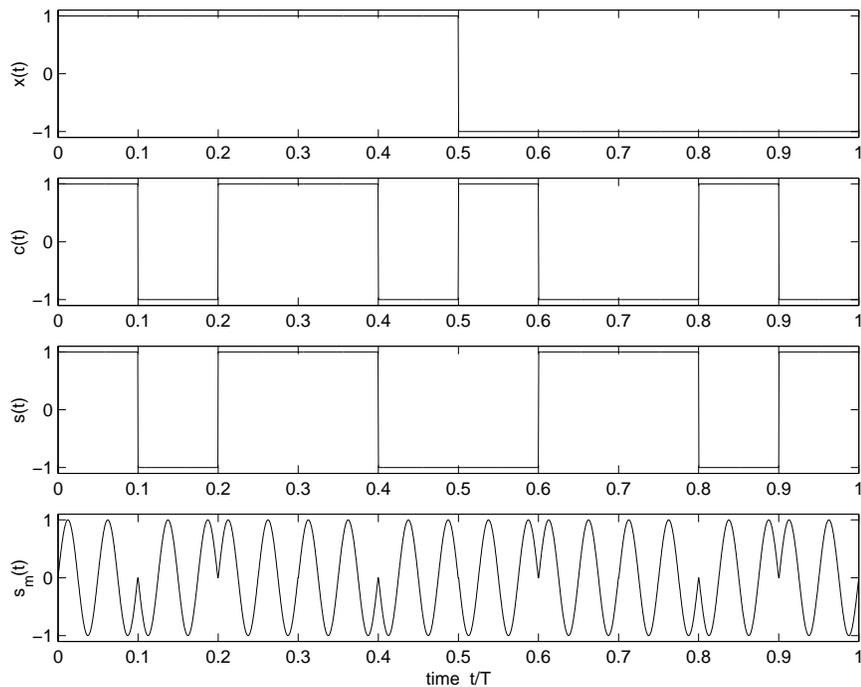


Figure 1.7: Spreading and carrier modulation (T is the symbol interval).

free space. In this thesis, we emphasize on four types of channel effects on the transmitted signal: additive noise, channel propagation delay, fading, and multipath propagation.

Thermal noise is the predominant noise and unavoidable source for all communication systems. Its characteristics (additive, white and Gaussian, giving rise to the name AWGN) are most often used to model the noise in the detection process and in the design of receivers. An AWGN channel is assumed in our work, which means the noise is a Gaussian random process with power spectral density $G_n(f) = N_0/2$ which is flat (constant) over the channel bandwidth.

Signals will experience propagation delays when arriving at the receiver end as long as the transmitter and receiver are not placed at the same location. The delay is directly proportional to the distance between the transmitter and receiver, thus vary with time in wireless communication systems due to the mobility of the mobile users.

The presence of reflecting objects and scatterers in the channel will introduce multipath propagation. When a signal leaves the transmitting antenna, it can take a number of different paths with different delays to reach the receiver, as shown in Fig. 1.8.

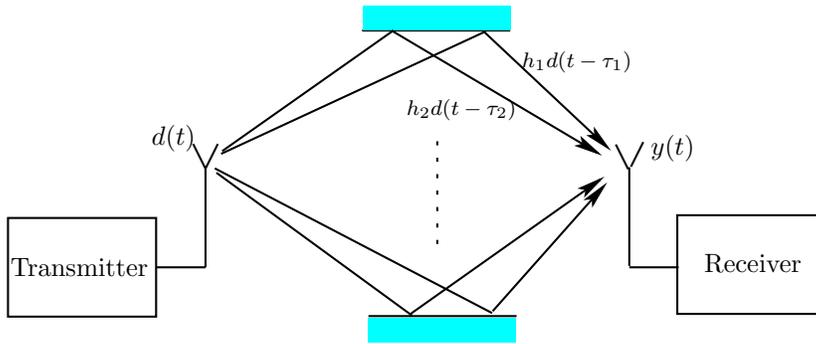


Figure 1.8: Multipath transmission.

The difference in delay will result in different phases between received signals. If the phase difference approaches 180 degree, the signals will add destructively, resulting in a phenomenon called signal fading; while if the phase difference approaches 0, they will add constructively. If the duration of a modulated symbol is much greater than the time spread of the propagation path delays, e.g., in narrow-band transmission systems, the phase variations between different frequency components in the transmitted signal will be small and they will all undergo very similar amount of fading. This is so called flat fading. On the other hand, if the symbol duration is of the same order or even smaller than the multipath delay spread, e.g., in wide-band transmission systems, the frequency components in the transmitted signal will experience different phase shifts along different paths. The channel does not have a constant frequency response over the bandwidth of the transmitted signal, it creates frequency-selective fading on the received signal. When this occurs, the received signal includes multiple version of the transmitted waveform which are attenuated (faded) in time. The channel can thus be modeled with several filter taps which represent attenuation along each of the delay path [9]. Time dispersion of the transmitted symbols results in intersymbol interference (ISI). Let us consider a L -path model illustrated in Fig. 1.8. The received signal $y(t)$ is given by a sum of delayed components

$$y(t) = \sum_{i=1}^L h_i d(t - \tau_i)$$

where $h_i, i = 1, 2, \dots, L$ is the fading coefficient of the i^{th} path, $d(t)$ is the transmitted waveform, and τ_i is the propagation delay of the i^{th} path.

When either the transmitter or receiver is moving within a multipath environment, the path lengths will vary with time, so will the relative phases between signals. The channel gain changes over time and is denoted as

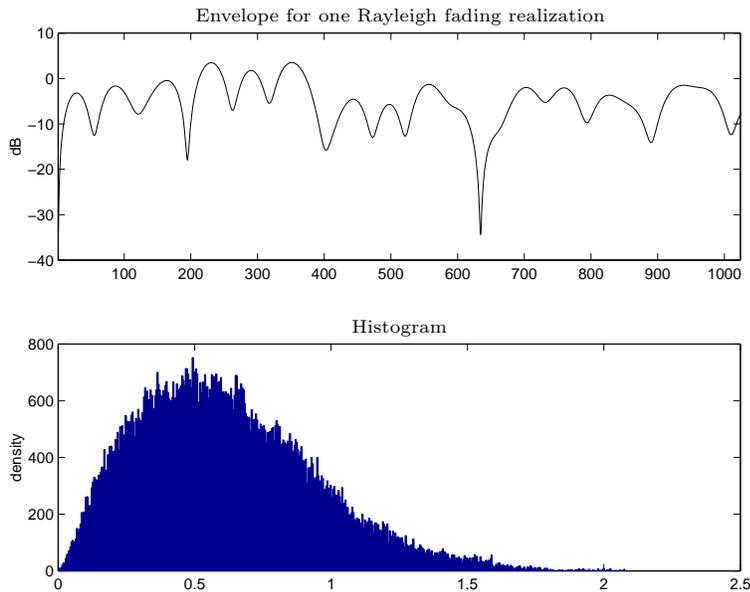


Figure 1.9: Rayleigh fading envelope and histogram, the normalized Doppler frequency is $f_d T = 0.01$.

$h_i(t)$. Without line-of-sight path, the channel gains are usually assumed to be Rayleigh distributed with the classical “bath tub” power spectrum [10], i.e., $h_i(t)$ is complex circular Gaussian process with autocorrelation function $E[h_i^*(t)h_i(t+\tau)] = P_i J_0(2\pi f_d \tau)$ where P_i is the average received power from the i^{th} path, f_d is the maximum Doppler frequency, and $J_0(x)$ is the zeroth order Bessel function of the first kind. A typical fading envelope and histogram are shown in Fig. 1.9.

Next, we shall take a look at the basic elements of the receiver which recovers the message signal contained in the received signal. The functional blocks in the receiver include downmixing, despreading, demodulation, and decoding as depicted in Fig. 1.1. They are more or less logically the inverse operations of the transmitter in reverse order.

1.1.6 Downmixing and despreading

Downmixing is the inverse process of carrier modulation, also called carrier demodulation. It performs frequency translation to transform bandpass signals to baseband signals. This process is illustrated in Fig. 1.10. The

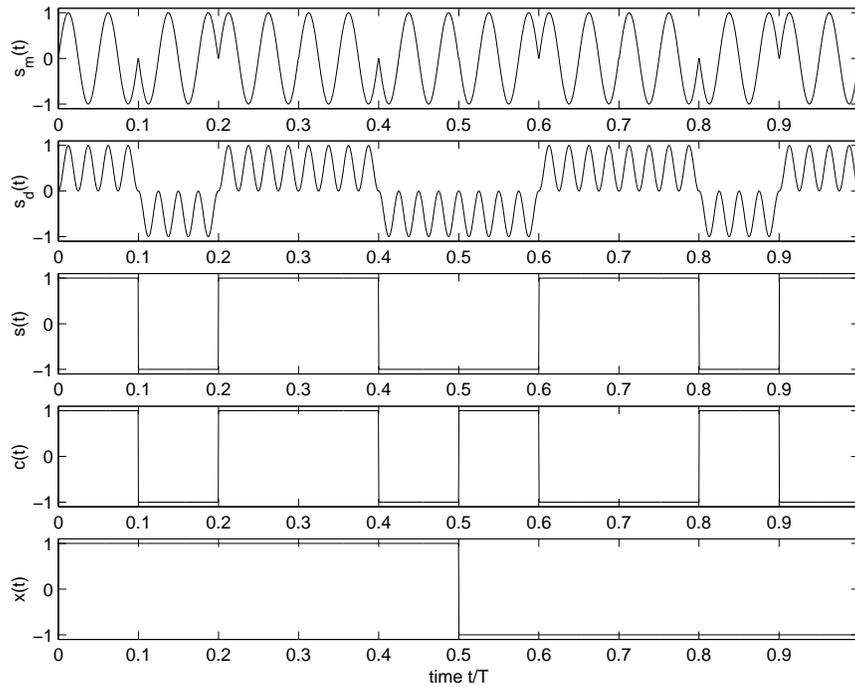


Figure 1.10: Downmixing and despreading (T is the symbol interval)

received signal³ $r(t) = s_m(t)$ is demodulated by a carrier $\cos(\omega_c t)$

$$s_d(t) = s_m(t) \cos(\omega_c t) = x(t)c(t) \cos^2(\omega_c t) = \frac{1}{2}x(t)c(t)[1 + \cos(2\omega_c t)]$$

The double frequency component in the above equation is filtered out by the LPF as shown in Fig. 1.6. In this way, the baseband signal $s(t) = x(t)c(t)$ is extracted from the sinusoidal carrier.

Despreading or descrambling is the process to retrieve the information bearing signal $x(t)$ from the scrambled signal $s(t) = x(t)c(t)$ at the receiver. It is performed as illustrated in Fig. 1.10. The baseband signal $s(t)$ is multiplied with a replica of the spreading signal $c(t)$ generated by the PN code generator at the receiver, which is synchronized to the PN code in the received signal, such that $s(t)c(t) = x(t)c^2(t) = x(t)$, since $c^2(t) = 1$ for all t .

³Strictly speaking, the received signal should have the form:

$r(t) = \sum_i s_m(t - \tau_i)h_i(t) + n(t)$, where τ_i is the propagation delay of the i^{th} path, $h_i(t)$ is its corresponding fading process, and $n(t)$ is the AWGN. In order to get a basic understanding of the principles of the spread spectrum communications, we only consider the ideal situation here and ignore the propagation delays, noise, fading and all other kinds of channel impairments.

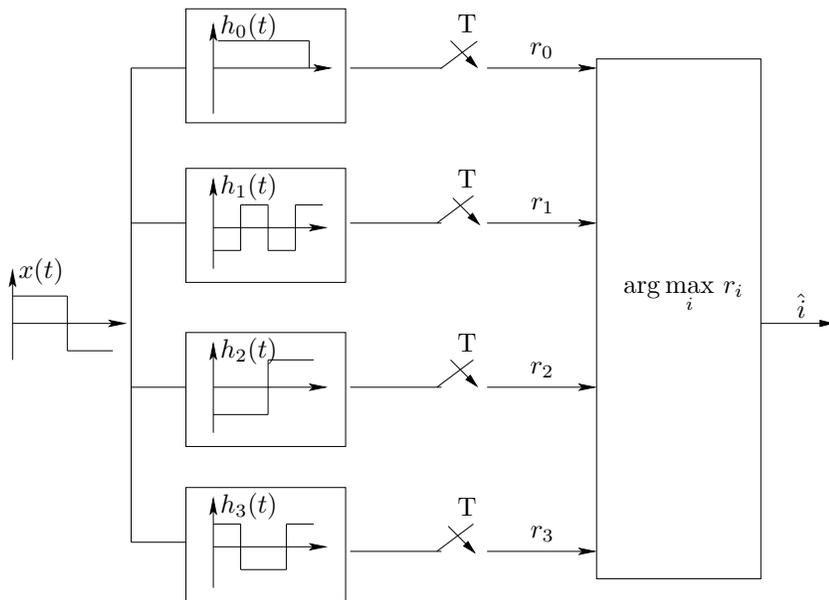


Figure 1.11: Demodulator

1.1.7 Demodulation

The demodulator performs the waveform-to-sample transformation, and makes symbol decision. Here, the waveform refers to the despread signal, in our example is $x(t)$. The demodulator consists of one or a bank of correlators or matched filters (MF) followed by samplers which sample the output of the correlators or MFs at the end of each symbol duration T . The samples are used by the decision device to make decisions on which of the M possible symbols was transmitted.

We use Fig. 1.11 to demonstrate the demodulation process. Suppose we have at the input of the demodulator the signal $x(t)$. It is filtered by a set of M MFs, each matched to the waveform of one of the M symbols ($M = 4$ in this example). The outputs of the MFs are compared and the symbol with the largest output is taken as the transmitted symbol.

In this case, r_2 has larger value than other samples due to the complete match between $x(t)$ and $h_2(t)$. Thus we make the decision that the third symbol ($\hat{i} = 2$) corresponding to the bit pattern (10) was transmitted by the transmitter.

1.1.8 Channel decoding

Viterbi algorithm is the best known implementation of the maximum likelihood decoding for convolutional codes. It can be illustrated with trellis

diagram, which is a way to show the transition between various states as the time evolves. The Viterbi decoder finds a path through the trellis that is at minimum distance from a given sequence. Decoding can be hard-decision decoding or soft-decision decoding, depending on whether Hamming distance or Euclidean distance is used as the metric to be minimized. In hard-decision decoding of convolutional codes, we choose a path through the trellis whose codeword is at minimum Hamming distance from the received sequence; while in soft-decision decoding, we choose a path whose codeword is at minimum Euclidean distance from the received sequence.

Fig. 1.12 shows an example of decoding process. Assume information bits 101000 were convolutionally coded into sequence 11, 10, 00, 10, 11, 00, which was transmitted over the channel. For simplicity, we demonstrate the hard-decision decoding, in which the Hamming distance is used for branch metric. After making hard decision on the received signal, the quantized received sequence becomes 11, 11, 00, 10, 10. The errors occur for the underlined bits due to the channel impairments (noise, fading, etc.). Starting from all-zero state s_0 , the decoder computes metrics (the Hamming distance) between all possible paths and the received sequence. The metrics are cumulative along the nodes of each path. When two branches enter the same node, the one with lower accumulated metric remains as a survivor and the other branch is eliminated. At the end, the path with minimum distance to the received sequence is selected to be the optimum trellis path. The information sequence will then be determined.

In our example, the path drawn with bold lines through the trellis is selected. The codeword for this path is 11, 10, 00, 10, 11, 00, which is at Hamming distance 3 from the received sequence. The corresponding information sequence is 101000. Three bits errors in the codeword are corrected and the original information sequence is recovered.

When the channel propagation delay, MAI, noise, multipath, and fading are considered as in the real systems, the demodulation and decoding processes become much more sophisticated. These issues will be treated in the following chapters.

1.2 Thesis Outline

This thesis includes two parts: the uncoded system and coded system. System model is introduced at the beginning of each part. Part one consists of Chapter 2 – 7, where we consider the orthogonal modulated system without convolutional encoding. Part two consists of Chapter 8 – 10, where we consider the orthogonal modulated system with convolutional encoding. Different linear and nonlinear iterative data detection schemes are presented in Chapter 3. Compared to the conventional receiver and other noncoherent multiuser detectors, coherent multiuser detection schemes achieve much better performance provided that the channels are accurately esti-

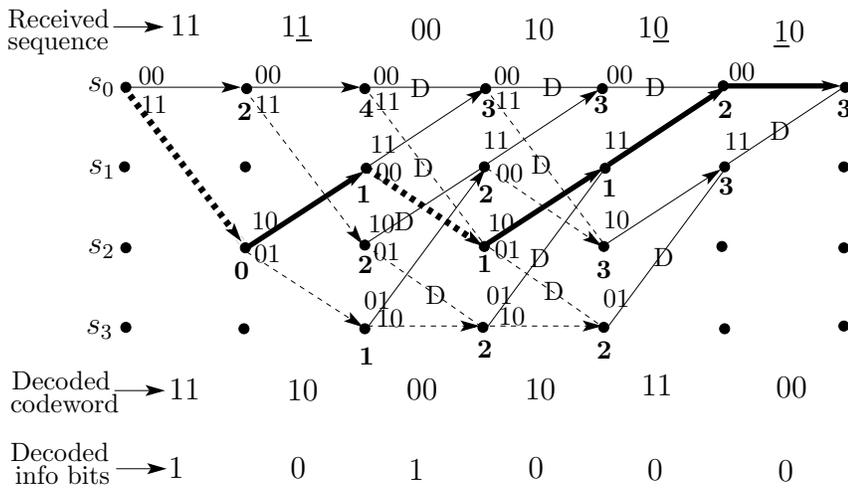


Figure 1.12: Trellis diagram for Viterbi decoding. The solid lines denote state transitions given input (information bit) 0, and the dashed lines denote state transitions given input 1. The corresponding code bits are written underneath each transition line. The branches with letter 'D' are deleted after metric comparison. The number under each node is the accumulated metric value. The path drawn with bold line is the final survivor's path. The last two 0s of the information bits are used to return the encoder to all-zero state, so we only considered the 0 inputs to the encoder in drawing the trellis in the last two stages.

mated. It is proved that PIC is an approximate ML approach to data detection. The performance of multistage PIC is evaluated theoretically in Chapter 4 and shown to be in fairly close agreement with the simulation results. The conventional interference cancellation receiver is subject to performance degradation due to incorrect decisions on interference subtracted from the received signal. To prevent error propagation from the decision feedback, two soft cancellation schemes, one based on maximum a posteriori (MAP) criterion, the other based on nonlinear minimum mean square error (MMSE) estimation are presented and proved to be superior to the conventional PIC using hard decision feedback. It is also shown that the soft information should be used for both interference cancellation and channel estimation in order to achieve the utmost performance.

In Chapter 5, we present a joint approach to channel estimation and data detection (interference cancellation in particular). We investigate channel estimation algorithms under different conditions. The maximum likelihood (ML) algorithm is shown to be efficient for estimating multipath Rayleigh fading channels when the underlying channels are completely unknown and when the fading processes are slow. With the knowledge of the noise and channel statistics, we can employ linear MMSE estimator to

yield more noise-resistant estimates. Other alternatives, like the first and second order Kalman filters which take into account the correlative nature of the Rayleigh fading channel are also introduced. The estimated channel information is used to enable coherent data detection to combat the detrimental effect of multipath propagation of the transmitted signal as well as multiuser interference.

The problem of estimating propagation delays is treated in Chapter 6. The Maximum Likelihood (ML) estimator and its unaffordable complexity for implementation are discussed. Several pilot-assisted acquisition algorithms, namely the whitened sliding correlator, MMSE estimator, subspace-based estimator, and approximate ML estimators are introduced. They reduce the computational complexity of the ML estimation from exponential to polynomial, and achieve good acquisition performance in presence of the MAI. This is in contrast to the conventional sliding correlator which does not work well in high-interference situations.

The effect of different pulse shaping filters on the system performance is analyzed in Chapter 7. We numerically evaluated and compared the performance of rectangular pulse and square root raised cosine pulses with different rolloff factors for conventional receiver and multiuser detectors. Based on the simulation results, some recommendations are made on how to choose RRC pulses in practical systems, e.g., the selection of rolloff factor, truncation length, etc..

Channel coding, iterative decoding and multiuser detection are general tools to design a reliable communication system. The principle of turbo processing seems to be a good way of combining these three powerful elements to achieve the maximum capacity of a system. In the second part of the thesis, we employ the turbo processing principle and study iterative decoding of a convolutionally coded and orthogonally modulated asynchronous CDMA system. In Chapter 9, we analyze and compare the performance and complexity of several strategies for demodulating and decoding without extrinsic information. It is shown that the replacement of the hard demodulator with a soft demodulator is most crucial in the improvement of system performance. Significant gains can also be obtained by integrating demodulation and decoding rather than performing them separately. The replacement of hard output channel decoder and hard decision directed interference cancellation and channel estimation with their soft counterparts further enhance the reliability of the system, however, the gain is less significant.

Orthogonal modulation is essentially a process of block coding with Walsh code. Chapter 10 aims at tackling the problem of iterative decoding of serially concatenated inner block code and outer convolutional code with extrinsic information. The (logarithm) maximum a posteriori probability criterion is used to derive the iterative decoding schemes. The soft output from inner block decoder is used as a priori information for the

outer decoder. The soft output from outer convolutional decoder is used for two purposes. In addition to be fed back to the inner decoder as extrinsic information, tentative hard/soft decisions can be made for interference cancellation and channel estimation. It is shown that the inner decoding with corrected extrinsic feedback or with non-extrinsic/extrinsic adaptation enables the system to operate reliably in the presence of severe multiuser interference.

1.3 Contributions

This monograph is based on the following papers which have been published or submitted to conferences or journals.

- Chapter 3
 - Pei Xiao and Erik Ström. “Multiuser Detection and Channel Estimation Algorithms for M-ary DS-CDMA Systems in Multipath Rayleigh Fading Channels”, In *Proceedings IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'2003*, vol. 2, pp. 1829–1834, Sept. 2003, Beijing, China.
 - Pei Xiao and Erik Ström. “Soft Interference Cancellation and Channel Estimation in Uncoded Asynchronous DS-CDMA System”. To appear in *Proceedings IEEE Semiannual Vehicular Technology Conference, VTC'2004 Spring*.
- Chapter 4
 - Pei Xiao and Erik Ström. “A Theoretical Evaluation of Parallel Interference Cancellation in M-ary Orthogonal Modulated Asynchronous DS-CDMA System over Multipath Rayleigh Fading Channels”. Submitted to *IEEE Transactions on Vehicular Technology*.
 - Pei Xiao and Erik Ström. “BER Performance Analysis of Multistage PIC Scheme in An Asynchronous DS-CDMA System with Long Scrambling Codes”. To appear in *Proceedings IEEE Semiannual Vehicular Technology Conference, VTC'2004 Spring*.
- Chapter 5
 - Pei Xiao and Erik Ström. “Joint Data Detection and Estimation of Time-Varying Multipath Rayleigh fading Channels in Asynchronous DS-CDMA Systems with Long Spreading Sequences”. Submitted to *European Transactions on Telecommunications*.
- Chapter 6

- Pei Xiao and Erik Ström. “Pilot-Aided Acquisition Algorithms for Asynchronous DS-CDMA Systems”. In *European Transactions on Telecommunications*, vol. 14(1), pp. 89–96, 2003.
- Pei Xiao and Erik Ström. “Applications of Maximum Likelihood Algorithm in Asynchronous CDMA Systems”. In *Proceedings IEEE International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’2002*, pp. 208–213, Sept. 2002, Prague, Czech Republic.
- Chapter 7
 - Pei Xiao, Erik Ström, Pål Orten and Arne Svensson. “Pulse shaping considerations in practical DS-CDMA systems”. In *Proceedings IEEE International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS’2002*, Nov. 2002, Kaohsiung, Taiwan.
- Chapter 9
 - Pei Xiao and Erik Ström. “Soft Demodulation and Decoding of Convolutionally Coded and Orthogonally Modulated DS-CDMA Signals in Frequency Selective Rayleigh Fading Channels”. Submitted to *IEEE Transactions on Vehicular Technology*.
 - Pei Xiao and Erik Ström. “Integrated Approach to Demodulation and Decoding of Convolutionally Coded and Orthogonally Modulated DS-CDMA Signals”. *Proceedings The 6th International Symposium on Wireless Personal Multimedia Communications, WPMC’2003*, vol. 1, pp. 122–126, Oct. 2003, Yokosuka, Japan.
- Chapter 10
 - Pei Xiao and Erik Ström. “Correction of Extrinsic Information for Iterative Decoding in Serially Concatenated DS-CDMA Systems”. Submitted to *IEEE Transactions on Wireless Communications*.

The following papers which were included in the previous Licentiate thesis are no longer included in this thesis:

1. Pei Xiao and Erik Ström. “Performance of Iterative DS-CDMA M-ary Demodulation in The Presence of Synchronization Errors”. In *Proceedings IEEE Semiannual Vehicular Technology Conference*, pp. 1703–1707, May, 2001, Rhodes, Greece.

2. Pei Xiao and Erik Ström. “Synchronization Algorithms for Iterative Demodulated M-ary DS-CDMA Systems”. In *Proceedings IEEE Global Telecommunications Conference*, pp. 1371–1375, Nov. 2001, Texas, USA.
3. Pei Xiao and Erik Ström. “Acquisition of Orthogonal Modulated Signals in Rayleigh-Fading Channels”. In *Proceedings European Wireless*, pp. 811–816, Feb. 2002, Florence, Italy.
4. Pei Xiao and Erik Ström. “Delay Estimation and Data Detection in Long-Code DS-CDMA System”. In *Proceedings The 9th Asia-Pacific Conference on Communication*, vol. 2, pp. 584–589, Sept. 2003, Penang, Malaysia.

Part I

Uncoded System

Throughout the first part of this thesis, we study the orthogonally modulated DS-CDMA system without convolutional encoding. The orthogonal modulation is accomplished by Walsh code which combines the advantages of spreading and coding to achieve improved performance for spread spectrum (CDMA) systems. It was shown in [11, 12] that M -ary signaling improves bandwidth efficiency significantly compared to binary signaling in fading and nonfading channels, and the efficiency further improves as the order of multipath diversity increases.

Fig. 2.1 shows the signal path for the k^{th} user. The k^{th} user's j^{th} symbol is denoted by $i_k(j) \in \{0, 1, \dots, M-1\}$, and mapped into $\mathbf{w}_{i_k(j)} \in \{\mathbf{w}_0, \dots, \mathbf{w}_{M-1}\}$, which is one of the by M orthogonal signal alternatives. The Walsh codeword $\mathbf{w}_{i_k(j)} \in \{+1, -1\}^M$ is repetition encoded into

$$\mathbf{s}_k(j) = \text{rep}\{\mathbf{w}_{i_k(j)}, N/\log_2(M)\} \in \{+1, -1\}^N \quad (2.1)$$

where $\text{rep}\{\cdot, \cdot\}$ denotes the repetition encoding operation where its first argument is the input bits and the second one is the repetition factor.

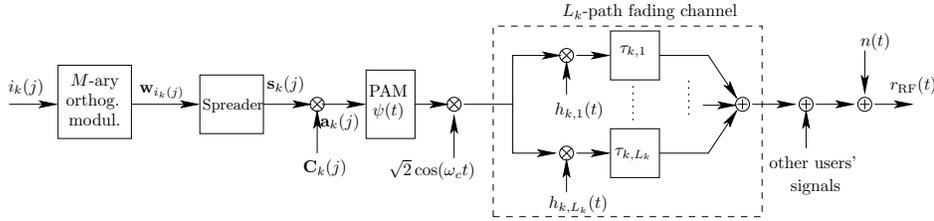


Figure 2.1: Block diagram of the transmitter.

Table 2.1: Mapping between input bits and Walsh codewords. The Walsh chip sequence \mathbf{s}_m is derived by repeating (spreading) each bit of \mathbf{w}_m by $N/8$ times.

info bits	index	Walsh codeword
	m	\mathbf{w}_m
+1 +1 +1	0	$\mathbf{w}_0 : +1 +1 +1 +1 +1 +1 +1 +1$
+1 +1 -1	1	$\mathbf{w}_1 : +1 +1 +1 +1 -1 -1 -1 -1$
+1 -1 +1	2	$\mathbf{w}_2 : +1 +1 -1 -1 +1 +1 -1 -1$
+1 -1 -1	3	$\mathbf{w}_3 : +1 +1 -1 -1 -1 -1 +1 +1$
-1 +1 +1	4	$\mathbf{w}_4 : +1 -1 +1 -1 +1 -1 +1 -1$
-1 +1 -1	5	$\mathbf{w}_5 : +1 -1 +1 -1 -1 +1 -1 +1$
-1 -1 +1	6	$\mathbf{w}_6 : +1 -1 -1 +1 +1 -1 -1 +1$
-1 -1 -1	7	$\mathbf{w}_7 : +1 -1 -1 +1 -1 +1 +1 -1$

Therefore, each bit of the Walsh codeword is spread (repetition coded) into $N_c = N/M$ chips, and each Walsh symbol is represented by N chips and denoted as $\mathbf{s}_k(j) \in \{\mathbf{s}_0, \dots, \mathbf{s}_{M-1}\}$. Table 2.1 shows the mapping between input bits and Walsh codewords \mathbf{w}_m and Walsh sequences \mathbf{s}_m for $M = 8$.

The Walsh sequence $\mathbf{s}_k(j)$ is then scrambled (randomized) by a scrambling code unique to each user to form the transmitted chip sequence

$$\mathbf{a}_k(j) = \mathbf{C}_k(j)\mathbf{s}_k(j) \in \{+1, -1\}^N$$

where $\mathbf{C}_k(j) \in \{-1, 0, +1\}^{N \times N}$ is a diagonal matrix whose diagonal elements correspond to the scrambling code for the k^{th} user's j^{th} symbol. The purpose of scrambling is to separate users. In this thesis, we focus on the use of long codes, e.g., the scrambling code differs from symbol to symbol.

Long spreading (scrambling) codes have been included in several leading standard proposals for 3G cellular networks [13, 14]. The use of long codes ensures that all the users achieve on the average the same performance. They makes MAI more white noise like, thus better combats the interference and other manner of channel impairments [15]. On the other hand, they inevitably destroy the cyclostationarity of CDMA signals and make the system time-varying, which disables applicability of many of the existing channel estimation, synchronization, and detection approaches developed for short code CDMA systems.

The scrambled sequence $\mathbf{a}_k(j)$ is pulse amplitude modulated using a unit-energy chip waveform $\psi(t)$ to form the baseband signal, i.e., $s_k(t) = \sum_n a_k(n)\psi(t - nT_c)$, where T_c is the chip duration and $T = NT_c$ is the symbol duration. For simplicity, we assume that $\psi(t)$ is a rectangular pulse with support $t \in [0, T_c)$ in most parts of this thesis; However, in Chapter 7,

we consider the use of square-root raised cosine pulses and compare their performance with the rectangular pulse.

The baseband signal is multiplied with a carrier and transmitted over a Rayleigh fading channel with L_k resolvable paths with time-varying complex channel gains $h_{k,1}(t), h_{k,2}(t), \dots, h_{k,L_k}(t)$ and delays $\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L_k}$. We assume, without loss of generality, that $\tau_{k,1} < \tau_{k,2} < \dots < \tau_{k,L_k}$. The received signal is the sum of all users' contributions plus additive white Gaussian noise with power spectral density $N_0/2$. The passband signal, $r_{\text{RF}}(t)$ is formed according to Fig. 2.1, and the complex envelope¹ of the received signal can be written as

$$r(t) = n(t) + \sum_{k=1}^K \sum_{l=1}^{L_k} h_{k,l}(t - \tau_{k,l}) s_k(t - \tau_{k,l})$$

where $n(t)$ has the second moments $\text{E}[n(t)n(s)] = 0$ and $\text{E}[n(t)n^*(s)] = N_0\delta(t-s)$, and $\delta(\cdot)$ is the Dirac delta function. The power of $h_{k,l}(t)$ is denoted by $P_{k,l} = \text{E}[|h_{k,l}(t)|^2]$.

The output from the chip-matched filter is denoted by $y(t) = r(t) * \psi(-t)$ and is sampled every T_c seconds to yield

$$\begin{aligned} y(iT_c) &= r(t) * \psi(-t)|_{t=iT_c} \\ &= \nu(iT_c) + \sum_n \sum_{k=1}^K \sum_{l=1}^{L_k} h_{k,l}(t - \tau_{k,l}) a_k(n) \psi(t - nT_c - \tau_{k,l}) * \psi(-t)|_{t=iT_c} \end{aligned}$$

where $\nu(t) = n(t) * \psi(-t)$, the noise sample $\nu(iT_c)$ is a zero-mean complex Gaussian random variable with second moments $\text{E}[\nu(iT_c)\nu^*(jT_c)] = N_0\delta[i-j]$ and $\text{E}[\nu(iT_c)\nu(jT_c)] = 0$, where $\delta[\cdot]$ is the Kronecker delta function.

Let $p_{k,l}$ and $\delta_{k,l} \in [0, 1)$ be the integer and fractional parts of the delay $\tau_{k,l}$, i.e., $\tau_{k,l} = (p_{k,l} + \delta_{k,l})T_c$. The vector $\mathbf{r}(k, j) \in \mathbb{C}^{N_k}$ corresponding to the k^{th} user's j^{th} symbol contains $N_k = N + p_{k,L_k} - p_{k,1}$ samples of $y(iT_c)$ and can be written in the following forms²

$$\begin{aligned} \mathbf{r}(k, j) &= \mathbf{A}(k, j)\mathbf{h}(j) + \mathbf{n}(k, j) \\ &= \mathbf{X}_k(j)\mathbf{h}_k(j) + \text{ISI}(k, j) + \text{MAI}(k, j) + \mathbf{n}(k, j) \end{aligned} \quad (2.2)$$

As seen from Fig. 2.2, $y(iT_c)$ consists of contributions from all users' path signals and the additive noise. The $\mathbf{n}(k, j)$ vector is a vector of the

¹The passband signal, $r_{\text{RF}}(t)$, can be written in terms of the complex envelope as $r_{\text{RF}}(t) = \sqrt{2}\text{Re}\{r(t)e^{j\omega_c t}\}$, where ω_c is the carrier frequency.

²Assume channel gains remain constant during one symbol interval.

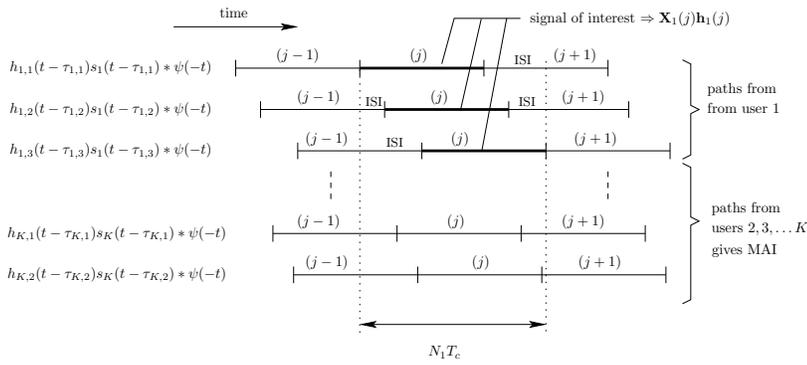


Figure 2.2: Sketch of contributions from the users' paths to the received signal. The signal during the indicated time interval is represented by $\mathbf{r}(1, j)$.

noise samples $\nu(iT_c)$

$$\mathbf{n}(k, j) = \left[\nu([jN + p_{k,1} + 1]T_c) \quad \cdots \quad \nu([jN + p_{k,L_k} + N + 1]T_c) \right]^T$$

Each column of the matrix $\mathbf{A}(k, j)$ represents the contribution from each path and is a shifted version of the appropriate user's chip sequence (the shift is due to the path delay). The columns of $\mathbf{A}(k, j)$ are weighted together by $\mathbf{h}(j)$, whose elements are the path gains of all users' paths. From Fig. 2.2, we see that $\mathbf{r}(k, j)$ can be written as the sum of four terms: the signal of interest, the intersymbol interference (ISI), the multiple access interference (MAI), and the noise. The signal of interest is the part of $y(iT_c)$ that is due to the k^{th} user's j^{th} symbol. In Fig. 2.2, the signal of interest for user 1 is marked with bold lines. The columns of the matrix $\mathbf{X}_k(j)$ are essentially the shifted versions of the chips due to the k^{th} user's j^{th} symbol, one column per path. The columns of $\mathbf{X}_k(j)$ are weighted together by the vector $\mathbf{h}_k(j)$, whose elements are the path gains of the k^{th} user's paths. The contribution only from the k^{th} user's j^{th} symbol can be written as $\mathbf{X}_k(j)\mathbf{h}_k(j)$.

The notations used in this thesis are introduced as follows: \mathbf{I}_r is the $r \times r$ identity matrix, and $\mathbf{0}_r$ is the $r \times 1$ all-zero vector. The vector $\mathbf{1}_r$ denotes all-ones vector $[1 \ 1 \ \cdots \ 1]^T$ of length r . The transpose, conjugate transpose, and 2-norm of a vector \mathbf{x} are denoted by \mathbf{x}^T , \mathbf{x}^* , and $\|\mathbf{x}\| = \sqrt{\mathbf{x}^* \mathbf{x}}$, respectively. The superscript operator $(\)^*$ is the conjugate transpose operation when applied to matrices, and simply the conjugate when applied to scalars. The $(m, n)^{\text{th}}$ element of a matrix \mathbf{A} is denoted by $[\mathbf{A}]_{m,n}$. The n^{th} element of a vector \mathbf{a} is denoted as $[\mathbf{a}]_n$. The symbols \mathbb{R} and \mathbb{C} denote the real field and complex field, respectively.

The matrix $\mathbf{A}(k, j) \in \mathbb{R}^{N_k \times L_{\text{tot}}}$, (L_{tot} is the total number of paths of

all users, i.e., $L_{\text{tot}} = \sum_{k=1}^K L_k$) is defined as

$$\begin{aligned} \mathbf{A}(k, j) &= [\mathbf{A}_1(k, j) \quad \cdots \quad \mathbf{A}_K(k, j)] \\ \mathbf{A}_i(k, j) &= [\mathbf{a}_{i,1}(k, j) \quad \cdots \quad \mathbf{a}_{i,L_k}(k, j)] \\ [\mathbf{a}_{i,l}(k, j)]_n &= (1 - \delta_{i,l})a_i(jN + p_{k,1} + n - p_{i,l}) \\ &\quad + \delta_{i,l}a_i(jN + p_{k,1} + n - p_{i,l} - 1) \end{aligned} \quad (2.3)$$

The channel vector $\mathbf{h}(j) \in \mathbb{C}^{L_{\text{tot}}}$ is defined as

$$\begin{aligned} \mathbf{h}(j) &= [\mathbf{h}_1^T(j) \quad \mathbf{h}_2^T(j) \quad \cdots \quad \mathbf{h}_K^T(j)]^T \\ \mathbf{h}_i(j) &= [h_{i,1}(jT) \quad h_{i,2}(jT) \quad \cdots \quad h_{i,L_i}(jT)]^T \end{aligned} \quad (2.4)$$

From (2.2) and (2.3), we can derive

$$\begin{aligned} \mathbf{r}(k, j) &= [\mathbf{A}_1(k, j) \cdots \mathbf{A}_K(k, j)][\mathbf{h}_1^T(j) \cdots \mathbf{h}_K^T(j)]^T + \mathbf{n}(k, j) \\ &= \mathbf{A}_1(k, j)\mathbf{h}_1^T(j) + \cdots + \mathbf{A}_K(k, j)\mathbf{h}_K^T(j) + \mathbf{n}(k, j) \\ &= \sum_{i=1}^K \mathbf{s}_i(k, j) + \mathbf{n}(k, j) = \mathbf{S}(k, j)\mathbf{1}_{L_{\text{tot}}} + \mathbf{n}(k, j) \\ \mathbf{S}(k, j) &= [\mathbf{A}_1\mathbf{h}_1^T \cdots \mathbf{A}_K\mathbf{h}_K^T] = [\mathbf{S}_1(k, j) \cdots \mathbf{S}_K(k, j)] \in \mathbb{C}^{N_k \times L_{\text{tot}}} \\ \mathbf{S}_i(k, j) &= [\mathbf{a}_{i,1}h_{i,1} \cdots \mathbf{a}_{i,L_i}h_{i,L_i}] = [\mathbf{s}_{i,1}(k, j) \cdots \mathbf{s}_{i,L_i}(k, j)] \in \mathbb{C}^{N_k \times L_i} \end{aligned}$$

where $\mathbf{s}_{i,l}(k, j)$ represents the contribution from the i^{th} user's l^{th} path. It is the product of the channel gain and shifted version of chip sequence transmitted by the i^{th} user. The vector $\mathbf{s}_i(k, j) = \mathbf{S}_i(k, j)\mathbf{1}_{L_i}$ represents the contribution from the i^{th} user's L_i paths to $\mathbf{r}(k, j)$. For the k^{th} user, however, some of the elements in $\mathbf{S}_k(k, j)$ are due to other symbols than the j^{th} symbol, e.g., $\mathbf{S}_k(k, j)$ contains intersymbol interference.

To summarize, the vector $\mathbf{r}(k, j)$ captures the total transmitted energy due to the transmission of the k^{th} user's j^{th} symbol. The contribution to $\mathbf{r}(k, j)$ due to the k^{th} user, including ISI, is $\mathbf{S}_k(k, j)\mathbf{1}_{L_k}$, and the contribution excluding ISI is $\mathbf{X}_k(j)\mathbf{h}_k(j)$.

For simplicity of notation we will sometimes suppress the index k and/or j from $\mathbf{r}(k, j)$, $\mathbf{A}(k, j)$, $\mathbf{n}(j)$, $\mathbf{X}_k(j)$ and $\mathbf{h}_k(j)$, etc., whenever no ambiguity arises.

The task of the demodulator is to detect the information symbols from all users, i.e., detect $i_k(j)$ or equivalently, the transmitted Walsh codewords $\mathbf{w}_{i_k(j)}$ for $k = 1, 2, \dots, K$, $j = 1, 2, \dots, L_b$ (L_b is the block length) given the observation $\mathbf{r}(k, j)$. The decision on the k^{th} user's j^{th} symbol, is found as

$$\hat{m} = \arg \max_{m \in \{0, \dots, M-1\}} z_k(m)$$

where $z_k(m)$ is the decision statistic from the symbol matched filter or multiuser detector, based on the condition that the m^{th} Walsh symbol is transmitted from user k .

The conventional detection technique is to form the soft decision by correlating the received signal with the M possible transmitted waveforms. Without the knowledge of the fading processes, the receiver has to use an equal gain combining scheme instead of an optimum maximum ratio combining (MRC) scheme, and the soft decision is formed in a path-by-path noncoherent manner as

$$z_k(m) = \sum_{l=1}^{L_k} |\mathbf{x}_{k,l,m}^*(j) \mathbf{r}(k, j)|^2 \quad (3.1)$$

where $\mathbf{x}_{k,l,m}(j)$ is the transmitted chip sequence due to the k^{th} user's j^{th} symbol from the l^{th} path based on the hypothesis that the m^{th} Walsh symbol is transmitted, and is formed by \mathbf{s}_m scrambled with $\mathbf{C}_k(j)$ and compensated with the path delay $\tau_{k,l}$. It is the l^{th} column corresponding

to the k^{th} user's l^{th} path in the matrix $\mathbf{X}_{k,m}(j)$, which is defined as

$$\mathbf{X}_{k,m}(j) = \begin{bmatrix} \mathbf{x}_{k,1,m}(j) & \cdots & \mathbf{x}_{k,l,m}(j) & \cdots & \mathbf{x}_{k,L_k,m}(j) \end{bmatrix}$$

This simple scheme is particularly useful in the beginning of the detection process when the estimates of the fading channel are lacking, we must therefore carry out the detection in a noncoherent manner. However, it has poor performance in multiuser environments since it considers multiple access interference (MAI) as additive noise and the knowledge about MAI is not exploited in any way. A CDMA cellular communication system is inherently interference limited. This is due to the difficulty of maintaining orthogonality on the reverse link between code channels used by independent mobile stations, which transmit asynchronously. This form of interference limits the uplink capacity severely. An effective tool to increase the capacity of interference-limited CDMA systems is multiuser detection (MUD), a method of jointly detecting all the users in the system. In the following, we will describe how the soft decisions are formed for different multiuser detectors starting from a nonlinear cancellation scheme.

3.1 Parallel Interference Cancellation (PIC)

Multistage interference cancellation schemes are known to be simple and effective techniques for mitigation of MAI in DS-CDMA systems. They offer key advantages over linear multiuser detectors¹ for practical CDMA systems in their ability to operate with long spreading codes, asynchronous reception, multipath channels, high dimensional modulation schemes etc. [16].

Interference cancellation has been the subject of study in several papers. For instance, parallel and successive interference cancellation were presented in [17, 18]. The interference is estimated and subtracted from the received signal before detection is done. Iterative schemes for demodulating M -ary orthogonal signaling formats in DS-CDMA systems were proposed in [16, 19], using nonlinear MMSE and PIC, respectively.

Here, we mainly consider the PIC scheme used in [19]. The basic principle is that once the transmitted signals are estimated for all the users at the previous iteration, interference can be removed by subtracting the estimated signals of the interfering users from the received signal $\mathbf{r}(k, j)$ to form a new signal vector $\mathbf{r}'(k, j)$ for demodulating the signal transmitted from user k , i.e.,

$$\mathbf{r}'(k, j) = \mathbf{r}(k, j) - \hat{\mathbf{A}}^{(p-1)}(k, j)\hat{\mathbf{h}}^{(p-1)}(j) + \hat{\mathbf{X}}_k^{(p-1)}(j)\hat{\mathbf{h}}_k^{(p-1)}(j)$$

¹A receiver is considered linear if the decision function is a linear function of the received signal.

where $\mathbf{r}'(k, j) \in \mathbb{C}^{N_k}$ denotes the interference canceled version of $\mathbf{r}(k, j)$ after subtracting the contributions from all the other users using decision feedback at the $(p-1)^{th}$ stage. The vector $\hat{\mathbf{A}}^{(p-1)}(k, j)\hat{\mathbf{h}}^{(p-1)}(j)$ represents the estimated contribution from all the users calculated by using the estimated data matrix $\hat{\mathbf{A}}^{(p-1)}(k, j)$ and channel vector $\hat{\mathbf{h}}^{(p-1)}(j)$ estimated at the $(p-1)^{th}$ stage. The vector $\hat{\mathbf{X}}_k^{(p-1)}(j)\hat{\mathbf{h}}_k^{(p-1)}(j)$ is the estimated contribution from all paths of user k . The data matrix $\hat{\mathbf{A}}^{(p-1)}(k, j)$ is obtained by substituting data estimates $\hat{i}_k^{(p-1)}$, $k = 1, 2, \dots, K$ into equation (2.3), and assuming the propagation delays are known. The channel estimate $\hat{\mathbf{h}}^{(p-1)}(j)$ can be obtained with channel estimation algorithms presented in Chapter 5.

In the beginning of the detection process, e.g., at the first iteration stage, the estimates of the fading channel are lacking, we can use noncoherent MF [as defined by (3.1)] to get an initial data estimates $\hat{i}_k^{(1)}$, which can be used for channel estimation and interference cancellation in the following stages. We then iterate for $p = 2, 3, \dots, N_{\text{iter}}$ to estimate data $\hat{i}_k^{(p)}$ which denote the data estimates at the p^{th} iteration. The soft decision at the p^{th} PIC stage is formed as

$$\begin{aligned} z_k^{(p)}(m) &= \text{Re}\{\hat{\mathbf{h}}_k^{*(p-1)}(j)\mathbf{X}_{k,m}^*\mathbf{r}'(k, j)\} \\ &= \text{Re}\{\hat{\mathbf{h}}_k^{*(p-1)}(j)\mathbf{X}_{k,m}^*[\mathbf{r}(k, j) - \hat{\mathbf{A}}^{(p-1)}(k, j)\hat{\mathbf{h}}^{(p-1)}(j) \\ &\quad + \hat{\mathbf{X}}_k^{(p-1)}(j)\hat{\mathbf{h}}_k^{(p-1)}(j)]\} \end{aligned} \quad (3.2)$$

With the channel estimate $\hat{\mathbf{h}}_k^{(p-1)}$, we can combine the hypothesized contributions from all the paths of the same user. The soft metric need not to be computed in a path-by-path fashion like we did for the noncoherent MF demodulator.

Compared to nonlinear cancellation techniques, relatively little work has been done in exploring linear receivers for the system in question. In the remainder of this chapter, we present some linear schemes suitable for demodulating DS-CDMA signals with orthogonal modulation and compare their performance with nonlinear receiver, e.g., PIC. We start with an approximate maximum likelihood receiver, then derive some other interference filtering algorithms.

3.2 Approximate Maximum Likelihood (AML) Detection

In the studied system, the noise is complex Gaussian. Given the received observation $\mathbf{r}(k, j)$, the log-likelihood function of the received vector conditioned on a realization of the fading channels and transmitted data can

be expressed as

$$\text{constant} - \frac{1}{N_0} \|\mathbf{r}(k, j) - \mathbf{A}(k, j)\mathbf{h}(j)\|^2$$

From the above equation, one can see that maximization of this log-likelihood function is equivalent to minimization of the function

$$\|\mathbf{r}(j) - \mathbf{A}(k, j)\mathbf{h}(j)\|^2 \quad (3.3)$$

For simplicity of notation, we will suppress explicit dependence on k, j , and p of the various quantities, whenever no ambiguity arises. In particular, we will use \mathbf{r} , \mathbf{n} , \mathbf{A} , $\mathbf{X}_{k,m}$, \mathbf{h}_k , and z_k to denote $\mathbf{r}(k, j)$, $\mathbf{n}(k, j)$, $\mathbf{A}^{(p-1)}(k, j)$, $\mathbf{X}_{k,m}(k, j)$, $\mathbf{h}_k^{(p-1)}(j)$, and $z_k^{(p)}(m)$, respectively.

It is shown in [19] that the ML estimate of the fading channel vector \mathbf{h} is $\hat{\mathbf{h}} = \mathbf{A}^\dagger \mathbf{r}$, where $\mathbf{A}^\dagger = (\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^*$ denotes the left pseudoinverse of \mathbf{A} (assuming \mathbf{A} has full column rank). Substituting $\hat{\mathbf{h}}$ into (3.3), we derive the ML estimation of the data matrix \mathbf{A} as the minimizer of

$$\begin{aligned} \|\mathbf{r} - \mathbf{A}\mathbf{h}\|^2 \Big|_{\hat{\mathbf{h}}=\mathbf{A}^\dagger \mathbf{r}} &= \|\mathbf{r} - \mathbf{A}\mathbf{A}^\dagger \mathbf{r}\|^2 = \|(\mathbf{I} - \mathbf{A}\mathbf{A}^\dagger)\mathbf{r}\|^2 = \|\mathbf{P}_\mathbf{A}^\perp \mathbf{r}\|^2 \\ &= \text{trace}\{\|\mathbf{P}_\mathbf{A}^\perp \mathbf{r}\|^2\} = \text{trace}\{[\mathbf{P}_\mathbf{A}^\perp \mathbf{r}]^* \mathbf{P}_\mathbf{A}^\perp \mathbf{r}\} \\ &= \text{trace}\{\mathbf{r}^* \mathbf{P}_\mathbf{A}^\perp \mathbf{P}_\mathbf{A}^\perp \mathbf{r}\} = \text{trace}\{\mathbf{r}^* \mathbf{P}_\mathbf{A}^\perp \mathbf{r}\} \\ &= \text{trace}\{\mathbf{P}_\mathbf{A}^\perp \mathbf{r} \mathbf{r}^*\} = \text{trace}\{\mathbf{P}_\mathbf{A}^\perp \check{\mathbf{R}}\} \end{aligned}$$

where $\mathbf{P}_\mathbf{A}^\perp = \mathbf{I}_N - \mathbf{A}\mathbf{A}^\dagger$ is the null space of \mathbf{A} , $\check{\mathbf{R}}$ is the sample auto-correlation matrix defined as $\check{\mathbf{R}} = \mathbf{r} \mathbf{r}^*$. The ML detector can then be expressed as

$$\hat{i}_k^{\text{AML}} = \arg \min_{\mathbf{A}} \text{trace}\{\mathbf{P}_\mathbf{A}^\perp \check{\mathbf{R}}\} \quad (3.4)$$

This algorithm needs the estimate of the whole matrix \mathbf{A} at a time. Under the assumption $N_k < 2N$ for all k , the matrix \mathbf{A} is a function of $i_k(j)$; $i_l(m)$, $l = 1, \dots, k-1, k+1, \dots, K$; $m = j-1, j, j+1$. Therefore, the computational complexity of the ML detector define by (3.4) grows exponentially with the number of users K . A sub-optimum solution that reduces the complexity from exponential to polynomial is the multistage ML detector. We use the conventional detector for the first stage to get initial estimates of the transmitted data. Then the ML algorithm switches to the decision directed mode. To detect the data transmitted by the k^{th} user at the p^{th} stage, we replace \mathbf{A} in equation (3.3) with $\hat{\mathbf{A}}$ defined as

$$\hat{\mathbf{A}} = [\hat{\mathbf{A}}_1 \ \dots \ \mathbf{X}_{k,m} \ \dots \ \hat{\mathbf{A}}_K] \quad (3.5)$$

where the matrices $\hat{\mathbf{A}}_1(j), \hat{\mathbf{A}}_2(j), \dots, \hat{\mathbf{A}}_K(j)$ are estimated MAI at the k^{th}

user's j^{th} symbol interval.

Let us denote $\tilde{\mathbf{A}}$ as the MAI matrix formed as $\tilde{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_1 & \cdots & \hat{\mathbf{A}}_{k-1} & \hat{\mathbf{A}}_{k+1} & \cdots & \hat{\mathbf{A}}_K \end{bmatrix}$. The matrix $\tilde{\mathbf{A}}$ is similar in form to $\hat{\mathbf{A}}$ except that the columns due to the k^{th} user are deleted. Similarly, we delete the estimate of the k^{th} user's channel vector $\hat{\mathbf{h}}_k$ from $\hat{\mathbf{h}}$ and form the channel vector $\tilde{\mathbf{h}}$ corresponding to the MAI: $\tilde{\mathbf{h}} = \begin{bmatrix} \hat{\mathbf{h}}_1^T & \hat{\mathbf{h}}_2^T & \cdots & \hat{\mathbf{h}}_{k-1}^T & \hat{\mathbf{h}}_{k+1}^T & \cdots & \hat{\mathbf{h}}_K^T \end{bmatrix}^T$. The expression of the approximate ML algorithm – equations (3.3) and (3.5) can now be expanded as

$$\begin{aligned} \|\mathbf{r} - \hat{\mathbf{A}}\hat{\mathbf{h}}\|^2 &= [\mathbf{r} - \tilde{\mathbf{A}}\tilde{\mathbf{h}} - \mathbf{X}_{k,m}\hat{\mathbf{h}}_k]^* [\mathbf{r} - \tilde{\mathbf{A}}\tilde{\mathbf{h}} - \mathbf{X}_{k,m}\hat{\mathbf{h}}_k] \\ &= \|\mathbf{r} - \tilde{\mathbf{A}}\tilde{\mathbf{h}}\|^2 - 2\text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* [\mathbf{r} - \tilde{\mathbf{A}}\tilde{\mathbf{h}}]\} \\ &\quad + \|\hat{\mathbf{h}}_k \mathbf{X}_{k,m}\|^2 \end{aligned}$$

The first term of the above expression is irrelevant to the choice of $\mathbf{X}_{k,m}$, and the third term is equal for all $m, m = 1, \dots, M$. Minimization of the above decision function is therefore equivalent to maximization of

$$z_k(m) = \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* [\mathbf{r} - \tilde{\mathbf{A}}\tilde{\mathbf{h}}]\}$$

which is exactly the same form as the PIC algorithm derived in [19]. We can conclude that the PIC is an approximate ML approach to data detection. Compared to the original ML detector, we will lose some performance by performing a suboptimum search for the minimizer of the criterion function in (3.4), since the search may end up in a local minima.

3.3 Iterative Interference Suppression (IIS)

An interference cancellation receiver estimates and subtracts interference from the received vector before detection. Interference suppression, on the other hand, removes the estimated interference from \mathbf{r} by filtering. The justification for using suppression instead of cancellation is that an erroneously estimated symbol would increase interference when using cancellation. However, when using suppression, an erroneously estimated symbol will cause the suppression of a non-existing signal. This will lead to some suppression of the desired signal, but the overall penalty may be less than in the cancellation case.

To construct the suppression filter we need to know (or estimate) the structure of the interference. In the following, we will define two filters, one for the case when the fading is unknown (or it is desirable to avoid channel estimation for complexity or other reasons) and one for the case when the fading is known (or estimated). Consider the matrix $\mathbf{U} \in \mathbb{R}^{N_k \times (L_{\text{tot}} - L_k)}$

defined as

$$\mathbf{U} = \begin{bmatrix} \mathbf{A}_1 & \cdots & \mathbf{A}_{k-1} & \mathbf{A}_{k+1} & \cdots & \mathbf{A}_K \end{bmatrix} \quad (3.6)$$

which is formed from \mathbf{A} by deleting the columns that are due to the k^{th} user. We can suppress the interference by projecting \mathbf{r} on the null space of \mathbf{U} which can be computed as $\mathbf{P}_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\dagger}$, where $\mathbf{U}^{\dagger} = (\mathbf{U}^*\mathbf{U})^{-1}\mathbf{U}^*$ denotes the left pseudoinverse of \mathbf{U} (assuming \mathbf{U} has full column rank). This implies that $\mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{A}_i = \mathbf{0}$ for all $i \neq k$, and thus

$$\mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{r} = \mathbf{P}_{\mathbf{U}}^{\perp}[\mathbf{A}\mathbf{h} + \mathbf{n}] = \sum_{i=1}^k \mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{A}_i\mathbf{h}_i + \mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{n} = \mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{A}_k\mathbf{h}_k + \mathbf{P}_{\mathbf{U}}^{\perp}\mathbf{n}$$

We can do interference suppression without knowing or estimating the fading. That leads to the noncoherent version of the IIS receiver

$$z_k(m) = \sum_{l=1}^{L_k} \left(|\mathbf{x}_{k,l,m}^* \mathbf{P}_{\hat{\mathbf{U}}}^{\perp} \mathbf{r}| - \frac{1}{2} \|\mathbf{P}_{\hat{\mathbf{U}}}^{\perp} \mathbf{x}_{k,l,m}\|^2 \right) \quad (3.7)$$

where $\hat{\mathbf{U}}$ is estimated interference matrix at the previous iteration stage.

If an estimate of the fading vector \mathbf{h} is available, the MAI matrix \mathbf{U} can be formed in different ways as follows

$$\mathbf{U} = [\mathbf{s}_1 \cdots \mathbf{s}_{k-1} \mathbf{s}_{k+1} \cdots \mathbf{s}_K] \in \mathbb{C}^{N_k \times (K-1)} \quad (3.8)$$

$$\mathbf{U} = [\mathbf{S}_1 \cdots \mathbf{S}_{k-1} \mathbf{S}_{k+1} \cdots \mathbf{S}_K] \in \mathbb{C}^{N_k \times (L_{\text{tot}} - L_k)} \quad (3.9)$$

$$\mathbf{U} = [\mathbf{s}_1 \cdots \mathbf{s}_{k-1} (\mathbf{s}_k - \mathbf{X}_{k,i_k} \mathbf{h}_k) \mathbf{s}_{k+1} \cdots \mathbf{s}_K] \in \mathbb{C}^{N_k \times K} \quad (3.10)$$

where $\mathbf{s}_i = \mathbf{S}_i \mathbf{1}_{L_i}$ is the contribution from the i^{th} user. A coherent version of the IIS receiver can now be formulated as

$$z_k(m) = \text{Re} \left\{ \hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{P}_{\hat{\mathbf{U}}}^{\perp} \mathbf{r} \right\} - \frac{1}{2} \left\| \mathbf{P}_{\hat{\mathbf{U}}}^{\perp} \mathbf{X}_{k,m} \hat{\mathbf{h}}_k \right\|^2 \quad (3.11)$$

Among the different ways of constructing suppression filter, we found out from experiments that the one constructed from the null space of (3.9) gives the best performance, and is therefore employed in our simulations.

3.4 Whitened Matched Filter (WMF)

As we know, the conventional matched filter achieves the best performance in the AWGN single user channel or in strict orthogonal synchronous channel. It is not a good choice for multiuser detection in which interference must be taken into account in addition to the white Gaussian noise. Interference combined with Gaussian noise does not have a Gaussian distribution. A way to work around this problem is to whiten the combined

interference and noise, which can be achieved by preprocessing the received vector \mathbf{r} with the matrix $\mathbf{R}^{-1/2}$, where \mathbf{R} is the auto-correlation matrix of \mathbf{r} defined as $\mathbf{R} = \text{E}[\mathbf{r}\mathbf{r}^*]$, and can be estimated as

$$\hat{\mathbf{R}} = \hat{\mathbf{A}}\mathbf{h}\mathbf{h}^*\hat{\mathbf{A}}^* + N_0\mathbf{I}_N$$

The whitened vector ${}^2\mathbf{R}^{-1/2}\mathbf{r}$ has a “white” correlation matrix

$$\text{E}[(\mathbf{R}^{-1/2}\mathbf{r})(\mathbf{R}^{-1/2}\mathbf{r})^*] = \mathbf{R}^{-1/2}\mathbf{R}\mathbf{R}^{-*/2} = \mathbf{I}$$

The matrix $\mathbf{R}^{-1/2}$ is obtained from \mathbf{R} by Cholesky factorization [20].

This idea leads to another kind of linear interference suppression technique, namely, the whitened matched filter (WMF). A coherent WMF can be formed by correlating the whitened received vector with each candidates vector $\{\hat{\mathbf{R}}^{-1/2}\mathbf{X}_{k,m}\hat{\mathbf{h}}_k\}$

$$\begin{aligned} z_k(m) &= \text{Re} \left\{ (\hat{\mathbf{R}}^{-1/2}\mathbf{X}_{k,m}\hat{\mathbf{h}}_k)^*\hat{\mathbf{R}}^{-1/2}\mathbf{r} \right\} - \frac{1}{2}\|\hat{\mathbf{R}}^{-1/2}\mathbf{X}_{k,m}\hat{\mathbf{h}}_k\|^2 \\ &= \text{Re} \left\{ \hat{\mathbf{h}}_k^*\mathbf{X}_{k,m}^*\hat{\mathbf{R}}^{-1}\mathbf{r} \right\} - \frac{1}{2}\hat{\mathbf{h}}_k^*\mathbf{X}_{k,m}^*\hat{\mathbf{R}}^{-1}\mathbf{X}_{k,m}\hat{\mathbf{h}}_k \end{aligned} \quad (3.12)$$

From above we know that all coherent data detectors require estimates of the fading processes, i.e., an estimate of \mathbf{h} . Several channel estimation algorithms are presented in Chapter 5. All of them are decision-directed and can be inserted into the coherent iteration loops presented in this chapter. The estimation procedure at the p^{th} iteration uses the data estimates from the previous stage, i.e., $\hat{i}_k^{(p-1)}$.

3.5 Soft Interference Cancellation

The conventional interference cancellation receiver is subject to performance degradation due to incorrect decisions on interference that are subtracted from the received signal. To prevent error propagation from the decision feedback, soft interference cancellation was proposed, e.g., in [21, 22] for convolutionally coded systems. In this case, the soft information is readily available from the soft-output channel decoder. For uncoded systems, the soft information has to be derived by some other means. Some soft interference cancellation algorithms for the uncoded asynchronous DS-SS system with orthogonal signaling formats are presented in this chapter. We then further extend the use of derived soft information for channel estimation in Chapter 5. The rationale is that the hard cancellation and channel estimation tends to increase the interference and propagate errors

²The matrix \mathbf{R} is symmetric positive definite as long as $N_0 > 0$. Hence, there exist a symmetric positive definite matrix \mathbf{Q} such that $\mathbf{Q}\mathbf{Q} = \mathbf{R}$, and we will use the notation $\mathbf{Q} = \mathbf{R}^{1/2}$, $\mathbf{Q}^{-1} = \mathbf{R}^{-1/2}$, $\mathbf{Q}^* = \mathbf{R}^{*/2}$, and $(\mathbf{Q}^{-1})^* = \mathbf{R}^{-*/2}$.

with incorrect decision feedback; while with soft cancellation and channel estimation, the soft estimate of an erroneously estimated symbol usually has small value, and does not make much contribution to the feedback, therefore error propagation is avoided.

3.5.1 MAP based soft PIC

The soft reliability value for the n^{th} bit of the Walsh codeword, $\mathbf{w}_k^n(j)$, $n = 0, 1, \dots, \log_2(M) - 1$ can be directly derived from the received vector $\mathbf{r}(k, j)$. In the following discussion, we use $M = 8$ as an example. From Table 2.1, we know that bits +1 and -1, are equally probable, i.e., $P(\mathbf{w}_k^n(j) = +1) = P(\mathbf{w}_k^n(j) = -1)$, for $n = 1, \dots, 7$. A posteriori log-likelihood ratio (LLR) for a transmitted +1 and a transmitted -1 in the bit sequence $\{\mathbf{w}_k^n(j)\}$ is defined as [23]

$$\begin{aligned} \lambda(\mathbf{w}_k^n(j)) &= \ln \frac{f(\mathbf{w}_k^n(j) = +1|\mathbf{r})}{f(\mathbf{w}_k^n(j) = -1|\mathbf{r})} = \ln \frac{f(\mathbf{r}|\mathbf{w}_k^n(j) = +1)P(\mathbf{w}_k^n(j) = +1)}{f(\mathbf{r}|\mathbf{w}_k^n(j) = -1)P(\mathbf{w}_k^n(j) = -1)} \\ &= \ln \frac{f(\mathbf{r}|\mathbf{w}_k^n(j) = +1)}{f(\mathbf{r}|\mathbf{w}_k^n(j) = -1)} = \ln \frac{\sum_{m:\mathbf{w}_k^n(j)=+1} f(\mathbf{r}|\mathbf{w}_m)}{\sum_{m:\mathbf{w}_k^n(j)=-1} f(\mathbf{r}|\mathbf{w}_m)} \end{aligned}$$

In the above equation, we denote $m : \mathbf{w}_k^n(j) = \pm 1$ as the set of Walsh codewords $\{\mathbf{w}_m\}$ that correspond to the code bit $\mathbf{w}_k^n(j) = \pm 1$. Typically, one term will dominate each sum, which suggests

$$\lambda(\mathbf{w}_k^n(j)) \approx \ln \frac{\max_{m:\mathbf{w}_k^n(j)=+1} f(\mathbf{r}|\mathbf{w}_m)}{\max_{m:\mathbf{w}_k^n(j)=-1} f(\mathbf{r}|\mathbf{w}_m)}$$

In the above equation, \mathbf{r} can be replaced by its interference canceled version \mathbf{r}' for better performance. In case of perfect cancellation, \mathbf{r}' only contains the contribution from the k^{th} user plus original additive Gaussian noise $\mathbf{n} \in \mathbb{C}^{N_k}$ with PDF $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I}_{N_k})$, i.e., $\mathbf{r}' = \mathbf{X}_k \mathbf{h}_k + \mathbf{n}$. Therefore,

$$\begin{aligned} f(\mathbf{r}'|\mathbf{w}_m) &= \frac{1}{(\pi N_0)^{N_k}} \exp\left(-\frac{\|\mathbf{r}' - \mathbf{X}_{k,m} \mathbf{h}_k\|^2}{N_0}\right) \\ \lambda(\mathbf{w}_k^n) &\approx \ln \frac{\max_{m:\mathbf{w}_k^n=+1} f(\mathbf{r}'|\mathbf{w}_m)}{\max_{m:\mathbf{w}_k^n=-1} f(\mathbf{r}'|\mathbf{w}_m)} = \ln \frac{\exp(-\|\mathbf{r}' - \mathbf{X}^+ \mathbf{h}_k\|^2/N_0)}{\exp(-\|\mathbf{r}' - \mathbf{X}^- \mathbf{h}_k\|^2/N_0)} \\ &= \frac{1}{N_0} \{ \|\mathbf{r}' - \mathbf{X}^- \mathbf{h}_k\|^2 - \|\mathbf{r}' - \mathbf{X}^+ \mathbf{h}_k\|^2 \} \\ &= \frac{2}{N_0} \text{Re} \{ \mathbf{h}_k^* \mathbf{X}^+ \mathbf{r}' - \mathbf{h}_k^* \mathbf{X}^- \mathbf{r}' \} \end{aligned} \quad (3.13)$$

where \mathbf{X}^+ denotes the $\mathbf{X}_{k,m}$ that corresponds to $\max_{m:\mathbf{w}_k^n(j)=+1} f(\mathbf{r}|\mathbf{s}_m)$,

and \mathbf{X}^- is defined similarly. Comparing (3.13) with (3.2), one can see that the added complexity by deriving soft values rather than making hard decisions is minor, also that the soft value for $\mathbf{w}_k^n(j)$ can be expressed as

$$\lambda(\mathbf{w}_k^n(j)) \approx \frac{2}{N_0} \left[\max_{m:\mathbf{w}_k^n(j)=+1} \{z_k(m)\} - \max_{m:\mathbf{w}_k^n(j)=-1} \{z_k(m)\} \right]$$

where $z_k(m) = \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{r}'\}$ is the sof decision derived by the PIC algorithm (see (3.2) in Section 3.1) based on the hypothesis that the m^{th} Walsh symbol is transmitted from user k .

The original \mathbf{h}_k is unknown, and it has to be estimated. In (3.13), we should replace it with its estimate $\hat{\mathbf{h}}_k$ instead. An estimate of the channel vector $\hat{\mathbf{h}}_k$ can be obtained using the channel estimation algorithms described in Chapter 5. The constant $2/N_0$ can be omitted since it is just a scaling factor for all LLR values, therefore, does not have any effect on the decision.

The Walsh codewords are listed in Table 2.1. The first bit is always +1, therefore, its LLR value $\lambda(\mathbf{w}_k^0(j)) = \infty$. For other bits, we know from Table 2.1 that $\mathbf{w}_k^1(j) = +1$ holds for $m = 0, 1, 2, 3$ and $\mathbf{w}_k^1(j) = -1$ holds for $m = 4, 5, 6, 7$. Therefore, the soft metric for the second bit of the Walsh codeword can be computed as

$$\lambda(\mathbf{w}_k^1(j)) \approx \max\{z_k(0), z_k(1), z_k(2), z_k(3)\} - \max\{z_k(4), z_k(5), z_k(6), z_k(7)\}$$

Similarly,

$$\lambda(\mathbf{w}_k^2(j)) \approx \max\{z_k(0), z_k(1), z_k(4), z_k(5)\} - \max\{z_k(2), z_k(3), z_k(6), z_k(7)\}$$

$$\lambda(\mathbf{w}_k^3(j)) \approx \max\{z_k(0), z_k(1), z_k(6), z_k(7)\} - \max\{z_k(2), z_k(3), z_k(4), z_k(5)\}$$

$$\lambda(\mathbf{w}_k^4(j)) \approx \max\{z_k(0), z_k(2), z_k(4), z_k(6)\} - \max\{z_k(1), z_k(3), z_k(5), z_k(7)\}$$

$$\lambda(\mathbf{w}_k^5(j)) \approx \max\{z_k(0), z_k(2), z_k(5), z_k(7)\} - \max\{z_k(1), z_k(3), z_k(4), z_k(6)\}$$

$$\lambda(\mathbf{w}_k^6(j)) \approx \max\{z_k(0), z_k(3), z_k(4), z_k(7)\} - \max\{z_k(1), z_k(2), z_k(5), z_k(6)\}$$

$$\lambda(\mathbf{w}_k^7(j)) \approx \max\{z_k(0), z_k(3), z_k(5), z_k(6)\} - \max\{z_k(1), z_k(2), z_k(4), z_k(7)\}$$

Once the LLR value is derived, the soft estimate (expected value given the received observation) for each bit of the Walsh codeword can be computed by

$$\begin{aligned} \text{E}[\mathbf{w}_k^n(j)|\mathbf{r}] &= (+1) \times P\{\mathbf{w}_k^n(j) = +1|\mathbf{r}\} + (-1) \times P\{\mathbf{w}_k^n(j) = -1|\mathbf{r}\} \\ &= (+1) \frac{e^{\lambda(\mathbf{w}_k^n(j))}}{1 + e^{\lambda(\mathbf{w}_k^n(j))}} + (-1) \frac{e^{-\lambda(\mathbf{w}_k^n(j))}}{1 + e^{-\lambda(\mathbf{w}_k^n(j))}} \\ &= \tanh\{\lambda(\mathbf{w}_k^n(j))/2\} \end{aligned} \quad (3.14)$$

Using equation (2.1), the soft estimate $\text{E}[\mathbf{s}_k^q|\mathbf{r}]$ for each Walsh chip

$\mathbf{s}_k^q, q = 1, \dots, N$ is derived by spreading (repetition encoding) the soft bit of Walsh codeword $E[\mathbf{w}_k^n(j)|\mathbf{r}], n = 1, \dots, M$. The repetition factor is N/M .

When $E(\mathbf{s}_k|\mathbf{r})$, the soft estimate of the transmitted sequence is available, we can carry out soft IC. The new vector after soft cancellation and decision statistic with MAP-PIC can be expressed as

$$\begin{aligned} \mathbf{r}' &= \mathbf{r} - E[\mathbf{y}|\mathbf{r}] + E[\mathbf{X}_k|\mathbf{r}]\hat{\mathbf{h}}_k \\ z_k^{\text{MAP}}(m) &= \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{r}'\} \end{aligned} \quad (3.15)$$

where $E[\mathbf{y}|\mathbf{r}] = E[\mathbf{A}|\mathbf{r}]\hat{\mathbf{h}}$, and the columns of $E[\mathbf{A}|\mathbf{r}]$, $E[\mathbf{X}_k|\mathbf{r}]$ are derived by scrambling $E[\mathbf{s}_k|\mathbf{r}]$ with \mathbf{C}_k and compensating with path delays.

3.5.2 Nonlinear MMSE based soft PIC

Nonlinear MMSE interference cancellation (NMIC) in DS-CDMA systems with BPSK and M -ary orthogonal modulation was proposed in [16], a multistage MMSE linear interference canceler that minimizes the power of residual cancellation error for each user was proposed to mitigate the effect of incorrect cancellation. The algorithm was originally derived for a single path asynchronous channel. Here, we extend its application to the multipath environments, and derive a logarithmic version of the NMIC algorithm which is suitable for practical implementation.

The composite received signal vector can be modeled as the sum of signals from different users, i.e.,

$$\mathbf{r} = \mathbf{r}_1 + \mathbf{r}_2 + \dots + \mathbf{r}_K + \mathbf{n}$$

Minimizing the residual error for the k^{th} user is equivalent to minimizing $E\{\|\mathbf{r}_k - \hat{\mathbf{r}}_k(\mathbf{y}_k)\|^2\}$, where \mathbf{y}_k is the matched filter (coherent version in our case) output in vector form for the k^{th} user. The solution to this nonlinear MMSE estimation problem is the conditional mean estimate

$$\begin{aligned} \hat{\mathbf{r}}_k &= E\{\mathbf{r}_k|\mathbf{y}_k\} = \sum_{m=1}^M b_{k,m} \mathbf{X}_{k,m} \mathbf{h}_k \\ b_{k,m} &= P[i_k(j) = m|\mathbf{y}_k] \end{aligned}$$

From the above equation, one can see that the MMSE estimate of \mathbf{r}_k is a weighted sum of all the M orthogonal signals. The interference canceled vector is obtained as

$$\mathbf{r}'_k = \mathbf{r} - \sum_{i \neq k} \hat{\mathbf{r}}_i = \mathbf{X}_k \mathbf{h}_k + \mathbf{r}_c + \mathbf{n} = \mathbf{X}_k \mathbf{h}_k + \mathbf{w}$$

where \mathbf{r}_c stands for cancellation residual, and the vector \mathbf{w} is defined as

$\mathbf{w} = \mathbf{r}_c + \mathbf{n}$. The m^{th} element of \mathbf{y}_k , denoted by $y_{k,m}$ is formed by

$$y_{k,m} = (\mathbf{X}_{k,m} \hat{\mathbf{h}}_k)^* \mathbf{r}'_k \quad (3.16)$$

$$\approx \begin{cases} \sum_{l=1}^{L_k} \|\mathbf{x}_{k,l,m} h_{k,l}\|^2 + \sum_{l=1}^{L_k} (\mathbf{x}_{k,l,m} h_{k,l})^* \mathbf{w}, & \text{if } i_k(j) = m \\ \sum_{l=1}^{L_k} (\mathbf{x}_{k,l,m} h_{k,l})^* \mathbf{w}, & \text{if } i_k(j) \neq m \end{cases}$$

$$= \begin{cases} N\check{P}_k + w, & \text{if } i_k(j) = m \\ w, & \text{if } i_k(j) \neq m \end{cases} \quad (3.17)$$

where $\check{P}_k = \sum_{l=1}^{L_k} |h_{k,l}|^2$ stands for the instantaneous (and time-varying) received power for the k^{th} user and $w = \sum_{l=1}^L (\mathbf{x}_{k,l,m} h_{k,l})^* \mathbf{w}$ stands for the noise plus residual interference component in the decision statistic. Provided that the processing gain and the number of users are large, we can model w as complex Gaussian random variable $w \sim \mathcal{CN}(0, \sigma_w^2)$. The approximation in (3.16) is due to the facts that perfect channel estimation is assumed so that $\hat{\mathbf{h}}_k = \mathbf{h}_k$ and that the autocorrelation of the same user's scrambling codes is assumed to approximate delta function so that the cross-correlation terms $(\mathbf{x}_{k,i,m} h_{k,i})^* \mathbf{x}_{k,j,m} h_{k,j}$, $i \neq j$ approximate zero and can be omitted.

The conditional pdf is thus

$$\begin{aligned} f(\mathbf{y}_k | i_k(j) = m) &= \frac{1}{(\pi \sigma_w^2)^M} \exp \left[-\frac{|y_{k,m} - N\check{P}_k|^2}{\sigma_w^2} - \sum_{\substack{j=1 \\ j \neq m}}^M \frac{|y_{k,j}|^2}{\sigma_w^2} \right] \\ &= \frac{1}{(\pi \sigma_w^2)^M} \exp \left[\frac{-N^2 \check{P}_k^2 + 2N\check{P}_k \text{Re}\{y_{k,m}\}}{\sigma_w^2} - \sum_{j=1}^M \frac{|y_{k,j}|^2}{\sigma_w^2} \right] \end{aligned} \quad (3.18)$$

Note that

$$\begin{aligned} P[i_k(j) = m | \mathbf{y}_k] &= \frac{f[\mathbf{y}_k | i_k(j) = m] \cdot P[i_k(j) = m]}{f[\mathbf{y}_k]} \\ f[\mathbf{y}_k] &= \sum_{q=1}^M P[i_k(j) = q] \cdot f[\mathbf{y}_k | i_k(j) = q] \\ P[i_k(j) = 1] &= \dots = P[i_k(j) = M] = \frac{1}{M} \end{aligned}$$

and combining the above equations with (3.18) yields

$$\begin{aligned}
b_{k,m} &= P[i_k(j) = m | \mathbf{y}_k] = \frac{f[\mathbf{y}_k | i_k(j) = m]}{\sum_{q=1}^M f[\mathbf{y}_k | i_k(j) = q]} \\
&= \frac{\exp\left[-\frac{N^2 \check{P}_k^2}{\sigma_w^2}\right] \cdot \exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,m}\}}{\sigma_w^2}\right]}{\exp\left[-\frac{N^2 \check{P}_k^2}{\sigma_w^2}\right] \cdot \sum_{q=1}^M \exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,q}\}}{\sigma_w^2}\right]} \\
&= \frac{\exp\left[2N \check{P}_k \operatorname{Re}\{y_{k,m}\} / \sigma_w^2\right]}{\sum_{q=1}^M \exp\left[2N \check{P}_k \operatorname{Re}\{y_{k,q}\} / \sigma_w^2\right]} \tag{3.19}
\end{aligned}$$

If $\mathbf{r}_c = \mathbf{0}$, i.e., the cancellation residual is neglected by assuming perfect cancellation, the noise plus interference variance can be approximated as $\sigma_w^2 \approx \sigma_n^2 = N \sum_{l=1}^{L_k} |h_{k,l}|^2 N_0 = \check{P}_k N N_0$, leading to the solution

$$\begin{aligned}
b_{k,m} &= \frac{\exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,m}\}}{\sigma_n^2}\right]}{\sum_{q=1}^M \exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,q}\}}{\sigma_n^2}\right]} = \frac{\exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,m}\}}{\check{P}_k N N_0}\right]}{\sum_{q=1}^M \exp\left[\frac{2N \check{P}_k \operatorname{Re}\{y_{k,q}\}}{\check{P}_k N N_0}\right]} \\
&= \frac{\exp[2\operatorname{Re}\{y_{k,m}\} / N_0]}{\sum_{q=1}^M \exp[2\operatorname{Re}\{y_{k,q}\} / N_0]} \tag{3.20}
\end{aligned}$$

Unfortunately, we found out that direct implementation of (3.20) leads to a numerically unstable algorithm, which was not pointed out in [16]. The problem can be tackled by performing the MMSE estimation in the log domain, similar to the idea presented in [24] for iterative decoding. With Log-NMIC algorithm, the coefficient $b_{k,m}$ associated with each orthogonal component is calculated as

$$b_{k,m} = \exp\left[\frac{2\operatorname{Re}\{y_{k,m}\}}{N_0} - \max^*\left(\frac{2\operatorname{Re}\{y_{k,1}\}}{N_0}, \dots, \frac{2\operatorname{Re}\{y_{k,M}\}}{N_0}\right)\right] \tag{3.21}$$

where the function $\max^*(\cdot)$ is defined as

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|})$$

which is max operation compensated with a correction term $\ln(1+e^{-|x-y|})$. Also

$$\max^*(x, y, z) = \max^*[\max^*(x, y), z]$$

The NMIC algorithm can be expressed as

$$\begin{aligned} \hat{\mathbf{r}}_i &= \mathbb{E}\{\mathbf{r}_i|\mathbf{y}_i\} = \sum_{m=1}^M b_{i,m} \mathbf{X}_{i,m} \mathbf{h}_i \\ \mathbf{r}'_k &= \mathbf{r} - \sum_{i \neq k} \hat{\mathbf{r}}_i \\ z_k^{\text{NMIC}}(m) &= \text{Re}\{\mathbf{y}_k\} = \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{r}'_k\} \end{aligned} \quad (3.22)$$

The above scheme assumes perfect cancellation and zero cross-correlation which is not the case in practice and leads to suboptimal solution. Now, we derive an adaptive algorithm [16] that takes the imperfect cancellation and non-zero cross-correlation into account and adaptively estimates the noise plus interference variance σ_w^2 . We know that

$$\mathbb{E}[\|\mathbf{y}_k\|^2] = \sum_{q=1}^M \mathbb{E}[|y_{k,q}|^2] = N^2 P_k^2 + M \sigma_w^2$$

Therefore

$$\sigma_w^2 = \frac{\mathbb{E}[\|\mathbf{y}_k\|^2] - N^2 P_k^2}{M}$$

The performance of NMIC algorithm can be improved if σ_n^2 in (3.20) is replaced by $\hat{\sigma}_w^2$, which is adaptively estimated by averaging $\frac{\|\mathbf{y}_k\|^2 - N^2 P_k^2}{M}$ over the whole block of symbols. This leads to adaptive Log-NMIC algorithm

$$b_{k,m} = \exp \left[\frac{2\text{Re}\{y_{k,m}\}}{\hat{\sigma}_w^2 / N P_k} - \max^* \left(\frac{2\text{Re}\{y_{k,1}\}}{\hat{\sigma}_w^2 / N P_k}, \dots, \frac{2\text{Re}\{y_{k,M}\}}{\hat{\sigma}_w^2 / N P_k} \right) \right] \quad (3.23)$$

3.6 Numerical results

In our simulations, each user transmits one of $M = 8$ Walsh codes spread to a total length of $N = 64$ chips. The effective spreading of the system is $N/\log_2 M = 64/3$ chips per bit. Different users are separated by different scrambling codes $\mathbf{C}_k(j)$ which are random, and differ from symbol to symbol (long-code system).

For simplicity, the simulated system is assumed to be chip-synchronous, i.e., all path delays are assumed to be multiples of T_c . This represents the worst-case interference scenario [25]. However, the system is asynchronous

on the symbol level. The channel gain $h_{k,l}(t)$ is a complex circular Gaussian process with autocorrelation function $E[h_{k,l}^*(t)h_{k,l}(t+\tau)] = P_{k,l}J_0(2\pi f_d\tau)$ and $J_0(x)$ is the zeroth order Bessel function of the first kind. The normalized Doppler frequency is assumed to be $f_dT = 0.01$. Perfect slow power control is assumed in the sense that $P_k = \sum_{l=1}^{L_k} P_{k,l}$, the average received power, is equal for all users. The channel coefficients are normalized so that each user has unity gain, i.e., $P_k = \sum_{l=1}^{L_k} P_{k,l} = 1$. The simulation results are averaged over random distributions of fading, noise, delay, and scrambling code through numerous Monte-Carlo runs.

The above parameter setting also applies to the subsequent experiments in the following chapters unless otherwise stated.

The conventional MF is used for the first stage of the multiuser detection to account for the fact that channel estimates are not yet available at the initial iteration. In the following stages, both interference cancellation/suppression and channel estimation are carried out in decision directed mode using the detected data from the previous iteration. Channel estimation is conducted with the maximum likelihood algorithm introduced in Section 5.2. A FIR filter of length 19 is used for channel smoothing.

The performance of different multiuser detectors is compared in Fig. 3.1. The system is simulated for a $K = 12$ -user system with 3-path channels, $L_k = L = 3$ for all k . The spacing between the three paths of each user is set to $2T_c$. The results are obtained after three iteration stages, initialized with conventional noncoherent first stage. The noncoherent IIS does not show much improvement compared to the conventional receiver, while other coherent schemes achieve considerable performance gain. We can also see that PIC performs better than IIS and WMF. Considering the fact that when long spreading (scrambling) codes are used, which is the case in most of the practical systems, the linear filters (e.g., IIS, WMF) update at a symbol rate, the inverse of \mathbf{U} in (3.11), \mathbf{R} in (3.12) has to be re-calculated every time, which significantly increases the computational complexity. On the contrary, PIC does not deal with any matrix inversion, thus considerably reduces the complexity compared to linear detectors. That makes PIC a more attractive detection algorithm in long-code CDMA systems.

The multistage approximate ML detector presented in Section 3.2 is not simulated. It, however, should have the same performance as PIC, since we have proved in Section 3.2 that these two schemes finally converge, although they detect data in different approaches (AML detection is linear scheme, while PIC is nonlinear). The advantage of using ML detection is the avoidance of estimating fading process, which is needed by PIC. While the performance of PIC can be improved considerably when combined with channel smoothing, which is not possible for the AML detector.

Fig. 3.2 shows the performance of multistage PIC for single-path and 3-path channels respectively. The iterations are initialized with a conventional noncoherent stage. One can observe that the gap between PIC and

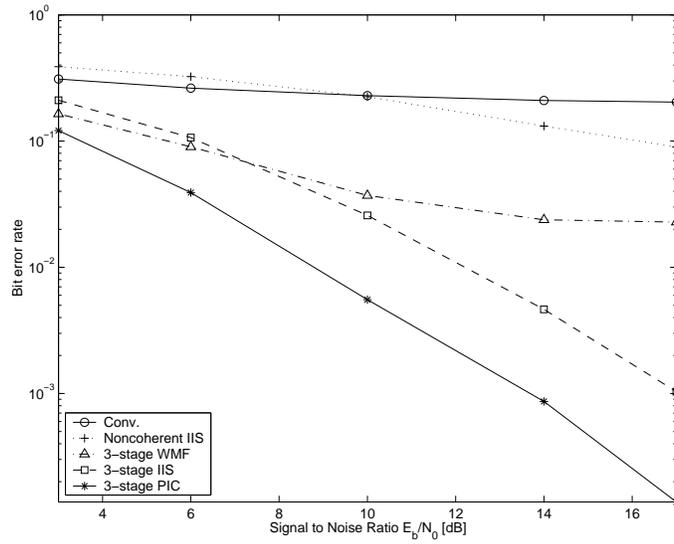


Figure 3.1: Performance comparison of different multiuser detectors.

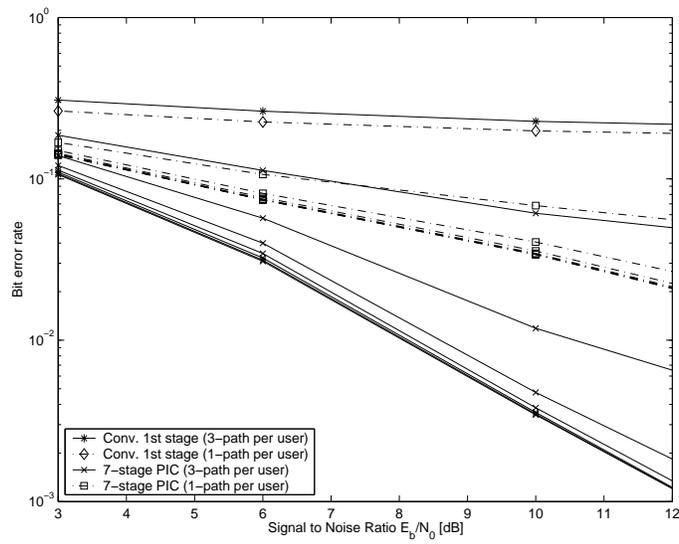


Figure 3.2: Performance of multi-stage PIC.

the conventional receiver becomes larger as SNR increases, which means we need to maintain a reasonable level of signal to noise ratio in order to achieve considerable capacity gain by applying multiuser detectors. The multistage PIC in 3-path case achieves lower error probabilities compared to the single-path case due to the diversity gain. However, in both cases, their performance gets saturated after 3 or 4 iterations.

The conventional PIC and the proposed soft PIC algorithms are compared numerically in Fig. 3.3 and 3.4. The number of path per user is set to $L = 3$. The number of users is set to $K = 21$, which means a fully loaded system. As indicated by the Fig. 3.3, the soft cancellation alone yields limited performance improvement, while the gain is much more obvious when the soft channel estimation is performed in conjunction with soft cancellation. Refer to Section 5.6 for soft channel estimation algorithms. Among the two soft PIC algorithms, the adaptive Log-NMIC performs slightly better than the MAP-PIC at low SNR; however, their performance is essentially the same in high SNR region, both achieve up to a 2 dB gain compared to the conventional PIC in a 21-user system. The gain achieved by soft channel estimation is less prominent for non-adaptive NMIC. However, the performance of adaptive NMIC is much superior to non-adaptive NMIC, as shown in Fig. 3.4.

Fig. 3.5 shows the convergence property of the conventional and soft PIC scheme (the MAP version). They both take 6 or 7 iterations to converge. The gain at each iteration becomes bigger as the SNR increases; the gap between them increases as the iteration goes on. The conclusion is that we need to perform enough stages and maintain a sufficiently high SNR to be able to fully benefit from the soft cancellation and soft channel estimation processes.

3.7 Summary

In this chapter, we compared the performance of different linear, nonlinear multiuser detectors. The conclusion is that nonlinear multistage PIC is better choice in practical systems with long spreading codes, in consideration of both performance and computational complexity. We showed that PIC is an approximate ML approach to data detection. In case of perfect cancellation, it is optimum multiuser detector in the sense of maximum likelihood data detection. The knowledge of the channel is essential to increase the system capacity. We need knowledge of the complex channel gains to do interference mitigation and coherent combining. Coherent detection gives better performance than noncoherent detection when the channel gains are accurately estimated.

The conventional interference cancellation receiver is subject to performance degradation due to incorrect decisions on interference subtracted from the received signal. To improve the performance of interference can-

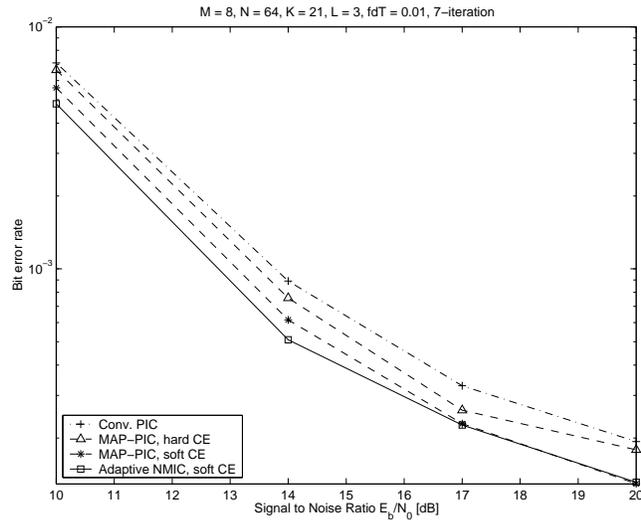


Figure 3.3: Performance of different IC schemes. The curves are plotted for the 7th stage IC.

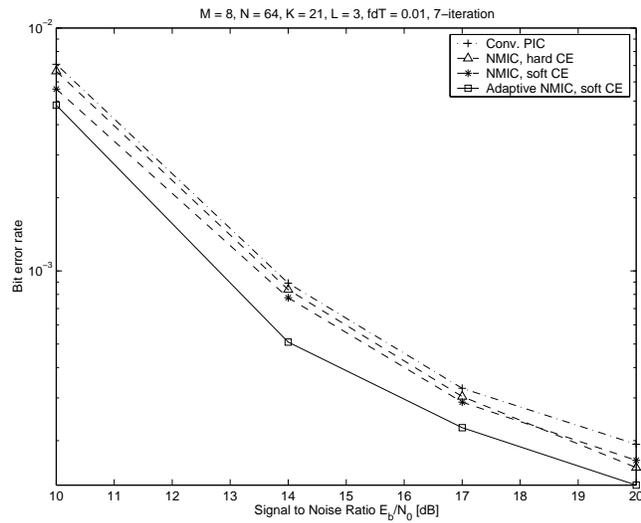


Figure 3.4: Performance of different IC schemes. The curves are plotted for the 7th stage IC.

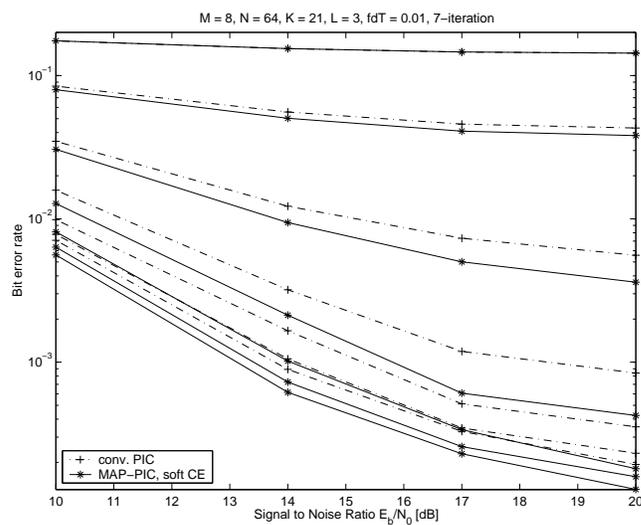


Figure 3.5: Convergence property of IC schemes. Topmost curve represents the noncoherent first stage and the second curve from top represents the first stage IC, the bottommost curve represents the 7th stage IC.

cellation in the uncoded system, we derived two soft cancellation schemes, one based on MAP criterion, the other based on nonlinear MMSE estimation. They are shown to be superior to the conventional PIC scheme with minor increase in complexity. Furthermore, the best system performance (2 dB gain in a 21-user system) is observed when the derived soft information is also used for channel estimation.

In this part of the thesis, we tackle the problem of theoretical evaluation for the multistage parallel interference cancellation (PIC) scheme in a DS-CDMA system with orthogonal modulation and long scrambling codes, which has not been addressed in the previous literature. The studied system operates on the reverse link in a time-varying multipath Rayleigh fading channel. By applying the Central Limit Theorem to multiple access interference (MAI) and intersymbol interference (ISI), as well as identically distributed chips from a single interferer, the bit error rate (BER) performance of the PIC scheme at any stage can be recursively computed from the signal-to-noise ratio, number of users, the number of paths per user, processing gain of the CDMA system, as well as the average received power of each path. For completeness, the BER expression is derived for chip synchronous and chip asynchronous systems over both equal and unequal power multipath channels. The proposed analysis is validated by the Monte-Carlo simulations and proved to be accurate, and it gives insight into the performance and capacity one can expect from the PIC based receivers under different situations. For instance, the analytical results can be used to examine the convergence property, multipath diversity gains as well as near-far resistance of the PIC scheme.

4.1 Introduction

The performance of orthogonal modulated DS-CDMA system with non-coherent and coherent combining was evaluated analytically in [26, 27] and in [28], respectively. The performance of interference canceler for

short-code CDMA systems with BPSK signaling was investigated, e.g., in [29, 30, 31, 32]. An adaptive multistage PIC scheme was analyzed in [29], and a closed form expression for BER performance is presented for the system operating over AWGN channels. The BER expressions are extended to derive asymptotic limits on the performance of interference cancellation as the number of cancellation stages approaches infinity, demonstrating a fundamental limit on the performance that can be expected from the multistage PIC scheme. In [31], an analytical BER expression for an adaptive multistage interference canceler was presented using an improved Gaussian approximation. The inclusion of second order statistics of MAI allows better performance prediction in cases where interference power has a random distribution, and it can be used to evaluate the performance of multistage PIC in arbitrary fading environments.

However, to our best knowledge, no results on the performance analysis of PIC for long-code CDMA systems in general, and PIC for orthogonal modulated CDMA systems in particular are available in the existing literature. The previous performance evaluation only relied on the use of simulation techniques. In this chapter, we provide an analytical approach to assess the performance of PIC for the system in question.

4.2 Theoretical Analysis For Equal Power Diversity Branches

In this section of performance analysis, we assume equal power among different paths of each user. The analysis in case of unequal gain among different diversity branches is discussed in Section 4.3.

Noncoherent equal gain combining expressed by (3.1) in Chapter 3 is used for the first stage of the PIC scheme to account for the fact that channel estimates are not yet available at the initial iteration. In the following stages, both interference cancellation and channel estimation are carried out in decision directed mode using the detected data from the previous iteration. The PIC algorithm is specified by (3.2) in Section 3.1. Channel estimation is conducted with the maximum likelihood algorithm (see equation (5.1) in Chapter 5).

4.2.1 Analysis for non-coherent first stage

To evaluate the probability of error, without loss of generality, let us assume that the j^{th} symbol transmitted from the k^{th} user is the first Walsh symbol and the channel gain remains constant during one symbol interval. The

decision statistic expressed in (3.1) can be reformed as

$$z_k(m) = \begin{cases} \sum_{l=1}^{L_k} |d_{k,l} + m_{k,l} + i_{k,l} + n_{k,l}|^2, & \text{if } m = 1 \\ \sum_{l=1}^{L_k} |m_{k,l} + i_{k,l} + n_{k,l}|^2, & \text{if } m \neq 1 \end{cases} \quad (4.1)$$

where $d_{k,l} = Nh_{k,l}$, $m_{k,l}$, $i_{k,l}$, and $n_{k,l}$ stand for the desired signal, contribution from MAI, ISI, and noise, respectively.

In [33], the long pseudonoise sequences were modeled as a random binary sequence which led to the strong justification of an accurate Gaussian approximation based on the central limit theorem. In this way, an equivalent noise-power spectral density for interference can be defined, which gives immediate insight into the degree of interference present in the receivers. This is the approach we take here to compute the variance of $m_{k,l}$ and $i_{k,l}$. If the processing gain is large enough, both MAI and ISI terms can be modeled as independent zero mean complex Gaussian random vectors and they are uncorrelated with the noise vector. Therefore, for the k^{th} user's l^{th} receiver branch, the interference plus noise variance is

$$\begin{aligned} \sigma^2 &= \sigma_m^2 + \sigma_i^2 + \sigma_n^2 \\ \sigma_m^2 &= \text{var}[m_{k,l}] = \text{E}[|m_{k,l}|^2] \\ \sigma_i^2 &= \text{var}[i_{k,l}] = \text{E}[|i_{k,l}|^2] \\ \sigma_n^2 &= \text{var}[n_{k,l}] = \text{E}[|n_{k,l}|^2] \end{aligned}$$

Note that $\sigma^2, \sigma_m^2, \sigma_i^2$ depend on k and l , which is not explicitly indicated in order to simplify notations. The noise variance can be easily computed as

$$\sigma_n^2 = \sum_{n=1}^N N_0 = NN_0$$

According to [33], for direct-sequence systems with long spreading sequences, the elements (chips) of each ISI or MAI sequence corresponding to the m^{th} user's i^{th} path, can be approximated as statistically independent and each element can be treated as a zero mean Gaussian random variable with variance

$$\frac{2}{3}\text{E}[|h_{m,i}|^2] = \frac{2}{3}P_{m,i}$$

for chip asynchronous systems, and

$$\text{E}[|h_{m,i}|^2] = P_{m,i}$$

for chip synchronous systems given $\psi(t)$ is a rectangular pulse. For chip

asynchronous system, time delays are assumed to be uniform distributed over $[0, T_c]$, where T_c is the chip interval. Consequently, we can derive the variance of the MAI and ISI as

$$\sigma_i^2 = \begin{cases} \frac{2}{3} \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \sum_{n=1}^N \mathbb{E}[\|h_{k,i}\|^2] = \frac{2N}{3} \sum_{\substack{i=1 \\ i \neq l}}^{L_k} P_{k,i}, \\ \text{for chip asynchronous systems} \end{cases}$$

$$\sigma_i^2 = \begin{cases} \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \sum_{n=1}^N \mathbb{E}[\|h_{k,i}\|^2] = N \sum_{\substack{i=1 \\ i \neq l}}^{L_k} P_{k,i}, \\ \text{for chip synchronous systems} \end{cases}$$

$$\sigma_m^2 = \begin{cases} \frac{2}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \sum_{n=1}^N \mathbb{E}[\|h_{m,i}\|^2] = \frac{2N}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} P_{m,i}, \\ \text{for chip asynchronous systems} \end{cases}$$

$$\sigma_m^2 = \begin{cases} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \sum_{n=1}^N \mathbb{E}[\|h_{m,i}\|^2] = N \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} P_{m,i}, \\ \text{for chip synchronous systems} \end{cases}$$

It is worth noticing that a chip asynchronous system is more resistant to MAI and ISI a chip synchronous system. In case of equal gain among different diversity branches, i.e., $P_{k,1} = P_{k,2} = \dots = P_{k,L_k} = P$, the interference variance does not differ from path to path. For the first stage noncoherent reception expressed in (4.1), the probability density function (pdf) of the decision statistics z_k is given by

$$f(z_k) = \begin{cases} \frac{1}{\sigma_1^{2L_k} (L_k-1)!} z_k^{L_k-1} e^{-\frac{z_k}{\sigma_1^2}}, & \text{if } m = 1 \\ \frac{1}{\sigma_2^{2L_k} (L_k-1)!} z_k^{L_k-1} e^{-\frac{z_k}{\sigma_2^2}}, & \text{if } m \neq 1 \end{cases}$$

which is central chi-square distribution with $2L_k$ degrees of freedom. The variances σ_1^2 and σ_2^2 are computed as

$$\sigma_1^2 = \mathbb{E}(|Nh_{k,l} + m_{k,l} + i_{k,l} + n_{k,l}|^2) = N^2P + \sigma_m^2 + \sigma_i^2 + \sigma_n^2$$

$$\sigma_2^2 = \sigma_m^2 + \sigma_i^2 + \sigma_n^2$$

The probability of making correct symbol decision for user k is calcu-

lated according to [1, p. 789] as

$$\mathcal{P}_{c,k} = \int_0^\infty \left[1 - e^{-z_k} \sum_{l=0}^{L_k-1} \frac{z_k^l}{l!} \right]^{M-1} \frac{z_k^{L-1}}{(1+\gamma)^L (L_k-1)!} \exp\left(-\frac{z_k}{1+\gamma}\right) dz_k \quad (4.2)$$

and the bit error probability is

$$\mathcal{P}_{b,k} = \frac{M}{2(M-1)} (1 - \mathcal{P}_{c,k}) \quad (4.3)$$

where $\gamma = N^2 P / \sigma_2^2$ is the average signal to interference plus noise ratio (SINR) per diversity branch.

4.2.2 Analysis for multistage PIC

The performance of coherent combining for single-user M-ary orthogonal systems with space diversity was analyzed in [28]. Here we extend its application to the analysis of PIC schemes in multiuser environments.

Let us assume that the first Walsh symbol was transmitted from the k^{th} user. The decision statistic expressed in (3.2) can be reformed after p stages of cancellations

$$\begin{aligned} z_k^{(p)}(m) &= \sum_{l=1}^{L_k} \text{Re}\{\hat{h}_{k,l}^* \mathbf{x}_{k,l,m} \mathbf{r}'\} \\ &= \begin{cases} \sum_{l=1}^{L_k} d_{k,l} + m_{k,l}^{(p)} + i_{k,l}^{(p)} + n_{k,l} = d + n_1^{(p)}, & \text{if } m = 1 \\ \sum_{l=1}^{L_k} m_{k,l}^{(p)} + i_{k,l}^{(p)} + n_{k,l} = n_m^{(p)}, & \text{if } m \neq 1 \end{cases} \end{aligned} \quad (4.4)$$

where $d = \sum_{l=1}^{L_k} d_{k,l} = N \sum_{l=1}^{L_k} h_{k,l} \hat{h}_{k,l}^*$ is the desired signal. The noise component for the l^{th} diversity branch is denoted by $n_{k,l}$. The contributions from MAI and ISI for the l^{th} diversity branch at the p^{th} stage are denoted by $m_{k,l}^{(p)}$ and $i_{k,l}^{(p)}$ respectively. Using the Gaussian approximation, n_1, n_2, \dots, n_M are zero-mean statistically independent Gaussian random variables with equal variance $(\sigma^2)^{(p)}/2$. The factor of 1/2 is due to the fact that the $\text{Re}(\cdot)$ operation in equation (4.4) removes the noise and interference present in the imaginary part of the decision statistics.

Let us denote $\mathcal{P}_{c,k}^{(p)}(x)$ as the probability that the receiver makes correct symbol decision for user k at the p^{th} stage conditioned on x , which is defined as $x = \frac{d}{\sigma^{(p)}}$. It is the probability that $z_k^{(p)}(1) = d + n_1^{(p)}$ is larger than each of the other $M - 1$ outputs $z_k^{(p)}(2) = n_2^{(p)}, z_k^{(p)}(3) = n_3^{(p)}, \dots, z_k^{(p)}(M) =$

$n_M^{(p)}$ [1]:

$$\begin{aligned}
\mathcal{P}_{c,k}^{(p)}(x) &= P_r \left(z_k^{(p)}(2) < z_k^{(p)}(1), z_k^{(p)}(3) < z_k^{(p)}(1), \dots, z_k^{(p)}(M) < z_k^{(p)}(1) | x \right) \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \exp \left[-\frac{1}{2} \left(y - \frac{\sqrt{2}d}{\sigma^{(p)}} \right)^2 \right] dy \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \exp \left[-\frac{1}{2} (y - \sqrt{2}x)^2 \right] dy \quad (4.5)
\end{aligned}$$

where the function $Q(x)$ is defined as $Q(x) = \frac{1}{2} \operatorname{erfc}(\frac{x}{\sqrt{2}})$. The interference plus noise variance at the p^{th} stage is defined as $(\sigma^2)^{(p)} = (\sigma_m^2)^{(p)} + (\sigma_i^2)^{(p)} + \sigma_n^2$. The noise term does not change between iterations and can be computed as $\sigma_n^2 = NN_0 \sum_{l=1}^{L_k} |h_{k,l}|^2$.

In the derivation of the variance of MAI, which changes at each iteration due to interference cancellation, we utilize some distinct feature of the Walsh code as depicted by Table 2.1 and 4.1. The new vector \mathbf{r}' is obtained by canceling other user's distribution path-by-path using the decision feedback from the $(p-1)^{\text{th}}$ stage. At the p^{th} stage, the probability of correct cancellation is $\mathcal{P}_{c,m}^{(p-1)} = 1 - \mathcal{P}_{e,m}^{(p-1)}$, where the interfering user $m = 1, \dots, K$, and $m \neq k$. The variance of the remaining MAI after correct cancellation (or cancellation residual) is of course zero. On the other hand, in case of erroneous cancellation, which occurs with probability $\mathcal{P}_{e,m}^{(p-1)}$, the cancellation residual is determined by the difference of two distinct Walsh symbols. Table 4.1 indicates that if a Walsh codeword is subtracted by another Walsh codeword, the resulting word $\Delta \mathbf{w}$ contains $\frac{M}{2}$ number of zeros and $\frac{M}{2}$ number of ± 2 s. Although Table 4.1 is not exhaustive, the rest of the words can be easily computed from Table 2.1 and shown to comply with the same rule. We use $M = 8$ as an example in these tables; however, the conclusion applies to any value of M . The cancellation residual for each path is formed by spreading $\Delta \mathbf{w}$ to a number of N chips (which consequently contains $\frac{N}{2}$ number of zeros and $\frac{N}{2}$ number of ± 2 s), scrambling with a random code, then multiplying the scrambled sequence with channel coefficient $h_{m,i}$, where $i = 1, 2, \dots, L_m$. The variance of MAI in the l^{th} diversity branch after cancellation is therefore

$$\begin{aligned}
(\sigma_m^2)_l^{(p)} &= (1 - \mathcal{P}_{e,m}^{(p-1)}) \cdot 0 + |h_{k,l}|^2 (\pm 2)^2 \frac{N}{2} \cdot \frac{2}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} \mathbb{E}[|h_{m,i}|^2] \\
&= |h_{k,l}|^2 \frac{4N}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}
\end{aligned}$$

Table 4.1: Difference between different Walsh codewords

$\mathbf{w}_0 - \mathbf{w}_1$	0 0 0 0 2 2 2 2				
$\mathbf{w}_0 - \mathbf{w}_2$	0 0 2 2 0 0 2 2	$\mathbf{w}_1 - \mathbf{w}_2$	0 0 2 2 -2 -2 0 0		
$\mathbf{w}_0 - \mathbf{w}_3$	0 0 2 2 2 2 0 0	$\mathbf{w}_1 - \mathbf{w}_3$	0 0 2 2 0 0 -2 -2	$\mathbf{w}_2 - \mathbf{w}_3$	0 0 0 0 2 2 -2 -2
$\mathbf{w}_0 - \mathbf{w}_4$	0 2 0 2 0 2 0 2	$\mathbf{w}_1 - \mathbf{w}_4$	0 2 0 2 -2 0 -2 0	$\mathbf{w}_2 - \mathbf{w}_4$	0 2 -2 0 0 2 -2 0
$\mathbf{w}_0 - \mathbf{w}_5$	0 2 0 2 2 0 2 0	$\mathbf{w}_1 - \mathbf{w}_5$	0 2 0 2 0 -2 0 -2	$\mathbf{w}_2 - \mathbf{w}_5$	0 2 -2 0 2 0 0 -2
$\mathbf{w}_0 - \mathbf{w}_6$	0 2 2 0 0 2 2 0	$\mathbf{w}_1 - \mathbf{w}_6$	0 2 2 0 -2 0 0 -2	$\mathbf{w}_2 - \mathbf{w}_6$	0 2 0 -2 0 2 0 -2
$\mathbf{w}_0 - \mathbf{w}_7$	0 2 2 0 2 0 0 2	$\mathbf{w}_1 - \mathbf{w}_7$	0 2 2 0 0 -2 -2 0	$\mathbf{w}_2 - \mathbf{w}_7$	0 2 0 -2 2 0 -2 0

for chip asynchronous systems, and

$$\begin{aligned}
(\sigma_m^2)_l^{(p)} &= (1 - \mathcal{P}_{e,m}^{(p-1)}) \cdot 0 + |h_{k,l}|^2 (\pm 2)^2 \frac{N}{2} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} \mathbb{E}[|h_{m,i}|^2] \\
&= |h_{k,l}|^2 2N \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}
\end{aligned}$$

for chip synchronous systems. The variance of MAI from all the diversity branches of user k can therefore be computed as

$$(\sigma_m^2)^{(p)} = \sum_{l=1}^{L_k} (\sigma_m^2)_l^{(p)} = \begin{cases} \sum_{l=1}^{L_k} |h_{k,l}|^2 \frac{4N}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}, \\ \text{for chip asynchronous systems} \\ \sum_{l=1}^{L_k} |h_{k,l}|^2 2N \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}, \\ \text{for chip synchronous systems} \end{cases}$$

Next, we derive the variance of the self interference for user k . For the l^{th} diversity branch, the i^{th} ISI vector ($i = 1, \dots, L_k$, $i \neq l$) due to the k^{th} user's j^{th} symbol (the desired symbol) spans $N - |p_{k,i} - p_{k,l}|$ chips. For chip asynchronous system, its variance is computed as $|h_{k,l}|^2 \cdot \frac{2}{3} (N - |p_{k,i} - p_{k,l}|) P_{k,i}$, for the same reasoning as stated in Section 4.2.1, and it does not change at each iteration. The ISI component due to some other symbol spans $|p_{k,i} - p_{k,l}|$ chips, it is canceled with decision feedback at each iteration. It can be treated in the same way as MAI, its variance is therefore $|h_{k,l}|^2 \cdot \frac{4}{3} |p_{k,i} - p_{k,l}| \mathcal{P}_{e,k}^{(p-1)} P_{k,i}$. The variances for chip synchronous system can be derived similarly. To ease understanding, an example of the ISI analysis is given in Figure 4.1. The variance the total ISI term can

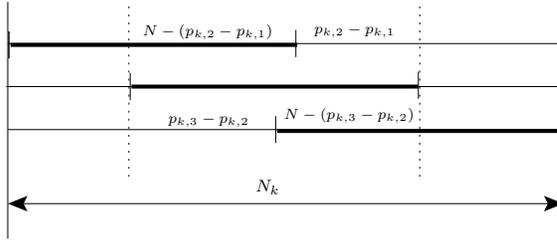


Figure 4.1: Sketch of ISI for the k^{th} user's 2nd path. The desired symbol spans N chips and is depicted with the bold line on each path. The processing window is $N_k = N + p_{k,L_k} - p_{k,1}$ chips. For the 2nd diversity branch, the ISI from the 1st path due to the desired symbol contains $N - (p_{k,2} - p_{k,1})$ chips; the ISI from the 1st path due to the other symbol contains $p_{k,2} - p_{k,1}$ chips. The ISI from the 3rd path due to the desired symbol contains $N - (p_{k,3} - p_{k,2})$ chips; the ISI from the 3rd path due to the other symbol contains $p_{k,3} - p_{k,2}$ chips.

therefore be computed as

$$(\sigma_i^2)^p = \begin{cases} \sum_{l=1}^{L_k} |h_{k,l}|^2 \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \left[\frac{2}{3} (N - |p_{k,i} - p_{k,l}|) P_{k,i} + \frac{4}{3} |p_{k,i} - p_{k,l}| \mathcal{P}_{e,k}^{(p-1)} P_{k,i} \right], \\ \text{for chip asynchronous systems} \\ \sum_{l=1}^{L_k} |h_{k,l}|^2 \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \left[(N - |p_{k,i} - p_{k,l}|) P_{k,i} + 2 |p_{k,i} - p_{k,l}| \mathcal{P}_{e,k}^{(p-1)} P_{k,i} \right], \\ \text{for chip synchronous systems} \end{cases}$$

$$= \begin{cases} \sum_{l=1}^{L_k} |h_{k,l}|^2 \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \frac{2}{3} \left[N + (2\mathcal{P}_{e,k}^{(p-1)} - 1) |p_{k,i} - p_{k,l}| \right] P_{k,i}, \\ \text{for chip asynchronous systems} \\ \sum_{l=1}^{L_k} |h_{k,l}|^2 \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \left[N + (2\mathcal{P}_{e,k}^{(p-1)} - 1) |p_{k,i} - p_{k,l}| \right] P_{k,i}, \\ \text{for chip synchronous systems} \end{cases}$$

Based on the above analysis, we derive the total noise plus interference variance as

$$(\sigma^2)^{(p)} = \sigma_n^2 + (\sigma_i^2)^{(p)} + (\sigma_m^2)^{(p)} = \sum_{l=1}^{L_k} |h_{k,l}|^2 (\alpha_l^2)^{(p)}$$

and $(\alpha_l^2)^{(p)}$ is defined as

$$(\alpha_l^2)^{(p)} = \begin{cases} NN_0 + \frac{2}{3} \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \left[N + (2\mathcal{P}_{e,k}^{(p-1)} - 1)|p_{k,i} - p_{k,l}| \right] P_{k,i} \\ \quad + \frac{4N}{3} \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}, \\ \text{for chip asynchronous systems} \\ \\ NN_0 + \sum_{\substack{i=1 \\ i \neq l}}^{L_k} \left[N + (2\mathcal{P}_{e,k}^{(p-1)} - 1)|p_{k,i} - p_{k,l}| \right] P_{k,i} \\ \quad + 2N \sum_{\substack{m=1 \\ m \neq k}}^K \sum_{i=1}^{L_m} \mathcal{P}_{e,m}^{(p-1)} P_{m,i}, \\ \text{for chip synchronous systems} \end{cases}$$

Assume accurate channel estimation, i.e., $\hat{h}_{k,l} \approx h_{k,l}$, then the desired signal $d = \sum_{l=1}^{L_k} d_{k,l} \approx N \sum_{l=1}^{L_k} |h_{k,l}|^2$. In case of equal power among different paths, i.e., $P_{k,1} = P_{k,2} = \dots = P_{k,L_k} = P$, then $(\alpha_1^2)^{(p)} = (\alpha_2^2)^{(p)} = \dots = (\alpha_{L_k}^2)^{(p)} = (\alpha^2)^{(p)}$. Denote

$$x = \frac{d}{\sigma^{(p)}} = \frac{N \sum_{l=1}^{L_k} |h_{k,l}|^2}{\alpha^{(p)} \sqrt{\sum_{l=1}^{L_k} |h_{k,l}|^2}} = \frac{N}{\alpha^{(p)}} \sqrt{\sum_{l=1}^{L_k} |h_{k,l}|^2}$$

$$z = x^2 = \frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} |h_{k,l}|^2$$

The random variable z is central chi-square distributed with $2L_k$ degrees of freedom and probability density function

$$p(z) = \frac{z^{L_k-1} \exp(-z/\gamma^{(p)})}{\gamma^{(p)L_k} (L_k - 1)!}; \quad 0 \leq z \leq \infty$$

where $\gamma^{(p)} = \frac{N^2}{(\alpha^2)^{(p)}} \mathbb{E}[|h_{k,l}|^2] = \frac{N^2 P}{(\alpha^2)^{(p)}}$ stands for the average SINR of each diversity branch. Consequently,

$$p(x) = \frac{2x^{2L_k-1} \exp(-x^2/\gamma^{(p)})}{\gamma^{(p)L_k} (L_k - 1)!}; \quad 0 \leq x \leq \infty$$

To obtain the error probability when x is random, we must average

$\mathcal{P}_{c,k}^{(p)}(x)$ given in (4.5) over the distribution of x , i.e.,

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)} &= \int_0^\infty \mathcal{P}_{c,k}^{(p)}(x)p(x)dx \\ &= \frac{1}{\sqrt{2\pi}} \int_0^\infty \int_{-\infty}^\infty [1 - Q(y)]^{M-1} \exp\left[-\frac{y^2}{2} + \sqrt{2}xy - x^2\right] \\ &\quad \cdot \frac{2x^{2L_k-1} \exp(-x^2/\gamma^{(p)})}{\gamma^{(p)L_k}(L_k-1)!} dy dx \end{aligned}$$

Following the procedure in [28], the BER at the p^{th} ($p > 1$) stage can be formulated as

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)} &= \frac{(2L_k-1)!}{\sqrt{2}(L_k-1)!(1+\gamma^{(p)})^{L_k}} \int_{-\infty}^\infty [1 - Q(y)]^{M-1} e^{-\frac{y^2}{2(1+\gamma^{(p)})}} \\ &\quad \cdot \operatorname{erfc}\left(2L_k-1, \frac{-y\sqrt{\gamma^{(p)}}}{\sqrt{2(1+\gamma^{(p)})}}\right) dy \\ \mathcal{P}_{b,k}^{(p)} &= \frac{M}{2(M-1)} \mathcal{P}_{e,k}^{(p)} = \frac{M}{2(M-1)} (1 - \mathcal{P}_{c,k}^{(p)}) \end{aligned} \quad (4.6)$$

where the symbol error probability $\mathcal{P}_{e,k}$ is initialized as

$$\mathcal{P}_{e,k}^{(1)} = 1 - \mathcal{P}_{c,k}^{(1)} = \frac{2\mathcal{P}_{b,k}^{(1)}(M-1)}{M}$$

and $\mathcal{P}_{b,k}^{(1)}$ is computed according to (4.2) and (4.3). The function $\operatorname{erfc}(m, x)$ is the m^{th} iterated integral of the $\operatorname{erfc}(x)$ function defined as [28]

$$\operatorname{erfc}(m, x) = \int_x^\infty \operatorname{erfc}(m-1, t) dt, \quad m = 0, 1, 2, \dots$$

It is initialized and iterated with the functions:

$$\begin{aligned} \operatorname{erfc}(-1, x) &= \frac{2}{\sqrt{\pi}} \exp(-x^2) \\ \operatorname{erfc}(0, x) &= \operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-t^2) dt \\ \operatorname{erfc}(m, x) &= \frac{1}{2m} \operatorname{erfc}(m-2, x) - \frac{x}{m} \operatorname{erfc}(m-1, x) \end{aligned}$$

Next, we derive an alternative way to simplify the computation of the error probability. Note that $\mathcal{P}_{c,k}^{(p)}$ is derived by taking the expectation of the function $\mathcal{P}_{c,k}^{(p)}(x)$ of the random variable x , i.e., $\mathcal{P}_{c,k}^{(p)} = \mathbb{E}[\mathcal{P}_{c,k}^{(p)}(x)] = \int_0^\infty \mathcal{P}_{c,k}^{(p)}(x)p(x)dx$. In [34], Holtzman introduced a simple and accurate

method to evaluate the expectation without carrying out the integration. First, we expand $\mathcal{P}_{c,k}^{(p)}(x)$ using a Taylor series in terms of central differences

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)}(x) &= \mathcal{P}_{c,k}^{(p)}(\mu_x) + (x - \mu_x) \left(\frac{\mathcal{P}_{c,k}^{(p)}(\mu_x + h) - \mathcal{P}_{c,k}^{(p)}(\mu_x - h)}{2h} \right) \\ &\quad + \frac{1}{2}(x - \mu_x)^2 \left(\frac{\mathcal{P}_{c,k}^{(p)}(\mu_x + h) - 2\mathcal{P}_{c,k}^{(p)}(\mu_x) + \mathcal{P}_{c,k}^{(p)}(\mu_x - h)}{h^2} \right) + \dots \end{aligned}$$

Let μ_x and σ_x^2 be the mean and variance of x , i.e., $\mu_x = \mathbb{E}[x]$ and $\sigma_x^2 = \mathbb{E}[(x - \mu_x)^2]$, then

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)} &= \mathbb{E}[\mathcal{P}_{c,k}^{(p)}(x)] \\ &\approx \mathcal{P}_{c,k}^{(p)}(\mu_x) + \frac{\sigma_x^2}{2} \left(\frac{\mathcal{P}_{c,k}^{(p)}(\mu_x + h) - 2\mathcal{P}_{c,k}^{(p)}(\mu_x) + \mathcal{P}_{c,k}^{(p)}(\mu_x - h)}{h^2} \right) \end{aligned}$$

It is shown in [34] that choosing $h = \sqrt{3}\sigma_x$ gives good accuracy, leading to the solution to our problem

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)}(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \exp\left[-\frac{1}{2}(y - \sqrt{2}x)^2\right] dy \\ \mathcal{P}_{c,k}^{(p)} &\approx \frac{2}{3}\mathcal{P}_{c,k}^{(p)}(\mu_x) + \frac{1}{6}\mathcal{P}_{c,k}^{(p)}(\mu_x + \sqrt{3}\sigma_x) + \frac{1}{6}\mathcal{P}_{c,k}^{(p)}(\mu_x - \sqrt{3}\sigma_x) \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \left\{ \frac{2}{3} \exp\left[-\frac{1}{2}(y - \sqrt{2}\mu_x)^2\right] \right. \\ &\quad \left. + \frac{1}{6} \exp\left[-\frac{1}{2}(y - \sqrt{2}(\mu_x + \sqrt{3}\sigma_x))^2\right] \right. \\ &\quad \left. + \frac{1}{6} \exp\left[-\frac{1}{2}(y - \sqrt{2}(\mu_x - \sqrt{3}\sigma_x))^2\right] \right\} dy \\ \mathcal{P}_{b,k}^{(p)} &= \frac{M}{2(M-1)} \mathcal{P}_{e,k}^{(p)} = \frac{M}{2(M-1)} (1 - \mathcal{P}_{c,k}^{(p)}) \end{aligned} \quad (4.7)$$

and μ_x and σ_x can be derived as

$$\mu_x = \mathbb{E}[x] = \int_0^{\infty} xp(x)dx = \frac{2}{\gamma^{(p)}L_k(L_k-1)!} \int_0^{\infty} x^{2L_k} \exp(-x^2/\gamma^{(p)})dx \quad (4.8)$$

$$\begin{aligned} \mathbb{E}[x^2] &= \mathbb{E}\left[\frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} |h_{k,l}|^2\right] = \frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} \mathbb{E}[|h_{k,l}|^2] = \frac{N^2 L_k P}{(\alpha^2)^{(p)}} \\ \sigma_x &= \sqrt{\mathbb{E}[x^2] - \mu_x^2} \end{aligned}$$

To obtain a closed form for μ_x expressed by (4.8), we use the formula [28],

$$\int_0^\infty \exp(-bx - ax^2)x^{s-1}dx = \frac{\sqrt{\pi}}{2}\Gamma(s)a^{-s/2} \exp\left[\frac{b^2}{4a}\right] \operatorname{erfc}\left(s-1, \frac{b}{2\sqrt{a}}\right) \quad (4.9)$$

where $\Gamma(s)$ is the gamma function [1]. Assigning $a = 1/\gamma^{(p)}$, $b = 0$, $s = 2L_k + 1$, we yield

$$\int_0^\infty x^{2L_k} \exp(-x^2/\gamma^{(p)})dx = \frac{\sqrt{\pi}}{2}(2L_k)! \left[\frac{1}{\gamma^{(p)}}\right]^{-\frac{2L_k+1}{2}} \operatorname{erfc}(2L_k, 0)$$

Thus,

$$\mu_x = \frac{\sqrt{\pi}(2L_k)!}{\gamma^{(p)L_k(L_k-1)!} \gamma^{(p)\frac{2L_k+1}{2}} \operatorname{erfc}(2L_k, 0)} = \sqrt{\pi\gamma^{(p)}} \frac{(2L_k)!}{(L_k-1)!} \operatorname{erfc}(2L_k, 0) \quad (4.10)$$

Since only the first and second order moment information is needed, the approach presented here can be easily extended to derive BER performance for systems operating over other multipath channels, e.g., the ones with lognormal or Nakagami distributions.

4.3 Theoretical Analysis For Unequal Power Diversity Branches

4.3.1 Analysis for non-coherent first stage

In case each path has unequal power, i.e., $P_{k,1} \neq P_{k,2} \neq \dots \neq P_{k,L_k}$ the decision statistic expressed in (4.1) can be formed as

$$z_k(m) = \begin{cases} U_1 = \sum_{l=1}^{L_k} u_1^l = \sum_{l=1}^{L_k} |Nh_{k,l} + m_{k,l} + i_{k,l} + n_{k,l}|^2, & \text{if } m = 1 \\ U_m = \sum_{l=1}^{L_k} u_m^l = \sum_{l=1}^{L_k} |m_{k,l} + i_{k,l} + n_{k,l}|^2, & \text{if } m \neq 1 \end{cases} \quad (4.11)$$

In case $m = 1$, each term $u_1^l = |Nh_{k,l} + m_{k,l} + i_{k,l} + n_{k,l}|^2$ is an independent central chi-square distributed random variable with 2 degrees of freedom and characteristic function $\psi_{u_1^l}(jv) = (1 - jv\gamma_l)^{-1}$, where $\gamma_l = E[|Nh_{k,l} + m_{k,l} + i_{k,l} + n_{k,l}|^2] = N^2P_{k,l} + \sigma_m^2 + \sigma_i^2 + \sigma_n^2$. The noise and interference variance is computed in the same way as in Section 4.2.1. As

a consequence of the statistical independence of u_1^l , $l = 1, 2, \dots, L_k$, the characteristic function of U_1 is

$$\begin{aligned}\psi_{U_1}(jv) &= \prod_{l=1}^{L_k} (1 - jv\gamma_l)^{-1} \\ &= \sum_{l=1}^{L_k} \left[\prod_{i=1, i \neq l}^{L_k} \left(1 - \frac{\gamma_i}{\gamma_l}\right)^{-1} \right] (1 - jv\gamma_l)^{-1} = \sum_{l=1}^{L_k} A_l (1 - jv\gamma_l)^{-1}\end{aligned}\quad (4.12)$$

where the coefficients of the partial fraction expansion $A_l = \prod_{i=1, i \neq l}^{L_k} (1 - \gamma_i/\gamma_l)^{-1}$ in equation (4.12) is based on the derivation in [35]. Taking the Fourier transform of (4.12), we obtain the pdf of U_1 as

$$p(U_1) = \sum_{l=1}^{L_k} \frac{A_l}{\gamma_l} \exp\left(-\frac{U_1}{\gamma_l}\right); \quad 0 \leq U_1 \leq \infty$$

Similarly,

$$p(U_m) = \sum_{l=1}^{L_k} \frac{B_l}{\beta_l} \exp\left(-\frac{U_m}{\beta_l}\right); \quad 0 \leq U_m \leq \infty, \quad m \neq 1$$

where

$$\begin{aligned}\beta_l &= \text{E}[|m_{k,l} + i_{k,l} + n_{k,l}|^2] = \sigma_m^2 + \sigma_i^2 + \sigma_n^2 \\ B_l &= \prod_{i=1, i \neq l}^{L_k} \left(1 - \frac{\beta_i}{\beta_l}\right)^{-1}\end{aligned}$$

The probability of making correct symbol decision can be computed as

$$\begin{aligned}\mathcal{P}_{c,k} &= P_r(U_2 < U_1, U_3 < U_1, \dots, U_M < U_1) \\ &= \int_0^\infty [P_r(U_2 < U_1)]^{M-1} p(U_1) dU_1 \\ P_r(U_2 < U_1) &= \int_0^{U_1} p(U_2) dU_2 = \int_0^{U_1} \sum_{l=1}^{L_k} \frac{B_l}{\beta_l} \exp\left(-\frac{U_2}{\beta_l}\right) dU_2 \\ &= \sum_{l=1}^{L_k} B_l \left[1 - \exp\left(-\frac{U_1}{\beta_l}\right)\right]\end{aligned}$$

Therefore, the BER for non-coherent first stage in unequal power mul-

tipath system is derived as

$$\begin{aligned} \mathcal{P}_{c,k} &= \int_0^\infty \left(\sum_{l=1}^{L_k} B_l \left[1 - \exp\left(-\frac{U_1}{\beta_l}\right) \right] \right)^{M-1} \sum_{l=1}^{L_k} \frac{A_l}{\gamma_l} \exp\left(-\frac{U_1}{\gamma_l}\right) dU_1 \\ \mathcal{P}_{b,k} &= \frac{M}{2(M-1)} \mathcal{P}_{c,k} = \frac{M}{2(M-1)} (1 - \mathcal{P}_{c,k}) \end{aligned} \quad (4.13)$$

4.3.2 Analysis for multistage PIC

In case each path has unequal power, the variable $x = \frac{d}{\sigma^{(p)}}$ is formed as

$$x = \frac{d}{\sigma^{(p)}} = \frac{N \sum_{l=1}^{L_k} |h_{k,l}|^2}{\sqrt{\sum_{l=1}^{L_k} |h_{k,l}|^2 (\alpha_l^2)^{(p)}}$$

The pdf of x is difficult to derive under such a circumstance because the numerator and denominator are not independent. However, if the self interference is small compared to noise and MAI, e.g., when the number of users K is much bigger than the number of paths L_k , which is usually the case, or when SNR is low, we can approximate $(\alpha_1^2)^{(p)} \approx (\alpha_2^2)^{(p)} \approx \dots \approx (\alpha_{L_k}^2)^{(p)} \approx (\alpha^2)^{(p)}$, then we can denote

$$z = x^2 = \sum_{l=1}^{L_k} z_l \approx \frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} |h_{k,l}|^2$$

and each term $z_l \approx \frac{N^2}{(\alpha^2)^{(p)}} |h_{k,l}|^2$ is an independent central chi-square distributed random variable with 2 degrees of freedom and characteristic function

$$\psi_{z_l}(jv) = (1 - jv\gamma_l^{(p)})^{-1}$$

where

$$\gamma_l^{(p)} = \frac{N^2}{(\alpha^2)^{(p)}} \mathbb{E}[|h_{k,l}|^2] = \frac{N^2 P_{k,l}}{(\alpha^2)^{(p)}}$$

As a consequence of the statistical independence of $z_l, l = 1, 2, \dots, L_k$, the characteristic function of z is

$$\begin{aligned} \psi_z(jv) &= \prod_{l=1}^{L_k} (1 - jv\gamma_l^{(p)})^{-1} = \sum_{l=1}^{L_k} \left[\prod_{i=1, i \neq l}^{L_k} \left(1 - \frac{\gamma_i^{(p)}}{\gamma_l^{(p)}} \right)^{-1} \right] (1 - jv\gamma_l^{(p)})^{-1} \\ &= \sum_{l=1}^{L_k} A_l^{(p)} (1 - jv\gamma_l^{(p)})^{-1} \end{aligned} \quad (4.14)$$

where $A_l^{(p)} = \prod_{i=1, i \neq l}^{L_k} (1 - \gamma_i^{(p)}/\gamma_l^{(p)})^{-1}$.

Taking the Fourier transform of (4.14), we obtain the pdfs of z and x as

$$\begin{aligned} p(z) &= \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)}} \exp \left[-\frac{z}{\gamma_l^{(p)}} \right]; \quad 0 \leq z \leq \infty \\ p(x) &= 2x \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)}} \exp \left[-\frac{x^2}{\gamma_l^{(p)}} \right]; \quad 0 \leq x \leq \infty \end{aligned} \quad (4.15)$$

To obtain the error probability when x is random, we must average $\mathcal{P}_{c,k}^{(p)}(x)$ given in (4.5) over the distribution of x , i.e.,

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)} &= \int_0^\infty \mathcal{P}_{c,k}^{(p)}(x) p(x) dx \\ &= \sqrt{\frac{2}{\pi}} \int_0^\infty \int_{-\infty}^\infty [1 - Q(y)]^{M-1} \exp \left[-\frac{y^2}{2} + \sqrt{2}xy - x^2 \right] \\ &\quad \cdot \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)}} x \exp \left(-\frac{x^2}{\gamma_l^{(p)}} \right) dy dx \\ &= \sqrt{\frac{2}{\pi}} \int_{-\infty}^\infty [1 - Q(y)]^{M-1} \exp \left[-\frac{y^2}{2} \right] dy \int_0^\infty \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)}} x \\ &\quad \cdot \exp \left(\sqrt{2}yx - \frac{\gamma_l^{(p)} + 1}{\gamma_l^{(p)}} x^2 \right) dx \end{aligned} \quad (4.16)$$

Recall that

$$\int_0^\infty \exp(-bx - ax^2) x^{s-1} dx = \frac{\sqrt{\pi}}{2} \Gamma(s) a^{-s/2} \exp \left[\frac{b^2}{4a} \right] \operatorname{erfc} \left(s - 1, \frac{b}{2\sqrt{a}} \right)$$

Assigning $a = \frac{\gamma_l^{(p)} + 1}{\gamma_l^{(p)}}$, $b = -\sqrt{2}y$, $s = 2$, equation (4.16) becomes

$$\begin{aligned} \mathcal{P}_{c,k}^{(p)} &= \sqrt{\frac{2}{\pi}} \int_{-\infty}^\infty [1 - Q(y)]^{M-1} \exp \left[-\frac{y^2}{2} \right] \frac{\sqrt{\pi}}{2} \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)}} \frac{\gamma_l^{(p)}}{\gamma_l^{(p)} + 1} \\ &\quad \cdot \exp \left[\frac{\gamma_l^{(p)} y^2}{2(\gamma_l^{(p)} + 1)} \right] \operatorname{erfc} \left(1, -y \sqrt{\frac{\gamma_l^{(p)}}{2(\gamma_l^{(p)} + 1)}} \right) dy \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)} + 1} \\
&\quad \cdot \exp \left[-\frac{y^2}{2(\gamma_l^{(p)} + 1)} \right] \operatorname{erfc} \left(1, -y \sqrt{\frac{\gamma_l^{(p)}}{2(\gamma_l^{(p)} + 1)}} \right) dy
\end{aligned}$$

The BER for multistage PIC in unequal power multipath system is derived as

$$\begin{aligned}
\mathcal{P}_{c,k}^{(p)} &= \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} [1 - Q(y)]^{M-1} \sum_{l=1}^{L_k} \frac{A_l^{(p)}}{\gamma_l^{(p)} + 1} \\
&\quad \cdot \exp \left[-\frac{y^2}{2(\gamma_l^{(p)} + 1)} \right] \operatorname{erfc} \left(1, -y \sqrt{\frac{\gamma_l^{(p)}}{2(\gamma_l^{(p)} + 1)}} \right) dy \\
\mathcal{P}_{b,k}^{(p)} &= \frac{M}{2(M-1)} \mathcal{P}_{e,k}^{(p)} = \frac{M}{2(M-1)} (1 - \mathcal{P}_{c,k}^{(p)}) \tag{4.17}
\end{aligned}$$

The approximation (4.7) still applies here, with μ_x and σ_x changed to

$$\begin{aligned}
\mu_x &= \mathbb{E}[x] = \int_0^{\infty} xp(x)dx = \sum_{l=1}^{L_k} \frac{2A_l^{(p)}}{\gamma_l^{(p)}} \int_0^{\infty} x^2 \exp(-x^2/\gamma^{(p)})dx \\
&= 2\sqrt{\pi} \operatorname{erfc}(2, 0) \sum_{l=1}^{L_k} A_l^{(p)} \sqrt{\gamma_l^{(p)}} \\
\mathbb{E}[x^2] &\approx \mathbb{E} \left[\frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} |h_{k,l}|^2 \right] = \frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} \mathbb{E}[|h_{k,l}|^2] = \frac{N^2}{(\alpha^2)^{(p)}} \sum_{l=1}^{L_k} P_{k,l} \\
\sigma_x &= \sqrt{\mathbb{E}[x^2] - \mu_x^2} \tag{4.18}
\end{aligned}$$

4.4 Analytical Results and Performance Comparison

Comparison between analysis and simulation is presented in this section. Figure 4.2 and 4.3 show the comparison between analytical and simulated results for a 15-user system. For simplicity, the simulated system is assumed to be chip-synchronous, i.e., all path delays are assumed to be multiples of T_c . However, the system is asynchronous on the symbol level. Perfect slow power control is assumed in the sense that $P_k = \sum_{l=1}^{L_k} P_{k,l}$, the average received power, is equal for all users. Different paths are assumed to have equal gain and the channel coefficients are normalized so that each user has unity gain, i.e., $P_{k,1} = P_{k,2} = \dots = P_{k,L_k}$ and $P_k = \sum_{l=1}^{L_k} P_{k,l} = 1$. The

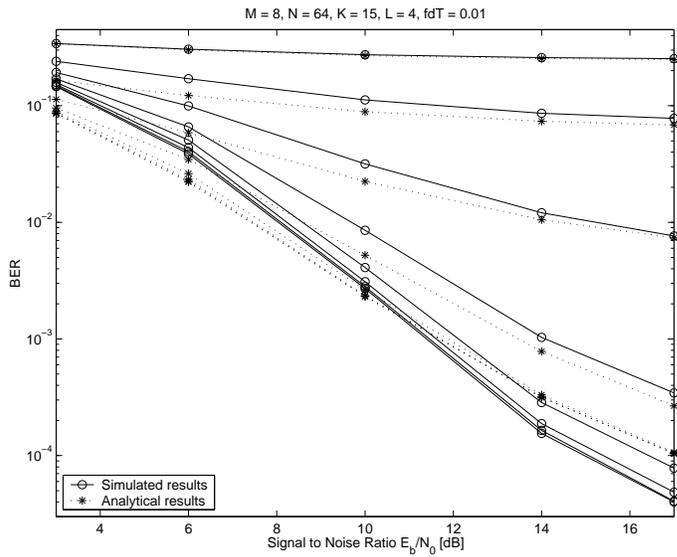
number of multipath channels L_k is set to be 4, ($L_k = L = 4$) for all k . The simulated PIC performance in Figure 4.2.b) and 4.3.b) is derived assuming perfect knowledge of the complex channel gains, e.g., the genie-aided case. We observe that the analysis obtained by (4.6) is more accurate for the genie-aided PIC and the analysis obtained by the approximation expressed by (4.7) is more accurate for the PIC scheme with channel estimation (CE). Since channel information has to be estimated in reality, the genie-aided case is not much of practical interest, we therefore focus on the PIC with CE and use (4.7) for the following analysis and comparisons.

Figures 4.4 – 4.7 show the comparison between analytical and simulated results for different number of users. The simulated curves precisely match the theoretical ones for the first noncoherent stage, which proves that Gaussian approximation is accurate to model MAI and ISI sequences as well as the elements of each interference sequence in long-code systems. The analysis starts to deviate slightly from simulations, but is still fairly accurate, after the first noncoherent stage. The theoretical analysis is a little pessimistic when the system is too lightly loaded, and a little optimistic when the system is too heavily loaded. From both simulation and analysis, one can observe that it takes PIC more stages to converge as K increases (the system becomes more heavily loaded). Seven stages (excluding the first noncoherent stage) ought be enough for the system to reach convergence in any case.

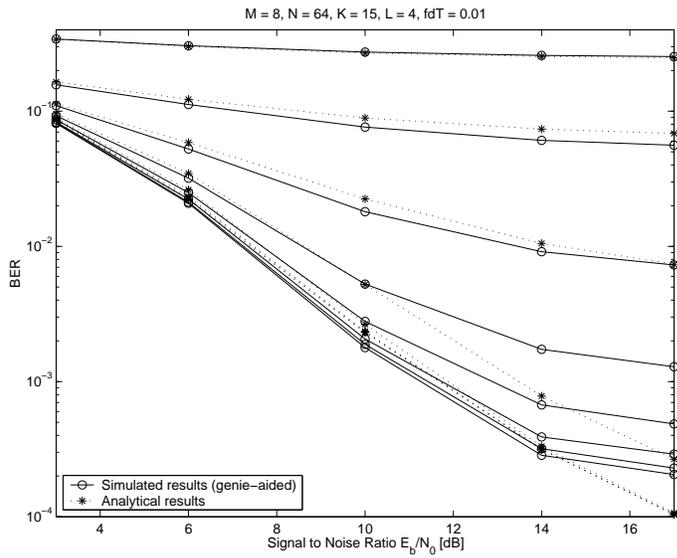
The convergence property of MAI elimination is studied analytically in Figure 4.8 for a 18-user system which is almost fully loaded considering the spreading factor equals $64/3$. We find that the variance of MAI approaches its limit quickly. Only 5 stages are required to achieve the desired performance. The interference can be more thoroughly removed at each iteration at high SNRs. A reasonable level of SNR therefore needs to be maintained in order to benefit from the PIC iteration process. The plot also indicates that MAI cannot be totally removed, thus the receiver cannot achieve single user performance in a heavily loaded system. Those findings agree with previous analysis and simulation results.

System capacity is illustrated in Figure 4.9 by plotting BER as a function of the number of users using both analytical and simulated results. It is clearly shown that analysis is in close agreement with simulation for BER above 10^{-4} . However, the analysis tends to over-estimate the MAI when the number of users is very small. Conversely, the MAI is under-estimated when there are too many active users. Compared with the topmost curve which represents the first noncoherent stage, the subsequent PIC stages significantly increase system capacity and BER performance as indicated by both analysis and simulation.

In Figure 4.10, we analyze the PIC with different degree of diversity (different number of paths). It can be seen that the system performance degrades for the first stage as the degree of diversity increases. The rea-

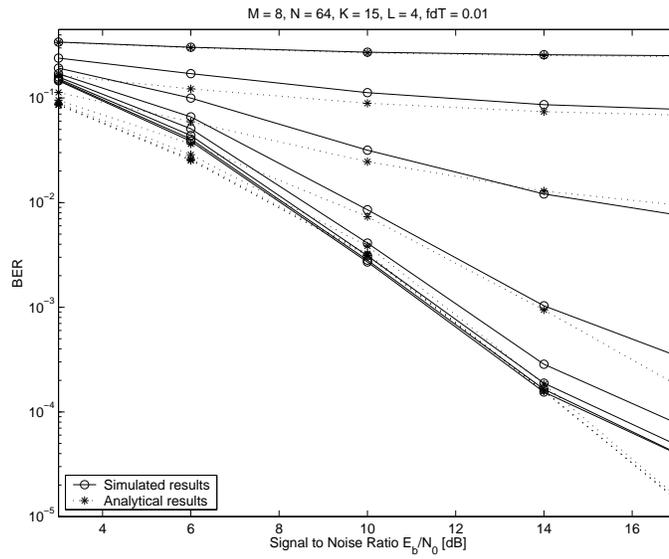


(a) Analysis vs. PIC with CE.

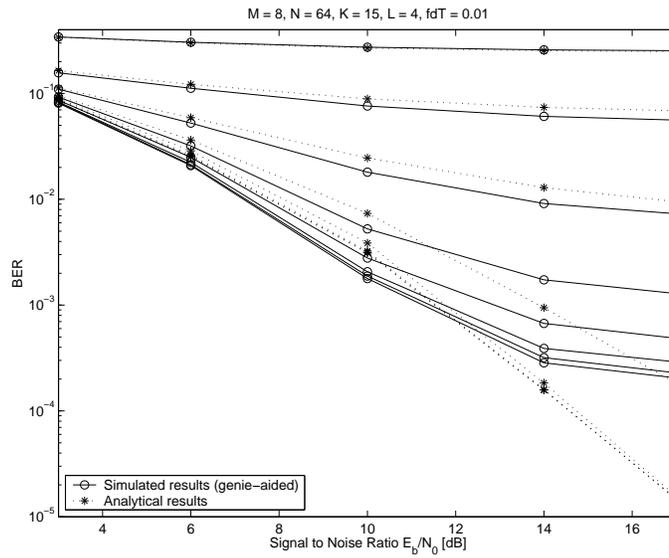


(b) Analysis vs. genie-aided PIC.

Figure 4.2: Analysis vs. simulation. Analytical BER is derived by (4.6). The number of users is $K = 15$. Topmost curve represents noncoherent first stage and the second curve from top represents the first stage PIC, the bottommost curve represents the 7th stage PIC.



(a) Analysis vs. PIC with CE.



(b) Analysis vs. genie-aided PIC.

Figure 4.3: Analysis vs. simulation. Analytical BER is derived by (4.7). The number of users is $K = 15$. Topmost curve represents noncoherent first stage and the second curve from top represents the first stage PIC, the bottommost curve represents the 7th stage PIC.

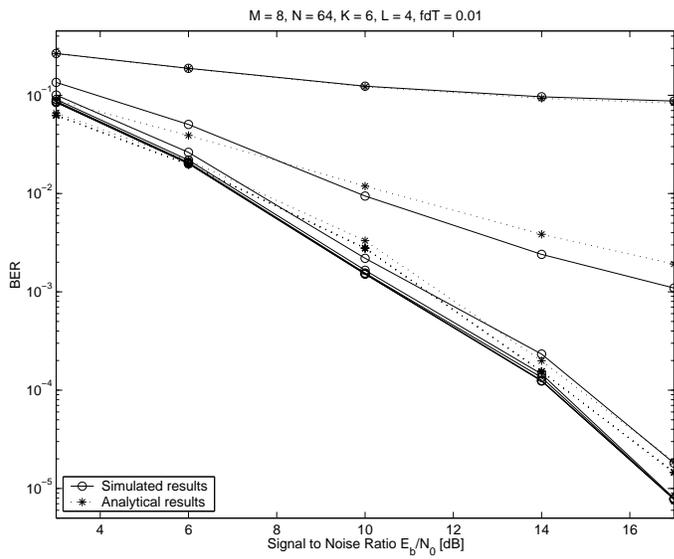


Figure 4.4: Analysis vs. simulation, $K = 6$.

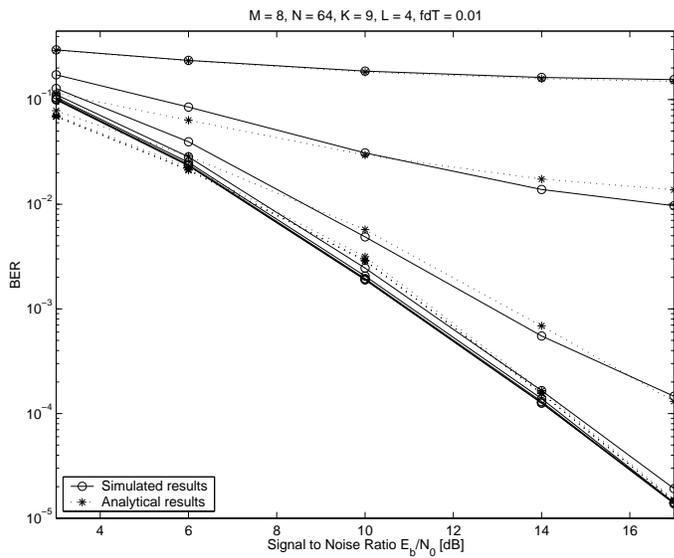


Figure 4.5: Analysis vs. simulation, $K = 9$.

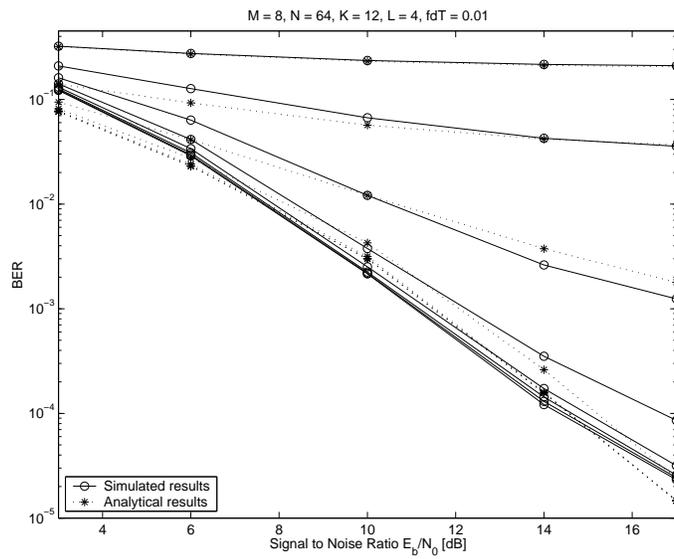


Figure 4.6: Analysis vs. simulation, $K = 12$.

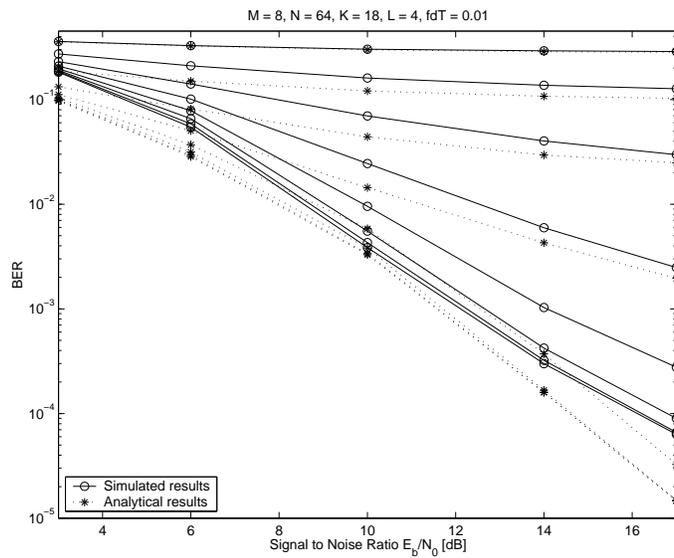


Figure 4.7: Analysis vs. simulation, $K = 18$.

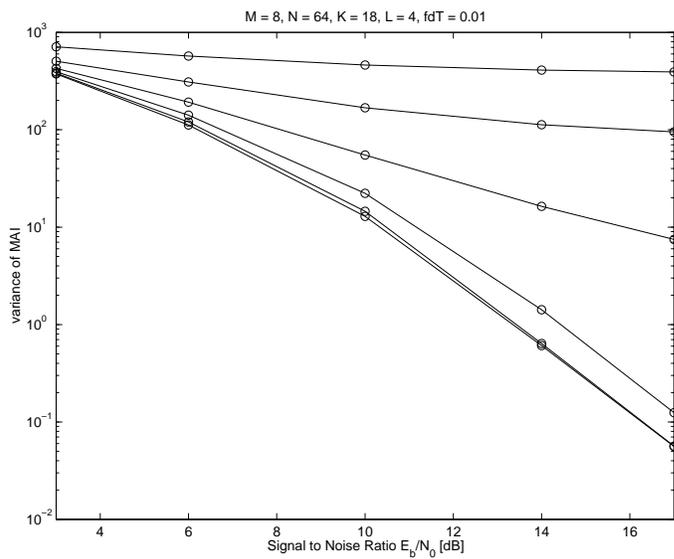


Figure 4.8: Analytical results for MAI variance at each stage of PIC.

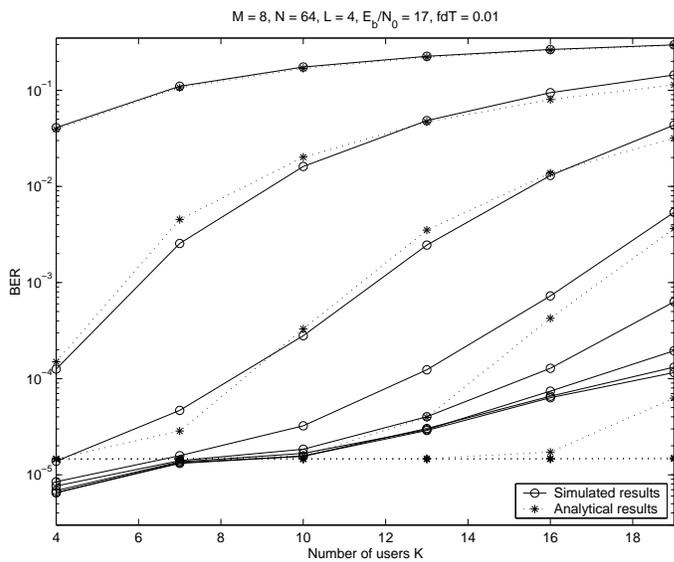
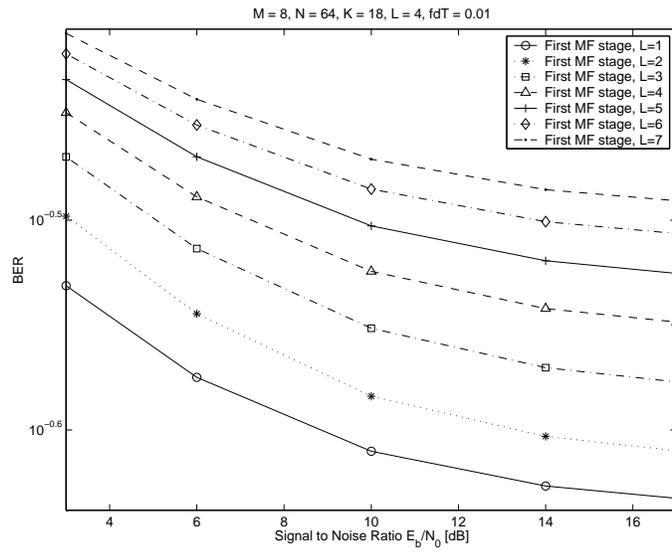
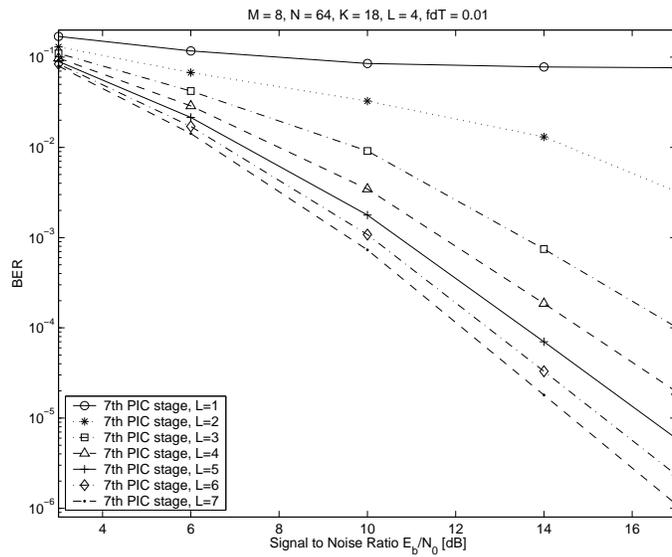


Figure 4.9: Analysis vs. simulation, system capacity with 7-stage PIC.



(a) Analytical BER for noncoherent first stage.

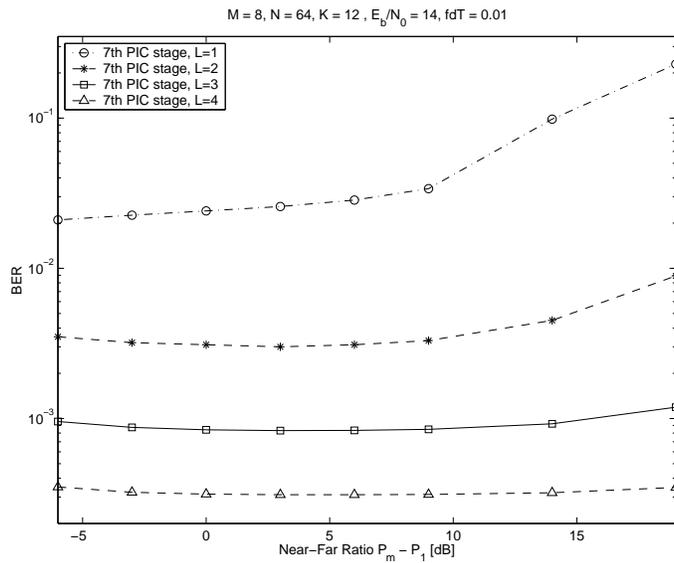
(b) Analytical BER for 7th PIC stage.**Figure 4.10:** Diversity gains achieved by PIC.

son is that with a noncoherent MF receiver, the interference is dominant and the multipath combining gain is not sufficient to compensate for the increased interference as the number of paths increases. However, for the following coherent PIC stages, the conclusion is opposite. The interference is effectively removed and the multipath gain becomes dominant. Furthermore, the cancellation residual and noise present in the imaginary part of the decision statistic are eliminated. Another discovery is that the first few taps exhibit big performance gain compared to single-path case, while the multipath gain gradually diminishes as the number of paths increases.

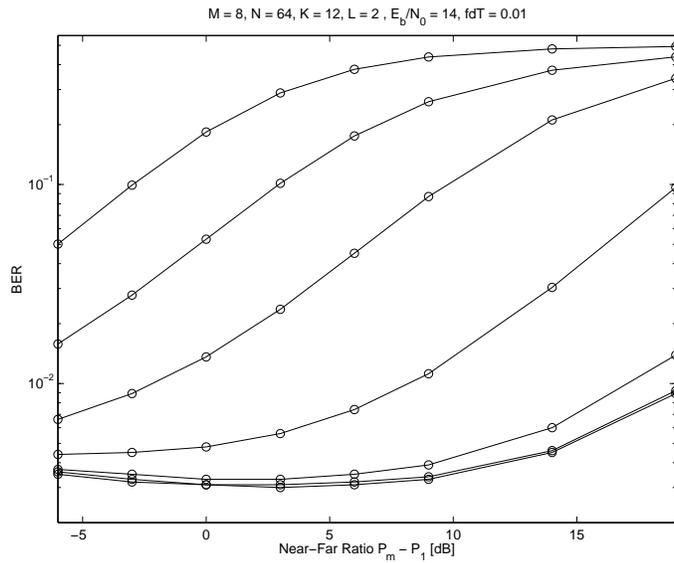
Ideal power control is assumed in the above discussion. The near-far robustness of the PIC algorithm is analytically examined in Figure 4.11 by plotting the resulting BER as a function of near-far ratio, which refers to the difference between the power of each of interfering user (it is assumed that $P_2 = P_3 = \dots = P_m = \dots = P_K$), and the power of the desired user P_1 (the first user is the user of interest). From Figure 4.11.a), we see that the PIC scheme in general is not sensitive to the variations in the interfering signal strengths and is near-far resistant. The only exception is for the single-path system in severe near far situation (when $P_m - P_1 > 10$ dB, i.e., the desired user is much weaker than the other interfering users), the system performance degrades. This concurs with the results shown in [36]. Figure 4.11.b) shows that the near-far robustness of the PIC scheme comes from interference cancellation process. The initial few stages do exhibit some degree of near-far problem, which will gradually vanish as the iteration goes on and the system reaches convergence. The rationale is that the error probability for strong interfering users is very low due to their high signal power level, we therefore have better chance to make correct cancellation and cancel their contributions, which greatly alleviates the near-far effect.

The performance of the PIC algorithm in presence of unequal power among different diversity branches is studied in Figure 4.12 for a 4-path channel. We use the analytical results (4.17) derived in Section 4.3 as well as its approximation expressed by (4.7) and (4.18). In this test, power control is assumed so that the average received power is equal for all users and each user has unity gain. However, the power difference between different paths is set to be $\Delta P_{k,l} = P_{k,4} - P_{k,3} = P_{k,3} - P_{k,2} = P_{k,2} - P_{k,1} = 0, 3, 6$ dB, respectively. Figure 4.12 shows that the PIC works the best when all the branches have equal power, i.e., when $\Delta P_{k,l} = 0$. The bigger deviation in power, the worse performance it gets.

The above analysis and comparisons are based on the assumption of chip synchronism. The system performance is compared between a chip-synchronous system and a chip-asynchronous system in Figure 4.13 with analytical approach. As expected, the latter system poses less interference and therefore has better performance. Chip synchronism does represent a worse-case interference scenario as stated in [25].

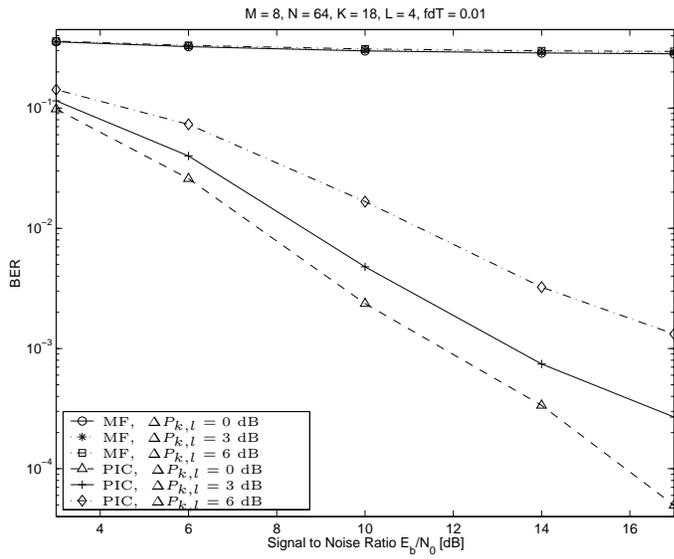


(a) Analytical BER vs. NFR with different number of paths.

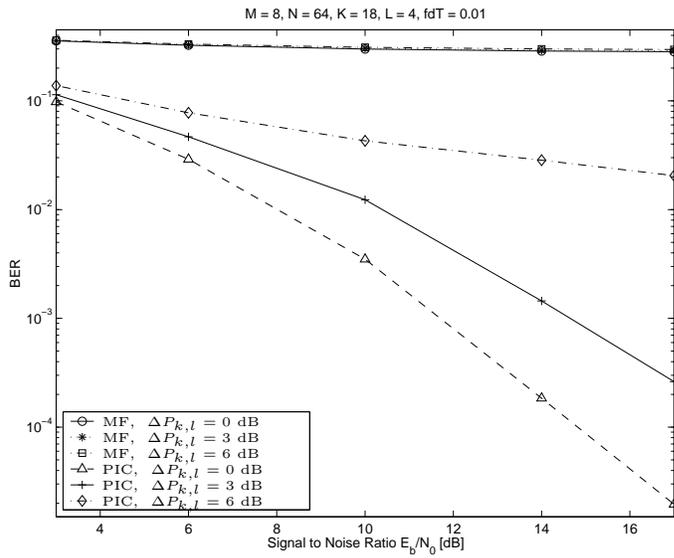


(b) Analytical BER vs. NFR at different stages of PIC.

Figure 4.11: Near-far effect for PIC.



(a) BER derived by (4.17).



(b) BER derived by (4.7) and (4.18).

Figure 4.12: PIC performance for unequal power diversity branches. MF curves represent the first noncoherent stage. PIC curves are plotted for the 7th stage.

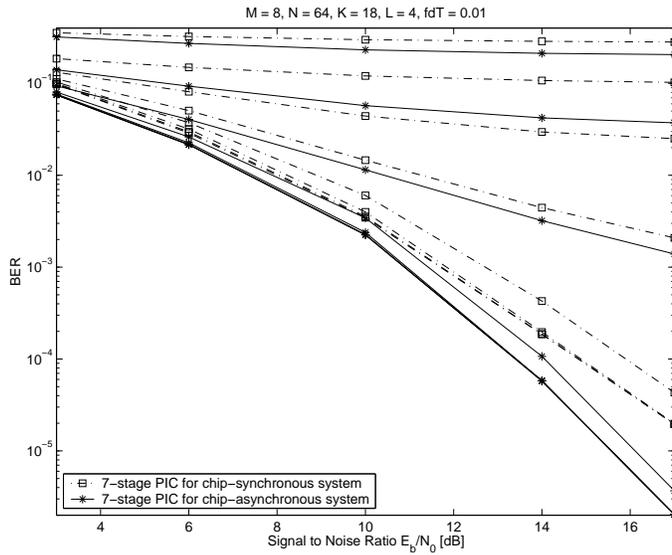


Figure 4.13: Chip-synchronism vs. chip-asynchronism.

4.5 Conclusions

BER performance of the multistage PIC scheme is theoretically analyzed in this chapter. We use the Central Limit Theorem to model MAI and ISI as Gaussian random processes. Comparison with the simulated results shows that the analysis is fairly accurate. A simplified method is also presented using only the mean and variance of SINR, leading to accurate approximations.

A close agreement is seen between analysis and simulation in most cases except for low BER (below 10^{-4}). The analysis tends to over estimate MAI in very lightly loaded systems, and under estimate MAI in very heavily loaded system. Considering the fact that the target BER for an uncoded system is usually above 10^{-4} , our analytical results are quite satisfactory. The presented analytical method provides an effective measure to predict BER performance and system capacity for the PIC scheme under investigation.

The PIC convergence property and multipath diversity gains are studied analytically. It is shown that the multistage PIC receiver effectively removes interference, the variance of MAI is reduced to asymptotic values with only a few stages of cancellation. It is also shown that multipath diversity gains can be achieved by the subsequent coherent stages rather than the first noncoherent stage. Interference cancellation and coherent combining are important techniques to combat MAI and multipath propagation.

Finally, the near-far effect of the PIC scheme and its performance in

presence of unequal power among different paths are examined using the analytical approach. The study shows that the PIC is near-far resistant. It can be used in practical systems even when strict power control is hard to obtain. We also learned that the PIC scheme achieves the best performance in presence of equal power among different diversity branches.

The performance of a communication system depends largely on its ability to retrieve an accurate measurement of the underlying channel. In this chapter, we present joint approach to data detection and channel estimation. The purpose of channel estimation is to enable M -ary orthogonal signals to be demodulated coherently and a Rake receiver to use a maximum ratio combining (MRC) scheme. For data detection, we mainly consider the use of interference cancellation technique which is suitable for CDMA systems with long codes. Different channel estimation schemes are evaluated and compared in terms of mean square error (MSE) of the channel estimation and the bit error rate (BER) performance. Based on our analysis and numerical results, some recommendations are made on how to choose appropriate channel estimators in practical systems.

5.1 Introduction

In addition to multiple access interference (MAI), CDMA systems also suffer from multipath fading. Mobile radio communication channels are time-varying channels. They are characterized by the presence of both delay and Doppler spread. Depending on the delay spread and the data rate, the channel may be approximately flat fading or frequency-selective fading. The latter one produces intersymbol interference (ISI). The received signal includes multiple versions of the transmitted waveform which are attenuated and delayed in time.

Accurate knowledge (or good estimate) of the underlying channel is essential for mitigating interference and the effect of multipath and fading. If the channel estimates are not reliable, the performance of algorithms such

as multiuser detectors and coherent Rake receivers degrade significantly. Channel estimation consequently is an important issue in mobile communications and good channel estimates have a very important impact on the overall performance of the system.

Several channel estimators, e.g., subspace-based estimators, and maximum likelihood estimators have been proposed, e.g., in [37, 38]. The authors only considered the use of short spreading codes. However, current and next generation CDMA systems use long spreading codes whose period is much larger than the symbol duration. For long code CDMA, several attempts have been made in obtaining channel estimates. In [39], a subspace-based algorithm for blind channel estimation of a synchronous CDMA downlink was proposed. It was shown that the estimation accuracy can be increased considerably using a decision feedback approach. However, a time invariant multipath channel was assumed in [39]. The time-varying nature of the fading channel prohibits the use of subspace algorithm, since the received signal is not constrained to any particular subspace if channel parameters are time-varying.

The estimation of channel parameters in a DS-CDMA system with M -ary orthogonal modulation, which is the main concern of this thesis, has been the subject of study in several papers (see, for instance, [16, 18, 19, 40]). The maximum likelihood (ML) channel estimator for long code CDMA systems over time-varying multipath channels was employed in [19]. In [16], a blind channel estimation strategy based on an adaptive Wiener filtering approach that yields unbiased channel estimates and low estimation variance for CDMA system using random codes was proposed. In [18, 40], it was shown that the channel parameters can be estimated with a maximum correlator output. Furthermore, the estimated parameters are used for the interference canceler with coherent detection, which results in an increase in system capacity.

Recall that the received signal vector is formed as $\mathbf{r} = \mathbf{A}\mathbf{h} + \mathbf{n}$. The task of a channel estimator is to estimate the fading vector \mathbf{h} given the received observation \mathbf{r} and the transmitted data. Depending on the form of the data that can be retrieved, channel estimation can be either decision directed or pilot aided. The former uses decision feedback loops and utilizes the decisions on the transmitted signals $\hat{\mathbf{A}}$ to extract the channel coefficients. The second approach makes the use of pilot symbols or channels (\mathbf{A} is known in this case). The use of pilots simplifies channel estimation with the penalty of wasting channel resources. In this chapter, we focus on the first approach and make an extensive investigation on different alternatives for estimating time-varying multipath Rayleigh fading channels in absence of pilot symbols.

We take an integrated approach such that channel estimation is coupled with data detection. All the channel estimators are decision-directed and can work in conjunction with the coherent data detection (interference

cancellation in particular). We emphasize on how to utilize the measured channel for detecting the transmitted data and how to use the detected data to enhance the channel estimation. The principle is that the accuracy of channel estimation depends on the accuracy of the data detection and vice versa. The channel is better estimated when the transmitted data are more accurately detected, the performance is improved by repeating this process in an iterative manner. To further improve the system performance, channel estimation using soft information is also proposed.

Depending on whether prior knowledge about the parameters to be estimated is used, statistical estimation can be conducted either with classical approach or Bayesian approach. Both methods are discussed in this chapter. The maximum likelihood channel estimator is derived based on the classical approach. In this case, the channel vector \mathbf{h} is assumed to be deterministic and unknown. While with a Bayesian approach, \mathbf{h} is assumed to be a random vector whose particular realization needs to be estimated. It improves the estimation accuracy by exploiting some prior knowledge about \mathbf{h} . The Bayesian philosophy leads to the linear MMSE estimator and the Kalman filter for channel estimation. Those channel estimation algorithms will be presented below.

5.2 Maximum Likelihood Channel Estimator

Given an estimate of the data matrix $\hat{\mathbf{A}}$ in (2.2), then \mathbf{h} can be computed as [19]

$$\begin{aligned}\hat{\mathbf{h}} &= \hat{\mathbf{A}}^\dagger \mathbf{r} \\ \hat{\mathbf{A}}^\dagger &= (\hat{\mathbf{A}}^* \hat{\mathbf{A}})^{-1} \hat{\mathbf{A}}^*\end{aligned}$$

where $\hat{\mathbf{A}}^\dagger$ denotes the left pseudoinverse of $\hat{\mathbf{A}}$ (assuming \mathbf{A} has full column rank). In case of correct decisions, i.e., $\hat{\mathbf{A}} = \mathbf{A}$, then

$$\hat{\mathbf{h}} = \mathbf{A}^\dagger \mathbf{r} = \mathbf{A}^\dagger (\mathbf{A} \mathbf{h} + \mathbf{n}) = \mathbf{h} + \mathbf{A}^\dagger \mathbf{n}$$

which is an unbiased estimate of \mathbf{h} . In Appendix 5.9.1, it is also proved that the ML estimator with $\hat{\mathbf{A}} = \mathbf{A}$ is an efficient estimator which attains the Cramer-Rao lower bound (CRLB).

This procedure will suffer from a so-called dimensionality problem. When the total number of paths of all the users is greater than the number of chips in the vector $\mathbf{r}(k, j)$, i.e., $L_{\text{tot}} \geq N$ ($L_{\text{tot}} = \sum_{k=1}^K L_k$), the matrix $\hat{\mathbf{A}}$ will not have full column rank and the above mentioned procedure will become useless. The problem can be resolved by stacking several $\mathbf{r}(j)$ vectors on top of each other and assume the channel remains static during several symbol intervals. In particular, suppose $\mathbf{h}(j) \approx \mathbf{h}(j+1)$, we can

then write

$$\begin{aligned} \begin{bmatrix} \mathbf{r}(j) \\ \mathbf{r}(j+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}(j) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(j+1) \end{bmatrix} \begin{bmatrix} \mathbf{h}(j) \\ \mathbf{h}(j+1) \end{bmatrix} + \begin{bmatrix} \mathbf{n}(j) \\ \mathbf{n}(j+1) \end{bmatrix} \\ &\approx \begin{bmatrix} \mathbf{A}(j) \\ \mathbf{A}(j+1) \end{bmatrix} \mathbf{h}(j) + \begin{bmatrix} \mathbf{n}(j) \\ \mathbf{n}(j+1) \end{bmatrix} \end{aligned}$$

The ML channel estimation algorithm using hard decision of the transmitted data can be reformulated as

$$\hat{\mathbf{h}}^{\text{ML}}(j) = \begin{bmatrix} \hat{\mathbf{A}}(j) \\ \hat{\mathbf{A}}(j+1) \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{r}(j) \\ \mathbf{r}(j+1) \end{bmatrix}, \quad (5.1)$$

which will produce usable estimates as long as $2N_k > L_{\text{tot}}$. Obviously, this scheme can be extended further by stacking several $\mathbf{r}(k, j)$ vectors on top of each other. In the derivation of CRLB in Appendix 5.9.1, we see that stacking also has the effect of noise averaging and tends to reduce the error of the channel estimation. However, for relatively fast fading channels, the stacking may have opposite effect and reduce the quality of the estimates.

5.3 Linear MMSE Channel Estimator

The linear MMSE algorithm for channel estimation computes a matrix \mathbf{W} , which is chosen to minimize the mean square error $E[\|\mathbf{h} - \mathbf{W}^* \mathbf{r}\|^2]$. The optimum matrix of \mathbf{W} under the MMSE criterion can be computed as

$$\begin{aligned} \mathbf{W}_{\text{MMSE}} &= \arg \min_{\mathbf{W}} E[\|\mathbf{h} - \mathbf{W}^* \mathbf{r}\|^2] = \mathbf{R}^{-1} \Phi \\ \mathbf{R} &= E[\mathbf{r} \mathbf{r}^*] = E[(\mathbf{A} \mathbf{h} + \mathbf{n})(\mathbf{A} \mathbf{h} + \mathbf{n})^*] \\ &= \mathbf{A} \mathbf{P} \mathbf{A}^* + N_0 \mathbf{I} \\ \Phi &= E[\mathbf{r} \mathbf{h}^*] = E[(\mathbf{A} \mathbf{h} + \mathbf{n}) \mathbf{h}^*] = \mathbf{A} \mathbf{P} \\ \mathbf{P} &= E[\mathbf{h} \mathbf{h}^*] = \text{diag}(P_{1,1}, P_{1,2}, \dots, P_{k,l}, \dots, P_{K,L_K}) \end{aligned}$$

where $P_{k,l}$ is the average received power from the k^{th} user's l^{th} path. Combining the above equations, the linear MMSE estimate of \mathbf{h} can be formulated as

$$\mathbf{h}^{\text{LMMSE}} = \mathbf{W}_{\text{MMSE}}^* \mathbf{r} = \Phi^* \mathbf{R}^{-1} \mathbf{r} = \mathbf{P}^* \mathbf{A}^* (\mathbf{A} \mathbf{P} \mathbf{A}^* + N_0 \mathbf{I})^{-1} \mathbf{r} \quad (5.2)$$

From (5.2), we see that the signal and noise power level $P_{k,l}$ and N_0 must be known or estimated to carry out LMMSE channel estimation. It is worth noticing that \mathbf{A} does not need to be full rank to ensure the

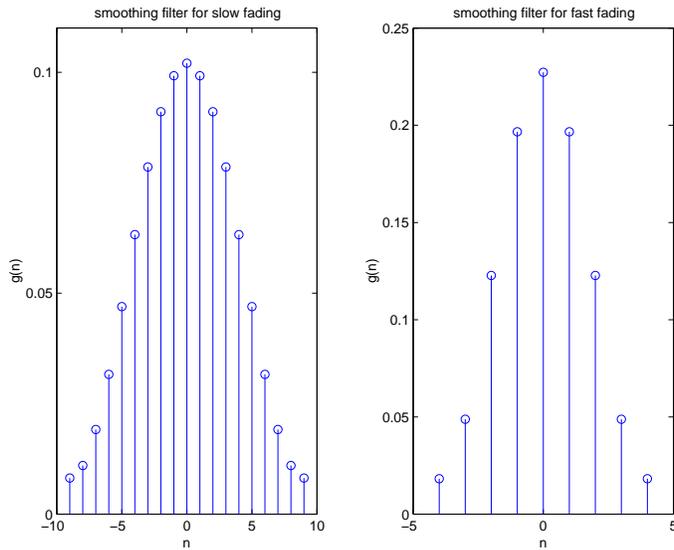


Figure 5.1: Smoothing filter impulse response.

invertibility of $(\mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I})$. Since we use a decision directed approach for channel estimation, the transmitted data \mathbf{A} is unknown, and has to be replaced by its estimate $\hat{\mathbf{A}}$ in the above equation.

It is shown in Appendix 5.9.2 that the Bayesian Gauss-Markov Theorem leads to the same LMMSE estimator. Furthermore, the estimation covariance matrix is also derived there.

The fading processes are lowpass in nature with bandwidths that are determined by the Doppler frequency. We should therefore be able to improve the estimates by lowpass filtering (smoothing). A simple smoothing procedure is to feed $\hat{\mathbf{h}}(j)$ through an FIR filter with impulse response $g(n)$ [19], which yields the smoothed channel gain vector $\bar{\mathbf{h}}(j)$ as

$$\bar{\mathbf{h}}(j) = \sum_{k=j-N_s}^{j+N_s} \hat{\mathbf{h}}(k)g(j-k). \quad (5.3)$$

Experiments indicate that exact form of the filter is not crucial. In our simulations, we use a smoothing filter derived from a Hamming window of length $2N_s + 1 = 19$ or 9 (for slow or fast fading channel), normalized such that $\sum_{k=-N_s}^{N_s} g(k) = 1$. The impulse response of the smoothing filters are plotted in Figure 5.1. As will be evidenced by numerical results, smoothing operation really improves the results of ML and LMMSE channel estimates, and subsequently leads to more reliable data detection.

5.4 First Order Kalman Filter

If \mathbf{h} and its statistics are completely unknown, the ML estimator is the best we can do. However, we know that channel gains are correlated in time. One way to exploit this knowledge is to use smoothing filter to improve estimation results as introduced above. An alternative approach is to consider \mathbf{h} to be a realization of a random process to be estimated in the development of channel estimation algorithm, which leads to the use of, e.g., a Kalman filter. The Kalman filter is a time-varying linear filter widely used in statistical estimation when the unknown parameters evolve in time according to some dynamic model [41]. It has been used, e.g., in [42, 43, 44, 45] for the purpose of channel estimation. Here, we extend the use of Kalman filter to estimation of multipath fading channels for orthogonal modulated CDMA systems.

We assume the channels are independently Rayleigh fading channels with the Clarke's power spectral density given by

$$S(f) = \begin{cases} \frac{K}{\sqrt{(1-(f/f_d)^2)}}, & |f| < f_d \\ 0, & |f| \geq f_d \end{cases} \quad (5.4)$$

where f_d is the maximum Doppler frequency. The channel gain $h_{k,l}(t)$ is a complex circular Gaussian process with autocorrelation function $E[h_{k,l}^*(t)h_{k,l}(t+\tau)] = P_{k,l}J_0(2\pi f_d\tau)$ and $J_0(x)$ is the zeroth order Bessel function of the first kind. Based on the relations

$$\int_{-f_d}^{f_d} \frac{K}{\sqrt{1-(f/f_d)^2}} df = \int_{-1}^1 \frac{K f_d}{\sqrt{1-(f/f_d)^2}} d(f/f_d) = P_{k,l}$$

$$\int \frac{dx}{\sqrt{1-x^2}} = \arcsin$$

we can readily derive the constant $K = \frac{P_{k,l}}{\pi f_d}$. The power spectrum of Clarke's model is illustrated in Fig. 5.2.

The power spectral density expressed in (5.4) can be approximated by a finite order autoregressive (AR) process. As illustrated in the left diagram of Fig. 5.3, the correlation between channel coefficients can simply be approximated by the following first-order vector Gauss-Markov model as

$$\mathbf{h}(j) = \mathbf{F}\mathbf{h}(j-1) + \mathbf{u}(j)$$

where $\mathbf{F} \in \mathbb{R}^{L_{\text{tot}} \times L_{\text{tot}}}$ is the state transition matrix and $\mathbf{u}(j) \in \mathbb{C}^{L_{\text{tot}}}$ is the driving noise vector (WGN sequence) with $E[\mathbf{u}(j)] = \mathbf{0}$ and covariance matrix \mathbf{Q} . The current input $\mathbf{h}(j)$ depends only on the state of the system at the previous symbol time $\mathbf{h}(j-1)$ and the current input $\mathbf{u}(j)$. The state $\mathbf{h}(j-1)$ accumulates the effect of all past inputs to the system.

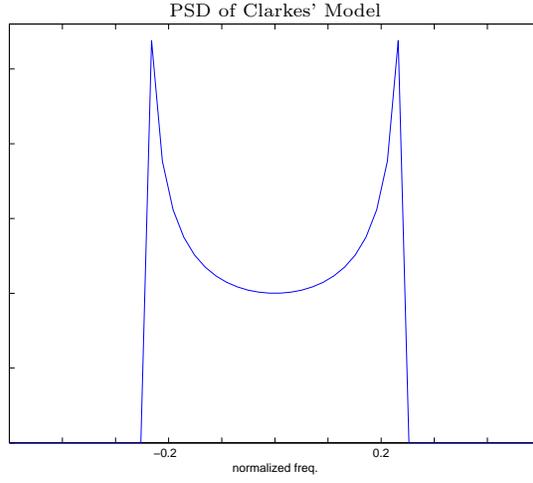


Figure 5.2: Power spectrum density of the Clarke's model. The normalized Doppler frequency is $f_d T = 0.25$.

Since different paths are uncorrelated with each other, we can let \mathbf{F} and \mathbf{Q} be diagonal matrices, i.e., $\mathbf{F} = \text{diag}(f_{1,1}, f_{1,2}, \dots, f_{k,l}, \dots, f_{K,L_K})$ and $\mathbf{Q} = \text{diag}(\sigma_{1,1}^2, \sigma_{1,2}^2, \dots, \sigma_{k,l}^2, \dots, \sigma_{K,L_K}^2)$. Using the notation $h_{k,l}[j] = h_{k,l}(jT)$, the vector Gauss-Markov model becomes L_{tot} independent scalar models: $h_{k,l}[j] = f_{k,l} h_{k,l}[j-1] + u_{k,l}[j]$. By solving the equations

$$\begin{aligned} \mathbb{E}\{h_{k,l}[j]h_{k,l}^*[j-1]\} &= f_{k,l}P_{k,l} + \mathbb{E}\{u_{k,l}[j]h_{k,l}^*[j-1]\} \\ \mathbb{E}\{|h_{k,l}[j]|^2\} &= f_{k,l}^2\mathbb{E}\{|h_{k,l}[j-1]|^2\} + \mathbb{E}\{|u_{k,l}[j]|^2\} \\ &= f_{k,l}^2P_{k,l} + \sigma_{k,l}^2 \end{aligned}$$

we obtain the coefficients for the first-order model

$$\begin{aligned} f_{k,l} &= \mathbb{E}\{h_{k,l}^*[j]h_{k,l}[j-1]\}/P_{k,l} = J_0(2\pi f_d T) \\ \sigma_{k,l}^2 &= P_{k,l}(1 - f_{k,l}^2) \end{aligned}$$

Note that the maximum Doppler frequency f_d can differ from path to path. As a special case, when all the paths from different users have the same received power and Doppler frequency, the matrices \mathbf{F} , \mathbf{Q} are simplified to $\mathbf{F} = f\mathbf{I}$, $\mathbf{Q} = \sigma_Q^2\mathbf{I}$. Some training method was proposed in [42] to estimate the state transition matrix \mathbf{F} , also the algorithm for tracking Doppler shift is addressed in [44]. Here we focus on the estimation of complex channel gains and assume that the Doppler shift is known after a training phase.

Recall that our observation vector (measurement model) is, from (2.2), $\mathbf{r}(j) = \mathbf{A}(j)\mathbf{h}(j) + \mathbf{n}(j) \in \mathbb{C}^N$, where $\mathbf{n}(j) \sim \mathcal{CN}(\mathbf{0}, \mathbf{C})$ and $\mathbf{C} = N_0\mathbf{I}_N$.

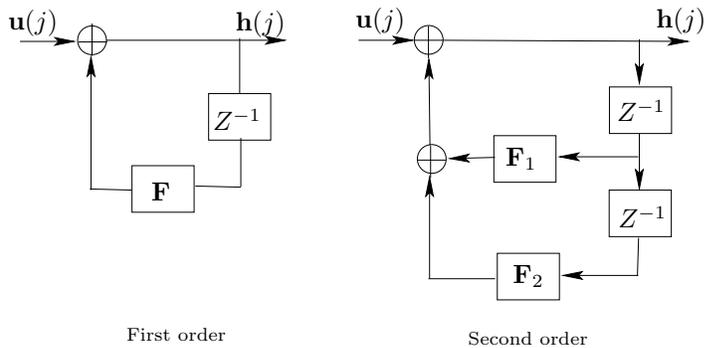


Figure 5.3: First and second order Gauss-Markov system model.

The channel estimate based on the previous received observations $\hat{\mathbf{h}}(j|j) = \mathbb{E}[\mathbf{h}(j)|\mathbf{r}(0), \mathbf{r}(1), \dots, \mathbf{r}(j)]$ can be obtained with the following Kalman recursive estimation procedure [41]

$$\begin{aligned}
 \hat{\mathbf{h}}(j|j-1) &= \mathbf{F}\hat{\mathbf{h}}(j-1|j-1) \\
 \mathbf{M}(j|j-1) &= \mathbf{F}\mathbf{M}(j-1|j-1)\mathbf{F}^T + \mathbf{Q} \\
 \mathbf{K}(j) &= \mathbf{M}(j|j-1)\mathbf{A}(j)^T[\mathbf{C} + \mathbf{A}(j)\mathbf{M}(j|j-1)\mathbf{A}(j)^T]^{-1} \\
 \hat{\mathbf{h}}(j|j) &= \hat{\mathbf{h}}(j|j-1) + \mathbf{K}(j)[\mathbf{r}(j) - \mathbf{A}(j)\hat{\mathbf{h}}(j|j-1)] \\
 \mathbf{M}(j|j) &= (\mathbf{I} - \mathbf{K}(j)\mathbf{A}(j))\mathbf{M}(j|j-1)
 \end{aligned} \tag{5.5}$$

From the above equation, we see that the minimum Bayesian MSE, $\mathbf{M}(j|j-1)$ is computed as an integral part of the estimator. The performance measure of the Kalman filter is therefore different from the estimators discussed earlier. In our experiments, the recursion is initialized by $\hat{\mathbf{h}}(-1|-1) = \mathbf{0}$ and $\mathbf{M}(-1|-1) = 100\mathbf{I}$, reflecting little knowledge of the initial stage. From (5.5), we see that the inversion of an $N \times N$ matrix is required to find the Kalman gain $\mathbf{K}(j)$.

5.5 Second Order Kalman Filter

The Kalman filter employed above is based on the first order AR model for the fading process. Further improvement in the Kalman channel estimator will be possible with increased complexity if a higher order AR model is used. As shown in Fig. 5.4, the second order AR model yields better approximation of the Doppler spectrum (shown in Fig. 5.2) than the first order AR model. The accuracy of the Kalman filter should improve.

As shown in the right diagram of Fig. 5.3, the correlation between chan-

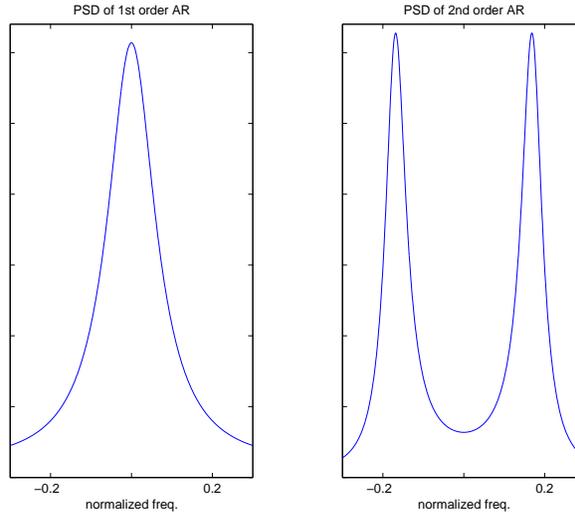


Figure 5.4: Power spectrum density of the AR models.

nel coefficients can be more closely approximated by

$$\mathbf{h}(j) = \mathbf{F}_1 \mathbf{h}(j-1) + \mathbf{F}_2 \mathbf{h}(j-2) + \mathbf{u}(j) \quad (5.6)$$

In both cases shown in Fig. 5.3, a model is established for a vector signal $\mathbf{h}(j)$ as the output of a linear time invariant system (\mathbf{F} , \mathbf{F}_1 , \mathbf{F}_2 are constant matrices) excited by a vector input $\mathbf{u}(j)$.

The second-order vector Gauss-Markov model and measurement model become

$$\begin{bmatrix} \mathbf{h}(j-1) \\ \mathbf{h}(j) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{F}_2 & \mathbf{F}_1 \end{bmatrix} \begin{bmatrix} \mathbf{h}(j-2) \\ \mathbf{h}(j-1) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{u}(j) \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{r}(j-1) \\ \mathbf{r}(j) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(j-1) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(j) \end{bmatrix} \begin{bmatrix} \mathbf{h}(j-1) \\ \mathbf{h}(j) \end{bmatrix} + \begin{bmatrix} \mathbf{n}(j-1) \\ \mathbf{n}(j) \end{bmatrix}$$

For simplicity, we assume all the paths have equal received power and Doppler frequency in the following derivation. However, the extension to unequal power and Doppler frequency is straightforward. Now, the covari-

ance matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_Q^2 \mathbf{I} \end{bmatrix}$ and the state of the system at the previous

symbol time is $\begin{bmatrix} \mathbf{h}(j-2) \\ \mathbf{h}(j-1) \end{bmatrix}$ and the state transition matrix $\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{F}_2 & \mathbf{F}_1 \end{bmatrix}$.

The covariance matrix of noise vector $\begin{bmatrix} \mathbf{n}(j-1) \\ \mathbf{n}(j) \end{bmatrix}$ becomes $\mathbf{C} = N_0 \mathbf{I}_{2N}$.

Since different paths are uncorrelated with each other, we can assume $\mathbf{F}_1 = f_1 \mathbf{I}$, $\mathbf{F}_2 = f_2 \mathbf{I}$, and f_1, f_2, σ_Q^2 can be derived by the relationship $h_{k,l}[j] = f_1 h_{k,l}[j-1] + f_2 h_{k,l}[j-2] + u_{k,l}[j]$. Let us denote

$$\begin{aligned} E\{h_{k,l}^*[j]h_{k,l}[j-1]\} &= J_0(2\pi f_d T)P_{k,l} = J_1 P_{k,l} \\ E\{h_{k,l}^*[j]h_{k,l}[j-2]\} &= J_0(4\pi f_d T)P_{k,l} = J_2 P_{k,l} \end{aligned} \quad (5.7)$$

Then the model parameters can be calculated from

$$\begin{aligned} E\{h_{k,l}[j]h_{k,l}^*[j-1]\} &= f_1 E\{|h_{k,l}[j-1]|^2\} + f_2 E\{h_{k,l}[j-2]h_{k,l}^*[j-1]\} \\ &\quad + E\{u_{k,l}[j]h_{k,l}^*[j-1]\} \implies J_1 = f_1 + f_2 J_1; \\ E\{h_{k,l}[j]h_{k,l}^*[j-2]\} &= f_1 E\{h_{k,l}[j-1]h_{k,l}^*[j-2]\} + f_2 E\{|h_{k,l}[j-2]|^2\} \\ &\quad + E\{u_{k,l}[j]h_{k,l}^*[j-2]\} \implies J_2 = f_1 J_1 + f_2; \\ E\{|h_{k,l}[j]|^2\} &= f_1^2 E\{|h_{k,l}[j-1]|^2\} + f_2^2 E\{|h_{k,l}[j-2]|^2\} + E\{|u_{k,l}[j]|^2\} \\ &\quad + f_1 f_2 E\{h_{k,l}[j-1]h_{k,l}^*[j-2] + h_{k,l}^*[j-1]h_{k,l}[j-2]\} \\ &= (f_1^2 + f_2^2 + 2f_1 f_2 J_1)P_{k,l} + \sigma_Q^2 \end{aligned}$$

By solving the above equations, we obtain

$$\begin{aligned} f_2 &= \frac{J_1^2 - J_2}{J_1^2 - 1}; \\ f_1 &= \frac{J_2 - f_2}{J_1} \\ \sigma_Q^2 &= (1 - f_1^2 - f_2^2 - 2f_1 f_2 J_1)P_{k,l} \end{aligned}$$

The procedure stated in (5.5) also applies to the second order Kalman filter with the following replacements:

$$\begin{aligned} \mathbf{A}(j) &\implies \begin{bmatrix} \mathbf{A}(j-1) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(j) \end{bmatrix}; & \mathbf{F} &\implies \begin{bmatrix} \mathbf{0} & \mathbf{I}_N \\ \mathbf{F}_2 & \mathbf{F}_1 \end{bmatrix} \\ \mathbf{Q} &\implies \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_Q^2 \mathbf{I} \end{bmatrix}; & \mathbf{h}(j) &\implies \begin{bmatrix} \mathbf{h}(j-1) \\ \mathbf{h}(j) \end{bmatrix} \end{aligned}$$

The complexity increase by implementing the second order Kalman filter instead of the first order is mainly due to the inversion of a $2N \times 2N$

($nN \times nN$ for the n^{th} order Kalman filter) matrix required to compute the Kalman gain.

In addition to estimation of time varying multipath coefficients, it was shown in [43] that the extended Kalman filter (EKF) can be used to estimate code delay. In this case, the observation (measurement) sequence is nonlinear in the state variables (the propagation delays), the ordinary Kalman filter equations cannot be used. Thus the observation sequence must be linearized to allow for a recursive estimation procedure using the Kalman filter equations. The EKF is a practical solution to this problem. It utilizes a Taylor's series approximation to obtain a linearized measurement sequence, at which point the ordinary filter equations can be employed.

5.6 Soft Channel Estimator

The ML channel estimation expressed by (5.1) can be reformulated by replacing \mathbf{A} with its soft estimate, which leads to soft version of the ML channel estimator

$$\hat{\mathbf{h}}_{\text{soft}}^{\text{ML}}(j) = \begin{bmatrix} \text{E}[\mathbf{A}(j)|\mathbf{r}(j)] \\ \text{E}[\mathbf{A}(j+1)|\mathbf{r}(j+1)] \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{r}(j) \\ \mathbf{r}(j+1) \end{bmatrix} \quad (5.8)$$

where $\text{E}[\mathbf{A}|\mathbf{r}]$ is soft estimate of \mathbf{A} derived based on (3.14) for MAP-PIC as explained in Section 3.5.1. For the NMIC algorithm, we replace (3.14) with $\text{E}[\mathbf{w}_k^n(j)|\mathbf{r}] = \sum_{m=1}^M b_{k,m} \mathbf{w}_m^n$. The rest of the derivation is the same.

Similarly, the soft version of the LMMSE channel estimator expressed by (5.2) can be formed as

$$\mathbf{h}_{\text{soft}}^{\text{LMMSE}} = \hat{\mathbf{P}}^* \text{E}[\mathbf{A}^*|\mathbf{r}] (\text{E}[\mathbf{A}|\mathbf{r}] \hat{\mathbf{P}} \text{E}[\mathbf{A}^*|\mathbf{r}] + N_0 \mathbf{I})^{-1} \mathbf{r} \quad (5.9)$$

We can also replace $\mathbf{A}(j)$ with $\text{E}[\mathbf{A}(j)|\mathbf{r}(j)]$ in (5.5) to obtain the soft versions of the Kalman filters.

Compare (5.1) vs. (5.8) for ML estimator, and (5.2) vs. (5.9) for LMMSE estimator, one can see the use of soft information in channel estimation itself does not introduce any additional complexity. It is the derivation of soft values that is little more complicated than making hard decisions as discussed earlier.

5.7 Numerical Results

In our simulations, the normalized Doppler frequency is set to $f_d T = 0.1$ in Fig. 5.12, 5.13 when the performance of ML and Kalman filter is examined in the presence of fast fading channels; and it is set to $f_d T = 0.01$ for slow fading channels in the rest of simulations. The system was simulated for

2-path channels, $L_k = L = 2$ or 3-path channels, $L_k = L = 3$ for all k . The spacing between the paths of each user is set to $2T_c$. The number of users is either $K = 8$ or $K = 12$.

Fig. 5.5 shows the original fading channel and the results of channel estimation with different schemes. The original estimates of ML and LMMSE estimators are noisy. By explicitly taking the correlation of the fading channel into account, Kalman filters result in more reliable channel estimates, especially if the fading channel is modeled as second order AR process. We can see from the plots that, after an initial transient, the Kalman filter quickly locks on the true channel values and tracks them closely. The imposition of a correlation constraint prevents the estimate of \mathbf{h} from fluctuating too widely in time. The same effect can be achieved by ML and LMMSE estimator by an additional lowpass filtering stage as shown in the plot at the lower right corner of Fig. 5.5. The quality of the estimated channel is greatly improved after applying the smoothing operation. (LMMSE estimates after smoothing are not shown since they are essentially the same as in the ML case).

Coupled with 5-stage PIC, different channel estimators without the smoothing are assessed and compared in Fig. 5.6 in terms of estimation MSE and BER performance. As expected, ML algorithm has the worst performance. The LMMSE estimator considers the noise effect and slightly improves the estimation results, especially in the low SNR region (the BER performance remains the same though). The Kalman filters take advantage of correlative nature of the fading channel and significantly improve the estimation results and BER performance. The PIC is also simulated with perfect channel estimates, i.e., when $\hat{\mathbf{h}}_k = \mathbf{h}_k$ (which is called genie-aided PIC) to see how close the performance of proposed channel estimators is to the ideal one. It is also evident from the figures that PIC with the second order Kalman filter has the closest performance to genie-aided case. It outperforms the first order Kalman filter with the penalty of higher computational complexity.

Different channel estimators combined with the smoothing filter are compared in Fig. 5.7. One can observe that the ML algorithm yields the smallest MSE and lowest BER after smoothing. The estimation MSE of LMMSE algorithm is a little higher, but the BER performance is essentially the same. The ML is preferable to LMMSE in that it does not need the knowledge of noise variance and the average received power of each signal path. No significant improvement is observed for the two Kalman filters after smoothing. Since the Kalman algorithm itself already takes the correlation into account, the additional smoothing does not make much difference. The ML estimator with channel smoothing appears to be the most favorable choice for estimating slow fading channels.

Based on the above experimental results, we come to important conclusions that the quality of the channel estimation directly translates into

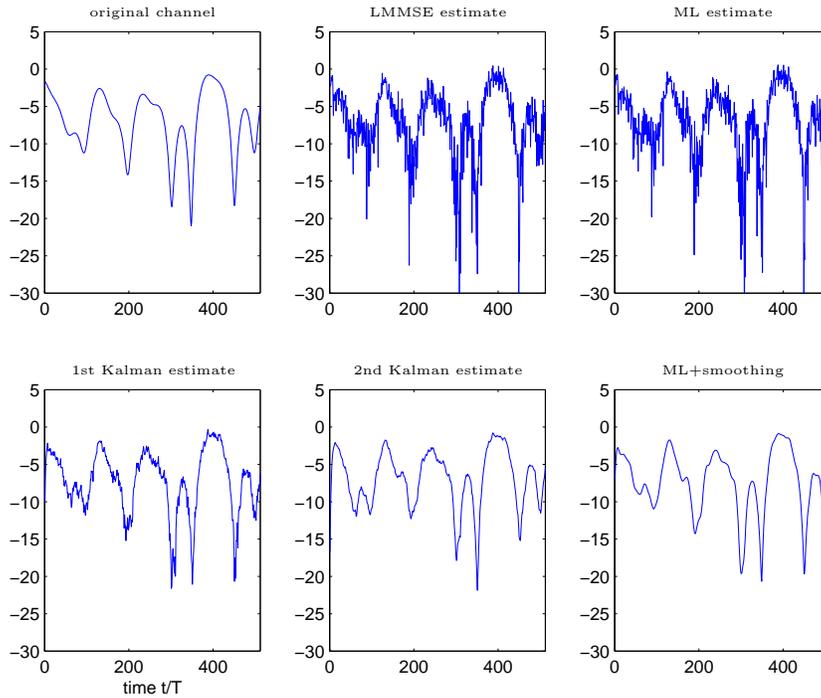
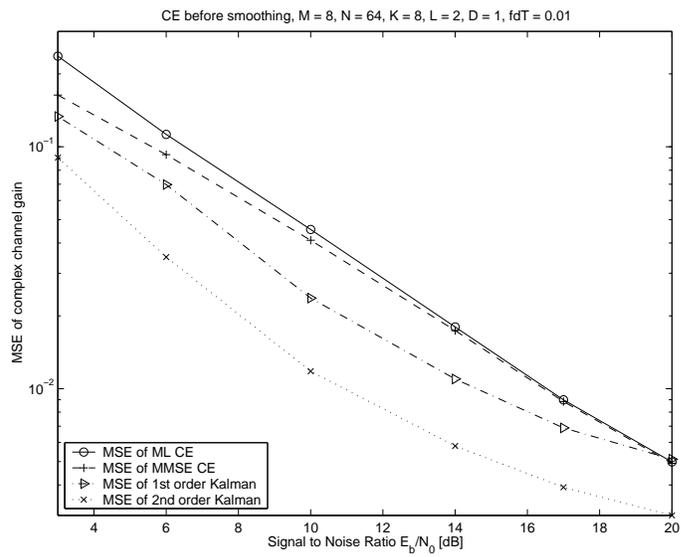


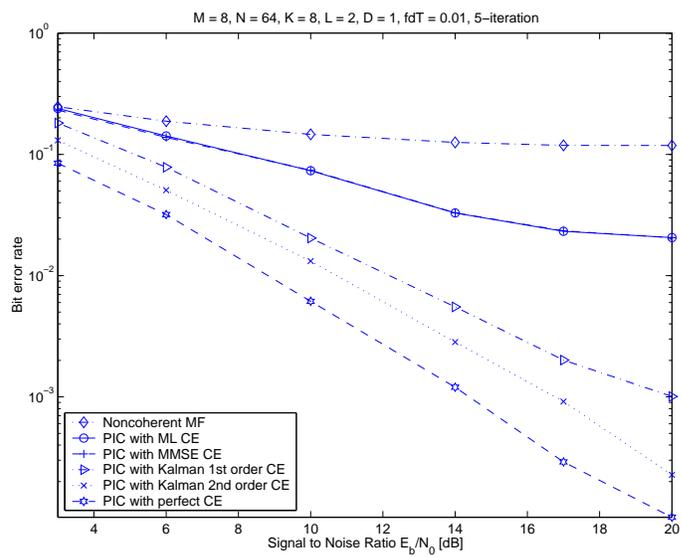
Figure 5.5: Different channel estimation algorithms ($f_d T = 0.01$, smoothing window length=19).

the performance of data detection and that the knowledge of the channel is crucial to the system performance, the coherent PIC with channel estimates significantly outperforms the non-coherent MF.

Fig. 5.8 and 5.9 demonstrate the convergence property of the joint ML channel estimator and PIC scheme. The channel estimation variance is measured with the ML estimator without smoothing; while, the PIC works jointly with the ML estimator and a length 19 smoothing filter. The number of users K is set to 12, the stacking factor of the received vectors D is set to 3 here. In Fig. 5.8, we examine the estimation variance (which is equivalent to MSE since ML estimator is unbiased) of the complex channel gain and magnitude of the channel gain for the ML estimator at each iteration and compare them to the CRLB. It is shown that the ML estimator converges after 3 iterations. Upon convergence, we can observe very close performance between ML estimation with decision feedback and the pilot aided approach assuming exact knowledge of the transmitted data. Only a small gap is noticed at very low SNR region. Theoretical CRLB is plotted for complex channel gain and magnitude and shown to be in close agreement with simulated estimation variance after the ML estimator is

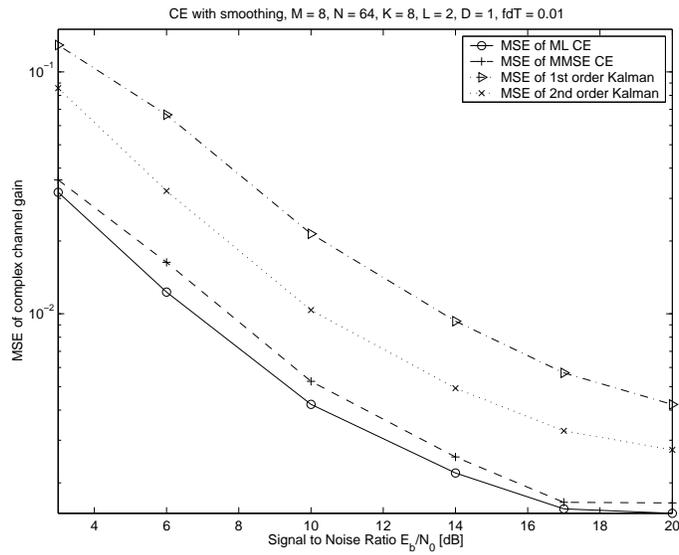


(a) MSE of CE vs. SNR.

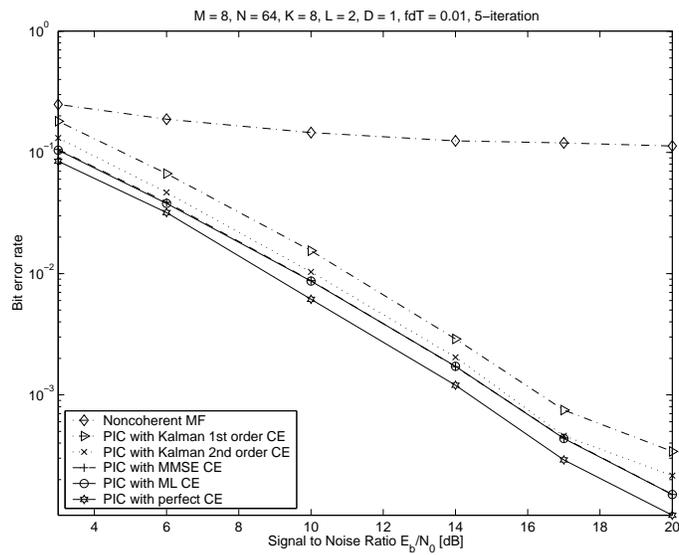


(b) BER vs. SNR.

Figure 5.6: Comparison of different channel estimators with 5-stage PIC without smoothing.

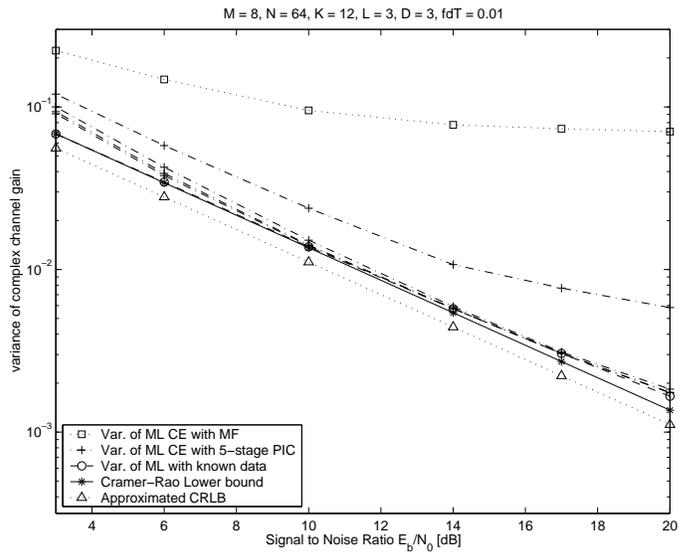


(a) MSE of CE vs. SNR.

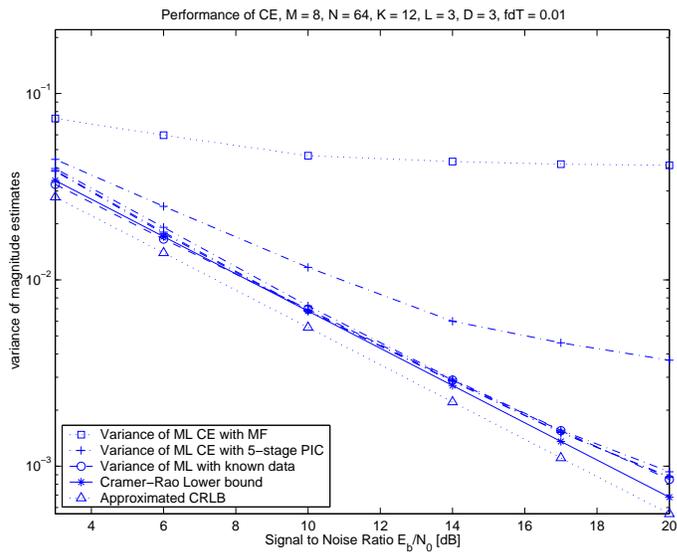


(b) BER vs. SNR.

Figure 5.7: Comparison of different channel estimators with 5-stage PIC with smoothing.



(a) Variance of the ML complex channel gain estimates vs. CRLB.



(b) Variance of magnitude of the ML complex channel gain estimates vs. CRLB.

Figure 5.8: Performance and convergence property of ML CE.

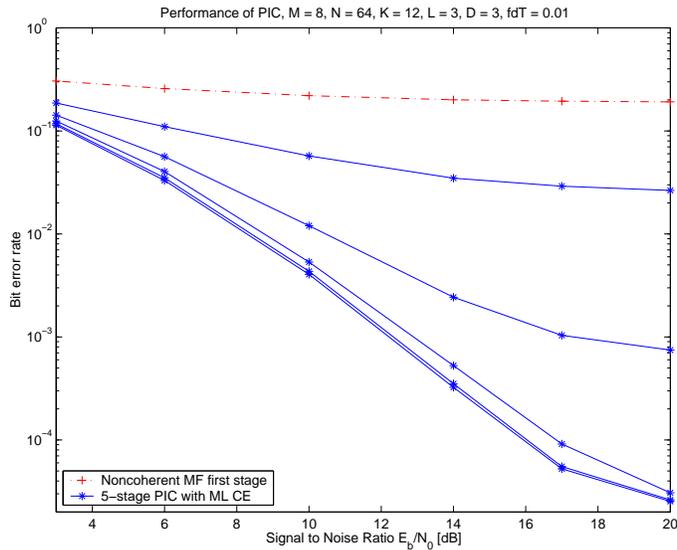


Figure 5.9: Performance and convergence property of PIC with ML CE.

converged. When the approximations expressed in (5.15) and (5.19) are used, the approximated CRLB exhibits a small discrepancy compared with the real CRLB in this two figures due to the fact that the approximations are based on the assumption of transmitted chip sequences from different users and different paths being orthogonal, which is too optimistic and not the case in reality. Fig. 5.9 shows that PIC converges after 5 iterations, it takes more iterations for PIC to reach convergence than channel estimators.

In Fig. 5.10, we examine the effect of the stacking factor D on the estimation results. As indicated by the CRLB, it seems that the larger D value, the smaller estimator error we will get. That would be the case if the channel is static. However, for the time-varying fading channel, the channel changes beyond the coherence time. Therefore, D value has to be chosen accordingly. From the plot, one can see that $D = 4$ appears to be the optimum value before smoothing, and $D = 2$ or $D = 3$ appears to be the optimum value after smoothing for the specific channel setting in question. The time-varying nature of the fading channel prohibits the use of a larger stacking. Also, the dependency between stacking and smoothing as shown by the simulation results has to be taken into account in the selection of the stacking factor D to achieve the best channel estimation and data detection performance.

Convergence property of the LMMSE channel estimator is examined in Fig. 5.11. Like the ML estimator, it takes 3 iterations to converge, at which point the estimation variance is close to its analytical value (5.24) and lower bound (5.26) derived in Appendix 5.9.2.

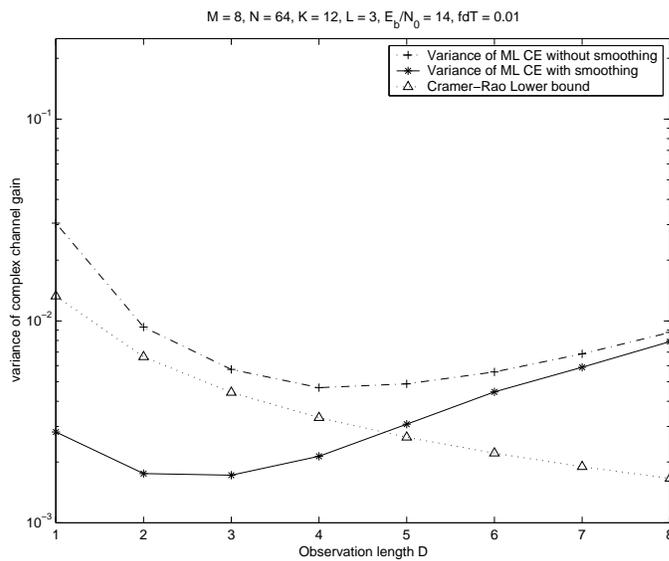


Figure 5.10: Performance of ML CE with different observation length.

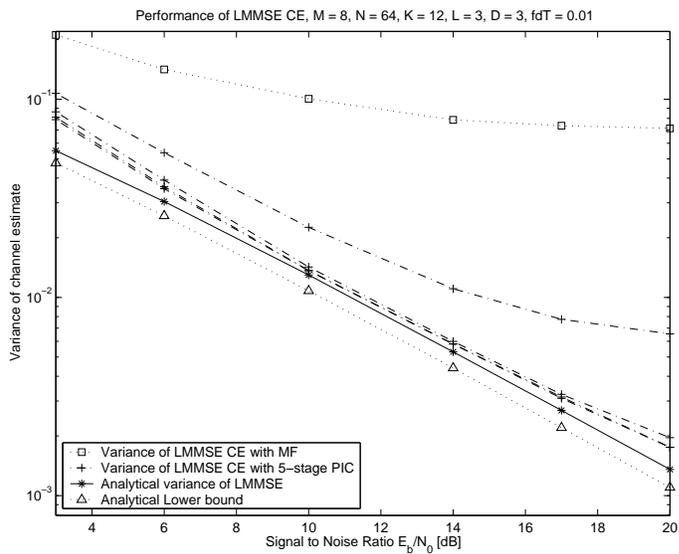


Figure 5.11: Performance of LMMSE CE.

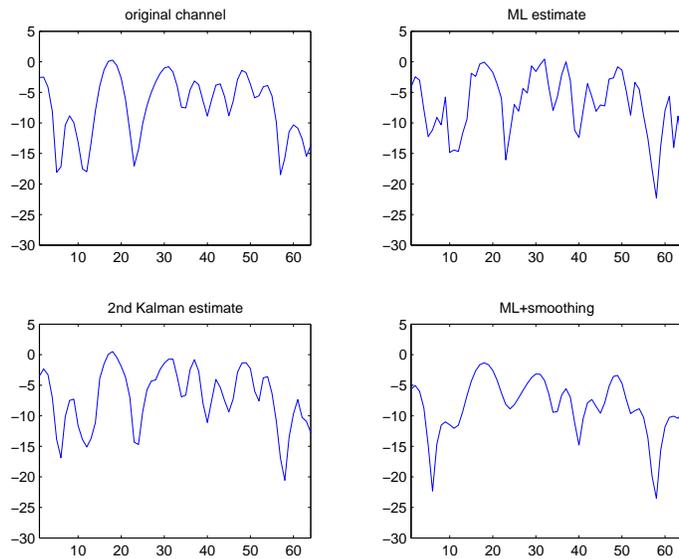


Figure 5.12: ML vs. Kalman CE in fast fading channels ($f_d T = 0.1$, smoothing window length=9).

The ML estimator and Kalman filter are compared in the fast fading channels when the normalized Doppler frequency $f_d T = 0.1$. The smoothing filter length is chosen to be 9 in this case. Fig. 5.12 shows that the second order Kalman filter outperforms the ML estimator under such circumstances. The latter can not keep track on the fast time-varying channel, and the additional filtering operation may have opposite effect, it can destroy the details of the channel information. This is also verified in Fig. 5.13, which shows that the ML estimator with smoothing yields higher estimation variance. However, somewhat surprisingly, the BER for the ML estimator is better with smoothing than without in high SNR region. The Kalman filter, on the contrary, is capable of tracking the fast fading channels, and achieves lower estimation error as well as better BER performance compared to the ML estimator.

Finally, the performance of soft IC and CE is tested in a 18-user system and illustrated in Fig. 5.14. The improvement by using soft IC alone is not noticeable until the SNR increases to $E_b/N_0 = 20$ dB, at which point the gain by applying soft cancellation is 0.4 dB, and it is further increased to 1 dB by applying soft CE. Apparently, in order to achieve the utmost performance, the soft information should be used for both interference cancellation and channel estimation.

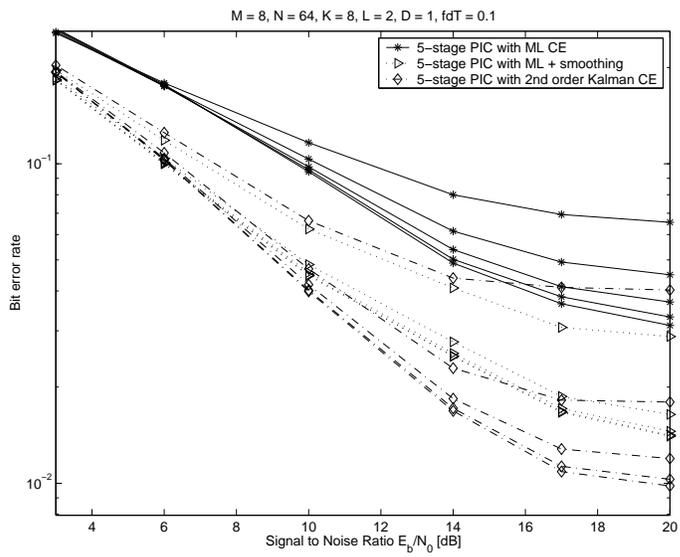
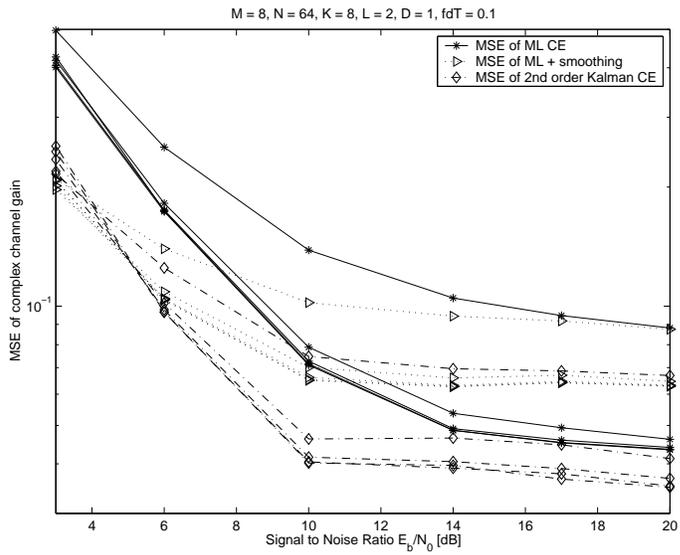


Figure 5.13: ML vs. Kalman CE with 5-stage PIC in fast fading channels.

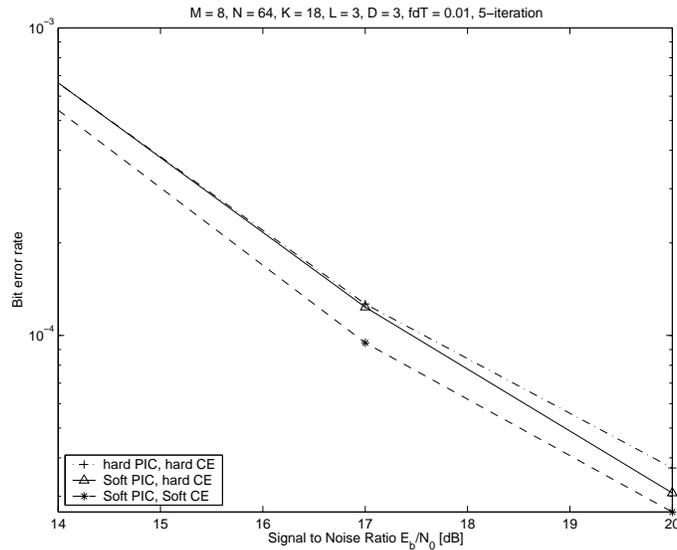


Figure 5.14: Hard vs. soft IC and CE.

5.8 Conclusions

Multuser detection techniques are widely used to combat the detrimental effects of multipath fading and MAI, which are the major impairments in CDMA communication systems. Most MUDs rely on accurate channel information, which needs to be estimated in practice. In this chapter, channel estimation and interference cancellation are presented jointly. We focus on the decision directed approach without inclusion of training sequences. The receiver estimates the channel parameters based on the detected data from feedback loops. The channel coefficients estimated by the presented algorithms are also used to regenerate the signal of each user for the multistage PIC scheme. The effectiveness of the iterative channel estimator is demonstrated in terms of the mean square error of the channel estimate as well as the BER performance of a multistage PIC detector based on the channel estimates. According to the results of our simulations, the performance of coherent demodulation using the proposed system is significantly improved in comparison with conventional noncoherent demodulation.

It is shown that ML estimator is efficient when the channel characteristic is unknown, and that when coupled with PIC, decision directed ML estimator with an additional smoothing filter performs better than the Kalman filters in slow fading channels. It closely tracks the optimum performance attainable by the pilot aided approach when the transmitted data are exactly known. However, the filtering process causes delay for channel estimation which is not desirable under some circumstances. Also

the ML estimator is subject to dimensionality problem when the number of users and paths increase. This problem can be tackled by stacking the received vectors provided the channel remains static during several symbol intervals. When this assumption is not valid, we can resort to the Kalman filter which is suitable for tracking fast fading channels. It takes both correlation and noise into account, does not have dimensionality problem and does not require an additional smoothing operation. Furthermore, the extended Kalman filter is capable of joint detection of propagation delays, fading channels and Doppler shift.

We also learned from experiments that soft IC and CE achieve superior performance compared to conventional IC and CE using hard decision feedback.

5.9 Appendix: Theoretical Analysis for Channel Estimation

5.9.1 Derivation of CRLB for ML channel estimator

Our main concern in this work is to estimate the complex channel gains. It is therefore of interest to establish a bound on the accuracy with which the channels can be estimated. If we restrict our attention to unbiased estimators, the natural performance measure is the error variance. The CRLB is a bound on the smallest covariance matrix that can be achieved by an unbiased estimator, $\hat{\mathbf{h}}$, of a parameter vector \mathbf{h} :

$$\begin{aligned} \mathbf{J}^{-1} &\leq \mathbf{C}_{\hat{\mathbf{h}}} = \mathbf{E} \left\{ (\mathbf{h} - \hat{\mathbf{h}})(\mathbf{h} - \hat{\mathbf{h}})^* \right\} \\ \mathbf{J} &= \mathbf{E} \left[\left(\frac{\partial \ln p(\mathbf{r}; \mathbf{h})}{\partial \mathbf{h}} \right) \left(\frac{\partial \ln p(\mathbf{r}; \mathbf{h})}{\partial \mathbf{h}} \right)^* \right] \end{aligned} \quad (5.10)$$

where $\mathbf{J} \in \mathbb{R}^{\text{tot} \times \text{tot}}$ is the Fisher information matrix and $\ln p(\mathbf{r}; \mathbf{h})$ is the log-likelihood function of the observed vector \mathbf{r} . In the derivation of the CRLB, we assume that the data \mathbf{A} as well as propagation delays are deterministic. We should therefore interpret the derived CRLB as being conditioned on the actual realization of the transmitted data and propagation delays.

Let us denote \mathbf{r} as the stacking of D observation vectors, i.e.,

$$\mathbf{r} = [\mathbf{r}^T(j) \ \mathbf{r}^T(j+1) \ \cdots \ \mathbf{r}^T(j+D-1)]^T \in \mathbb{C}^{DN}$$

and assume the stacking factor D is chosen such that the channel remains relatively static during the observation period.

The vector \mathbf{r} is formed by $\mathbf{r} = \mathbf{A}\mathbf{h} + \mathbf{n}$, and has PDF $\mathbf{r} \sim \mathcal{CN}(\mathbf{A}\mathbf{h}, N_0\mathbf{I})$

where

$$\begin{aligned}\mathbf{A} &= [\mathbf{A}^T(j) \ \mathbf{A}^T(j+1) \ \cdots \ \mathbf{A}^T(j+D-1)]^T \\ &= [\mathbf{a}_{1,1} \ \mathbf{a}_{1,2} \ \cdots \ \mathbf{a}_{k,l} \ \cdots \ \mathbf{a}_{K,L_K}] \\ \mathbf{n} &= [\mathbf{n}^T(j) \ \mathbf{n}^T(j+1) \ \cdots \ \mathbf{n}^T(j+D-1)]^T\end{aligned}\quad (5.11)$$

Its likelihood function and log-likelihood function are

$$\begin{aligned}p(\mathbf{r}; \mathbf{h}) &= \frac{1}{(\pi N_0)^{DN}} \exp \left[-\frac{(\mathbf{r} - \mathbf{A}\mathbf{h})^*(\mathbf{r} - \mathbf{A}\mathbf{h})}{N_0} \right] \\ &= \frac{1}{(\pi N_0)^{DN}} \exp \left[-\frac{(\mathbf{r}^* - \mathbf{h}^* \mathbf{A}^*)(\mathbf{r} - \mathbf{A}\mathbf{h})}{N_0} \right] \\ &= \frac{1}{(\pi N_0)^{DN}} \exp \left[-\frac{\|\mathbf{r}\|^2 - \mathbf{h}^* \mathbf{A}^* \mathbf{r} - \mathbf{r}^* \mathbf{A} \mathbf{h} + \mathbf{h}^* \mathbf{A}^* \mathbf{A} \mathbf{h}}{N_0} \right]\end{aligned}$$

$$\begin{aligned}\ln p(\mathbf{r}; \mathbf{h}) &= -DN \ln \pi - DN \ln N_0 \\ &\quad - \frac{\|\mathbf{r}\|^2 - \mathbf{h}^* \mathbf{A}^* \mathbf{r} - \mathbf{r}^* \mathbf{A} \mathbf{h} + \mathbf{h}^* \mathbf{A}^* \mathbf{A} \mathbf{h}}{N_0}\end{aligned}\quad (5.12)$$

Taking complex gradient [41] of $\ln p(\mathbf{r}; \mathbf{h})$ with respect to \mathbf{h} yields

$$\begin{aligned}\frac{\partial \ln p(\mathbf{r}; \mathbf{h})}{\partial \mathbf{h}} &= -\frac{1}{N_0} \frac{\partial [\|\mathbf{r}\|^2 - \mathbf{h}^* \mathbf{A}^* \mathbf{r} - \mathbf{r}^* \mathbf{A} \mathbf{h} + \mathbf{h}^* \mathbf{A}^* \mathbf{A} \mathbf{h}]}{\partial \mathbf{h}} \\ &= -\frac{1}{N_0} (\mathbf{A}^* \mathbf{A} \mathbf{h} - \mathbf{A}^* \mathbf{r})^*\end{aligned}\quad (5.13)$$

The above equality holds since

$$\begin{aligned}\frac{\partial \|\mathbf{r}\|^2}{\partial \mathbf{h}} &= \mathbf{0}; \quad \frac{\partial \mathbf{h}^* \mathbf{A}^* \mathbf{r}}{\partial \mathbf{h}} = \mathbf{0}; \\ \frac{\partial \mathbf{r}^* \mathbf{A} \mathbf{h}}{\partial \mathbf{h}} &= (\mathbf{A}^* \mathbf{r})^*; \quad \frac{\partial \mathbf{h}^* \mathbf{A}^* \mathbf{A} \mathbf{h}}{\partial \mathbf{h}} = (\mathbf{A}^* \mathbf{A} \mathbf{h})^*\end{aligned}$$

Thus we can derive,

$$\begin{aligned}\frac{\partial \ln p(\mathbf{r}; \mathbf{h})}{\partial \mathbf{h}^*} &= \left(\frac{\partial \ln p(\mathbf{r}; \mathbf{h})}{\partial \mathbf{h}} \right)^* \\ &= \frac{\mathbf{A}^* \mathbf{r} - \mathbf{A}^* \mathbf{A} \mathbf{h}}{N_0} \\ &= \frac{\mathbf{A}^* \mathbf{A}}{N_0} [(\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^* \mathbf{r} - \mathbf{h}] \\ &= \mathbf{J}(\mathbf{h}) [\hat{\mathbf{h}} - \mathbf{h}]\end{aligned}\quad (5.14)$$

This proves that the minimum variance unbiased (MVU) estimator of \mathbf{h} is

$$\hat{\mathbf{h}} = (\mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^* \mathbf{r}$$

which is equivalent to the ML estimator described in Section 5.2. It is efficient in that it attains CRLB. The Fisher information matrix $\mathbf{J}(\mathbf{h})$ and covariance matrix $\mathbf{C}_{\hat{\mathbf{h}}}$ of this unbiased estimator are

$$\begin{aligned} \mathbf{J}(\mathbf{h}) &= \frac{\mathbf{A}^* \mathbf{A}}{N_0} \\ \mathbf{C}_{\hat{\mathbf{h}}} &= \mathbf{J}^{-1}(\mathbf{h}) = N_0 (\mathbf{A}^* \mathbf{A})^{-1} \\ &\geq N_0 \text{diag}(\|\mathbf{a}_{1,1}\|^2, \|\mathbf{a}_{1,2}\|^2, \dots, \|\mathbf{a}_{K,L_K}\|^2)^{-1} \\ &= \frac{N_0}{DN} \mathbf{I}_{\text{tot}} \end{aligned} \quad (5.15)$$

The inequality in (5.15) holds with equality when columns of \mathbf{A} are orthogonal. The CRLB for the complex gain of the k^{th} user's l^{th} path is thus the $[(k-1)L+l]^{\text{th}}$ diagonal element of $\mathbf{J}^{-1}(\mathbf{h})$ when $L_k = L$ for all k , i.e.,

$$\begin{aligned} \text{var}(h_{k,l}) &= \text{diag}[\mathbf{C}_{\hat{\mathbf{h}}}]_{(k-1)L+l} \\ &= \text{diag}[\mathbf{J}^{-1}(\mathbf{h})]_{(k-1)L+l} \\ &\geq \frac{N_0}{DN} \end{aligned} \quad (5.16)$$

Now, we derive the CRLB for amplitude $g_{k,l}$ and phase $\psi_{k,l}$ of the complex channel gain $h_{k,l}$. First, partition \mathbf{h} as

$$\begin{aligned} \mathbf{h} &= [h_{1,1} \ h_{1,2} \ \dots \ h_{k,l} \ \dots \ h_{K,L_K}]^T \\ &= [g_{1,1} e^{j\psi_{1,1}} \ g_{1,2} e^{j\psi_{1,2}} \ \dots \ g_{k,l} e^{j\psi_{k,l}} \ \dots \ g_{K,L_K} e^{j\psi_{K,L_K}}]^T \end{aligned} \quad (5.17)$$

Let us define $\mathbf{e}_{k,l}$ as an $L_{\text{tot}} \times 1$ vector of all zeros except it has element 1 at the position where $h_{k,l}$ is located in \mathbf{h} . Taking the first derivative of \mathbf{h} with respect to amplitude $g_{k,l}$ and phase $\psi_{k,l}$ yields

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial g_{k,l}} &= [0 \ 0 \ \dots \ e^{j\psi_{k,l}} \ \dots \ 0]^T \\ &= e^{j\psi_{k,l}} \mathbf{e}_{k,l} \\ \frac{\partial \mathbf{h}}{\partial \psi_{k,l}} &= [0 \ 0 \ \dots \ jg_{k,l} \ \dots \ 0]^T \\ &= jg_{k,l} \mathbf{e}_{k,l} \end{aligned} \quad (5.18)$$

Assuming $\mathbf{a}_{m,n}$ and $\mathbf{a}_{i,j}$ are orthogonal, i.e., $\mathbf{a}_{m,n}^* \mathbf{a}_{i,j} = 0$ when $m \neq i$,

or $n \neq j$, and taking the expectation of the derivative, we obtain [39]

$$\begin{aligned}
\mathbb{E} \left\{ \frac{\partial^2 \ln p}{\partial g_{m,n} \partial g_{i,j}} \right\} &= -\frac{2}{N_0} \operatorname{Re} \left\{ \frac{\partial \mathbf{h}^*}{\partial g_{m,n}} \mathbf{A}^* \mathbf{A} \frac{\partial \mathbf{h}}{\partial g_{i,j}} \right\} \\
&= -\frac{2}{N_0} \operatorname{Re} \left\{ e^{-j\psi_{m,n}} e^{j\psi_{i,j}} \mathbf{e}_{m,n}^* \mathbf{A}^* \mathbf{A} \mathbf{e}_{i,j} \right\} \\
&= -\frac{2}{N_0} \operatorname{Re} \left\{ e^{-j\psi_{m,n}} e^{j\psi_{i,j}} \mathbf{a}_{m,n}^* \mathbf{a}_{i,j} \right\} = 0 \\
\mathbb{E} \left\{ \frac{\partial^2 \ln p}{\partial g_{k,l}^2} \right\} &= -\frac{2}{N_0} \operatorname{Re} \left\{ e^{-j\psi_{k,l}} e^{j\psi_{k,l}} \mathbf{e}_{k,l}^* \mathbf{A}^* \mathbf{A} \mathbf{e}_{k,l} \right\} \\
&= -\frac{2}{N_0} \|\mathbf{a}_{k,l}\|^2 = -\frac{2DN}{N_0} \\
\mathbb{E} \left\{ \frac{\partial^2 \ln p}{\partial \psi_{m,n} \partial \psi_{i,j}} \right\} &= -\frac{2}{N_0} \operatorname{Re} \left\{ \frac{\partial \mathbf{h}^*}{\partial \psi_{m,n}} \mathbf{A}^* \mathbf{A} \frac{\partial \mathbf{h}}{\partial \psi_{i,j}} \right\} \\
&= -\frac{2}{N_0} \operatorname{Re} \left\{ g_{m,n} g_{i,j} \mathbf{e}_{m,n}^* \mathbf{A}^* \mathbf{A} \mathbf{e}_{i,j} \right\} \\
&= -\frac{2}{N_0} \operatorname{Re} \left\{ g_{m,n} g_{i,j} \mathbf{a}_{m,n}^* \mathbf{a}_{i,j} \right\} = 0 \\
\mathbb{E} \left\{ \frac{\partial^2 \ln p}{\partial \psi_{k,l}^2} \right\} &= -\frac{2}{N_0} \operatorname{Re} \left\{ g_{k,l}^2 \mathbf{e}_{k,l}^* \mathbf{A}^* \mathbf{A} \mathbf{e}_{k,l} \right\} \\
&= -\frac{2\|\mathbf{a}_{k,l}\|^2 \check{P}_{k,l}}{N_0} = -\frac{2DN\check{P}_{k,l}}{N_0}
\end{aligned}$$

Therefore, we derive the variance of amplitude and phase estimate as

$$\operatorname{var}(g_{k,l}) \geq \frac{N_0}{2DN} \quad (5.19)$$

$$\operatorname{var}(\psi_{k,l}) \geq \frac{N_0}{2DN\check{P}_{k,l}} \quad (5.20)$$

We now extend to the case where in addition to \mathbf{h} , the noise variance N_0 is also unknown and derive CRLB under such condition. The parameter vector to be estimated is $\theta = [\mathbf{h} \ N_0]$. The Fisher information matrix is

$$\mathbf{J}(\theta) = \begin{bmatrix} -\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial \mathbf{h}^* \partial \mathbf{h}} \right] & -\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial \mathbf{h}^* \partial N_0} \right] \\ -\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial N_0 \partial \mathbf{h}} \right] & -\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial N_0^2} \right] \end{bmatrix}$$

The derivatives of the log-likelihood function expressed in (5.12) can be

easily found as

$$\begin{aligned}
\frac{\partial \ln p}{\partial \mathbf{h}^*} &= \frac{\mathbf{A}^* \mathbf{r} - \mathbf{A}^* \mathbf{A} \mathbf{h}}{N_0} \\
\frac{\partial \ln p}{\partial N_0} &= -\frac{DN}{N_0} + \frac{\|\mathbf{r} - \mathbf{A} \mathbf{h}\|^2}{N_0^2} \\
\frac{\partial^2 \ln p}{\partial \mathbf{h}^{*2}} &= -\frac{\mathbf{A}^* \mathbf{A}}{N_0} \\
\frac{\partial^2 \ln p}{\partial \mathbf{h}^* \partial N_0} &= \frac{\mathbf{A}^* \mathbf{A} \mathbf{h} - \mathbf{A}^* \mathbf{r}}{N_0^2} = \frac{\mathbf{A}^* (\mathbf{A} \mathbf{h} - \mathbf{r})}{N_0^2} \\
\frac{\partial^2 \ln p}{\partial N_0^2} &= \frac{DN}{N_0^2} - \frac{2\|\mathbf{r} - \mathbf{A} \mathbf{h}\|^2}{N_0^3}
\end{aligned} \tag{5.21}$$

Taking the negative expectation yields

$$\begin{aligned}
-\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial \mathbf{h}^{*2}} \right] &= \frac{\mathbb{E}[\mathbf{A}^* \mathbf{A}]}{N_0} = \frac{DN}{N_0} \mathbf{I} \\
-\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial \mathbf{h}^* \partial N_0} \right] &= -\mathbb{E} \left[\frac{\mathbf{A}^* (\mathbf{A} \mathbf{h} - \mathbf{r})}{N_0^2} \right] = -\mathbb{E} \left[\frac{\mathbf{A}^* \mathbf{n}}{N_0^2} \right] = \mathbf{0} \\
-\mathbb{E} \left[\frac{\partial^2 \ln p}{\partial N_0^2} \right] &= -\frac{DN}{N_0^2} + \frac{2\mathbb{E}[\|\mathbf{r} - \mathbf{A} \mathbf{h}\|^2]}{N_0^3} \\
&= -\frac{DN}{N_0^2} + \frac{2\mathbb{E}[\|\mathbf{n}\|^2]}{N_0^3} = \frac{DN}{N_0^2}
\end{aligned} \tag{5.22}$$

The Fisher information matrix becomes

$$\mathbf{J}(\theta) = \begin{bmatrix} \frac{DN}{N_0} \mathbf{I} & \mathbf{0} \\ 0 & \frac{DN}{N_0^2} \end{bmatrix}$$

So that

$$\begin{aligned}
\text{var}(\mathbf{h}) &\geq \frac{N_0}{DN} \mathbf{I} \\
\text{var}(N_0) &\geq \frac{N_0^2}{DN}
\end{aligned}$$

Note that the CRLB for \mathbf{h} is the same as for the case when N_0 is unknown due to the diagonal nature of the matrix.

5.9.2 Derivation of error covariance for LMMSE channel estimator

In the following, we derive the LMMSE channel estimator and its MSE with Bayesian Gauss-Markov Theorem. The observed data \mathbf{r} is modeled as $\mathbf{r} = \mathbf{A}\mathbf{h} + \mathbf{n}$ where $\mathbf{h} \in \mathbb{C}^{\text{tot}}$ is now assumed to be a random vector whose realization is to be estimated and has mean $\text{E}[\mathbf{h}] = \boldsymbol{\mu}_{\mathbf{h}} = \mathbf{0}$, and covariance matrix $\mathbf{C}_{\mathbf{h}} = \mathbf{P}$. The noise vector $\mathbf{n} \in \mathbb{C}^N$ has PDF $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I})$ and is independent of \mathbf{h} . The Bayesian MMSE estimate of \mathbf{h} that minimizes MSE averaged over all realizations of \mathbf{h} and \mathbf{r} is

$$\begin{aligned}\hat{\mathbf{h}} &= \text{E}(\mathbf{h}|\mathbf{r}) = \boldsymbol{\mu}_{\mathbf{h}} + \mathbf{C}_{\mathbf{h}}\mathbf{A}^*(\mathbf{A}\mathbf{C}_{\mathbf{h}}\mathbf{A}^* + N_0\mathbf{I})^{-1}(\mathbf{r} - \mathbf{A}\boldsymbol{\mu}_{\mathbf{h}}) \\ &= \mathbf{P}\mathbf{A}^*(\mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I})^{-1}\mathbf{r}\end{aligned}\quad (5.23)$$

The performance of this estimator is measured by the error $\mathbf{e} = \mathbf{h} - \hat{\mathbf{h}}$, whose mean is zero and covariance matrix is [41]

$$\begin{aligned}\mathbf{C}_{\mathbf{e}} &= \text{E}(\mathbf{e}\mathbf{e}^T) \\ &= \mathbf{C}_{\mathbf{h}} - \mathbf{C}_{\mathbf{h}}\mathbf{A}^*(\mathbf{A}\mathbf{C}_{\mathbf{h}}\mathbf{A}^* + N_0\mathbf{I})^{-1}\mathbf{A}\mathbf{C}_{\mathbf{h}} \\ &= \mathbf{P} - \mathbf{P}\mathbf{A}^*(\mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I})^{-1}\mathbf{A}\mathbf{P}\end{aligned}\quad (5.24)$$

To simplify the derivation, we assume all the paths have the same received power P . In this case, $\mathbf{P} = P\mathbf{I}$. Using the matrix inversion lemma [41]

$$(\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{D})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1})^{-1}\mathbf{D}\mathbf{A}^{-1}$$

we have

$$\begin{aligned}(\mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I})^{-1} &= \frac{\mathbf{I}}{N_0} - \frac{\mathbf{I}}{N_0}\mathbf{A}(\mathbf{A}^*\frac{\mathbf{I}}{N_0}\mathbf{A} + \mathbf{P}^{-1})^{-1}\mathbf{A}^*\frac{\mathbf{I}}{N_0} \\ &= \frac{\mathbf{I}}{N_0} - \frac{\mathbf{I}}{N_0}\mathbf{A}\left[\frac{\mathbf{A}^*\mathbf{A}}{N_0} + \frac{\mathbf{I}}{P}\right]^{-1}\frac{\mathbf{A}^*}{N_0} \\ &\leq \frac{\mathbf{I}}{N_0} - \frac{\mathbf{I}}{N_0}\mathbf{A}\left[\frac{DN\mathbf{I}}{N_0} + \frac{\mathbf{I}}{P}\right]^{-1}\frac{\mathbf{A}^*}{N_0} \\ &= \frac{\mathbf{I}}{N_0} - \frac{\mathbf{A}}{N_0}\left[\frac{DN}{N_0} + \frac{1}{P}\right]^{-1}\frac{\mathbf{A}^*}{N_0} \\ &= \frac{\mathbf{I}}{N_0} - \frac{P}{N_0(DNP + N_0)}\mathbf{A}\mathbf{A}^*\end{aligned}\quad (5.25)$$

The covariance matrix can then be computed as

$$\begin{aligned}
\mathbf{C}_e &\geq \mathbf{P} - \mathbf{P}\mathbf{A}^* \left(\frac{\mathbf{I}}{N_0} - \frac{P}{N_0(DNP + N_0)} \mathbf{A}\mathbf{A}^* \right) \mathbf{A}\mathbf{P} \\
&= \mathbf{P} - \frac{\mathbf{P}\mathbf{A}^*\mathbf{A}\mathbf{P}}{N_0} + \frac{P}{N_0(DNP + N_0)} \mathbf{P}\mathbf{A}^*\mathbf{A}\mathbf{A}^*\mathbf{A}\mathbf{P} \\
&\geq P\mathbf{I} - \frac{P^2DN}{N_0} \mathbf{I} + \frac{P^3D^2N^2}{N_0(DNP + N_0)} \mathbf{I} \\
&= \left[\frac{PN_0 - P^2DN}{N_0} + \frac{P^3D^2N^2}{N_0(DNP + N_0)} \right] \mathbf{I} \tag{5.26}
\end{aligned}$$

The error covariance matrix is also the the minimum MSE matrix [41]

$$\text{MSE}(\mathbf{h}) = \mathbf{C}_e \geq \left[\frac{PN_0 - P^2DN}{N_0} + \frac{P^3D^2N^2}{N_0(DNP + N_0)} \right] \mathbf{I} \tag{5.27}$$

6.1 Introduction

Accurate synchronization of the chip timing is required to achieve the low bit error rate. The impact of synchronization errors on orthogonally modulated and BPSK modulated systems was studied in [46, 47, 48]. It was shown that errors in the delay estimates would drastically degrade the system performance.

Synchronization of orthogonally modulated signals with long spreading sequences was explored in [49]. An adaptive algorithm was proposed to estimate the synchronization errors in synchronous CDMA systems. Based on the estimates, remedial actions are taken to alleviate the performance degradation caused by sampling the received signals at the incorrect timing. Simulation results show considerable capacity gains when the proposed algorithms are applied on erroneously sampled signals. The algorithms proposed in [49] are only applicable in the downlink scenario in which all the users transmit in a synchronous manner. On the uplink, different users transmit signals asynchronously, the propagation delays are therefore randomly distributed among different users.

Synchronization issues in asynchronous systems (on the uplink) were dealt with, e.g., in [50, 51, 52], etc.. However, the algorithms proposed in those papers only apply to CDMA systems with short spreading sequences and the modulation schemes other than orthogonal signalling.

In this chapter, we aim at solving the problem of uplink code acquisition¹ in multiuser environment in asynchronous long-code CDMA systems.

¹The term acquisition is used interchangeably with delay estimation when the propagation delay of the received signal is less than one symbol period T , which is assumed here.

First, the maximum likelihood (ML) estimator and its unaffordable complexity for implementation are discussed. Some suboptimal solutions, e.g., whitened sliding correlator, MMSE estimator, subspace-based estimator, and approximative ML estimators are then proposed to combat the multiple access interference (MAI) in the fading channels. The performance of these estimators is evaluated with the computer simulations and shown to have better acquisition performance than the standard sliding correlator. They also achieve reduced computational complexity compared to the ML estimator, while maintaining an acceptable performance degradation.

In the last two chapters of the first part of this thesis, we use a simplified scenario and consider single-path flat Rayleigh fading channels in order to facilitate algorithm development. However, the proposed algorithms can be extended to multipath channels. Also, for the purpose of deriving delay estimation algorithms, we rephrase notations as follows.

We form the received vector, $\mathbf{r}(j) \in \mathbb{C}^N$, due to transmission of the j^{th} symbol as

$$\begin{aligned} \mathbf{r}(j) &= [r([(j-1)N+1]T_c) \cdots r([(j-1)N+N]T_c)]^T \\ &= \mathbf{A}(j, \boldsymbol{\tau})\mathbf{h}(j) + \mathbf{n}(j) \end{aligned} \quad (6.1)$$

where the vector $\mathbf{h}(j) \in \mathbb{C}^K$ is defined by the complex channel gains as $\mathbf{h}(j) = [h_1(jT) \ h_2(jT) \ \cdots \ h_K(jT)]^T$. The data matrix $\mathbf{A}(j, \boldsymbol{\tau}) \in \mathbb{R}^{N \times K}$ can be expressed as

$$\mathbf{A}(j, \boldsymbol{\tau}) = \begin{bmatrix} \mathbf{a}_1(j, \tau_1) & \mathbf{a}_2(j, \tau_2) & \cdots & \mathbf{a}_K(j, \tau_K) \end{bmatrix} \quad (6.2)$$

where τ_k stands for the propagation delay of the k^{th} user. The vector $\boldsymbol{\tau}$ is defined as $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_K]^T \in \mathbb{R}^K$. Let us denote $\tau_k = (p_k + \delta_k)T_c$ (T_c is one chip interval) such that $p_k \in \{0, 1, \dots, N-1\}$ and $\delta_k \in [0, 1)$ stand for integer and fractional parts of the delay, respectively, and denote $\mathbf{b}_k(j) \in \mathbb{R}^N$, the transmitted chip sequence due to the k^{th} user as

$$\mathbf{b}_k(j) = \mathbf{C}_k(j)\mathbf{s}_k(j) \quad (6.3)$$

where $\mathbf{C}_k(j) \in \{-1, +1\}^{N \times N}$ is a diagonal matrix defined by the k^{th} user's scrambling code, and $\mathbf{s}_m \in \{-1, +1\}^{N \times 1}$ is the m^{th} column ($m = i_k(j)$) of the $N \times N$ Hadamard matrix. For the rectangular pulse shape employed in this work, each column of the matrix $\mathbf{A}(j, \boldsymbol{\tau})$ in (6.2) can be expressed as

$$\begin{aligned} \mathbf{a}_k(j, \tau_k) &= (1 - \delta_k) [\text{ds}(\mathbf{b}_k(j), p_k) + \text{us}(\mathbf{b}_k(j-1), N - p_k)] \\ &\quad + \delta_k [\text{ds}(\mathbf{b}_k(j), p_k + 1) + \text{us}(\mathbf{b}_k(j-1), N - p_k - 1)] \end{aligned} \quad (6.4)$$

where $\text{us}(\cdot)$, $\text{ds}(\cdot)$ stand for the up-shift and down-shift operators respec-

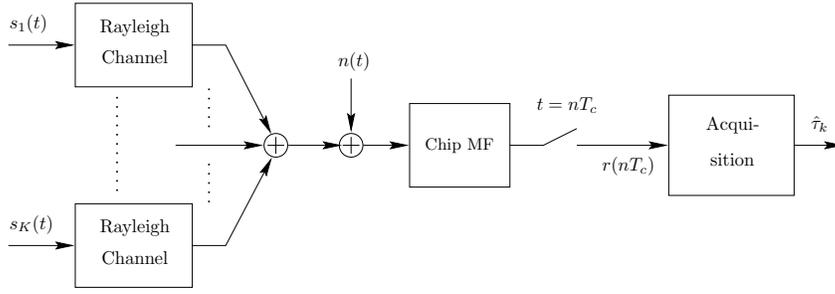


Figure 6.1: Receiver front-end.

tively

$$\begin{aligned} \text{us}([a_1 \cdots a_N]^T, q) &= [a_{N+1-q} \cdots a_N \ 0 \cdots 0]^T \\ \text{ds}([a_1 \cdots a_N]^T, q) &= [0 \cdots 0 \ a_1 \cdots a_{N-q}]^T \end{aligned}$$

All the acquisition schemes introduced in this chapter require a training sequence which is periodic for each user and with good cross-correlation property among the users. This can be fulfilled by assigning a distinct Walsh sequence to each user and spreading it with a short scrambling code to achieve low cross-correlation between the shifted versions of Walsh sequences, i.e., $\mathbf{b}_k(j) = \mathbf{C}_k(j)\mathbf{s}_k(j)$ remains unchanged and is transmitted repeatedly.

The delay estimation is the very first step to be conducted in the receiver. The task of the delay estimator in the receiver is to detect the propagation delays τ_k for $k = 1, 2, \dots, K$ (see Figure 6.1) given the received signal vector $\mathbf{r}(j)$, i.e., performing delay estimation jointly for all the users. For notation simplicity we sometimes suppress the symbol index j from $\mathbf{r}(j)$, $\mathbf{h}(j)$, $\mathbf{A}(j, \boldsymbol{\tau})$, etc., whenever no ambiguity arises.

The decision on the k^{th} user's delay τ_k is found as the minimizer of the cost function $J_k(\tau_k)$

$$\hat{\tau}_k = \arg \min_{\tau_k \in [0, T)} J_k(\tau_k)$$

We define the failure of acquisition (estimation outlier) to be the case when the estimated delay deviates from the true value by more than half chip interval, i.e. $|\hat{\tau}_k - \tau_k| > T_c/2$.

In the following, we shall introduce how the decision function $J_k(\tau_k)$ is derived for different acquisition algorithms.

6.2 ML Approach to Delay Estimation

The log-likelihood function of the received vector can be expressed as

$$\text{constant} - LN \ln N_0 - \frac{1}{N_0} \sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{A}(j, \boldsymbol{\tau})\mathbf{h}(j)\|^2$$

Maximization of this log-likelihood function is equivalent to minimization of the function

$$\sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{A}(j, \boldsymbol{\tau})\mathbf{h}(j)\|^2 \quad (6.5)$$

Substituting $\hat{\mathbf{h}}(j) = \mathbf{A}^\dagger \mathbf{r}(j)$ into (6.5), we yield the ML estimate of $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_K]^T$ as

$$\begin{aligned} \hat{\boldsymbol{\tau}}^{ML} &= \arg \min_{\boldsymbol{\tau}} \sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{A}(j, \boldsymbol{\tau})\mathbf{h}(j)\|^2 \Big|_{\hat{\mathbf{h}}(j) = \mathbf{A}^\dagger \mathbf{r}(j)} \\ &= \arg \min_{\boldsymbol{\tau}} \text{trace}\{\mathbf{P}_\mathbf{A}^\perp \hat{\mathbf{R}}\} \end{aligned}$$

where $\mathbf{P}_\mathbf{A}^\perp = \mathbf{I}_N - \mathbf{A}\mathbf{A}^\dagger$ is the orthogonal projection matrix onto the orthogonal complement to the subspace spanned by the columns of $\mathbf{A}(j, \boldsymbol{\tau})$ and $\hat{\mathbf{R}}$ is the sample autocorrelation matrix defined by

$$\hat{\mathbf{R}} = \frac{1}{L} \sum_{j=1}^L \mathbf{r}(j)\mathbf{r}^*(j) \quad (6.6)$$

With this ML delay estimator and most of the algorithms introduced later on, each user has to transmit the same pilot symbol during the training period so that the data matrix \mathbf{A} in $\mathbf{R} = \mathbf{E}[\mathbf{r}\mathbf{r}^*] = \mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I}_N$ remains unchanged. Therefore, the autocorrelation of the fading channel vector, after averaging, approximates the matrix \mathbf{P} , i.e.,

$$\frac{1}{L} \sum_{j=1}^L \mathbf{h}(j)\mathbf{h}^*(j) \approx \mathbf{P} = \text{diag}(P_1, P_2, \dots, P_K)$$

which leads to

$$\frac{1}{L} \sum_{j=1}^L \mathbf{r}(j)\mathbf{r}^*(j) \approx \mathbf{R}$$

The ML delay estimator finds the delays of all the users simultaneously. The disadvantage of this algorithm is the unaffordable computational complexity because it has to search over N^K points (K is the number of users,

N is spreading factor, and we assume the delays are within one symbol interval). While other estimators introduced below estimate the delay for one user at a time, therefore, the number of points to be searched goes down to NK . They achieve reduced computational complexity with an acceptable performance.

6.3 Conventional Sliding Correlator

The sliding correlator (SC) is the standard approach to propagation delay estimation. It treats the MAI as additive noise: $\mathbf{r} = h_k \mathbf{a}_k + \underbrace{\text{MAI} + \mathbf{n}}_{\text{noise}}$.

The received signal is correlated with time delayed versions of the training sequence, and the desired timing is the value of the time delay candidate that maximizes the correlation. Mathematically, this estimate is given by

$$\begin{aligned} \hat{\tau}_k^c &= \arg \max_{\tau_k \in [0, T)} \left| \frac{1}{L} \sum_{j=1}^L \frac{\mathbf{a}_k^*(j, \tau_k) \mathbf{r}(j)}{\|\mathbf{a}_k(j, \tau_k)\|} \right|^2 \\ &= \arg \max_{\tau_k \in [0, T)} \frac{|\mathbf{a}_k(j, \tau_k)^* \bar{\mathbf{r}}|^2}{\|\mathbf{a}_k(j, \tau_k)\|^2} \end{aligned}$$

or equivalently, it can be expressed as

$$\begin{aligned} \hat{\tau}_k^c &= \arg \min_{\tau_k \in [0, T)} J_{k,C}(\tau_k) \\ &= \arg \min_{\tau_k \in [0, T)} - \left| \frac{1}{L} \sum_{j=1}^L \frac{\mathbf{a}_k^*(j, \tau_k) \mathbf{r}(j)}{\|\mathbf{a}_k(j, \tau_k)\|} \right|^2 \\ &= \arg \min_{\tau_k \in [0, T)} - \frac{|\mathbf{a}_k(j, \tau_k)^* \bar{\mathbf{r}}|^2}{\|\mathbf{a}_k(j, \tau_k)\|^2} \end{aligned} \quad (6.7)$$

where $\bar{\mathbf{r}} = \frac{1}{L} \sum_{j=1}^L \mathbf{r}(j)$ is the sample mean of the received signal vector.

Recall that

$$\begin{aligned} \mathbf{a}_k(j, \tau_k) &= (1 - \delta_k) [\text{ds}(\mathbf{b}_k(j), p_k) + \text{us}(\mathbf{b}_k(j-1), N - p_k)] \\ &\quad + \delta_k [\text{ds}(\mathbf{b}_k(j), p_k + 1) + \text{us}(\mathbf{b}_k(j-1), N - p_k - 1)] \end{aligned}$$

We can see $\mathbf{a}_k(j, \tau_k)$ is piecewise linear in τ_k . In particular, for $\tau_k \in [pT_c, (p+1)T_c)$

$$\mathbf{a}_k(j, \tau_k) = \delta \mathbf{a}_k(t_1) + (1 - \delta) \mathbf{a}_k(t_0) \quad (6.8)$$

where $t_0 = pT_c$, $t_1 = (p+1)T_c$, p is an integer and $\delta = (\tau_k - t_0)/T_c$. By

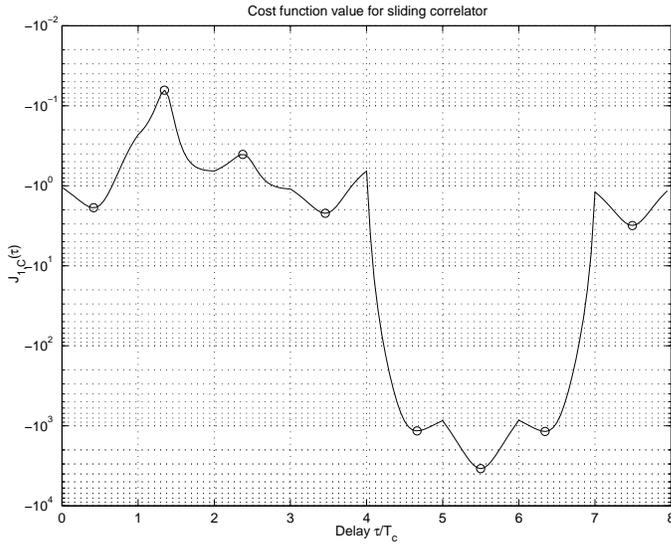


Figure 6.2: Sliding correlator function value for a single user, the real delay $\tau = 5.5T_c$, SNR= 20 dB.

substituting (6.8) into (6.7), we obtain

$$J_{k,C}(\tau_k) = -\frac{(K_1 + K_2 - 2K_3)\delta^2 + (2K_3 - 2K_1)\delta + K_1}{(2N - 2K_4)\delta^2 + (2K_4 - 2N)\delta + N} \quad (6.9)$$

where $K_1 = \bar{\mathbf{r}}^* \mathbf{a}_k(t_0) \mathbf{a}_k(t_0)^* \bar{\mathbf{r}}$, $K_2 = \bar{\mathbf{r}}^* \mathbf{a}_k(t_1) \mathbf{a}_k(t_1)^* \bar{\mathbf{r}}$, $K_3 = \text{Re}\{\bar{\mathbf{r}}^* \mathbf{a}_k(t_0) \mathbf{a}_k(t_1)^* \bar{\mathbf{r}}\}$, $K_4 = \mathbf{a}_k(t_0)^* \mathbf{a}_k(t_1)$.

From (6.9), we see that $J_{k,C}(\tau_k)$ can be written as a rational function of two polynomials of degree four. Furthermore, $J_{k,C}(\tau_k)$ is differentiable for $\tau_k \in [0, T)$ except at the points of $\tau_k = pT_c$ for an integer p . This suggests an efficient search procedure of the cost function which is illustrated in Figure 6.2. We form the set of candidate estimates \mathcal{T} as the union of the solution set to $\frac{dJ_{k,C}(\tau_k)}{d\tau_k} = 0$ for $\tau_k \in (pT_c, (p+1)T_c)$, $p = 0, 1, \dots, N-1$, and the set of end points $\{pT_c\}_{p=0}^{N-1}$. The final estimate, $\hat{\tau}_k$, is found as the member of \mathcal{T} corresponding to the smallest value of the cost function $J_{k,C}(\tau_k)$. As shown in Figure 6.2, $J_{k,C}(\tau)$ has the minimum value at the true delay $\tau = 5.5T_c$ in this single user case.

The advantages of the SC are the low computational complexity and good performance in single user situation. However, it is highly unreliable in presence of the MAI, even in two-user case, as demonstrated in Figure 6.3.

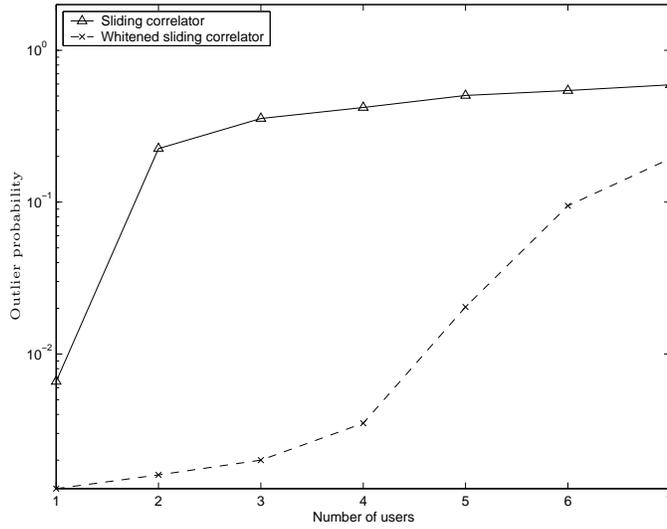


Figure 6.3: Performance of SC and WSC.

The simulated system for the following experiments is a $K = 6$ -user system. Simulation results are averaged over 500 ~ 3000 Monte-Carlo runs with $t = 10 \sim 300$ pilot symbols transmitted. Each Monte-Carlo run represents a particular realization of the noise and fading processes as well as randomly distributed propagation delays $\tau_k \in [0, T)$.

6.4 ML Single User Delay Estimator

The ML single user delay estimator also treats the MAI as additive noise. If we look into the received vector

$$\begin{aligned} \mathbf{r}(j) &= \sum_{k=1}^K \mathbf{a}_k(\tau_k) h_k + \mathbf{n} = \mathbf{a}_k(\tau_k) h_k + \text{MAI} + \mathbf{n} \\ &= \mathbf{a}_k(\tau_k) h_k + \mathbf{w} \end{aligned}$$

the contribution from the k^{th} user is $\mathbf{a}_k(\tau_k) h_k$, the vector \mathbf{w} represents the combined interference and noise. We suppress the symbol index j from $\mathbf{a}(j, \tau_k)$ from now on since it is constant and only a function of the delay τ_k during the acquisition stage.

Under the assumption that \mathbf{w} is a Gaussian random vector and that the data is known (training sequence), an ML estimator of τ_k can be found

by minimizing

$$J_{ML}(\tau_k) = \sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{a}_k(\tau_k)h_k\|^2 \quad (6.10)$$

The complex channel gain for the user k during the j^{th} symbol interval, h_k can be estimated as $h_k = \mathbf{a}_k^\dagger \mathbf{r}(j)$, where $\mathbf{a}_k^\dagger = (\mathbf{a}_k^* \mathbf{a}_k)^{-1} \mathbf{a}_k^* = \mathbf{a}_k^* / \|\mathbf{a}_k\|^2$ denotes the left pseudoinverse of $\mathbf{a}_k(\tau_k)$. The ML single user delay estimator can thus be formulated as

$$\begin{aligned} \hat{\tau}_k^{ML} &= \arg \min_{\tau_k} \sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{a}_k(\tau_k)h_k\|^2 \Big|_{h_k = \mathbf{a}_k^\dagger \mathbf{r}(j)} \\ &= \arg \min_{\tau_k} \text{trace}\{\mathbf{P}_{\mathbf{a}_k}^\perp \hat{\mathbf{R}}\} \end{aligned} \quad (6.11)$$

where $\mathbf{P}_{\mathbf{a}_k}^\perp = \mathbf{I}_N - \mathbf{a}_k \mathbf{a}_k^\dagger$ is the orthogonal projection matrix onto the orthogonal complement to the subspace spanned by the vector $\mathbf{a}_k(\tau_k)$.

If we denote $\mathbf{P}_{\mathbf{a}_k} = \mathbf{a}_k(\tau_k) \mathbf{a}_k^\dagger(\tau_k)$, the equation (6.10) can be reformed as

$$\begin{aligned} &\sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{a}_k(\tau_k)h_k\|^2 \Big|_{h_k = \mathbf{a}_k^\dagger \mathbf{r}(j)} \\ &= \sum_{j=1}^L \|\mathbf{r}(j) - \mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)\|^2 \\ &= \sum_{j=1}^L [\mathbf{r}(j) - \mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)]^* [\mathbf{r}(j) - \mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)] \\ &= \sum_{j=1}^L [\mathbf{r}^*(j) - \mathbf{r}^*(j) \mathbf{P}_{\mathbf{a}_k}] [\mathbf{r}(j) - \mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)] \\ &= \sum_{j=1}^L \{\|\mathbf{r}(j)\|^2 - \mathbf{r}^*(j) \mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)\} \\ &= \sum_{j=1}^L \{\|\mathbf{r}(j)\|^2 - \|\mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)\|^2\} \end{aligned}$$

Since $\|\mathbf{r}\|^2$ is irrelevant to the choice of τ_k , so the decision is solely decided by $\mathbf{P}_{\mathbf{a}_k} \mathbf{r}$, i.e., the projection of $\mathbf{r}(j)$ onto the the subspace spanned by the vector \mathbf{a}_k . Therefore, the ML single user delay estimator (6.11) can be expressed as

$$\hat{\tau}_k^{ML} = \arg \min_{\tau_k} \sum_{j=1}^L -\|\mathbf{P}_{\mathbf{a}_k} \mathbf{r}(j)\|^2$$

The projection of $\mathbf{r}(j)$ onto the the subspace spanned by the vector $\mathbf{a}_k(\tau_k)$ is the same as the normalized correlation of the vectors $\mathbf{r}(j)$ and $\mathbf{a}_k(\tau_k)$. Therefore, the ML delay estimator is identical to the conventional sliding correlator (SC), the standard approach to delay estimation expressed by (6.7).

6.5 Whitened Sliding Correlator

Like the standard receiver (i.e., the single user matched filter), the SC or the ML single user delay estimator is only optimal in the AWGN single user channel or in a strictly orthogonal synchronous channel. It performs poorly when MAI is present. A way to work around this problem is to whiten the received vector \mathbf{r} by preprocessing it with the matrix $\mathbf{R}^{-1/2}$ [51], where the matrix \mathbf{R} is the autocorrelation matrix of the received vector \mathbf{r} , i.e., $\mathbf{R} = \mathbf{E}[\mathbf{r}\mathbf{r}^*] = \mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I}_N$.

The received vector can be written as

$$\mathbf{r} = \mathbf{a}_k h_k + \underbrace{\sum_{\substack{l=1 \\ l \neq k}}^K \mathbf{a}_l h_l}_{\text{MAI}} + \mathbf{n} \quad (6.12)$$

The “whitened” received vector is

$$\mathbf{R}^{-1/2}\mathbf{r} = \mathbf{R}^{-1/2}(\mathbf{A}\mathbf{h} + \mathbf{n}) = \underbrace{h_k \mathbf{R}^{-1/2} \mathbf{a}_k}_{\text{desired signal}} + \underbrace{\mathbf{R}^{-1/2}(\text{MAI} + \mathbf{n})}_{\text{noise}}$$

where $\mathbf{R}^{-1/2}$ is obtained from \mathbf{R} by Cholesky factorization. The autocorrelation matrix \mathbf{R} is unknown and usually replaced by the sample autocorrelation matrix $\hat{\mathbf{R}}$ defined in equation (6.6).

The enhanced version of the SC, namely, the whitened sliding correlator (WSC) can be formulated as

$$J_{k,W}(\tau_k) = -\frac{|(\hat{\mathbf{R}}^{-1/2} \mathbf{a}_k(\tau_k))^* \hat{\mathbf{R}}^{-1/2} \bar{\mathbf{r}}|}{\|\hat{\mathbf{R}}^{-1/2} \mathbf{a}_k(\tau_k)\|^2} = -\frac{|\mathbf{a}_k^*(\tau_k) \hat{\mathbf{R}}^{-1} \bar{\mathbf{r}}|}{\mathbf{a}_k^*(\tau_k) \hat{\mathbf{R}}^{-1} \mathbf{a}_k(\tau_k)} \quad (6.13)$$

In order to calculate the delay analytically as we did earlier, we use $J_{k,W}^2$ to replace $J_{k,W}$ of equation (6.13) in order to make the numerator

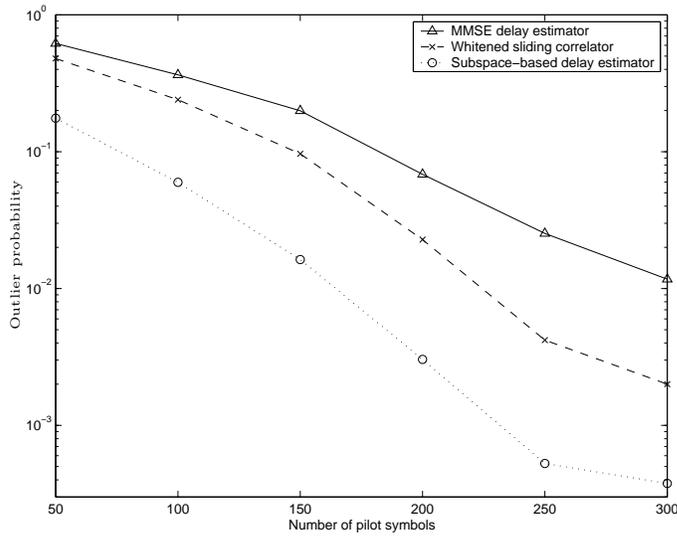


Figure 6.4: Acquisition error probability as function of the number of training symbols.

differentiable

$$\begin{aligned}
 \hat{\tau}_k^w &= \arg \min_{\tau_k \in [0, T]} -J_{k, W}^2(\tau_k) \\
 &= \arg \min_{\tau_k \in [0, T]} -\frac{\|K_2 - K_1\|^2 \delta^2 + X_0 \delta + \|K_1\|^2}{X_1 \delta^4 + X_2 \delta^3 + X_3 \delta^2 + X_4 \delta + K_3^2} \quad (6.14)
 \end{aligned}$$

where $K_1 = \mathbf{a}_k(t_0)^* \mathbf{R}^{-1} \mathbf{r}$, $K_2 = \mathbf{a}_k(t_1)^* \mathbf{R}^{-1} \mathbf{r}$, $K_3 = \mathbf{a}_k(t_0)^* \mathbf{R}^{-1} \mathbf{a}_k(t_0)$, $K_4 = \mathbf{a}_k(t_1)^* \mathbf{R}^{-1} \mathbf{a}_k(t_1)$, $K_5 = \text{Re}\{\mathbf{a}_k(t_0)^* \mathbf{R}^{-1} \mathbf{a}_k(t_1)\}$, $X_0 = [K_1^*(K_2 - K_1) + K_1(K_2 - K_1)^*]$, $X_1 = (K_3 + K_4 - 2K_5)^2$, $X_2 = 4(K_3 + K_4 - 2K_5)(K_5 - K_3)$, $X_3 = 2(K_3 + K_4 - 2K_5)K_3$, $X_4 = 4(K_5 - K_3)K_3$.

The delay estimate $\hat{\tau}_k$ is derived using the same procedure as defined in Section 6.3, i.e., finding the solutions to $dJ_{k, W}^2(\tau_k)/d\tau_k = 0$ for $\tau_k \in (pT_c, (p+1)T_c)$, $p = 0, 1, \dots, N-1$, together with the set of end points $\{pT_c\}_{p=0}^{N-1}$, then searching for a global minimum. Figure 6.3 shows that the acquisition performance of the WSC is greatly improved in the presence of MAI compared to the conventional SC.

The number of pilot symbols needs to be sufficiently large in order to get an accurate estimate of \mathbf{R} . Figure 6.4 shows that the acquisition error probability decreases when more pilot symbols are used. Long training sequences are needed to combat noise and fading.

6.6 Linear MMSE Delay Estimator

The linear MMSE delay estimation is similar to the MMSE approach to channel estimation introduced in Section 5.3. The receiver computes a receiver matrix $\mathbf{W} \in C^{N \times N}$, which is chosen to minimize the mean square error $E[\|\mathbf{a}_k(\tau_k)h_k - \mathbf{W}^*\mathbf{r}\|^2]$. The receiver matrix can be computed using adaptive filtering techniques

$$\begin{aligned}\mathbf{W}_{\text{MMSE}} &= \arg \min_{\mathbf{W}} E[\|\mathbf{a}_k(\tau_k)h_k - \mathbf{W}^*\mathbf{r}\|^2] = \mathbf{R}^{-1}\mathbf{U} \\ \mathbf{U} &= E[\mathbf{r}(\mathbf{a}_k(\tau_k)h_k)^*] = E[\mathbf{a}_k(\tau_k)h_k h_k^* \mathbf{a}_k^*(\tau_k)] = P_k \mathbf{a}_k(\tau_k) \mathbf{a}_k^*(\tau_k)\end{aligned}\quad (6.15)$$

If we look into the received vector \mathbf{r} in (6.12), the contribution from the k^{th} user is $\mathbf{a}_k h_k$. This implies the rationale behind the equation (6.15) is that we would like to find a matrix \mathbf{W} which can filter out all the interference and noise from the received observation so that in the ideal situation, the received vector after filtering only contains the k^{th} user's contribution. Once the receiver matrix is computed, the delay can be estimated by correlating this matrix with the received vector. Thus, the MMSE approach to delay estimation can be formulated as

$$\begin{aligned}\hat{\tau}_k^{\text{MMSE}} &= \arg \min_{\tau_k \in [0, T)} J_{k, \text{MMSE}}(\tau_k) \\ &= \arg \min_{\tau_k \in [0, T)} - \frac{\|\mathbf{W}_{\text{MMSE}}^* \bar{\mathbf{r}}\|^2}{\|\mathbf{W}_{\text{MMSE}}\|^2} \\ &= \arg \min_{\tau_k \in [0, T)} - \frac{\|(\hat{\mathbf{R}}^{-1} \mathbf{a}_k(\tau_k) \mathbf{a}_k^*(\tau_k))^* \bar{\mathbf{r}}\|^2}{\|\hat{\mathbf{R}}^{-1} \mathbf{a}_k(\tau_k) \mathbf{a}_k^*(\tau_k)\|^2}\end{aligned}\quad (6.16)$$

where $\bar{\mathbf{r}}$ and $\hat{\mathbf{R}}$ are the sample mean of the received vector and autocorrelation matrix, respectively. The analytical expression of $J_{k, \text{MMSE}}$ is omitted to conserve space.

As depicted in Figure 6.4, the MMSE performs a little worse than the WSC when the number of pilots is small; However, the performance gap becomes bigger when the number of pilots increases.

6.7 Subspace-Based Delay Estimator

The subspace-based approach to delay estimation was initially proposed in [50, 53]. Recall that

$$\mathbf{R} = E[\mathbf{r}\mathbf{r}^*] = \mathbf{A}E[\mathbf{h}\mathbf{h}^*]\mathbf{A}^* + N_0\mathbf{I}_N = \mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I}_N$$

where $\mathbf{A}(\boldsymbol{\tau}) = [\mathbf{a}_1(\tau_1) \cdots \mathbf{a}_K(\tau_K)]$, $\mathbf{P} = \text{diag}(P_1, P_2, \dots, P_K)$. We assume that \mathbf{A} and \mathbf{P} in the above equation have full rank² and define the signal subspace as the column space of \mathbf{A} , i.e., $\text{range}(\mathbf{A})$, and the noise subspace as the orthogonal complement to $\text{range}(\mathbf{A})$. Since \mathbf{A} has rank K (the number of users), the signal subspace will have dimensionality K . It can be shown that $\text{range}(\mathbf{A}\mathbf{P}\mathbf{A}^*) = \text{range}(\mathbf{A})$. Therefore, the matrix $\mathbf{A}\mathbf{P}\mathbf{A}^*$ has K nonzero positive eigenvalues: $\{\lambda_1, \lambda_2, \dots, \lambda_K\}$, and $N - K$ zero eigenvalues. Let \mathbf{E}_s denote the matrix formed by the eigenvectors corresponding to the K nonzero eigenvalues and \mathbf{E}_n denote the matrix formed by the eigenvectors corresponding to the $N - K$ zero eigenvalues. The hermitian and positive definite matrices $\mathbf{A}\mathbf{P}\mathbf{A}^*$ and $\mathbf{R} = \mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I}_N$ can be decomposed into signal subspace, i.e., $\text{range}(\mathbf{E}_s) \in \mathbb{C}^{N \times K}$ and noise subspace, i.e., $\text{range}(\mathbf{E}_n) \in \mathbb{C}^{N \times (N-K)}$ as

$$\mathbf{A}\mathbf{P}\mathbf{A}^* = \begin{bmatrix} \mathbf{E}_s & \mathbf{E}_n \end{bmatrix} \begin{bmatrix} \tilde{\Lambda}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{E}_s & \mathbf{E}_n^* \end{bmatrix} = \mathbf{E}_s \tilde{\Lambda}_s \mathbf{E}_s^* + \mathbf{0} \mathbf{E}_n \mathbf{E}_n^*$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{E}_s & \mathbf{E}_n \end{bmatrix} \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \Lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{E}_s & \mathbf{E}_n^* \end{bmatrix} = \mathbf{E}_s \Lambda_s \mathbf{E}_s^* + \Lambda_n \mathbf{E}_n \mathbf{E}_n^*$$

where $\tilde{\Lambda}_s = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_K)$, $\Lambda_s = \tilde{\Lambda}_s + N_0\mathbf{I}_K = \text{diag}(\lambda_1 + N_0, \lambda_2 + N_0, \dots, \lambda_K + N_0)$, $\Lambda_n = N_0\mathbf{I}_{N-K} = \text{diag}(N_0, \dots, N_0)$.

Since signal subspace and noise subspace are orthogonal to each other, the columns of $\mathbf{A}(\boldsymbol{\tau})$, i.e., $\mathbf{a}_1(\tau_1), \dots, \mathbf{a}_k(\tau_k), \dots, \mathbf{a}_K(\tau_K)$, lie in the signal subspace, and are therefore orthogonal to the columns of \mathbf{E}_n . Given the knowledge of \mathbf{R} , we can get the noise subspace \mathbf{E}_n out of it. The delay τ_k can then be found as the solution to

$$\|\mathbf{E}_n^* \mathbf{a}_k(\tau_k)\| = 0$$

In practice, the autocorrelation matrix is unknown and is therefore estimated by the sample autocorrelation matrix, $\hat{\mathbf{R}}$, and an estimate of \mathbf{E}_n is found in the eigenvalue decomposition of $\hat{\mathbf{R}}$

$$\hat{\mathbf{R}} = \frac{1}{L} \sum_{j=1}^L \mathbf{r}(j) \mathbf{r}^*(j) = \hat{\mathbf{E}}_s \hat{\Lambda}_s \hat{\mathbf{E}}_s^* + \hat{\mathbf{E}}_n \hat{\Lambda}_n \hat{\mathbf{E}}_n^*$$

where the columns of $\hat{\mathbf{E}}_n$ are the eigenvectors corresponding to the $N - K$

²The matrix $\mathbf{A}(\boldsymbol{\tau})$ will have full rank if and only if its columns $\mathbf{a}_1(\tau_1), \mathbf{a}_2(\tau_2), \dots, \mathbf{a}_K(\tau_K)$ are linearly independent for all possible realizations of $\boldsymbol{\tau}$. This is desirable and most likely the case for a DS-SS system in which the orthogonality of different users' signature waveforms is maintained to a maximum extent to keep their mutual interference sufficiently low. The matrix \mathbf{P} has full rank if $P_k > 0$ for all k , which is obviously the case, since we assume all the users are active.

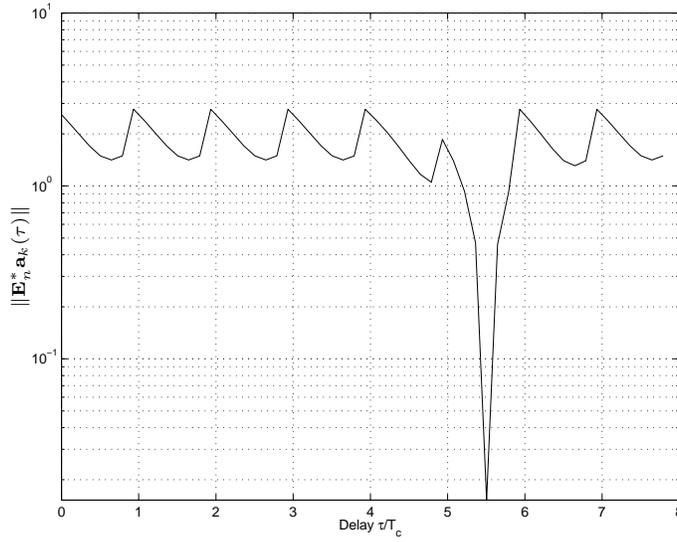


Figure 6.5: The cost function value for subspace-based algorithm, the real delay $\tau_k = 5.5T_c$, SNR= 20 dB.

smallest eigenvalues of $\hat{\mathbf{R}}$. Note that the columns of \mathbf{A} will now only be approximately orthogonal to the columns of $\hat{\mathbf{E}}_n$. However, the algorithm still works if we pick up the candidate delay value corresponding to minimum correlation between $\hat{\mathbf{E}}_n$ and $\mathbf{a}_k(\tau_k)$ as shown in Figure 6.5. In the simulation, the real delay of the signal transmitted by the k^{th} user is $5.5T_c$, the cost function has minimum value exactly at this point.

The above idea leads to the subspace-based delay estimator formulated as follows

$$\hat{\tau}_k^s = \arg \min_{\tau_k \in [0, T]} J_{k,S}(\tau_k) = \arg \min_{\tau_k \in [0, T]} \frac{\|\hat{\mathbf{E}}_n^* \mathbf{a}_k(\tau_k)\|^2}{\|\mathbf{a}_k(\tau_k)\|^2} \quad (6.17)$$

$J_{k,S}$ can be written analogously to (6.9) and (6.14) in order to estimate the delay analytically

$$J_{k,S}(\tau_k) = \frac{(K_1 + K_2 - 2K_3)\delta^2 + (2K_3 - 2K_1)\delta + K_1}{(2N - 2K_4)\delta^2 + (4K_4 - 2N)\delta + N} \quad (6.18)$$

where $K_1 = \|\hat{\mathbf{E}}_n^* \mathbf{a}_k(t_0)\|^2$, $K_2 = \|\hat{\mathbf{E}}_n^* \mathbf{a}_k(t_1)\|^2$, $K_3 = \text{Re}\{\mathbf{a}_k^*(t_0)\hat{\mathbf{E}}_n \hat{\mathbf{E}}_n^* \mathbf{a}_k(t_1)\}$, $K_4 = \mathbf{a}_k^*(t_0)\mathbf{a}_k(t_1)$.

Figure 6.4 shows this subspace-based algorithm outperforms the WSC and MMSE estimator. The performance of all the algorithms is improved with longer training sequence. Figure 6.6 shows the SNR performance of

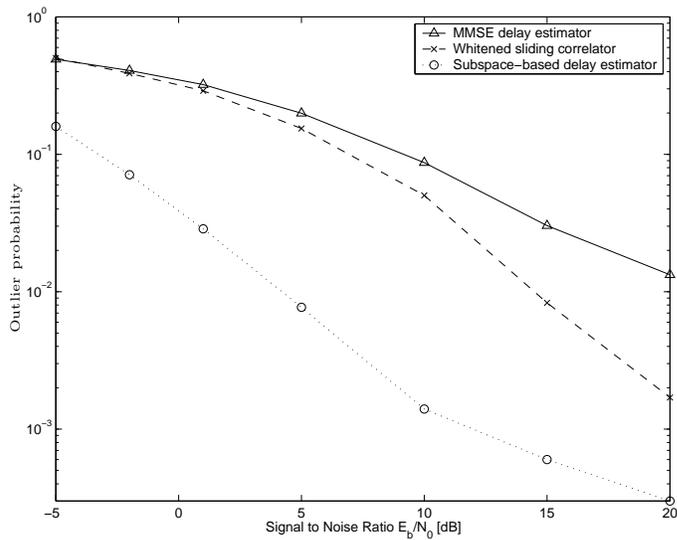


Figure 6.6: Acquisition error probability as function of SNR.

these acquisition algorithms. As expected, strong noise deteriorates the performance of all the delay estimators. A reasonable level of signal to noise ratio has to be maintained for the algorithms to work.

Equal average power among different users is assumed in the above discussion. However, fast and accurate power control is hard to obtain, and it is therefore likely that the users have different power levels. We did some simulations in the case that the users have different power levels. Figure 6.7 presents results of delay estimators as function of the near-far ratio (NFR) in case $t = 300$ pilot symbols. The NFR refers to the difference in power between each interfering user (assuming $P_2 = P_3 = \dots = P_K$), and the power P_1 of the desired user. As illustrated in Figure 6.7, the subspace-based estimator exhibits better near-far resistance compared to the WSC and MMSE delay estimator when NFR is less than 10 dB.

The problem with the estimators presented above is the slow convergence (large overhead), they all need a long training sequence to combat noise and fading and get an accurate estimate of the autocorrelation matrix. On the other hand, the overwhelming complexity of the ML delay estimator makes it unfeasible for implementation. It requires a search in K -dimensional space, thus the complexity grows exponentially with the number of users K .

In the following, some approximation methods to reduce the complexity of the ML algorithm are introduced. The general principle is to use the single user ML algorithm as the first stage to get an initial estimates of the delays. In this case, the MAI is treated as additive noise. After obtaining

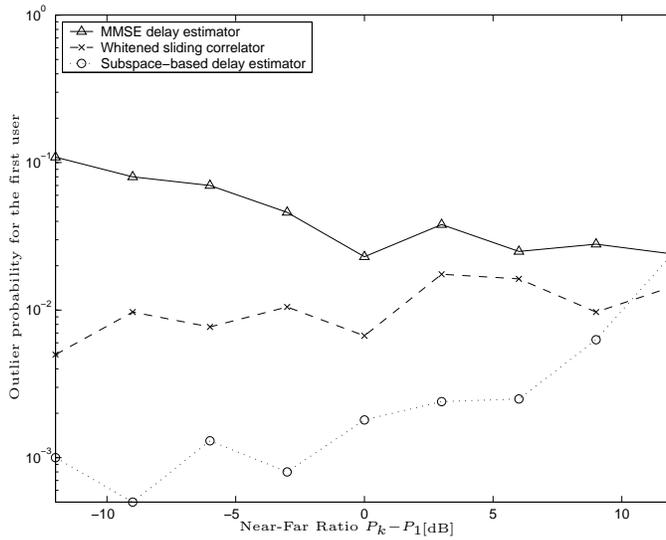


Figure 6.7: Acquisition error probability as function of NFR ($t = 300, E_b/N_0 = 20$ dB).

the initial estimates, we switch to decision directed mode and improve the results at each iteration stage, using the ideas similar to successive interference cancellation (SIC) or parallel interference cancellation (PIC). The suggested approximative ML (AML) algorithms requires much less pilot symbols, thus achieve faster convergence compared to the previously introduced schemes.

6.8 AML(1): Successive ML Delay Estimator

In [54], a hierarchic way to construct an ML approximation for delay estimation was proposed. A revised version of this algorithm adopted to the system in question is as follows:

1. Evaluate $J(\tau_k) = \text{trace}\{\mathbf{P}_{\mathbf{a}_k}^\perp \hat{\mathbf{R}}\}$ for $k = 1, 2, \dots, K$. Choose $\mathbf{a}_{n_1}(\hat{\tau}_{n_1})$ corresponding to the minimum trace value, which means we fix the delay estimate $\hat{\tau}_{n_1}$ for the user n_1 .

2. Form the matrices $\mathbf{A}_k = [\mathbf{a}_k(\tau_k) \mathbf{a}_{n_1}(\hat{\tau}_{n_1})]$ for $k = 1, 2, \dots, K$ and $k \neq n_1$. Compute $\mathbf{P}_{\mathbf{A}_k}^\perp = \mathbf{I}_N - \mathbf{A}_k \mathbf{A}_k^\dagger$, evaluate $J(\tau_k) = \text{trace}\{\mathbf{P}_{\mathbf{A}_k}^\perp \hat{\mathbf{R}}\}$ for $k = 1, 2, \dots, K$ and $k \neq n_1$. Choose $\mathbf{a}_{n_2}(\hat{\tau}_{n_2})$ corresponding to the minimum trace value.

3. Form the matrices $\mathbf{A}_k = [\mathbf{a}_k(\tau_k) \mathbf{a}_{n_1}(\hat{\tau}_{n_1}) \mathbf{a}_{n_2}(\hat{\tau}_{n_2})]$ for $k = 1, 2, \dots, K$ and $k \neq n_1, n_2$. Compute $\mathbf{P}_{\mathbf{A}_k}^\perp = \mathbf{I}_N - \mathbf{A}_k \mathbf{A}_k^\dagger$, evaluate

$J(\tau_k) = \text{trace}\{\mathbf{P}_{\hat{\mathbf{A}}_k}^\perp \hat{\mathbf{R}}\}$, and choose $\mathbf{a}_{n_3}(\hat{\tau}_{n_3})$ corresponding to the minimum trace value.

4. Repeat step 3 until all the users' delays are estimated and fixed.

The idea of this scheme is similar to that of the SIC. We fix the delay of one user at each step, that user's signal is reconstructed and used for detecting the next user's delay.

6.9 AML(2): Parallel ML Delay Estimator

The complexity of the ML algorithm can also be reduced in an iterative multistage manner like PIC. We use the conventional delay estimator (SC) to get initial estimates of delays for all the users and enter the iteration loop. The subsequent stages differ from the successive estimator introduced earlier. Instead of fixing one user's delay at a time, we fix the delays of all the interfering users simultaneously (in parallel), using the estimates derived from last iteration in order to estimate the delay for the user of interest, e.g., the k^{th} user. At the p^{th} iteration stage, the parallel ML delay estimator can be expressed as

$$\begin{aligned} \hat{\tau}_k^{(p)} &= \arg \min_{\tau_k} \sum_{j=1}^L \|\mathbf{r}(j) - \hat{\mathbf{A}}^{(p-1)}(\tau_k) \mathbf{h}(j)\|^2 \Big|_{\mathbf{h}(j) = \hat{\mathbf{A}}^\dagger \mathbf{r}(j)} \\ &= \arg \min_{\tau_k} \text{trace}\{\mathbf{P}_{\hat{\mathbf{A}}}^\perp \hat{\mathbf{R}}\} \end{aligned} \quad (6.19)$$

where $\mathbf{P}_{\hat{\mathbf{A}}}^\perp = \mathbf{I}_N - \hat{\mathbf{A}}\hat{\mathbf{A}}^\dagger$ is the orthogonal projection matrix onto the orthogonal complement to the subspace spanned by the columns of $\hat{\mathbf{A}}^{(p-1)}(\tau_k)$, which is the estimated data matrix at the $(p-1)^{\text{th}}$ stage, and is defined as

$$\hat{\mathbf{A}}^{(p-1)}(\tau_k) = [\mathbf{a}_1(\hat{\tau}_1^{(p-1)}) \cdots \mathbf{a}_k(\tau_k) \cdots \hat{\mathbf{a}}_K(\hat{\tau}_K^{(p-1)})]$$

where $\hat{\tau}_1^{(p-1)}, \dots, \hat{\tau}_K^{(p-1)}$ are the estimated delays for the interfering user $1, \dots, K$ at the $(p-1)^{\text{th}}$ stage.

In contrast to the original ML algorithm which jointly detects all the users, these two approximate ML algorithms detect the delay of one user at a time, thus greatly reduce the computational complexity compared to a full search of the ML criterion function. Like the SC, they have the property of fast convergence, require much shorter training sequence than the WSC, MMSE and subspace-based algorithms, thus significantly reduce the overhead induced by the training.

The results of the 7-stage parallel ML delay estimation are demonstrated in Figure 6.8. Clearly, the performance is improved at each iteration. However, it tends to get saturated at the 7^{th} stage.

The comparison among different estimators is illustrated in Figure 6.9.

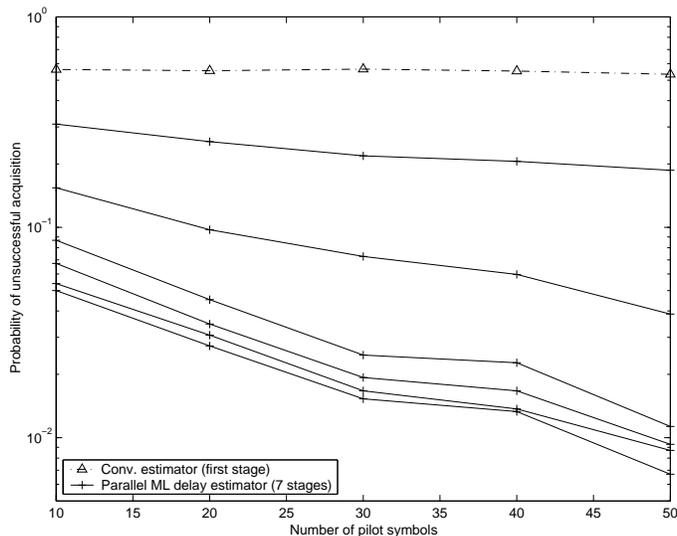


Figure 6.8: Acquisition performance of the multistage parallel ML delay estimator.

The parallel ML estimator performs slightly better than the successive one. They both achieve good acquisition performance with a relatively short training sequence. As expected, the subspace-based estimator does not work with short training sequences. Its performance improves considerably as the number of the pilots increases and eventually becomes better than the AML estimators when more than 250 pilots are used.

6.10 AML(3): Combined SML and PML

Since the successive ML (SML) and parallel ML (PML) estimator have different mechanisms for searching the optimum values of the propagation delays, we tend to think that when these two are coupled together, the combined algorithm has better chance to get out of the local minimum and approach the global optimum attainable by the original ML algorithm. That is indeed the case as shown in Figure 6.10. In the experiment, the successive scheme is used as the first stage followed by 3-stage parallel scheme. The combined algorithm performs better than each individual one. For instance, with 50 pilot symbols, the acquisition error probability goes down to 0.003 when these two methods are coupled together; while the acquisition error probability of each individual scheme is around 0.01.

To shorten the length of the training sequence, we can use the AML estimator instead of the others, e.g., the subspace-based estimator. It would be interesting to see the performance loss as a result of the reduction of

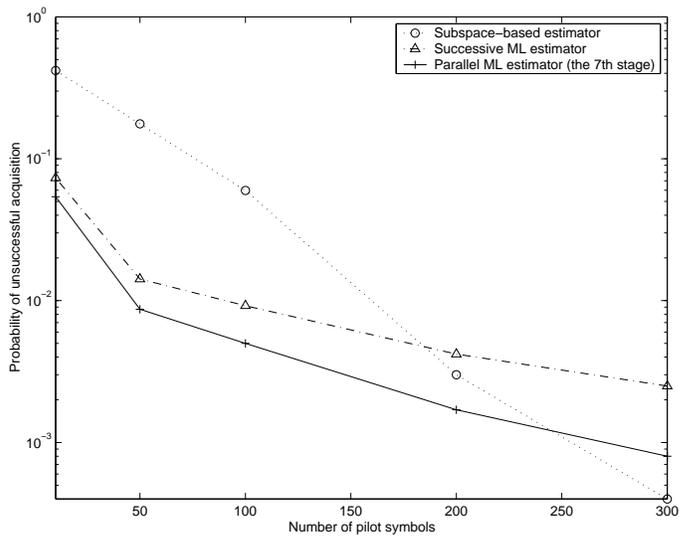


Figure 6.9: Comparison between subspace-based estimator and AML estimators.

the overhead. Figure 6.11 and 6.12 illustrate the performance comparison between the subspace-based estimator with 300 pilots and the combined ML algorithm with 40 pilots. The probability of unsuccessful acquisition is plotted against SNR and NFR respectively. Apparently, the gap between these two is not very significant. For instance, if we allow 3 – 4 dB loss in SNR at acquisition error probability of 10^{-2} , then the number of pilots can be dropped from 300 to 40. The gap between these two can be further reduced by slightly increasing the number of pilots for the AML algorithm.

6.11 Summary

In this chapter, several pilot-assisted multi-user acquisition algorithms, namely the WSC, MMSE, subspace-based, and AML estimators were introduced. Compared to the ML estimator, they reduce the computational complexity from exponential to polynomial, which makes them more feasible for implementation. We consider the use of periodic training sequence to obtain an estimate of autocorrelation matrix of the received observation vector in the presence of fading and AWGN. Simulation results show that they achieve good acquisition performance in presence of the MAI. This is in contrast with the conventional SC which does not work well in multiuser environments. All of them are non-coherent synchronization algorithms. The extension to their coherent variants is rather straightforward by incorporating the channel estimates into the equations of the acquisition algo-

rithms, as we did in the derivation of the coherent demodulation schemes in Chapter 3. We would expect some performance gain accomplished by channel estimation in this case.

By comparison, the subspace-based algorithm in general gives more reliable and more accurate estimate of delays at the expense of large overhead induced by the training. While the overhead is greatly reduced by the iterative AML algorithms at the cost of increased computational complexity which is needed for iteration process. If the receiver has fast and powerful signal processing capability, which is the case in the base station, the AML algorithms are preferred. Like the subspace-based algorithm, the MMSE estimator and WSC also need a long training sequence. They might have good performance when the channel is corrupted with colored noise, and would be preferable choices under such circumstances. This however, needs to be verified with further experiments.

All the algorithms introduced in this chapter require users' training at the same time, they are therefore applicable when we have a dedicated pilot channel for synchronization. This should be feasible because in some 3G standard, the I-channel is employed for transmitting data, while Q-channel is the control channel reserved for synchronization and channel estimation.

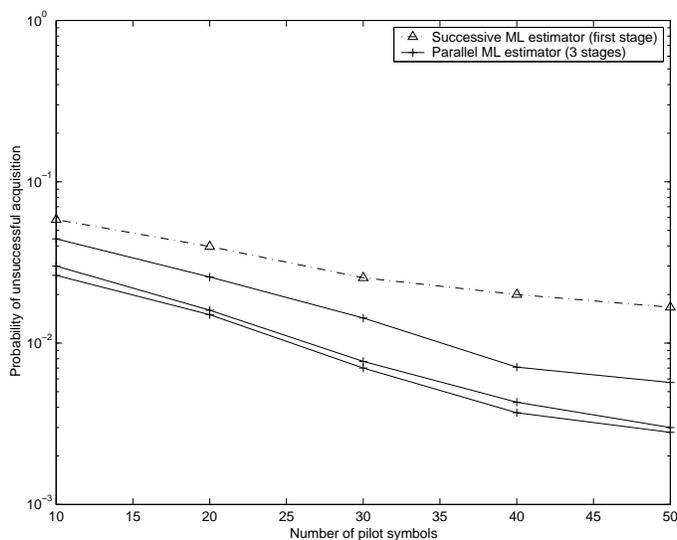


Figure 6.10: Combined successive and parallel ML delay estimator.

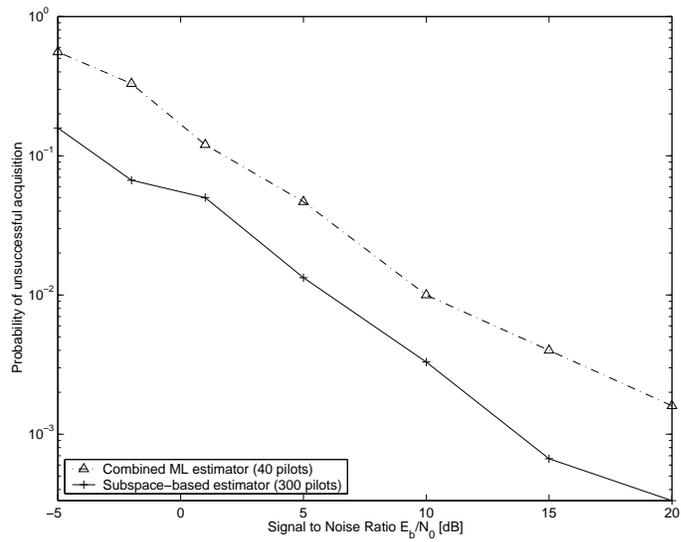


Figure 6.11: Comparison of SNR performance between subspace-based estimator and AML estimator.

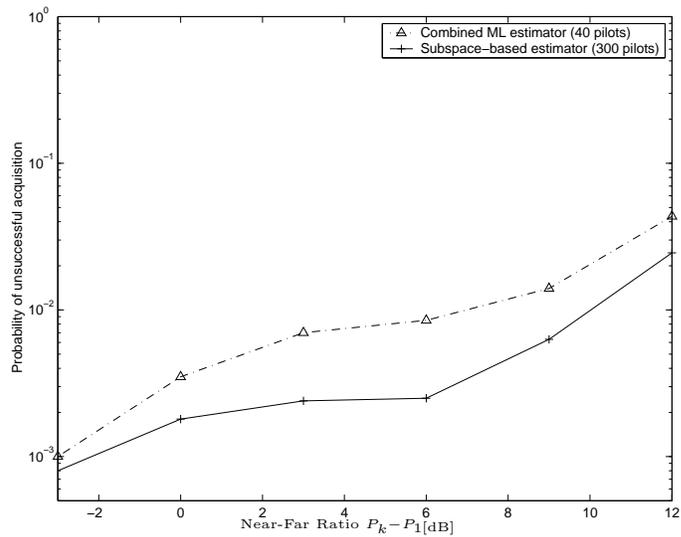


Figure 6.12: Comparison of near-far resistance between subspace-based estimator and AML estimator.

Most work examining the performance of DS-CDMA systems only considers the use of rectangular chip waveforms since the transmitted signal can be formed, and the chip matched filter can be implemented readily. This simplifies the simulation models, and considerably reduces the simulation complexity. Due to these reasons, the rectangular chip pulses are used in our previous simulations. However, they are mainly of interest to academic research. More spectrally efficient waveforms are employed in practical systems in order to satisfy the bandwidth limitations imposed by the channels and to limit the out-of-band power. Therefore, it would be important to study the behavior of the receiver algorithms when spectrally efficient pulse shapes, e.g., square root raised cosine (RRC) waveforms (see Section 1.1.2.2) are used for modulation.

The topic of chip waveform design was studied in several papers. For instance, in [55], optimum pulse shapes are designed to minimize the MAI given a restriction on out-of-band power, and chip pulses are limited to the chip duration T_c . In [56], chip waveforms, such as Blackman and Kaiser pulses, are examined and their performance in a micro-cellular packet mobile radio system is compared.

Square root raised cosine chip pulses are used in 3G systems, such as W-CDMA [57]. The effect of RRC on the bit error rate of the conventional receiver in the systems with periodic short spreading sequences was evaluated in [58], for AWGN and Rayleigh fading channels. However, the study on the effect of pulse shaping for the long-code CDMA systems, and especially for multiuser detectors is still lacking. In this part of the work, we numerically evaluate and compare the performance of rectangular pulse and square root raised cosine pulses with different rolloff factors for conventional receiver and multiuser detectors. Based on the simulation results,

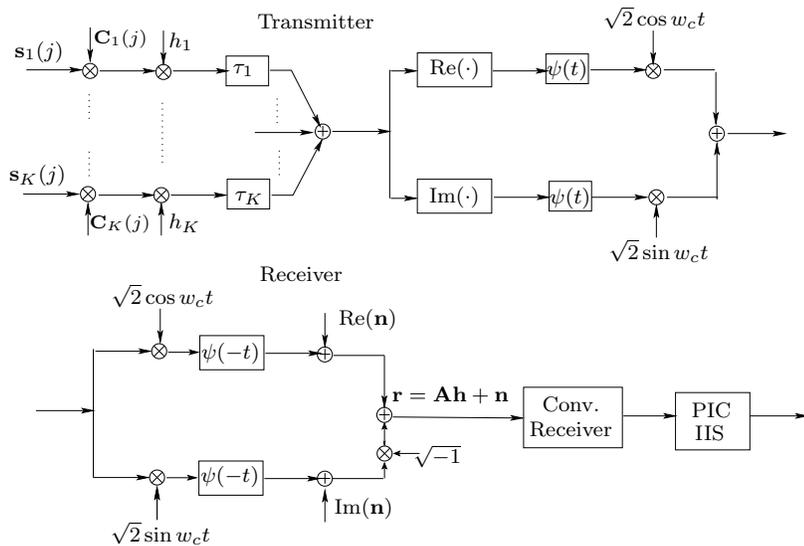


Figure 7.1: Simulation model 1 : passband scenario.

some recommendations are made on how to choose RRC pulses in practical systems, e.g., the selection of rolloff factor, truncation length, etc..

7.1 Simulation Models

The two models used for pulse shaping simulations are illustrated in Figure 7.1 and 7.2. The first one is passband model, similar to the transmitter block diagram depicted in Figure 2.1 except that we only consider single-path flat fading channel and exchange the order of pulse shaping and summation of users' signals in the transmitter to simplify the computer simulations. In this way, only two instead of $2K$ filtering operations are needed (one for I channel – the data channel, the other one for Q channel). The same principle holds for the baseband model in which the carrier modulation in the transmitter and frequency down-conversion (downmixing) in the receiver are omitted. These two models would produce the same results. For simplicity, we mainly consider the baseband equivalent model for our simulations.

It is concluded in [58] that a DS-CDMA system is rather robust against adjacent channel interference. Hence, we don't consider the presence of adjacent systems with carriers in the vicinities of the one used in the simulated system.

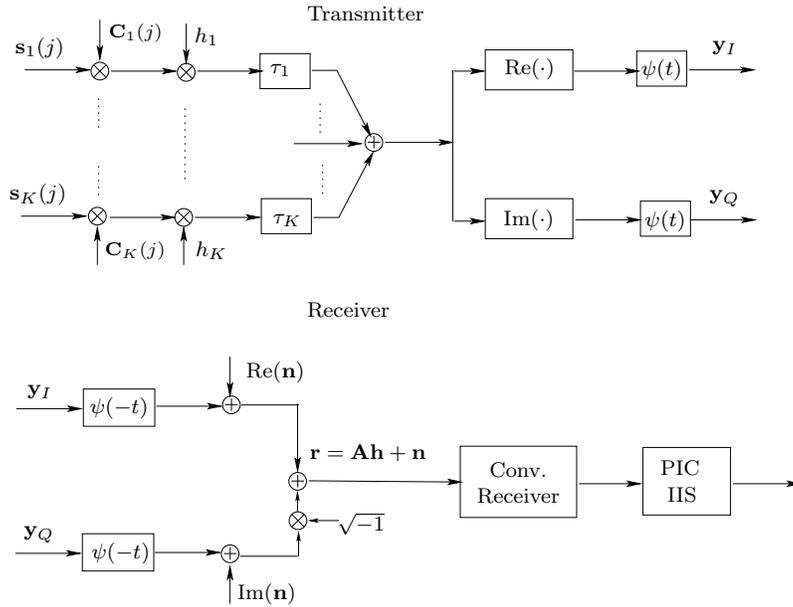


Figure 7.2: Simulation model 2 : baseband equivalent.

7.2 Numerical Results

Figure 7.3 shows the different pulses used in our simulations. The impulse response of the RRC filter is [57]

$$\psi(t) = \frac{\sin[\pi t(1-\alpha)/T_c] + 4\alpha t \cos[\pi t(1+\alpha)/T_c]/T_c}{\pi t [1 - (4\alpha t/T_c)^2]/T_c} \quad (7.1)$$

We use RRC pulses with rolloff factors $\alpha = 0.1, 0.5, 0.9$, truncated to $T_r = 4T$ chip intervals on each side of the peak. The rectangular pulse has the spectral deficiency of infinite bandwidth occupancy, which can be observed from its spectrum plot. Given the same truncation length, the sidelobes reduce as the rolloff factor α increases. However, the excess bandwidth also increases with α .

Figure 7.4, 7.5, and 7.6 show the performance of the conventional receiver, PIC, and IIS, respectively, for different pulses under the assumption of perfect synchronization. In those experiments, the parameters are configured as follows: the number of users $K = 6$, the oversampling rate $\eta = 8$, truncation interval $T_r = 4T$.

Figure 7.4 shows that all the pulses have the same performance for the conventional matched filter receiver (MF). This comes as no surprise considering that the MF is only optimal in WGN channel, it is very sensitive to MAI. When the MF is used, MAI will be the main cause for the degra-

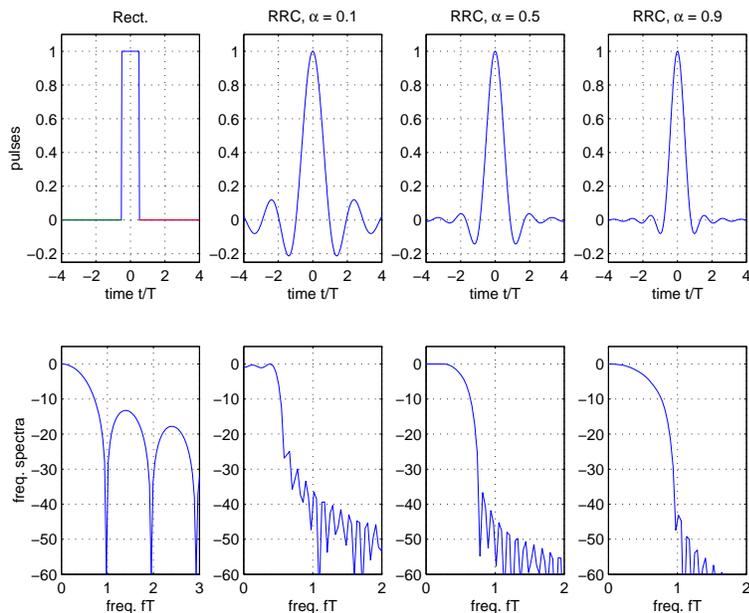


Figure 7.3: Different pulses (upper plots) and their corresponding spectra (lower plots). The x-axis for pulses is normalized to the symbol interval T , the frequency axis for spectra is normalized to the symbol rate $1/T$, the magnitude of the spectra is plotted in dB scale. All pulses are normalized to a peak value of unity.

dation of the system performance. The choice of pulses will not change the amount of interference imposed on each user. That explains why the performance of the conventional receiver is irrelevant to the choice of pulses.

Fig. 7.5 and 7.6 show the performance of different pulses for 4-stage PIC and IIS respectively. In this case, MAI can be effectively canceled or suppressed by multiuser detection schemes. Noise and ISI will be dominant factors on the system performance. In the low SNR region, the noise is dominant, the users have the same amount of noise regardless of what pulses are chosen, therefore, all the pulses have approximately the same performance. In the high SNR region, the performance is mainly influenced by ISI because noise is weak. The RRC pulses with low rolloff factor (e.g., $\alpha = 0.1$) introduce more ISI due to larger out-of-band power caused by truncation, therefore perform slightly worse than the ones with higher rolloff factor (e.g., $\alpha = 0.5, 0.9$). When SNR increases, ISI becomes more and more dominant, the distinction becomes clearer.

Rectangular pulse gives a lower bound for the BER performance in case of perfect synchronization. It is finite in time, does not need any truncation, therefore, does not introduce any ISI if the channel is band unlimited.

The 4-stage PIC (Fig. 7.5) almost exhibits the identical results as IIS

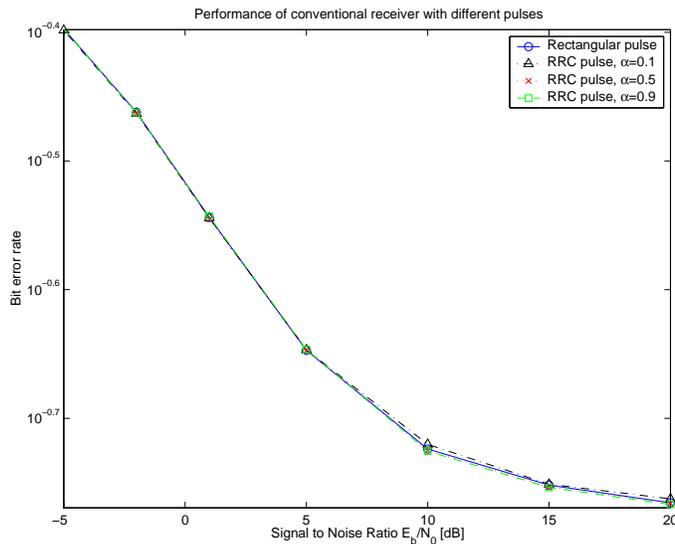


Figure 7.4: Performance of different pulses for conventional receiver ($\eta = 8$, $T_r = 4T$, $K = 6$).

(Fig. 7.6). It is reasonable to assume that the RRC pulses behave similarly for all the linear and nonlinear multiuser detectors. We shall use 4-stage PIC in all the subsequent experiments.

Perfect synchronization is assumed in the investigations we have done so far. To see the effect of imperfect synchronization on different pulse shapes, we measure their performance versus synchronization errors in Figure 7.7 and 7.8. We can see that the rectangular pulse is more sensitive to synchronization errors than the RRC pulses with large rolloff factor, e.g., $\alpha = 0.9$, but more robust to synchronization errors than the RRC pulses with small rolloff factor, e.g., $\alpha = 0.1$.

The eye diagrams are useful visual tools for evaluating the performance of different pulse shapes. They are generated using an oscilloscope connected to the output of the chip matched filter and before the sampler. Therefore, what we see on the oscilloscope is the convolution of the chip waveform $\psi(t) * \psi(-t)$. The convolution of a rectangular pulse is a triangular pulse, and the convolution of a RRC pulse becomes a raised cosine (RC) pulse. The oscilloscope is re-triggered at every symbol period. By relying on the persistence of a typical oscilloscope display, the result is an overlaying of consecutive received symbol waveforms which form 'eye' pattern on the screen. We re-create this effect with computer simulation and plot it in Figure 7.9.

As demonstrated in the diagrams, the eye pattern of the triangular pulse is more open compared to the eye pattern of the RC pulse with $\alpha = 0.1$.

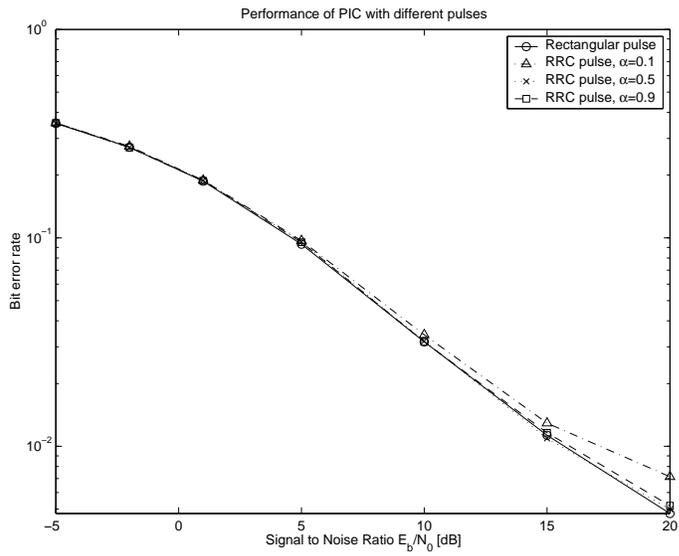


Figure 7.5: Performance of 4-stage PIC as function of SNR ($\eta = 8, T_r = 4T, K = 6$).

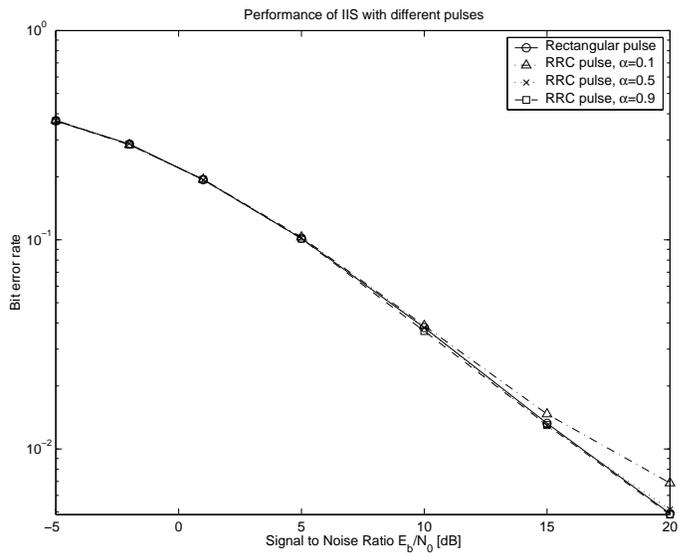


Figure 7.6: Performance of different pulses for 4-stage IIS ($\eta = 8, T_r = 4T, K = 6$).

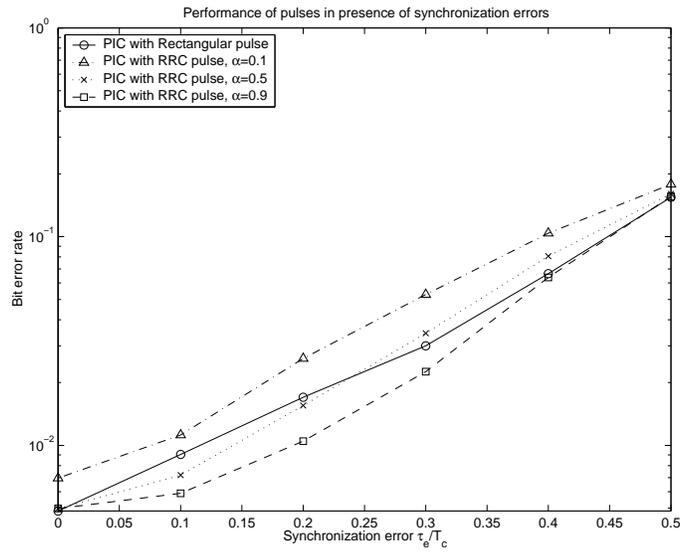


Figure 7.7: Performance of different pulses as function of synchronization errors ($\eta = 10, T_r = 4T, K = 6$).

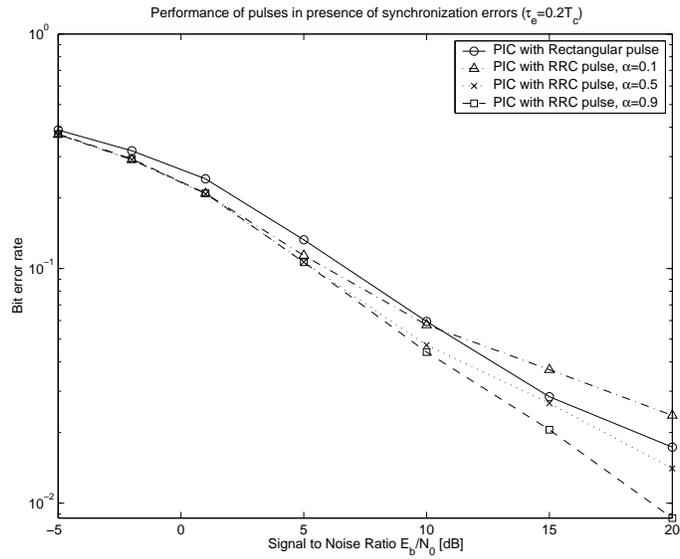


Figure 7.8: Performance of different pulses in presence of synchronization errors ($\eta = 10, T_r = 4T, K = 6, \tau_e = 0.2T_c$).

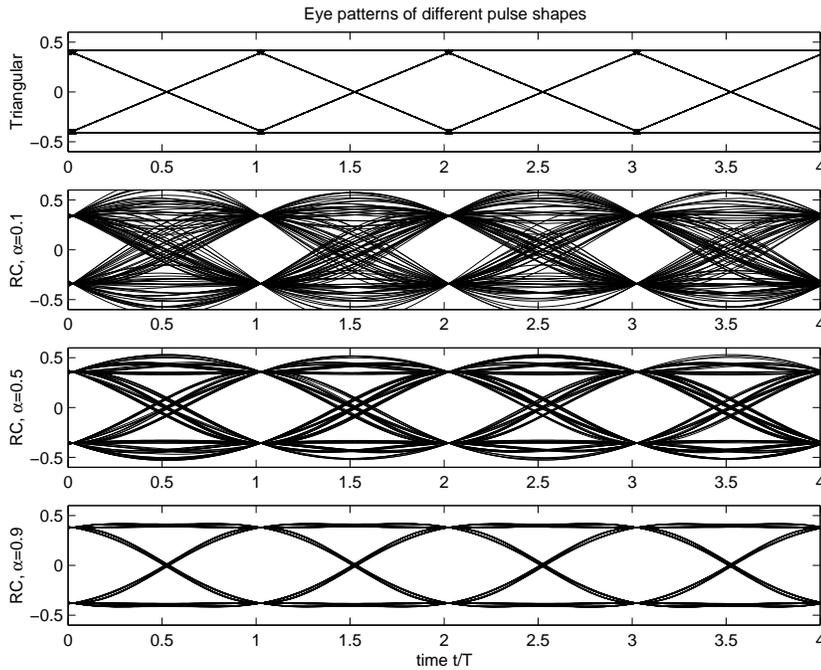


Figure 7.9: Eye diagrams of employed pulse shapes.

That explains why the rectangular pulse is more robust to synchronization errors than the RRC pulse with $\alpha = 0.1$. Because it has a wider eye opening, thus exhibits a larger margin against MAI and additive noise. The same argument can be made reversely to explain why the rectangular pulse is more sensitive to synchronization errors than the RRC pulse with $\alpha = 0.9$.

The eye opening of the RC pulse with $\alpha = 0.5$ is nearly as wide as the triangular pulse. It is, however, more flat around the optimum sampling point, thus more robust to small synchronization errors (up to $\tau_e = 0.25T_c$). This is also verified in Figure 7.8. When the synchronization error $\tau_e = 0.2T_c$, the BER curve for RRC pulse with $\alpha = 0.5$ lies below the one for rectangular pulse.

When the synchronization error τ_e reaches half chip interval, none of the pulses has any margin against MAI and additive noise. In this case, the decision is entirely determined by MAI and noise. Therefore all pulses converge to approximately the same error rate as demonstrated in Figure 7.7.

In Figure 7.10, we measure the performance of different pulses versus oversampling rate η to see what would be proper value of η in simulations when evaluating the performance of an RRC pulse. Apparently, η has to be no less than 5. The performance gets saturated when it goes beyond

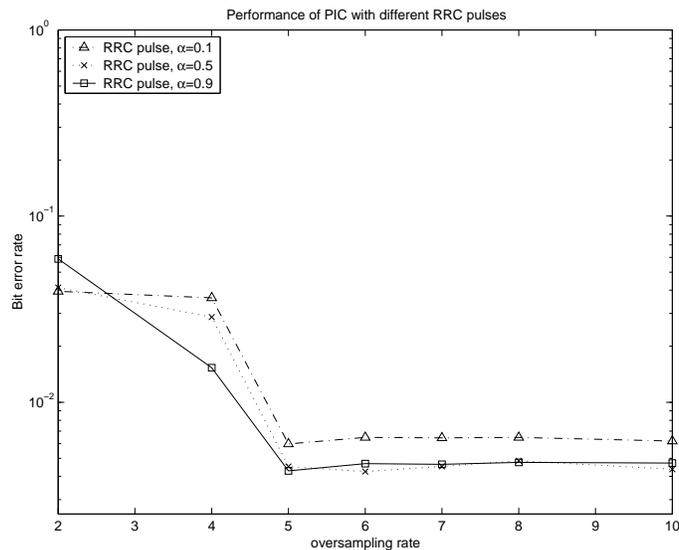


Figure 7.10: The impact of oversampling rate η on the simulation results (4-stage PIC, $T_r = 4T$, $K = 6$).

this point.

Note that all the RRC pulses have infinite length. We have to do truncation to make them realizable. In Figure 7.11, we measure the performance of different RRC pulses as function of the truncation interval T_r , attempting to find out a suitable truncation length for different pulses. As expected, the RRC pulses with smaller α value need longer truncation interval, and vice versa. Even $T_r = T$ is enough for a RRC pulse with rolloff factor $\alpha = 0.9$. Generally speaking, $T_r = 5T$ would be sufficient for all RRC pulses. In this simulation, no bandwidth constraint is imposed on the channel. Therefore, the out-of-band power caused by the truncation is not taken into account.

Shorter truncation length will reduce the complexity of pulse shaping filter, but also lead to larger sidelobes of the signal spectrum (increase of the out-of-band power as illustrated in Figure 7.12), which in turn will increase the intersymbol interference (ISI) and degrade the system performance. In order to test the impact of truncation in a more realistic scenario, we impose the bandwidth constraint and feed the signals through a bandpass channel with bandwidth of the pass band equal to $0.75/T$ on each side of the carrier frequency, which is approximately the bandwidth required by the RRC pulse with $\alpha = 0.5$ (its frequency response is depicted by the solid line in Fig. 7.13). Figure 7.14 shows similar results compared to the case of no bandwidth constraint (Figure 7.11). We can conclude that small amount of the out-of-band power can be tolerated without causing

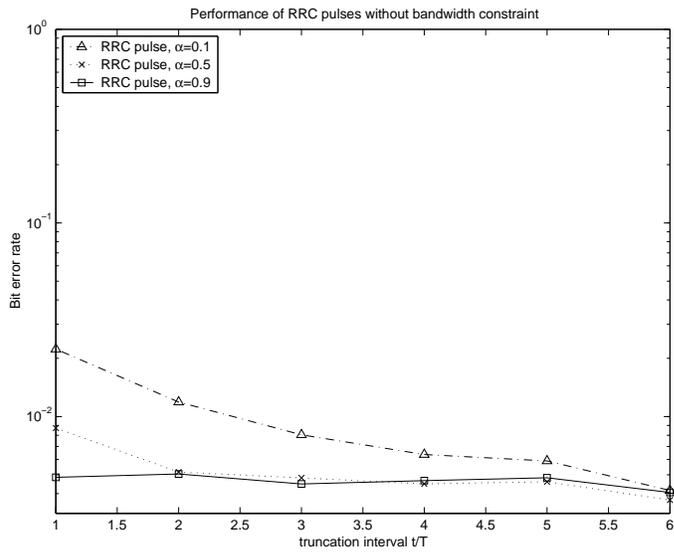


Figure 7.11: The impact of truncation on the performance of different pulses (4-stage PIC, $\eta = 8$, $K = 6$, no bandwidth constraint).

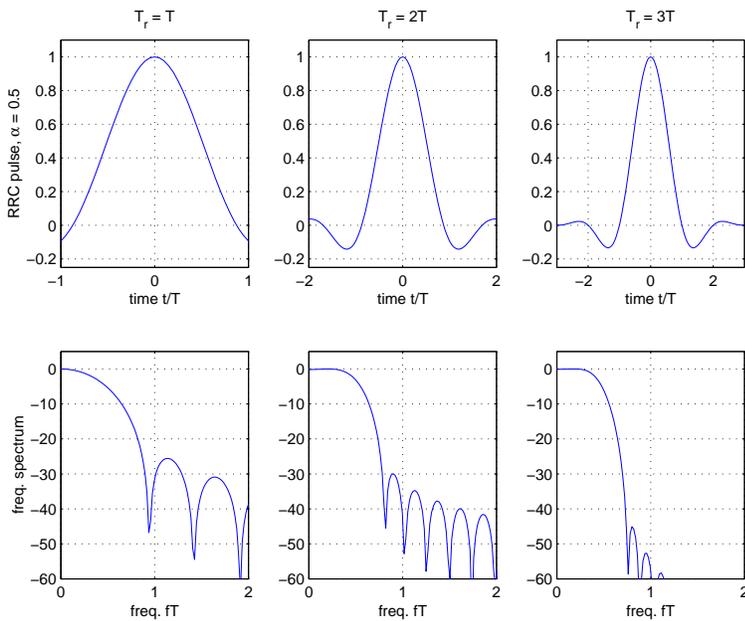


Figure 7.12: The impact of truncation on signal spectrum. The tested pulse is a RRC pulse with $\alpha = 0.5$.

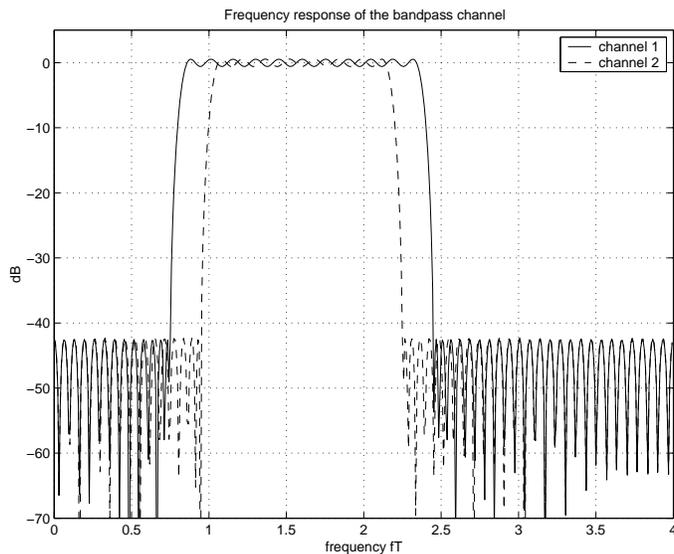


Figure 7.13: The bandpass channels used in the simulation.

noticeable performance degradation.

We then narrow down the pass band of the channel to $0.5/T$ on each side of the carrier frequency, which is approximately the bandwidth required by the RRC pulse with $\alpha = 0.1$ (see the dashed curve in Fig. 7.13). Figure 7.15 shows that the performance of the RRC pulse with $\alpha = 0.9$ becomes worse due to the relatively large out-of-band power. In this case, it is better to use RRC pulses with low rolloff factor value, like $\alpha = 0.1, 0.5$ in order to fit into the narrow bandwidth of the channel.

7.3 Summary

The selection of chip waveform affects not only the bandwidth efficiency, but also the performance of a DS-CDMA system. Pulses should be designed to minimize the MAI and ISI. The bandwidth constraint of practical systems generally precludes the use of rectangular pulse. However, we have justified in this work that employing rectangular pulse is an effective way of simplifying computer simulations when evaluating the performance of receiver algorithms, because it yields similar results to the ones produced by the RRC pulses. However, one has to be aware that the RRC pulses with low rolloff factors perform slightly worse, especially when the noise level is low. They are also more sensitive to synchronization errors. Due to those reasons, the RRC pulses with large rolloff factor are preferred in practice in order to make the systems more resistant to ISI and synchronization errors,

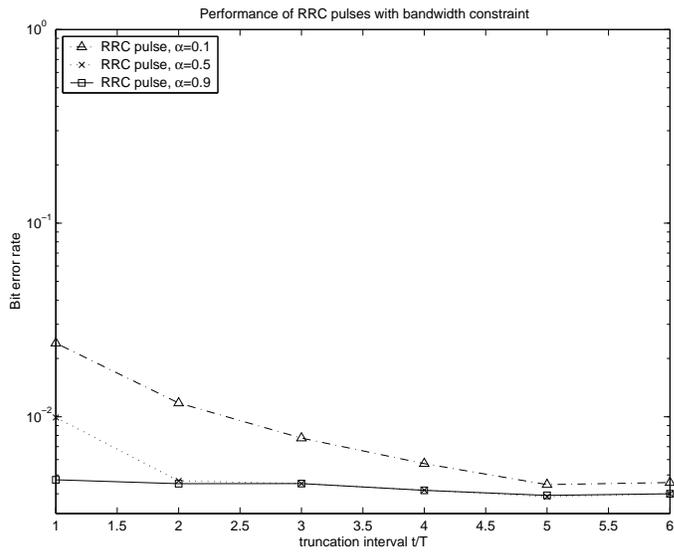


Figure 7.14: The impact of truncation on the performance of different pulses (4-stage PIC, $\eta = 8$, $K = 6$, bandpass channel 1 is imposed).

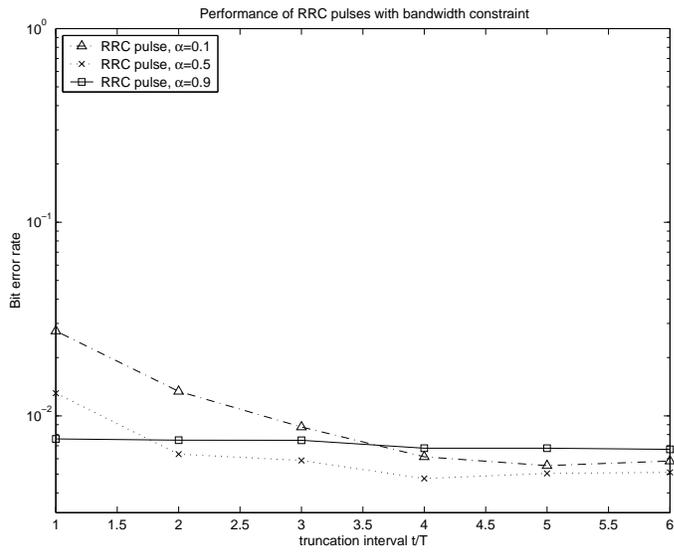


Figure 7.15: The impact of truncation on the performance of different pulses (4-stage PIC, $\eta = 8$, $K = 6$, bandpass channel 2 is imposed).

as well as to simplify the filter design as discussed below. The price to pay for the better performance is larger bandwidth requirement. The excess bandwidth increases linearly with the rolloff factor. This performance and bandwidth trade-off has to be considered when selecting chip waveforms.

We have shown that moderate out-of-band power caused by the truncation of RRC pulses can be tolerated without causing noticeable performance degradation. To minimize the filter complexity without loss of system performance, we recommend a truncation interval $T_r = T - 5T$ depending on the rolloff factor. In general, the higher value of rolloff factor, the shorter truncation length is needed, which means simpler filter design.

Although the simulation results are derived with long-code CDMA systems, we believe the conclusions drawn above are widely applicable, e.g., for short-code CDMA systems as well as non-spread spectrum systems.

Part II

Coded System

In the second part of this thesis, we study the orthogonally modulated DS-CDMA system with convolutional encoding over time-varying frequency selective Rayleigh fading channels. Convolutional codes are employed to further improve the performance and power efficiency of the system. It is believed that CDMA systems exhibit their full potential, when combined with forward error correction coding (FEC) [59].

The block diagram of the transmitter is shown in the upper part of Fig. 8.1. The k^{th} user's l^{th} information bit is denoted as $b_k[l] \in \{+1, -1\}$ ($k = 1, \dots, K$, $l = 1, \dots, L_b$, and L_b is the block length). The information bits are convolutionally encoded into code bits $\{u_k[l^n]\} \in \{+1, -1\}$, where $u_k[l^n]$ denotes the n^{th} code bit due to $b_k[l]$. For example, in case of a rate $1/3$ code, $b_k[l]$ is encoded into $u_k[l^0], u_k[l^1], u_k[l^2]$.

Code bits are subsequently interleaved and each block of $\log_2 M$

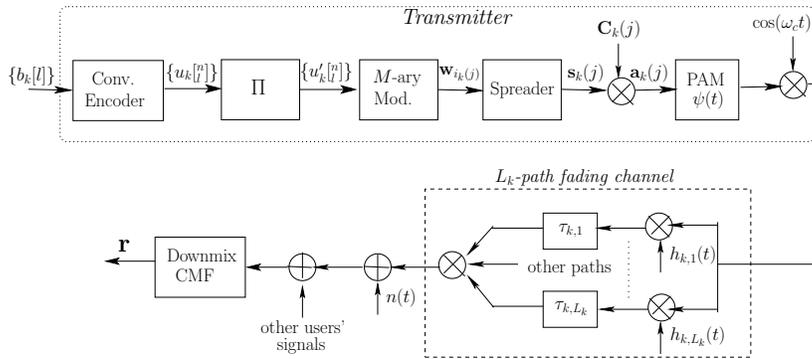


Figure 8.1: Transmitter, channel and receiver front end.

Table 8.1: Mapping between input bits and Walsh codewords. The Walsh chip sequence \mathbf{s}_m is derived by repeating (spreading) each bit of \mathbf{w}_m by $N/8$ times.

Code bits	Symbol index	Walsh codeword
$u'_k[l] \quad u'_k[l] \quad u'_k[l]$	$m = i_k(j)$	\mathbf{w}_m
+1 +1 +1	0	$\mathbf{w}_0 : +1 \quad +1$
+1 +1 -1	1	$\mathbf{w}_1 : +1 \quad +1 \quad +1 \quad +1 \quad -1 \quad -1 \quad -1 \quad -1$
+1 -1 +1	2	$\mathbf{w}_2 : +1 \quad +1 \quad -1 \quad -1 \quad +1 \quad +1 \quad -1 \quad -1$
+1 -1 -1	3	$\mathbf{w}_3 : +1 \quad +1 \quad -1 \quad -1 \quad -1 \quad -1 \quad +1 \quad +1$
-1 +1 +1	4	$\mathbf{w}_4 : +1 \quad -1 \quad +1 \quad -1 \quad +1 \quad -1 \quad +1 \quad -1$
-1 +1 -1	5	$\mathbf{w}_5 : +1 \quad -1 \quad +1 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1$
-1 -1 +1	6	$\mathbf{w}_6 : +1 \quad -1 \quad -1 \quad +1 \quad +1 \quad -1 \quad -1 \quad +1$
-1 -1 -1	7	$\mathbf{w}_7 : +1 \quad -1 \quad -1 \quad +1 \quad -1 \quad +1 \quad +1 \quad -1$

coded and interleaved bits $\{u'_k[l]\} \in \{+1, -1\}$ is mapped into $\mathbf{w}_{i_k(j)} \in \{\mathbf{w}_0, \dots, \mathbf{w}_m, \dots, \mathbf{w}_{M-1}\}$, which is one of the M Walsh codewords. The subscript $i_k(j) \in \{0, 1, \dots, M-1\}$ denotes the k^{th} user's j^{th} Walsh symbol index. The index of the $\log_2 M$ systematic bits of each Walsh codeword $\mathbf{w}_{i_k(j)}$ is given by

$$i = \frac{M}{2^{s+1}}, \quad s = 0, 1, \dots, \log_2 M - 1$$

In case $M = 8$, the mapping rule is given in Table 8.1 below. Three systematic bits are $\mathbf{w}_{i_k(j)}^1, \mathbf{w}_{i_k(j)}^2, \mathbf{w}_{i_k(j)}^4$, where $\mathbf{w}_{i_k(j)}^p$ denotes the p^{th} bit of the codeword. The columns corresponding to the systematic bits are highlighted in the table.

The interleaver and deinterleaver are denoted as Π and Π^{-1} , respectively, in Fig. 8.1 and the following figures. The purpose of interleaving is to separate adjacent code bits in time so that, ideally, each code bit will experience independent fading.

The system model presented here differs from the one in Chapter 2 in how the Walsh symbols are formed. The remaining part is almost identical. It is, however, still repeated below to ease reading.

The Walsh codeword, $\mathbf{w}_{i_k(j)} \in \{+1, -1\}^M$, is then repetition encoded into

$$\mathbf{s}_k(j) = \text{rep}\{\mathbf{w}_{i_k(j)}, N/\log_2(M)\} \in \{+1, -1\}^N \quad (8.1)$$

where $\text{rep}\{\cdot, \cdot\}$ denotes the repetition encoding operation, where its first argument is the input bits and the second one is the repetition factor. Therefore, each bit of the Walsh codeword is spread (repetition coded) into $N_c = N/M$ chips, and each Walsh symbol is represented by N chips and denoted as $\mathbf{s}_k(j) \in \{\mathbf{s}_0, \dots, \mathbf{s}_{M-1}\}$.

The Walsh sequence $\mathbf{s}_k(j)$ is then scrambled (randomized) by a scrambling code unique to each user to form the transmitted chip sequence

$$\mathbf{a}_k(j) = \mathbf{C}_k(j)\mathbf{s}_k(j) \in \{+1, -1\}^N \quad (8.2)$$

where $\mathbf{C}_k(j) \in \{-1, 0, +1\}^{N \times N}$ is a diagonal matrix whose diagonal elements correspond to the scrambling code for the k^{th} user's j^{th} symbol. We still consider the use of long codes, e.g., the scrambling code differs from symbol to symbol. The purpose of scrambling is to separate users. From equation (8.2), one can see that scrambling is accomplished by chip-wise multiplication, and does not introduce any rate change.

The scrambled sequence $\mathbf{a}_k(j)$ is pulse amplitude modulated using a unit-energy chip waveform $\psi(t)$ to form the baseband signal. For simplicity, we assume that $\psi(t)$ is a rectangular pulse with support $t \in [0, T_c)$ (the chip duration is denoted by T_c , its relation with symbol duration T is $T = NT_c$); however, the proposed methods in this thesis can be extended for other waveforms, e.g., square-root raised cosine pulses.

The baseband signal is multiplied with a carrier and transmitted over a Rayleigh fading channel with noise power spectral density $N_0/2$ and with L_k resolvable paths, having time-varying complex channel gains $h_{k,1}(t), h_{k,2}(t), \dots, h_{k,L_k}(t)$ and delays $\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L_k}$ (see the lower part of Fig. 8.1). The received signal is the sum of K users' signals plus additive white complex Gaussian noise $n(t)$. After frequency down-conversion and chip matched filtering (CMF), the received signal $\mathbf{r}(k, j) \in \mathbb{C}^{N_k}$ corresponding to the k^{th} user's j^{th} transmitted Walsh sequence $\mathbf{s}_k(j)$ can be written in vector form as

$$\begin{aligned} \mathbf{r}(k, j) &= \mathbf{A}(k, j)\mathbf{h}(j) + \mathbf{n}(k, j) \\ &= \mathbf{X}_k(j)\mathbf{h}_k(j) + \text{ISI}(k, j) + \text{MAI}(k, j) + \mathbf{n}(k, j) \end{aligned} \quad (8.3)$$

where the columns of the matrix $\mathbf{A}(k, j)$ are the delayed version of transmitted chip sequences $\mathbf{a}_k(j)$ for $k = 1, 2, \dots, K$, one column per path. The length of the processing window N_k , is larger than the symbol interval N to account for the asynchronous and multipath nature of the channel. The columns are weighted together by $\mathbf{h}(j)$, whose elements are the path gains of all users' paths. The received vector $\mathbf{r}(k, j)$ can be written as the sum of four terms: the signal of interest $\mathbf{X}_k(j)\mathbf{h}_k(j)$, the intersymbol interference (ISI), the multiple access interference (MAI), and the noise represented by $\mathbf{n}(k, j)$ which is a vector of complex noise samples with zero mean and variance N_0 . The columns of the matrix $\mathbf{X}_k(j)$ are essentially the shifted versions of the chips due to the k^{th} user's j^{th} symbol, one column per path (the shift is determined by the path delay). The vector

$$\mathbf{h}_k(j) = [h_{k,1}(jT) \ h_{k,2}(jT) \ \dots \ h_{k,l}(jT) \ \dots \ h_{k,L_k}(jT)]^T$$

corresponds to the channel gains of the k^{th} user's paths, it is part of $\mathbf{h}(j)$ as shown in (2.4).

To facilitate reading, the frequently used acronyms and notations are summarized in Table 8.2.

Table 8.2: Acronyms and notations used in Part two of the thesis

CE	channel estimation
MAP	maximum a posteriori
VA	Viterbi algorithm
HIVA	hard input VA
SIVA	soft input VA
SOVA	soft output VA
MF	matched filter
IC	interference cancellation
IS	interference suppression
HDIC	hard (decision) interference cancellation
SDIC	soft (decision) interference cancellation
HDIS	hard (decision) interference suppression
SDIS	soft (decision) interference suppression
SISO	soft-input, soft-output
LLR	log-likelihood ratio
$i_k(j)$	k^{th} user's j^{th} Walsh symbol index
$\mathbf{w}_{i_k(j)}$	k^{th} user's j^{th} Walsh codeword
$\mathbf{w}_{i_k(j)}^p$	p^{th} bit of $\mathbf{w}_{i_k(j)}$
$b_k[l]$	k^{th} user's l^{th} information bit
$u_k[l]^n$	n^{th} code bit due to $b_k[l]$
$u'_k[l]^n$	interleaved version of $u_k[l]^n$
$\mathbf{C}_k(j)$	k^{th} user's scrambling matrix for the j^{th} symbol
$\mathbf{s}_k(j)$	k^{th} user's j^{th} transmitted chip sequence
$h_{k,l}$	l^{th} path complex channel gain for k^{th} user's j^{th} symbol
$\lambda(\cdot, ; I)$	unconstrained LLR at input of a SISO device
$\lambda(\cdot, ; O)$	modified LLR at output of a SISO device
$L(\cdot)$	soft input or output of non-SISO devices

In conventional systems, M -ary symbol demodulation and convolutional decoding are conducted separately in the receiver, only hard decisions are passed between these two blocks. In this chapter, we propose several iterative schemes based on some soft demodulation and decoding algorithms. Instead of making hard decision on the transmitted M -ary symbols from the received observations, we compute the reliability value for the code bits from which orthogonal symbols are formed. This soft information is then deinterleaved and decoded. The detected bits (either hard or soft estimates) can be fed back to demodulator for channel estimation and multiuser detection. For channel decoding, the soft-output MAP (maximum a posteriori) or Log-MAP algorithm can be used instead of the VA (Viterbi algorithm) for better performance. Maximum achievable performance for the system is obtained by iterating this soft demodulation and VA/Log-MAP decoding process. The performance of different strategies are evaluated numerically and proved to achieve substantial performance gain compared to the conventional hard decision based scheme, especially when the soft demodulator is assisted by decision directed channel estimation and interference cancellation/suppression techniques, and also when demodulation and decoding are performed jointly in an iterative manner. Additional enhancement is noticed when VA is replaced by Log-MAP decoder. It is also shown that the interference cancellation and channel estimation based on the soft decision feedback further improves the system performance compared to the hard decision directed approach with minor increase in the complexity.

9.1 Introduction

The conventional receiver in a DS-CDMA system consists of a bank of matched filters, each matched to a particular user's signature waveform. In such a receiver, multiple access interference (MAI) is considered as noise and is not exploited by the receiver. Multiuser detection is an effective tool to increase the capacity of interference-limited CDMA systems and alleviate some technical requirements, such as power control. Under a wide range of conditions, e.g., low user correlations, MUD allows uncoded asynchronous DS-CDMA systems to achieve performance comparable to that of the single-user system [60].

Several iterative MUD schemes were proposed in the first part of this thesis for uncoded M -ary orthogonal systems with affordable complexity and performance much better than the standard receiver, especially in high-capacity networks in which the interference from other users is large. In order to fully explore the potential of multiuser detectors, we need to acquire accurate measurements of the fading channel to do coherent detection or interference cancellation. We showed that the use of iterative multiuser detection with decision-directed channel estimation provides substantial capacity gains compared to the conventional receiver.

Convolutional coding is employed in this system to improve the error correcting capability and power efficiency of the system. Combined with FEC coding, MUD can overcome its limitations in highly correlated multiuser systems [61]. Therefore, in some proposed systems, MUD is employed in conjunction with FEC coding to obtain greater capacity and throughput.

The problem of joint multiuser detection and decoding was treated, e.g., in [21, 22, 60, 61, 62, 63, 64]. Soft interference cancellation, linear MMSE filtering, or trellis based Log-MAP multiuser detector, etc. were proposed in those papers to reduce the deteriorative effect of interference before single user decoding is done. However, the algorithms developed in the above papers are constrained to uncascaded systems with a single convolutional code, and the issue of joint detection/decoding and channel estimation is not investigated except in [21] where a soft input MMSE channel estimation algorithm was proposed. If soft information is to be used for channel estimation and interference cancellation, a serially concatenated system would be rather different from the non-concatenated systems in that the soft values are not directly available for all the inner code bits from the outer decoder. In particular, in our case, we can only extract the soft information for the systematic bits of the Walsh codewords from the Log-MAP decoder, which necessitates the design of a soft modulator to derive the soft estimates for parity bits. This chapter is devoted to the research on those unexplored topics. With emphasis on the development of soft demodulation algorithms, we investigate different approaches to iterative soft demodulation and decoding for this serially concatenated CDMA system with orthogonal modulation and convolutional coding in a multi-user

environment over multipath fading channels. Hard/soft decision-directed interference mitigation and channel estimation are proposed to improve the reliability of the demodulation process. Different channel decoding algorithms, e.g., VA and Log-MAP algorithms are considered and their performance compared. Two alternatives of interfacing inner demodulator and outer decoder, namely partitioned approach and integrated approach [62], are studied. The former one treats demodulation and decoding separately; while the latter one performs them together in the iteration process.

9.2 Demodulation and Decoding Schemes

The task of the receiver is to detect the information bits transmitted from all users, i.e., detect $b_k[l]$ ($l = 1, 2, \dots, L_b$ and $k = 1, 2, \dots, K$) given the observation $\mathbf{r}(k, j)$, for $k = 1, 2, \dots, K$ and $j = 1, 2, \dots, \frac{L_c}{\log_2 M}$. To this end, first we need to demodulate the received signal to transmitted Walsh sequence $\mathbf{s}_k(j)$ or directly to the code bits $\{u'_k[l]\}$ which are subsequently convolutionally decoded to obtain an estimate of $\{b_k[l]\}$. Four different demodulation and decoding strategies are studied in this chapter and outlined in Fig. 9.1. The first one is the conventional scheme with partitioned hard demodulation and HIVA decoding; the second one is still partitioned approach, but with soft demodulation and SIVA; the third scheme is an integrated approach with soft demodulation and SIVA decoding; the last one is an integrated approach with soft demodulation and Log-MAP decoding. Different iterative multistage demodulators are used for different schemes. The third layer of the tree diagram indicates the types of demodulator applicable for each scheme. One can see from the diagram that the MF-based noncoherent demodulation is always used at the initial stage; HDIC/HDIS, SDIC/SDIS based soft demodulators can be employed at the subsequent stages for both scheme 2 and 4. However, the hard decisions used for HDIC/HDIS and soft decisions used for SDIC/SDIS are derived differently for these two schemes. The hard version of scheme 2 (4) is named scheme 2.1 (4.1), the soft version is named scheme 2.2 (4.2). Readers are referred to Table 8.2 for the acronyms and abbreviations. The design and implementation of different algorithms are discussed next.

9.2.1 Scheme 1: partitioned hard demodulation, HIVA decoding

The conventional scheme is delineated in Fig. 9.2. Based on the received observation $\mathbf{r}(k, j)$, the receiver makes hard decisions on the transmitted sequences $\hat{\mathbf{s}}_k(j) \in \{\mathbf{s}_0, \dots, \mathbf{s}_{M-1}\}$ using symbol matched filter or more advanced multiuser detectors. The detected symbols are converted to bits $\{\hat{u}'_k[l]\}$. In case $M = 8$, the conversion rule is specified in Table 8.1. The

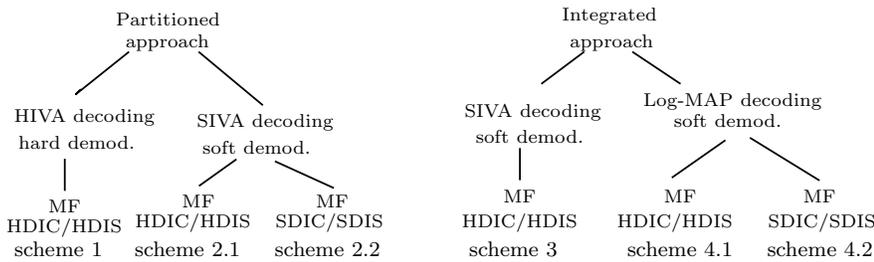


Figure 9.1: Different demodulation and decoding schemes.

converted bits $\{\hat{u}'_k[l]\}$ are then deinterleaved and decoded with hard input VA (HIVA) decoder.

In case a multistage multiuser detector is used, the iteration is only inside the demodulation block. The performance is improved in an iterative manner due to the fact that the channel is more accurately measured and interference is better detected (meaning better interference mitigation), as the iteration goes on.

Since only hard decisions are passed from demodulator to decoder, performance loss is inevitable. It is a well known fact that for VA, hard-decision decoding is 2 – 3 dB inferior to soft-decision decoding in an additive white Gaussian noise channel and the gap is even greater in presence of fading. In order to enable soft decoding, we propose three soft demodulation algorithms in Section 9.3. Different ways of interfacing the soft demodulator and channel decoder are introduced below. They are presented in order of increased complexity and improved performance.

9.2.2 Scheme 2: partitioned soft demodulation, SIVA decoding

Instead of estimating the transmitted M -ary sequence $\mathbf{s}_k(j)$ and then converting them into bits $\{u'_k[l]\}$, we can directly derive soft reliability value for each bit $u'_k[l]$ from the received vector $\mathbf{r}(k, j)$. Assuming bits +1 and

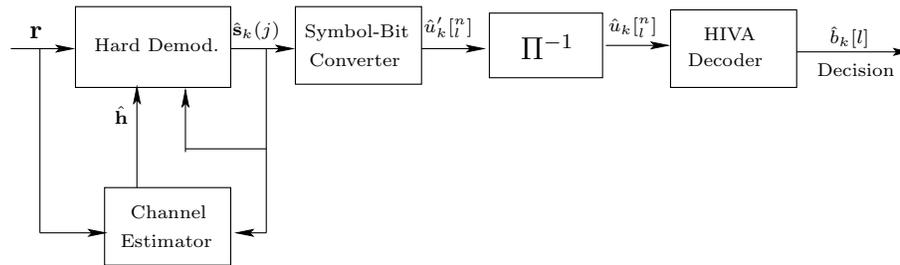


Figure 9.2: Conventional demodulation and decoding (scheme 1).

-1 , are equally probable, i.e., $P(u'_k[l] = +1) = P(u'_k[l] = -1)$, a posteriori log-likelihood ratio (LLR) for a transmitted $+1$ and a transmitted -1 in the bit sequence $\{u'_k[l]\}$ is defined as [23]

$$\begin{aligned} \lambda(u'_k[l]; O) &= \ln \frac{f(u'_k[l] = +1 | \mathbf{r})}{f(u'_k[l] = -1 | \mathbf{r})} = \ln \frac{f(\mathbf{r} | u'_k[l] = +1) P(u'_k[l] = +1)}{f(\mathbf{r} | u'_k[l] = -1) P(u'_k[l] = -1)} \\ &= \ln \frac{f(\mathbf{r} | u'_k[l] = +1)}{f(\mathbf{r} | u'_k[l] = -1)} = \ln \frac{\sum_{m: u'_k[l] = +1} f(\mathbf{r} | \mathbf{w}_m)}{\sum_{m: u'_k[l] = -1} f(\mathbf{r} | \mathbf{w}_m)} \\ &= \ln \frac{\sum_{m: u'_k[l] = +1} f(\mathbf{r} | \mathbf{s}_m)}{\sum_{m: u'_k[l] = -1} f(\mathbf{r} | \mathbf{s}_m)} \end{aligned} \quad (9.1)$$

For the purpose of designing receiver algorithms, we make some approximation assuming that the conditional probability $f(\mathbf{r} | \mathbf{s}_m)$ or equivalently, $f(\mathbf{r} | \mathbf{w}_m)$ can be approximated as $f(\mathbf{r} | \mathbf{s}_m) = A \exp\{B z_k(m)\}$, where $z_k(m)$ is the decision statistic from demodulator, based on the hypothesis that the m^{th} Walsh symbol is transmitted from user k , and A and B are some constants. We then derive the LLR for code bits under this assumption. The derivation of $z_k(m)$ for different soft demodulators is thoroughly studied in Section 9.3. Here, we only describe the basic principles of soft demodulation. In the above equation, we denote $m : u'_k[l] = \pm 1$ as the set of Walsh sequences $\{\mathbf{s}_m\}$ that correspond to the code bit $u'_k[l] = \pm 1$. Typically, one term will dominate each sum in (9.1), which suggests the “dual-maxima” rule [65]

$$\begin{aligned} \lambda(u'_k[l]; O) &\approx \ln \frac{\max_{m: u'_k[l] = +1} f(\mathbf{r} | \mathbf{s}_m)}{\max_{m: u'_k[l] = -1} f(\mathbf{r} | \mathbf{s}_m)} \\ &= \max_{m: u'_k[l] = +1} \{z_k(m)\} - \max_{m: u'_k[l] = -1} \{z_k(m)\} \end{aligned} \quad (9.2)$$

In case $M = 8$, the k^{th} user's j^{th} Walsh codeword $\mathbf{w}_{i_k(j)}$, or equivalently, the Walsh sequence $\mathbf{s}_k(j)$ corresponds to 3 code bits: $u'_k[l] = u'_k[l]^0, u'_k[l]^1, u'_k[l]^2$. We know from Table 8.1 that $u'_k[l]^0 = +1$ holds for $m = 0, 1, 2, 3$ and $u'_k[l]^0 = -1$ holds for $m = 4, 5, 6, 7$. Therefore, the soft metric for the first bit of a 3-bit block can be computed as

$$\lambda(u'_k[l]^0; O) \approx \max\{z_k(0), z_k(1), z_k(2), z_k(3)\} - \max\{z_k(4), z_k(5), z_k(6), z_k(7)\}$$

Similarly,

$$\lambda(u'_k[l]^1; O) \approx \max\{z_k(0), z_k(1), z_k(4), z_k(5)\} - \max\{z_k(2), z_k(3), z_k(6), z_k(7)\}$$

$$\lambda(u'_k[l]^2; O) \approx \max\{z_k(0), z_k(2), z_k(4), z_k(6)\} - \max\{z_k(1), z_k(3), z_k(5), z_k(7)\}$$

We use the notations $\lambda(\cdot; I)$ and $\lambda(\cdot; O)$ at the input and output ports of

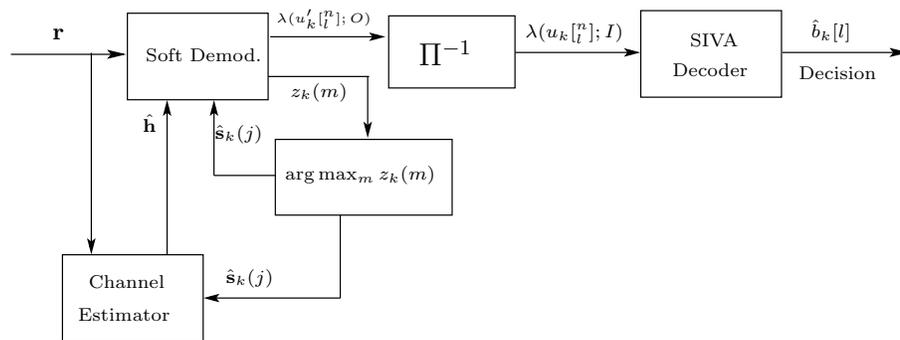


Figure 9.3: Partitioned soft demodulation with HDIC/HDIS, SIVA decoding (scheme 2.1).

a SISO. They refer to the unconstrained LLRs when the second argument is I , and modified LLRs according to the code constraints when it is O . The second argument I or O is sometimes omitted to simplify notation whenever no ambiguity arises. Other soft values are denoted by $L(\cdot)$. They are usually soft inputs or outputs of non-SISO devices.

The scheme based on this soft decision rule is shown in Fig. 9.3. Note that there are two outputs from the soft demodulator: the soft decision about the transmitted Walsh symbol $z_k(m)$ and the LLRs for code bits $\lambda(u'_k[l]^n; O)$. The former one is used to make hard decision on the Walsh symbol index $i_k(j)$ or transmitted Walsh sequence $\mathbf{s}_k(j)$

$$\hat{i}_k(j) = \arg \max_{m \in \{0, \dots, M-1\}} z_k(m)$$

or

$$\hat{\mathbf{s}}_k(j) = \arg \max_{\mathbf{s}_m} f(\mathbf{r}|\mathbf{s}_m) \quad (9.3)$$

The above two equations are equivalent under the assumption stated earlier. The estimated sequence $\hat{\mathbf{s}}_k(j)$ is needed for estimating the multipath complex channel gains and HDIC/HDIS based soft demodulation.

The deinterleaved LLR value $\lambda(u_k[l]^n; I) = \Pi^{-1}\{\lambda(u'_k[l]^n; O)\}$ is delivered as soft input to the channel decoder.

Compared to the traditional scheme illustrated in Fig. 9.2, a soft demodulator has replaced the hard demodulator so that soft-input VA (SIVA) decoding can be implemented. Since the soft demodulator directly outputs the soft value for the code bits $\{u'_k[l]^n\}$, the symbol-to-bit converter is not needed, which slightly simplifies the receiver structure.

The use of hard decision for $\mathbf{s}_k(j)$ for interference cancellation makes the system vulnerable to error propagation. The probability of error propagation can be reduced through the feedback of soft information instead of

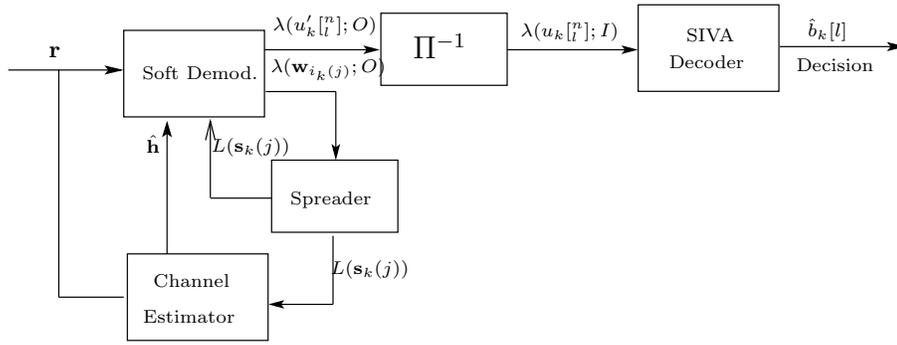


Figure 9.4: Partitioned soft demodulation with SDIC/SDIS, SIVA decoding (scheme 2.2).

hard decisions [21]. Fig. 9.4 shows the scenario of soft demodulation and channel estimation (CE) using $L(\mathbf{s}_k(j))$, some soft estimate of the Walsh sequence $\mathbf{s}_k(j)$. The derivation of $L(\mathbf{s}_k(j))$ is discussed below for scheme 2.2 and also in Section 9.2.4 for scheme 4.2. In this case, we need not only the LLRs for code bits $\{u_k^{[n]}\}$, but also for the Walsh codeword $\mathbf{w}_{i_k(j)}$. For systematic bits of $\mathbf{w}_{i_k(j)}$, the LLRs simply are

$$\begin{aligned} \lambda(\mathbf{w}_{i_k(j)}^1; O) &= \lambda(u_k^{[0]}; O); & \lambda(\mathbf{w}_{i_k(j)}^2; O) &= \lambda(u_k^{[1]}; O); \\ \lambda(\mathbf{w}_{i_k(j)}^4; O) &= \lambda(u_k^{[2]}; O) \end{aligned}$$

The first parity bit is always +1, therefore, its LLR value $\lambda(\mathbf{w}_{i_k(j)}^0; O) = \infty$. For the other parity bits

$$\begin{aligned} \lambda(\mathbf{w}_{i_k(j)}^3; O) &\approx \max\{z_k(0), z_k(1), z_k(6), z_k(7)\} - \max\{z_k(2), z_k(3), z_k(4), z_k(5)\} \\ \lambda(\mathbf{w}_{i_k(j)}^5; O) &\approx \max\{z_k(0), z_k(2), z_k(5), z_k(7)\} - \max\{z_k(1), z_k(3), z_k(4), z_k(6)\} \\ \lambda(\mathbf{w}_{i_k(j)}^6; O) &\approx \max\{z_k(0), z_k(3), z_k(4), z_k(7)\} - \max\{z_k(1), z_k(2), z_k(5), z_k(6)\} \\ \lambda(\mathbf{w}_{i_k(j)}^7; O) &\approx \max\{z_k(0), z_k(3), z_k(5), z_k(6)\} - \max\{z_k(1), z_k(2), z_k(4), z_k(7)\} \end{aligned} \quad (9.4)$$

The soft estimates $\{L(\mathbf{s}_k(j))\}$ are derived by spreading $\{\lambda(\mathbf{w}_{i_k(j)}; O)\}$ using equation (8.1). Clearly, the added complexity by deriving soft values rather than making hard decisions is minor if we compare (9.4) with (9.3). The use of $L(\mathbf{s}_k(j))$ in soft demodulation will be discussed shortly in Section 9.3.

All the above schemes are not optimized in the sense that the demodulator can not benefit from information derived from the channel decoder. Demodulation and decoding are strictly partitioned into two blocks. In the following, we present the integrated approach, where the problem of joint soft demodulation and decoding is approached by expanding the iteration

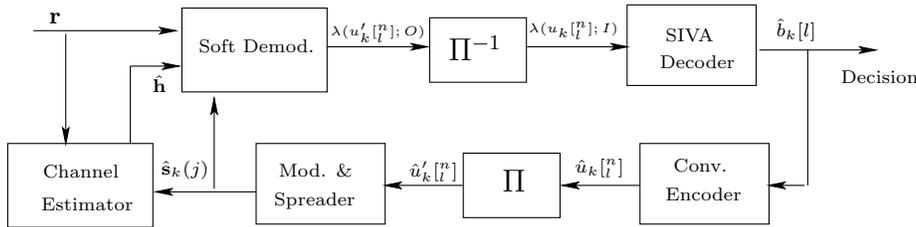


Figure 9.5: Integrated soft demodulation and SIVA decoding (scheme 3).

loop over the concatenation of demodulation and decoding blocks.

9.2.3 Scheme 3: integrated soft demodulation and SIVA decoding

The first integrated scheme is illustrated in Fig. 9.5. As we did earlier, the computed soft metrics $\{\lambda(u'_k[l]; O)\}$ are de-interleaved and decoded to get the estimated information bits $\{\hat{b}_k[l]\}$. Then we go through the same encoding, interleaving and modulation processes as in the transmitter to yield an estimate of the code bits $\{\hat{u}'_k[l]\}$ and Walsh sequences $\hat{s}_k(j)$.

Here we still use SIVA decoder to yield hard decision on $\{b_k[l]\}$. It differs from the aforementioned algorithms in that the decisions from the channel decoder are fed back to the demodulator. The estimate of $\hat{s}_k(j)$ needed for channel estimation and interference mitigation is not delivered from the demodulator itself, but from the output of the channel decoder. As will be clear later on, spanning the iteration loop over the two blocks is really crucial in improving the system performance. The price to pay is the added complexity mainly due to the channel decoding at every iteration instead of doing it once for all. Going through convolutional encoding, interleaving and modulation processes every time also slightly increase the complexity.

9.2.4 Scheme 4: integrated soft demodulation and Log-MAP decoding

A further enhancement idea is to replace the VA decoder with some SISO decoder which produces soft output for both information bits $\{b_k[l]\}$ and code bits $\{u_k[l]\}$. Based on the soft input $\lambda(u_k[l]; I)$ and the trellis structure of the convolutional code, the k^{th} user's SISO channel decoder computes a posteriori LLR of each information bit $\lambda(b_k[l]; O)$ and each code

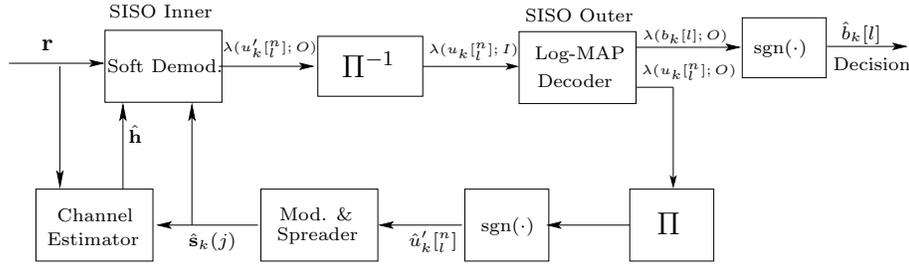


Figure 9.6: Integrated soft demodulation with HDIC/HDIS and Log-MAP decoding (scheme 4.1).

bit $\lambda(u_k[l]; O)$ as

$$\lambda(b_k[l]; O) = \ln \frac{P[b_k[l] = +1 | \lambda(u_k[l]; I)]}{P[b_k[l] = -1 | \lambda(u_k[l]; I)]} \quad (9.5)$$

$$\lambda(u_k[l]; O) = \ln \frac{P[u_k[l] = +1 | \lambda(u_k[l]; I)]}{P[u_k[l] = -1 | \lambda(u_k[l]; I)]} \quad (9.6)$$

where $\lambda(b_k[l]; O)$ is used to make decision on the transmitted information bit at the final iteration, while $\lambda(u_k[l]; O)$ is used for channel estimation and interference cancellation/suppression in the demodulator at the next iteration.

Several SISO algorithms can be used to compute the channel decoder outputs (9.5) and (9.6). For estimating the states or outputs of a Markov process, the symbol-by-symbol MAP algorithm is optimal. It differs from VA in the optimality criterion. The VA minimizes the frame or packet error probability and the MAP algorithm minimizes symbol error probability [66]. The MAP algorithm searches for the most probably transmitted symbol, given the received vector. It, however, poses numerical representation problems, and requires a large number of additions and multiplications. Max-Log-MAP solves the numerical problem and reduces the computational complexity, but are suboptimal especially at low SNR region. A further simplification yields the soft-output Viterbi algorithm (SOVA), it has simpler structure but inferior performance compared to Max-Log-MAP. By complementing the $\max(\cdot)$ operation with a correction function, Log-MAP algorithm avoids the approximations in the Max-Log-MAP and is equivalent to (true) symbol-by-symbol MAP, but without its numerical problems. Therefore, for the purpose of this study we consider the use of Log-MAP, which will be explained in detail in Section 10.2. For a complete treatment on different SISO algorithms, their similarities, differences and performance comparisons, readers are recommended to consult [67].

The improved iterative decoding schemes are shown in Fig. 9.6 and Fig. 9.7. The demodulator and decoder are each implemented with a SISO

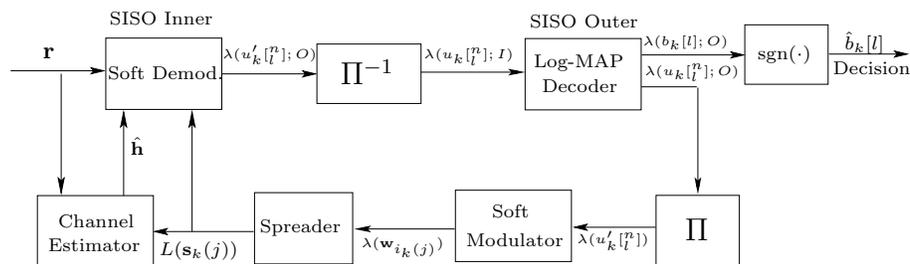


Figure 9.7: Integrated soft demodulation with SDIC/SDIS and Log-MAP decoding (scheme 4.2).

algorithm and operate in an iterative feedback mode where the information derived by the channel decoder is fed back to demodulator. The soft metrics from demodulator are then deinterleaved and decoded to carry on with the next iteration.

As common practice in iterative decoding, the soft outputs $\{\lambda(u_k[l]; O)\}$ from the Log-MAP decoder can be interleaved and fed back to the SISO inner soft demodulator as extrinsic information. This issue is thoroughly treated in Chapter 10, where we show that extrinsic information really helps improve the quality of demodulator if handled properly. Several enhancement ideas were proposed to make more efficient use of extrinsic information. However, the main focus in this chapter is the soft demodulation without extrinsic feedback. The soft outputs from the Log-MAP decoder are used by the demodulator only for the purposes of channel estimation and interference mitigation. Our intention is to perform an extensive investigation on different approaches to improve the demodulation and decoding performance, without the complications of extrinsic information.

Scheme 4.1 shown in Fig. 9.6, is similar to scheme 3, shown in Fig. 9.5, except that the SIVA decoder is replaced by a Log-MAP decoder and the code bits $\{\hat{u}'_k[l]\}$ are estimated by making hard decisions on the outputs of the channel decoder, i.e., $\{\hat{u}'_k[l]\} = \text{sgn}(\Pi\{\lambda(u_k[l]; O)\})$. A Log-MAP decoder is more complex than a VA decoder due to the need for forward and backward recursions and calculation of correction terms for the $\max(\cdot)$ operation. Refer to [67] for a thorough discussion on the complexity of different optimal and sub-optimal decoding algorithms.

Scheme 4.2 shown in Fig. 9.7, differs from scheme 4.1 in that it uses soft information $L(\mathbf{s}_k(j))$ for soft demodulation and channel estimation with the intention to reduce the likelihood of error propagation. Instead of estimating $\mathbf{s}_k(j)$, we can derive the soft estimate of $\mathbf{s}_k(j)$ by feeding $\lambda(u'_k[l]) = \Pi\{\lambda(u_k[l]; O)\}$ into a soft modulator which computes $\lambda(\mathbf{w}_{i_k(j)})$, the LLRs of the codeword $\mathbf{w}_{i_k(j)}$, then derive $L(\mathbf{s}_k(j))$ by repetition encoding (spreading) $\lambda(\mathbf{w}_{i_k(j)})$. Next, we shall explain how the soft modulator is implemented.

In case $M = 8$, we can see from Table 8.1 that the parity bits are formed by three systematic bits $\mathbf{w}_{i_k(j)}^1, \mathbf{w}_{i_k(j)}^2, \mathbf{w}_{i_k(j)}^4$ as¹

$$\begin{aligned}\mathbf{w}_{i_k(j)}^0 &= +1; & \mathbf{w}_{i_k(j)}^3 &= \mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^2; & \mathbf{w}_{i_k(j)}^5 &= \mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^4; \\ \mathbf{w}_{i_k(j)}^6 &= \mathbf{w}_{i_k(j)}^2 \oplus \mathbf{w}_{i_k(j)}^4; & \mathbf{w}_{i_k(j)}^7 &= \mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^2 \oplus \mathbf{w}_{i_k(j)}^4\end{aligned}$$

The LLRs for systematic bits are

$$\lambda(\mathbf{w}_{i_k(j)}^1) = \lambda(u'_k[l]^0); \quad \lambda(\mathbf{w}_{i_k(j)}^2) = \lambda(u'_k[l]^1); \quad \lambda(\mathbf{w}_{i_k(j)}^4) = \lambda(u'_k[l]^2)$$

Considering the fact that the interleaver breaks the memory of the convolutional encoding process, the bits $u'_k[l]^0, u'_k[l]^1, u'_k[l]^2$ can be modeled as statistically independent random variables. Also assume that they are independent conditioned on the received signal, then the LLRs for parity bits can thus be computed according to [23] by

$$\begin{aligned}\lambda(\mathbf{w}_{i_k(j)}^3) &= \lambda(\mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^2) \\ &= 2\operatorname{arctanh} \left\{ \tanh(\lambda(u'_k[l]^0)/2) \cdot \tanh(\lambda(u'_k[l]^1)/2) \right\} \\ &\approx \operatorname{sgn}\{\lambda(u'_k[l]^0)\} \cdot \operatorname{sgn}\{\lambda(u'_k[l]^1)\} \cdot \min\{|\lambda(u'_k[l]^0)|, |\lambda(u'_k[l]^1)|\} \\ \lambda(\mathbf{w}_{i_k(j)}^5) &= \lambda(\mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^4) \\ &= 2\operatorname{arctanh} \left\{ \tanh(\lambda(u'_k[l]^0)/2) \cdot \tanh(\lambda(u'_k[l]^2)/2) \right\} \\ &\approx \operatorname{sgn}\{\lambda(u'_k[l]^0)\} \cdot \operatorname{sgn}\{\lambda(u'_k[l]^2)\} \cdot \min\{|\lambda(u'_k[l]^0)|, |\lambda(u'_k[l]^2)|\} \\ \lambda(\mathbf{w}_{i_k(j)}^6) &= \lambda(\mathbf{w}_{i_k(j)}^2 \oplus \mathbf{w}_{i_k(j)}^4) \\ &= 2\operatorname{arctanh} \left\{ \tanh(\lambda(u'_k[l]^1)/2) \cdot \tanh(\lambda(u'_k[l]^2)/2) \right\} \\ &\approx \operatorname{sgn}\{\lambda(u'_k[l]^1)\} \cdot \operatorname{sgn}\{\lambda(u'_k[l]^2)\} \cdot \min\{|\lambda(u'_k[l]^1)|, |\lambda(u'_k[l]^2)|\} \\ \lambda(\mathbf{w}_{i_k(j)}^7) &= \lambda(\mathbf{w}_{i_k(j)}^1 \oplus \mathbf{w}_{i_k(j)}^2 \oplus \mathbf{w}_{i_k(j)}^4) \\ &= 2\operatorname{arctanh} \left\{ \prod_{n=0}^2 \tanh(\lambda(u'_k[l]^n)/2) \right\} \\ &\approx \prod_{n=0}^2 \operatorname{sgn}\{\lambda(u'_k[l]^n)\} \cdot \min_{n=0,1,2} \{|\lambda(u'_k[l]^n)|\}\end{aligned} \tag{9.7}$$

The approximation in (9.7) can be further approximated by omitting the

¹In order to ease understanding, we use $M = 8$ as an example for the derivation of the soft demodulation and soft cancellation algorithms. However, the extension of the proposed algorithms to other M values is straightforward.

$\min\{\cdot\}$ operations, which yields

$$\begin{aligned}
\hat{\mathbf{w}}_{i_k(j)}^0 &= +1; \quad \hat{\mathbf{w}}_{i_k(j)}^1 = \text{sgn}\{\lambda(u'_k[l^0])\}; \\
\hat{\mathbf{w}}_{i_k(j)}^2 &= \text{sgn}\{\lambda(u'_k[l^1])\}; \quad \hat{\mathbf{w}}_{i_k(j)}^4 = \text{sgn}\{\lambda(u'_k[l^2])\}; \\
\hat{\mathbf{w}}_{i_k(j)}^3 &= \hat{\mathbf{w}}_{i_k(j)}^1 \oplus \hat{\mathbf{w}}_{i_k(j)}^2 = \text{sgn}\{\lambda(u'_k[l^0])\} \cdot \text{sgn}\{\lambda(u'_k[l^1])\}; \\
\hat{\mathbf{w}}_{i_k(j)}^5 &= \hat{\mathbf{w}}_{i_k(j)}^1 \oplus \hat{\mathbf{w}}_{i_k(j)}^4 = \text{sgn}\{\lambda(u'_k[l^0])\} \cdot \text{sgn}\{\lambda(u'_k[l^2])\} \\
\hat{\mathbf{w}}_{i_k(j)}^6 &= \hat{\mathbf{w}}_{i_k(j)}^2 \oplus \hat{\mathbf{w}}_{i_k(j)}^4 = \text{sgn}\{\lambda(u'_k[l^1])\} \cdot \text{sgn}\{\lambda(u'_k[l^2])\} \\
\hat{\mathbf{w}}_{i_k(j)}^7 &= \hat{\mathbf{w}}_{i_k(j)}^1 \oplus \hat{\mathbf{w}}_{i_k(j)}^2 \oplus \hat{\mathbf{w}}_{i_k(j)}^4 = \prod_{n=0}^2 \text{sgn}\{\lambda(u'_k[l^n])\} \quad (9.8)
\end{aligned}$$

which are the hard decisions made in the modulator in Fig. 9.6. Compared to hard decisions expressed by (9.8), the soft derivation only slightly increases the computational complexity by introducing $\min\{\cdot\}$ operations as shown in (9.7).

Channel estimation using the soft estimates of $\mathbf{s}_k(j)$ is introduced in Section 5.6. We shall see how they can be used for soft demodulation in Section 9.3.

9.3 Soft demodulation algorithms

The improved algorithms discussed above all require the design of soft demodulator that can produce soft outputs to enable soft input channel decoding. In a serially concatenated system, the quality of the inner demodulation or decoding is decisive for the system performance. The derivation of different soft demodulators are given below. For simplicity of notation we will suppress the index k and/or j from $\mathbf{s}_k(j)$, $\mathbf{C}_k(j)$, $\mathbf{r}(k, j)$, $\mathbf{A}(k, j)$, $\mathbf{n}(k, j)$, $\mathbf{X}_k(j)$ and $\mathbf{h}_k(j)$, etc., whenever no ambiguity arises.

9.3.1 Soft demodulation with matched filter

Let $\mathbf{r}_{k,l}$, ($l = 1, 2, \dots, L_k$) denote the delay aligned version of the received vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path and denote the vector $\tilde{\mathbf{r}}_{k,l} = [\tilde{r}_{k,l}(1) \ \tilde{r}_{k,l}(2) \ \dots \ \tilde{r}_{k,l}(N)] \in \mathbb{C}^N$ as $\mathbf{r}_{k,l}$ scrambled with the scrambling sequence \mathbf{C}_k . The descrambled vector $\tilde{\mathbf{r}}_{k,l}$ can be expressed as $\tilde{\mathbf{r}}_{k,l} = \mathbf{s}_k h_{k,l} + \mathbf{m}_{k,l} + \mathbf{i}_{k,l} + \mathbf{n}_{k,l}$, where $\mathbf{s}_k \in \{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{M-1}\}$ is the transmitted Walsh sequence. The desired signal vector $\mathbf{s}_k h_{k,l}$ is due to the contribution from k^{th} user's l^{th} path. The vectors $\mathbf{m}_{k,l}$ and $\mathbf{i}_{k,l}$ stand for the MAI and ISI term respectively.

If we approximate both the MAI and ISI terms as independent complex Gaussian random vectors: $\mathbf{m}_{k,l} \in \mathcal{CN}(\mathbf{0}, N_m \mathbf{I}_N)$ and $\mathbf{i}_{k,l} \in \mathcal{CN}(\mathbf{0}, N_i \mathbf{I}_N)$, then $\tilde{\mathbf{r}}_{k,l}$ can be reformed as $\tilde{\mathbf{r}}_{k,l} = \mathbf{s}_k h_{k,l} + \tilde{\mathbf{n}}_{k,l}$ where $\tilde{\mathbf{n}}_{k,l} = \mathbf{m}_{k,l} + \mathbf{i}_{k,l} + \mathbf{n}_{k,l} \in \mathbb{C}^N$ has PDF $\tilde{\mathbf{n}}_{k,l} \sim \mathcal{CN}(\mathbf{0}, N'_0 \mathbf{I}_N)$ and $N'_0 = N_0 + N_m + N_i$.

Also, $\tilde{\mathbf{n}}_{k,1}, \dots, \tilde{\mathbf{n}}_{k,L_k}$ are uncorrelated after despreading due to descrambling. Therefore,

$$\begin{aligned}
f(\mathbf{r}|\mathbf{s}_m) &= \prod_{l=1}^{L_k} f(\tilde{\mathbf{r}}_{k,l}|\mathbf{s}_m) \\
f(\tilde{\mathbf{r}}_{k,l}|\mathbf{s}_m) &= \frac{1}{(\pi N'_0)^N} \exp\left(-\frac{\|\tilde{\mathbf{r}}_{k,l} - \mathbf{s}_m h_{k,l}\|^2}{N'_0}\right) \\
\lambda_{\text{MF}}^C(u'_k[l^n]; O) &\approx \ln \frac{\max_{m:u'_k[l^n]=+1} \prod_{l=1}^{L_k} f(\tilde{\mathbf{r}}_{k,l}|\mathbf{s}_m)}{\max_{m:u'_k[l^n]=-1} \prod_{l=1}^{L_k} f(\tilde{\mathbf{r}}_{k,l}|\mathbf{s}_m)} \\
&= \ln \frac{\exp\left(-\sum_{l=1}^{L_k} \|\tilde{\mathbf{r}}_{k,l} - \mathbf{s}^+ h_{k,l}\|^2/N'_0\right)}{\exp\left(-\sum_{l=1}^{L_k} \|\tilde{\mathbf{r}}_{k,l} - \mathbf{s}^- h_{k,l}\|^2/N'_0\right)} \\
&= \frac{1}{N'_0} \sum_{l=1}^{L_k} \{\|\tilde{\mathbf{r}}_{k,l} - \mathbf{s}^- h_{k,l}\|^2 - \|\tilde{\mathbf{r}}_{k,l} - \mathbf{s}^+ h_{k,l}\|^2\} \\
&\approx \frac{2}{N'_0} \sum_{l=1}^{L_k} \text{Re} \{h_{k,l}^* \mathbf{s}^{+*} \tilde{\mathbf{r}}_{k,l} - h_{k,l}^* \mathbf{s}^{-*} \tilde{\mathbf{r}}_{k,l}\} \quad (9.9)
\end{aligned}$$

In equation (9.9), \mathbf{s}^+ denotes the Walsh sequence \mathbf{s}_m that corresponds to $\max_{m:u'_k[l^n]=+1} f(\mathbf{r}|\mathbf{s}_m)$, and \mathbf{s}^- is defined similarly. In the above equation, we can omit the constant $2/N'_0$ since it is just a scaling factor for all the LLR values, therefore, does not have any effect on the decision.

A non-coherent version of the MF soft demodulator can be obtained similarly in a path-by-path manner as

$$\lambda_{\text{MF}}^N(u'_k[l^n]; O) \approx \sum_{l=1}^{L_k} |\mathbf{s}^{+*} \tilde{\mathbf{r}}_{k,l}| - \sum_{l=1}^{L_k} |\mathbf{s}^{-*} \tilde{\mathbf{r}}_{k,l}| \quad (9.10)$$

Estimates of the complex channel gains $h_{k,l}$ are not needed to compute the LLR value for bit $u'_k[l^n]$ in the above equation. This is particularly useful in the beginning of the iteration process when the estimate of channel fading process is not yet available. On the contrary, the coherent MF demodulator expressed by (9.9) is not often used because the channel is unknown at the initial stage. In the subsequent stages, both channel and interference can be estimated in a decision directed mode, and more powerful coherent MUD (IC/IS) techniques can be used. This will be discussed in Section 9.3.2

and 9.3.3.

Referring to equation (9.2), $z_k(m) = \sum_{l=1}^{L_k} \text{Re}\{\hat{h}_{k,l}^* \mathbf{s}_m^* \tilde{\mathbf{r}}_{k,l}\}$ and $z_k(m) = \sum_{l=1}^{L_k} |\mathbf{s}_m^* \tilde{\mathbf{r}}_{k,l}|$ for the coherent and noncoherent MF demodulator, respectively. In the first decision metric, $\hat{h}_{k,l}$ is an estimate of $h_{k,l}$. The original complex channel gain in (9.9) is unknown, it has to be replaced by its estimated version.

9.3.2 Soft demodulation with hard/soft decision interference cancellation

Once the transmitted signals are estimated for all the users at the previous iteration, interference can be removed by subtracting the estimated signals of the interfering users from the received signal \mathbf{r} to form a new signal vector \mathbf{r}' for demodulating the signal transmitted from user k , i.e.,

$$\mathbf{r}'_{\text{hard}} = \mathbf{r} - \hat{\mathbf{y}} + \hat{\mathbf{X}}_k \hat{\mathbf{h}}_k \quad (9.11)$$

where $\mathbf{r} \in \mathbb{C}^{N_k}$ denote the received signal vector due to the transmission of the j^{th} symbol from the k^{th} user, and $\mathbf{r}'_{\text{hard}} \in \mathbb{C}^{N_k}$ is its interference canceled version after subtracting the contributions from all the other users using hard decision feedback. The vector $\hat{\mathbf{y}} = \hat{\mathbf{A}} \hat{\mathbf{h}}$ represents the estimated contribution from all the users calculated by using the data matrix $\hat{\mathbf{A}}$ and channel vector $\hat{\mathbf{h}}$ estimated at the previous iteration. The vector $\hat{\mathbf{X}}_k \hat{\mathbf{h}}_k$ is the estimated contribution from all paths of user k .

When $L(\mathbf{s}_k)$, the soft estimate of the transmitted sequence, is available from the either soft demodulator (see Section 9.2.2) or soft decoder (see Section 9.2.4), we can carry out soft IC. The rationale is that the hard IC tends to propagate errors and increase the interference with incorrect decision feedback; while with soft cancellation, an erroneously estimated symbol usually has small LLR, and hence the soft estimate of this symbol is small and does not make much contribution to the feedback, therefore error propagation is avoided.

If $\lambda(\mathbf{w}_{i_k(j)}^p)$ is derived from soft demodulator expressed by (9.4), according to the definition $\lambda(\mathbf{w}_{i_k(j)}^p) = \ln \frac{P(\mathbf{w}_{i_k(j)}^p = +1|\mathbf{r})}{P(\mathbf{w}_{i_k(j)}^p = -1|\mathbf{r})}$, the soft estimate (expected value given the received observation) for each bit of the Walsh codeword is computed by

$$\begin{aligned} E[\mathbf{w}_{i_k(j)}^p | \mathbf{r}] &= (+1) \times P\{\mathbf{w}_{i_k(j)}^p = +1 | \mathbf{r}\} + (-1) \times P\{\mathbf{w}_{i_k(j)}^p = -1 | \mathbf{r}\} \\ &= (+1) \frac{e^{\lambda(\mathbf{w}_{i_k(j)}^p)}}{1 + e^{\lambda(\mathbf{w}_{i_k(j)}^p)}} + (-1) \frac{e^{-\lambda(\mathbf{w}_{i_k(j)}^p)}}{1 + e^{-\lambda(\mathbf{w}_{i_k(j)}^p)}} \\ &= \tanh\{\lambda(\mathbf{w}_{i_k(j)}^p)/2\} \end{aligned}$$

If $\lambda(\mathbf{w}_{i_k(j)}^p)$ is derived from soft modulator expressed by (9.7), then the soft estimate of each bit can be computed as

$$\begin{aligned}
\mathbb{E}[\mathbf{w}_{i_k(j)}^0|\mathbf{r}] &= +1; \quad \mathbb{E}[\mathbf{w}_{i_k(j)}^1|\mathbf{r}] = \tanh\{\lambda(u'_k[l]^0)/2\}; \\
\mathbb{E}[\mathbf{w}_{i_k(j)}^2|\mathbf{r}] &= \tanh\{\lambda(u'_k[l]^1)/2\}; \quad \mathbb{E}[\mathbf{w}_{i_k(j)}^4|\mathbf{r}] = \tanh\{\lambda(u'_k[l]^2)/2\}; \\
\mathbb{E}[\mathbf{w}_{i_k(j)}^3|\mathbf{r}] &= \tanh\{\lambda(u'_k[l]^0)/2\} \cdot \tanh\{\lambda(u'_k[l]^1)/2\}; \\
\mathbb{E}[\mathbf{w}_{i_k(j)}^5|\mathbf{r}] &= \tanh\{\lambda(u'_k[l]^0)/2\} \cdot \tanh\{\lambda(u'_k[l]^2)/2\}; \\
\mathbb{E}[\mathbf{w}_{i_k(j)}^6|\mathbf{r}] &= \tanh\{\lambda(u'_k[l]^1)/2\} \cdot \tanh\{\lambda(u'_k[l]^2)/2\}; \\
\mathbb{E}[\mathbf{w}_{i_k(j)}^7|\mathbf{r}] &= \prod_{n=0}^2 \tanh\{\lambda(u'_k[l]^n)/2\}
\end{aligned} \tag{9.12}$$

Comparing with equation (9.8), it is evident that the additional complexity by computing $\mathbb{E}[\mathbf{w}_{i_k(j)}^p|\mathbf{r}]$ instead of making hard decision on $\mathbf{w}_{i_k(j)}^p$ is small. The complexity increase is due to the replacement of $\text{sgn}(\cdot)$ operation with $\tanh(\cdot)$.

The soft estimate $\mathbb{E}[\mathbf{s}_k^q|\mathbf{r}]$ for each Walsh chip $\mathbf{s}_k^q, q = 1, \dots, N$ is derived by spreading (repetition encoding) the soft bit of Walsh codeword $\mathbb{E}[\mathbf{w}_{i_k(j)}^p|\mathbf{r}], p = 1, \dots, M$. The repetition factor is N/M . The cancellation residual after soft cancellation becomes

$$\mathbf{r}'_{\text{soft}} = \mathbf{r} - \mathbb{E}[\mathbf{y}|\mathbf{r}] + \mathbb{E}[\mathbf{X}_k|\mathbf{r}]\hat{\mathbf{h}}_k \tag{9.13}$$

where $\mathbb{E}[\mathbf{y}|\mathbf{r}] = \mathbb{E}[\mathbf{A}|\mathbf{r}]\hat{\mathbf{h}}$, and the columns of $\mathbb{E}[\mathbf{A}|\mathbf{r}], \mathbb{E}[\mathbf{X}_k|\mathbf{r}]$ are derived by scrambling $\mathbb{E}[\mathbf{s}_k|\mathbf{r}]$ with \mathbf{C}_k and compensating with the path delays.

In case of perfect cancellation, \mathbf{r}' ($\mathbf{r}'_{\text{hard}}$ or $\mathbf{r}'_{\text{soft}}$) only contains the contribution from the k^{th} user plus original additive Gaussian noise $\mathbf{n} \in \mathbb{C}^{N_k}$ with PDF $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_{N_k})$, i.e., $\mathbf{r}' = \mathbf{X}_k\mathbf{h}_k + \mathbf{n}$. Therefore,

$$\begin{aligned}
f(\mathbf{r}'|\mathbf{s}_m) &= \frac{1}{(\pi N_0)^{N_k}} \exp\left(-\frac{\|\mathbf{r}' - \mathbf{X}_{k,m}\mathbf{h}_k\|^2}{N_0}\right) \\
\lambda_{\text{IC}}(u'_k[l]^n; O) &\approx \ln \frac{\max_{m:u'_k[l]^n=+1} f(\mathbf{r}'|\mathbf{s}_m)}{\max_{m:u'_k[l]^n=-1} f(\mathbf{r}'|\mathbf{s}_m)} \\
&= \ln \frac{\exp(-\|\mathbf{r}' - \mathbf{X}^+\mathbf{h}_k\|^2/N_0)}{\exp(-\|\mathbf{r}' - \mathbf{X}^-\mathbf{h}_k\|^2/N_0)} \\
&= \frac{1}{N_0} \{ \|\mathbf{r}' - \mathbf{X}^-\mathbf{h}_k\|^2 - \|\mathbf{r}' - \mathbf{X}^+\mathbf{h}_k\|^2 \} \\
&= \frac{2}{N_0} \text{Re} \{ \mathbf{h}_k^* \mathbf{X}^{+*} \mathbf{r}' - \mathbf{h}_k^* \mathbf{X}^{-*} \mathbf{r}' \}
\end{aligned} \tag{9.14}$$

where \mathbf{X}^+ denotes the $\mathbf{X}_{k,m}$ that corresponds to $\max_{m:u'_k[l]^n=+1} f(\mathbf{r}'|\mathbf{s}_m)$,

and \mathbf{X}^- is defined similarly.

The original \mathbf{h}_k is unknown, it has to be estimated. In (9.14), we should replace it with its estimates $\hat{\mathbf{h}}_k$ instead. An estimate of the channel vector $\hat{\mathbf{h}}_k$ can be obtained using detected data from previous iteration. Channel estimation was treated in Chapter 5. With the channel estimate $\hat{\mathbf{h}}_k$, we can combine the hypothesized contributions from all the paths of the same user. The soft metric need not to be computed in a path-by-path fashion like we did for the noncoherent MF demodulator.

9.3.3 Soft demodulation with hard/soft decision interference suppression

The idea of interference suppression (IS) is to suppress the estimated interference by filtering (orthogonal projection). The motivation for us to investigate the potential of IS is that an erroneously estimated symbol will lead to a doubling of the interference when using hard cancellation. However, when using suppression, an erroneously estimated symbol will cause the suppression of a non-existing signal. This will lead to some suppression of the desired signal, but the overall penalty may be less than in the cancellation case; the penalty will be even less when soft suppression is used as will be evidenced by the simulation results.

To construct the suppression filter, we need to know (or estimate) the structure of the interference. Consider the matrix $\mathbf{U} \in \mathbb{R}^{N_k \times (L_{\text{tot}} - L_k)}$ defined as

$$\mathbf{U} = \begin{bmatrix} \mathbf{A}_1 & \cdots & \mathbf{A}_{k-1} & \mathbf{A}_{k+1} & \cdots & \mathbf{A}_K \end{bmatrix}$$

which is formed from \mathbf{A} by deleting the columns that are due to the k^{th} user. The symbols N_k , L_{tot} , and L_k denote the processing window length, the total number of paths of all the users, and the number of paths for user k , respectively.

We can suppress the interference by projecting \mathbf{r} on the null space of \mathbf{U} which is computed as $\mathbf{P}_{\mathbf{U}}^\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^\dagger$, where $\mathbf{U}^\dagger = (\mathbf{U}^*\mathbf{U})^{-1}\mathbf{U}^*$ denotes the left pseudoinverse of \mathbf{U} (assume \mathbf{U} has full column rank). This implies that $\mathbf{P}_{\mathbf{U}}^\perp \mathbf{A}_i = \mathbf{0}$ for all $i \neq k$, and thus

$$\mathbf{P}_{\mathbf{U}}^\perp \mathbf{r} = \mathbf{P}_{\mathbf{U}}^\perp [\mathbf{A}\mathbf{h} + \mathbf{n}] = \sum_{i=1}^K \mathbf{P}_{\mathbf{U}}^\perp \mathbf{A}_i \mathbf{h}_i + \mathbf{P}_{\mathbf{U}}^\perp \mathbf{n} = \mathbf{P}_{\mathbf{U}}^\perp \mathbf{A}_k \mathbf{h}_k + \mathbf{P}_{\mathbf{U}}^\perp \mathbf{n}$$

The interference is suppressed by projecting \mathbf{r} on $\mathbf{P}_{\mathbf{U}}^\perp$ which is orthogonal to the subspace spanned by the MAI (columns of \mathbf{U}). It can be easily shown that $\mathbf{P}_{\mathbf{U}}^\perp = \mathbf{P}_{\mathbf{U}}^{\perp*}$, $\mathbf{P}_{\mathbf{U}}^{\perp 2} = \mathbf{P}_{\mathbf{U}}^\perp$, and $\mathbf{P}_{\mathbf{U}}^\perp = \mathbf{P}_{\mathbf{U}}^{\perp*} \mathbf{P}_{\mathbf{U}}^\perp$. Since a linear transformation of a Gaussian random variable is still a Gaussian random variable, and the original noise vector has the statistics $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I})$,

the covariance matrix for the projected noise vector $\tilde{\mathbf{n}} = \mathbf{P}_U^\perp \mathbf{n}$ is

$$\mathbb{E}[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^*] = \mathbb{E}[\mathbf{P}_U^\perp \mathbf{nn}^* \mathbf{P}_U^{\perp*}] = \mathbf{P}_U^\perp \mathbb{E}[\mathbf{nn}^*] \mathbf{P}_U^{\perp*} = \mathbf{P}_U^\perp (N_0 \mathbf{I}) \mathbf{P}_U^{\perp*} = N_0 \mathbf{P}_U^\perp \quad (9.15)$$

The conditional PDF can be approximated as

$$\begin{aligned} f(\mathbf{P}_U^\perp \mathbf{r} | \mathbf{s}_m) &\approx \frac{\exp[-(\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)^* (N_0 \mathbf{P}_U^\perp)^{-1} (\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)]}{(\pi)^{N_k} \det(N_0 \mathbf{P}_U^\perp)} \\ &= \frac{\exp\left[-\frac{(\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)^* (\mathbf{P}_U^\perp)^{-1} (\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)}{N_0}\right]}{(\pi)^{N_k} \det(N_0 \mathbf{P}_U^\perp)} \\ &= \frac{1}{(\pi)^{N_k} \det(N_0 \mathbf{P}_U^\perp)} \exp\left[-\frac{(\mathbf{r} - \mathbf{X}_{k,m} \mathbf{h}_k)^* \mathbf{P}_U^\perp (\mathbf{r} - \mathbf{X}_{k,m} \mathbf{h}_k)}{N_0}\right] \\ &= \frac{\exp\left[-\frac{(\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)^* (\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k)}{N_0}\right]}{(\pi)^{N_k} \det(N_0 \mathbf{P}_U^\perp)} \\ &= \frac{1}{(\pi)^{N_k} \det(N_0 \mathbf{P}_U^\perp)} \exp\left[-\frac{\|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k\|^2}{N_0}\right] \quad (9.16) \end{aligned}$$

The LLRs can thus be computed as

$$\begin{aligned} \lambda_{\text{IS}}(u'_k[l]; O) &\approx \ln \frac{\max_{m: u'_k[l] = +1} f(\mathbf{r} | \mathbf{s}_m)}{\max_{m: u'_k[l] = -1} f(\mathbf{r} | \mathbf{s}_m)} \\ &\approx \ln \frac{\max_{m: u'_k[l] = +1} \exp(-\|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k\|^2 / N_0)}{\max_{m: u'_k[l] = -1} \exp(-\|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}_{k,m} \mathbf{h}_k\|^2 / N_0)} \\ &= \ln \frac{\exp(-\|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}^+ \mathbf{h}_k\|^2 / N_0)}{\exp(-\|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}^- \mathbf{h}_k\|^2 / N_0)} \\ &= \frac{1}{N_0} \{ \|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}^- \mathbf{h}_k\|^2 - \|\mathbf{P}_U^\perp \mathbf{r} - \mathbf{P}_U^\perp \mathbf{X}^+ \mathbf{h}_k\|^2 \} \\ &= \frac{2}{N_0} \text{Re} \{ \mathbf{h}_k^* \mathbf{X}^{+*} \mathbf{P}_U^\perp \mathbf{r} - \mathbf{h}_k^* \mathbf{X}^{-*} \mathbf{P}_U^\perp \mathbf{r} \} \quad (9.17) \end{aligned}$$

In the above equation, \mathbf{h}_k and \mathbf{U} needs to be replaced with $\hat{\mathbf{h}}_k$ and $\hat{\mathbf{U}}$; the constant $2/N_0$ in (9.14) and (9.17) can be omitted for the same

reason as stated earlier. In addition to simplicity, another advantage of using max approximation to compute the LLR shown in (9.1) and (9.2) is the avoidance of estimating noise variance N_0 . The matrix $\hat{\mathbf{U}}$ is formed from $\hat{\mathbf{A}}$, the estimate of \mathbf{A} at the previous iteration by deleting the columns that are due to the k^{th} user.

In the same way as the soft IC is formed, the soft IS can be accomplished by using soft estimate of \mathbf{U} conditioned on the received observation:

$$\mathbf{E}[\mathbf{U}|\mathbf{r}] = \left[\mathbf{E}[\mathbf{A}_1|\mathbf{r}] \quad \cdots \quad \mathbf{E}[\mathbf{A}_{k-1}|\mathbf{r}] \quad \mathbf{E}[\mathbf{A}_{k+1}|\mathbf{r}] \quad \cdots \quad \mathbf{E}[\mathbf{A}_K|\mathbf{r}] \right]$$

The soft projection matrix becomes $\mathbf{P}_{\text{soft}} = \mathbf{I} - \mathbf{E}[\mathbf{U}|\mathbf{r}]\mathbf{E}[\mathbf{U}|\mathbf{r}]^\dagger$. The SDIS soft demodulation can be expressed by equation (9.17) by replacing $\mathbf{P}_{\hat{\mathbf{U}}}^\perp$ with \mathbf{P}_{soft} .

Referring to equation (9.2), $z_k(m) = \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{r}'\}$ and $z_k(m) = \text{Re}\{\hat{\mathbf{h}}_k^* \mathbf{X}_{k,m}^* \mathbf{P}\mathbf{r}\}$, respectively for IC and IS demodulator. Note that the computation of the projected vector $\mathbf{P}\mathbf{r}$ (\mathbf{P} can be either $\mathbf{P}_{\hat{\mathbf{U}}}^\perp$ for HDIS or \mathbf{P}_{soft} for SDIS) involves matrix inversions at a symbol rate since the MAI matrix \mathbf{U} differs from symbol to symbol. On the contrary, the vector \mathbf{r}' (either $\mathbf{r}'_{\text{hard}}$ for HDIC or $\mathbf{r}'_{\text{soft}}$ for SDIC) is derived just by subtraction operations. Therefore, the IC demodulator is less computationally intensive than the IS demodulator.

We need estimates of the complex channel gains to do coherent demodulation as discussed above. The maximum likelihood (ML) channel estimator is derived in Chapter 5 as

$$\hat{\mathbf{h}}_{\text{hard}}^{\text{ML}}(k, j) = \begin{bmatrix} \hat{\mathbf{A}}(k, j) \\ \cdots \\ \hat{\mathbf{A}}(k, j + D) \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{r}(k, j) \\ \cdots \\ \mathbf{r}(k, j + D) \end{bmatrix} \quad (9.18)$$

When $\mathbf{E}[\mathbf{A}|\mathbf{r}]$ is available, the soft version of the ML channel estimator can be formed as

$$\hat{\mathbf{h}}_{\text{soft}}^{\text{ML}}(k, j) = \begin{bmatrix} \mathbf{E}[\mathbf{A}(k, j)|\mathbf{r}(k, j)] \\ \cdots \\ \mathbf{E}[\mathbf{A}(k, j + D)|\mathbf{r}(k, j + D)] \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{r}(k, j) \\ \cdots \\ \mathbf{r}(k, j + D) \end{bmatrix} \quad (9.19)$$

As mentioned in Chapter 5, the estimation results can be further enhanced by applying smoothing operation on the original ML channel estimates. Comparing (9.18) vs. (9.19), one can see the use of soft information in channel estimation itself does not introduce any additional complexity. It is the derivation of soft values that is little more complicated than making

hard decisions as discussed in Section 9.2.2 and Section 9.2.4.

9.4 Numerical Results

In the simulations, we employ a rate $R_c = 1/3$ Maximum Free Distance (MFD) convolutional code [1] with constraint length 5 and generator polynomials (25, 33, 37) in octal form for all the users. Block interleaving is applied to the convolutionally encoded bits to decorrelate the fading effect. Each block of $\log_2 8 = 3$ interleaved bits from each user is then converted into one of $M = 8$ Walsh codewords spread to a total length of $N = 64$ chips. The number of chips per inner code bit is $N_c = N/M = 8$. If the orthogonal modulation is viewed as part of spreading, the effective spreading factor of the system is $N/\log_2 M = 64/3$ chips per convolutionally coded bit (64 chips per information bit).

Channels are independent multipath Rayleigh fading channels with the classical “bath tub” power spectrum. That is, the channel gain $h_{k,l}(t)$ is a complex circular Gaussian process with autocorrelation function $E[h_{k,l}^*(t)h_{k,l}(t+\tau)] = P_{k,l}J_0(2\pi f_d\tau)$ where f_d is the maximum Doppler frequency, $J_0(x)$ is the zeroth order Bessel function of the first kind, and $P_{k,l}$ is the average power of $h_{k,l}(t)$. The Doppler shifts on each of the multipath components are due to the relative motion between the base station and mobile units. Here, the normalized Doppler frequency is assumed to be $f_dT = 0.01$. Perfect slow power control is assumed in the sense that $P_k = \sum_{l=1}^{L_k} P_{k,l} = \sum_{l=1}^{L_k} E[|h_{k,l}|^2]$, the average received power, is equal for all users. The number of multipath channels L_k is set to be 3, ($L_k = L = 3$) for k . Channel estimation is conducted with the ML algorithm presented in Chapter 5, and channel smoothing is accomplished by an FIR filter derived from Hamming window of length 19.

The long scrambling codes \mathbf{C}_k are randomly assigned. The noise variance N_0 , and \mathbf{C}_k as well as path delays $\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L_k}$ are assumed to be known to the receiver. One simplifying assumption is made such that the delays of all the users’ paths are multiples of the chip duration. However, the presented algorithms are general and can be extended to include arbitrary delays. The block size is set to 1540 Walsh symbols, which corresponds to $1540 \times 3 = 4620$ code bits. The simulation results are averaged over random distributions of fading, noise, delay, and scrambling code with minimum of 10 blocks of data transmitted and at least 100 errors generated. To study the behavior of each algorithm, the number of iterations is usually set to 6 or 7, since it is observed that almost all the algorithms would converge after 5 or 6 iterations.

The above parameter setting also applies to the experiments in Chapter 10.

It should be emphasized that the interleaving design is essential for the system performance. We test the schemes discussed in Section 9.2

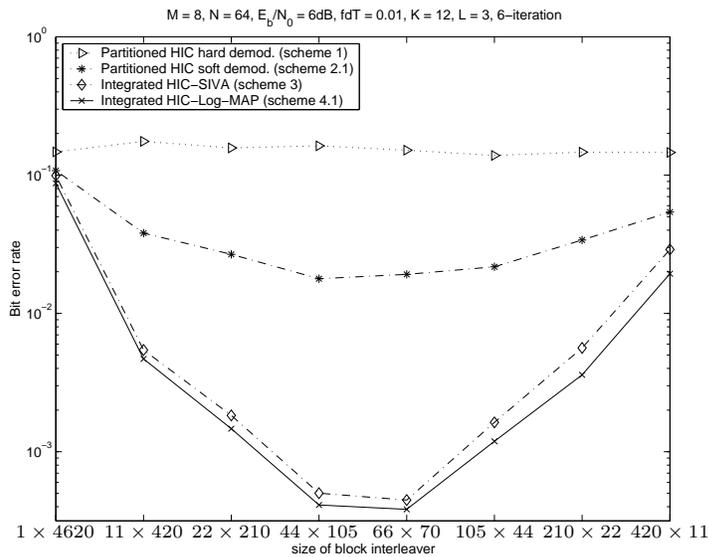


Figure 9.8: Performance of different block interleavers.

with different interleavers. The block size of 4620 code bits are passed through different block interleavers which exhibit large discrepancies in the results as demonstrated in Fig. 9.8, especially for the integrated schemes 3 and 4. The interleaver of size 66×70 in general works better than the others for the studied system. It is therefore chosen for all the subsequent experiments. Another expected discovery from Fig. 9.8 is that the system performance improves as the receiver complexity increases. Conventional receiver has lowest complexity and poorest performance, it is not able to operate reliably at $E_b/N_0 = 6$ dB in 12-user system. On the contrary, the integrated algorithms with Log-MAP decoding has highest computational complexity, but the performance is the best of all (with properly designed interleaver, BER can be kept well below 10^{-3} at $E_b/N_0 = 6$ dB with 6 iterations).

In Fig. 9.9 and 9.10, we compare the performance of all the discussed algorithms with HDIC and HDIS demodulators, respectively in a 12-user system. Apparently, in both cases, the most significant gain (over 2 dB at $\text{BER} < 10^{-2.5}$) is obtained by replacing hard demodulator with soft demodulator. By going from partitioned approach to integrated approach also gives substantial improvement (around 1 dB). The improvement by replacing VA with Log-MAP decoding is relatively small, but still noticeable. Comparing these two figures one can see that IC demodulator produces better results than IS demodulator. To achieve $\text{BER} = 10^{-4}$, $E_b/N_0 \approx 6.5$ dB is required with HDIC-Log-MAP; while $E_b/N_0 \approx 9$ dB is required with HDIS-Log-MAP. Therefore, the difference between IC and IS scheme is

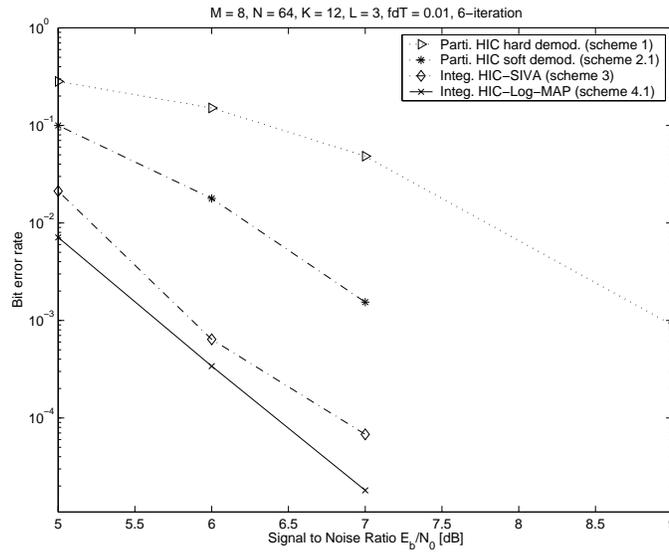


Figure 9.9: Different schemes with IC demodulator.

over 2 dB.

The behavior and convergence property of two integrated schemes are further investigated in Fig. 9.11 and Fig. 9.12. As indicated in Fig. 9.11, the performance gain by replacing SIVA decoding with Log-MAP decoding is obvious during the first 3 iterations; however, the gap tends to decrease as they reach convergence, at which point, the difference is 0.3 dB at BER of 10^{-4} . For both schemes, the gain at each iteration increases with increased SNR. A reasonable level of SNR needs to be maintained in order to benefit from the iteration process. With $E_b/N_0 = 9$ dB and target BER of 10^{-3} , the system capacity with IC soft demodulation and Log-MAP/SIVA decoding can be seen from Fig. 9.12, and is summarized in Table 9.1. The gain by iterations becomes smaller as the system becomes more heavily loaded. We observe from these plots that all the algorithms converge to maximum achievable performance after 4 – 6 iterations, beyond which improvement through the iteration process becomes insignificant.

Finally, the performance of soft cancellation/suppression and channel estimation are examined in Fig. 9.13, and 9.14. Compared with their hard counterparts, the gain by applying soft information for interference alleviation and channel estimation for partitioned/integrated approach is 0.1/0.2 dB for IC and 0.2/0.4 dB for IS. Although the enhancement is marginal, it has the tendency of increasing as SNR increases and as system becomes more heavily loaded. From the experiments, we noticed that the performance degradation introduced by the approximation in (9.7) is negligible.

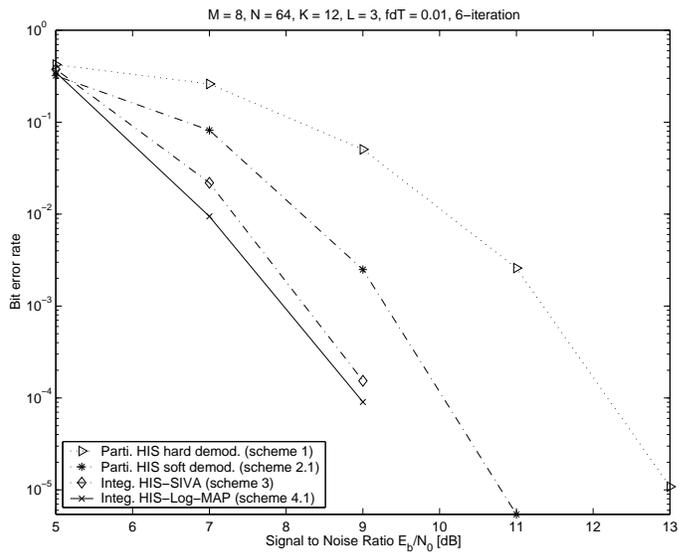


Figure 9.10: Different schemes with IS demodulator.

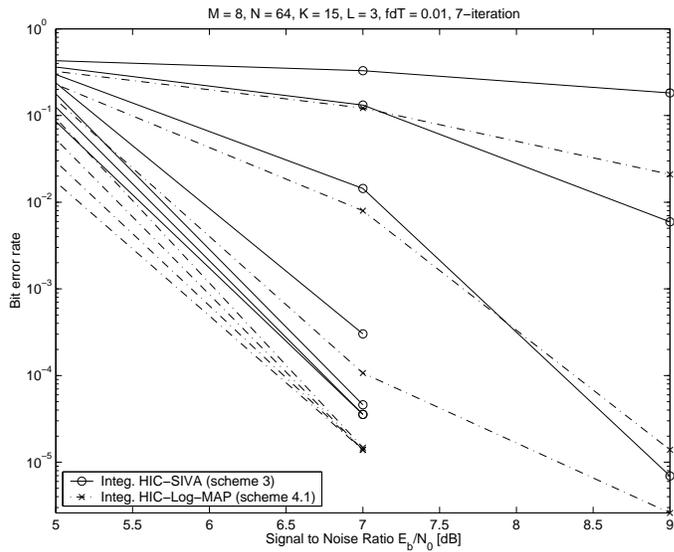


Figure 9.11: Comparison of two integrated approaches with SIVA/Log-MAP decoding.

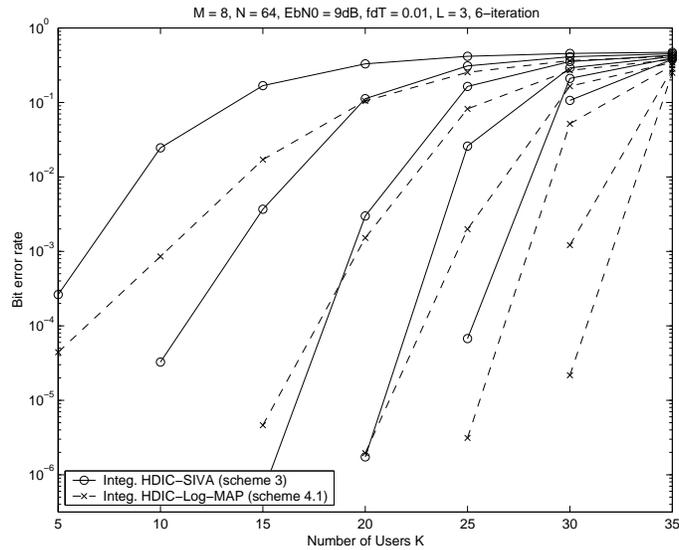


Figure 9.12: Test of system capacity.

Table 9.1: Test results for system capacity at $P_b = 10^{-3}$ and $E_b/N_0 = 9$ dB shown by Figure 9.12.

Number of iterations	1	2	3	4	5
Capacity with HDIC-Log-MAP (users)	10	19	23	27	30
Capacity with HDIC-SIVA (users)	6	14	19	23	27

9.5 Conclusions

In this chapter, we analyzed and compared the performance and complexity of several strategies for demodulating and decoding orthogonally modulated and convolutionally coded signals in frequency selective channels, with the emphasis placed on the development of soft demodulation algorithms which is the main contribution of this work. We have shown that of all the presented enhancement ideas, the replacement of hard demodulator with soft demodulator is most crucial in the improvement of system performance. The MF, IC, and IS based soft demodulators are introduced and compared. The noncoherent MF soft demodulator is usually used at the first stage of the iteration process to obtain an initial estimate of data for channel and interference estimation, which are needed for subsequent stages of HDIC/HDIS or SDIC/SDIS soft demodulation. The IC and IS demodulators differ in that the former one estimates and subtracts interference from the received vector before demodulation; the latter one, on the other hand, removes the estimated interference from the received observation by filtering. The IC soft demodulator is preferred because it achieves superior

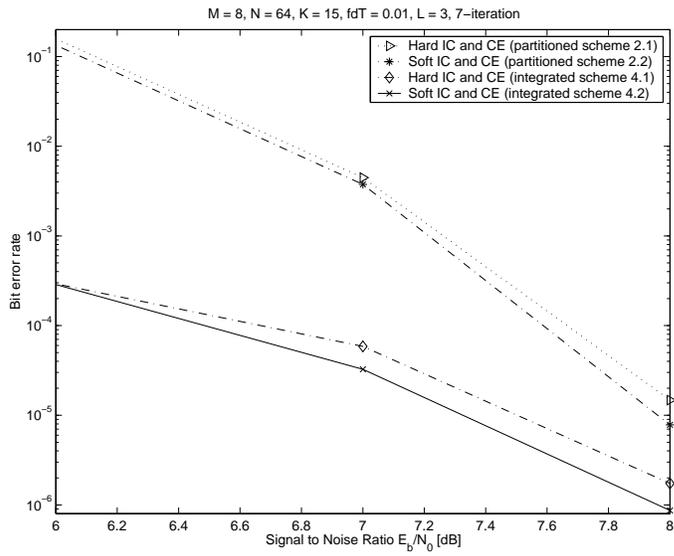


Figure 9.13: Hard vs. soft IC & CE.

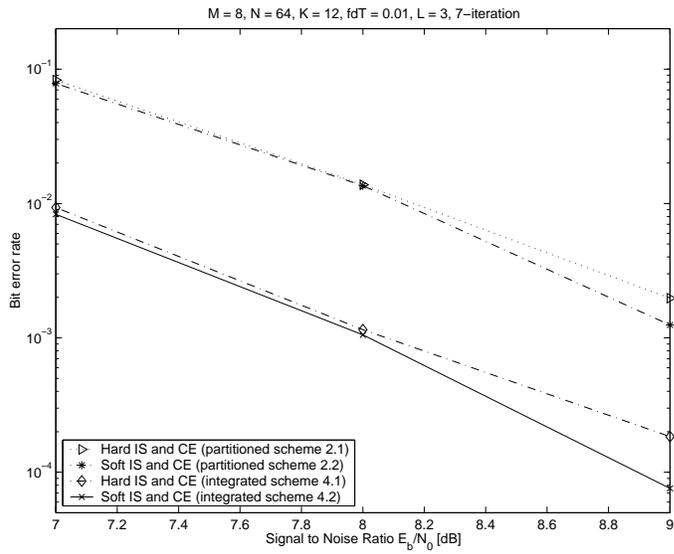


Figure 9.14: Hard vs. soft IS & CE.

performance with less complexity compared to the IS soft demodulator. In general, when long codes are employed, the use of linear demodulators becomes cumbersome, as in this case, the crosscorrelations between different users' signature sequences vary at the data rate.

In order to remove deteriorative effect of interference, we need an estimate of the fading channel. Both channel estimation and interference cancellation/suppression can be implemented using the hard/soft decisions either from the output of demodulator or channel decoder. In the former case, the demodulation and decoding blocks are partitioned as in the conventional system; in latter case, the two blocks are connected in an iterative fashion so that the output of the channel decoder can be fed back to the demodulator. The integrated approach outperforms the partitioned approach, at cost of higher complexity. The increase in complexity is due to the need of going through channel decoding, modulation, sometimes also encoding processes at each iteration to derive an estimate of transmitted sequences. Convolutional decoder can be implemented either with VA or Log-Map algorithm. Noticeable gain is obtained by replacing VA with Log-MAP, especially at the first few iterations. It is attributed to fact that Log-MAP is designed to minimize symbol (bit) error probability, thus yielding better estimate of information bits as well as better estimate of code bits for channel estimation and interference mitigation. Besides, Log-MAP decoding also enables SDIC/SDIS and soft CE, which are impossible with VA decoding. However, the gain comes with penalty of the increased complexity and memory consumption due to the implementation of forward and backward recursions in the decoding process. In order to utilize the soft outputs from Log-MAP decoder to improve the performance of the soft demodulator, a soft modulator is proposed in this chapter to derive an soft estimate of transmitted chip sequence for SDIC/SDIS and soft CE. Based on the numerically results and analysis, we conclude that the use of soft information for IC/IS and CE can slightly improve the system performance with minor increase in complexity compared to IC/IS and CE using hard decisions.

Different strategies are examined and compared numerically, and their behaviors interpreted in this chapter. To summarize, we conclude that significant gains are obtained by replacing the hard demodulator with a MUD assisted soft demodulator and also by integrating demodulation and decoding rather than performing them separately. The replacement of hard output channel decoder and hard decision directed IC/IS, CE with their soft counterparts further enhance the reliability of the system, however, the gain is less significant. The general conclusion is that the performance improves with increased computational complexity. This performance/complexity tradeoff has to be considered in the receiver design of a practical CDMA system.

ITERATIVE DECODING WITH EXTRINSIC
INFORMATION

M -ary orthogonal modulation is essentially a process of block coding. This chapter aims at tackling the problem of joint iterative decoding of serially concatenated inner block code and outer convolutional code in multiuser environments. The (logarithm) maximum a posteriori probability, (Log)-MAP criterion is used to derive the iterative decoding schemes. In our system, the soft output from inner block decoder is used as a priori information for the outer decoder. The soft output from outer convolutional decoder is used for two purposes. First, it may be fed back to the inner decoder as extrinsic information for the systematic bits of the Walsh codeword. Secondly, it is utilized for channel estimation and multiuser detection. We also show that the inner decoding can be accomplished without extrinsic information, and in some cases, e.g., when the system is heavily loaded, yields better performance than the decoding with unprocessed extrinsic information. This implies the need for correcting the extrinsic information obtained from outer decoder. Different schemes are examined and compared numerically and it is shown that iterative decoding with properly corrected extrinsic information or with non-extrinsic/extrinsic adaptation enables the system to operate reliably in the presence of severe multiuser interference, especially when the inner decoding is assisted by decision directed channel estimation and interference cancellation techniques. Additional gain is noticed when soft information rather than hard decision feedback is used for channel estimation and interference cancellation.

10.1 Introduction

Turbo codes represent an important advancement in the area of power efficient communications. The practical importance of turbo codes stems from the fact that they enable reliable communications at signal-to-noise ratios close to the channel capacity with simple component codes, yet admit high performance iterative soft decoding algorithms with complexity not significantly higher than that of the decoder for single constituent code.

In a conventional communication receiver, only bits, or hard-decisions are passed between the subsystems. Information is lost and becomes unavailable to the subsequent stages whenever hard-decisions are made. Also, preceding stages can not benefit from the information derived by the following stages. The interface between each subsystem can be greatly improved by applying “turbo processing principle” which was first employed for decoding parallel concatenated convolutional codes, known as Turbo codes. With turbo processing, each subsystem is implemented with a soft-input, soft-output (SISO) algorithm, such as MAP or Log-MAP. Soft decision values, typically in the form of log-likelihood ratios (LLRs), are passed down the chain and refined by the subsequent stages. The soft output of the final stage is then fed back to the first stage and a second iteration of the processing is initiated. Several iterations of turbo processing can be executed to improve performance.

The turbo principle is a general strategy of iterative feedback decoding or detection [68], and can be used in a more general way than just for the decoding of parallel concatenated convolutional codes. It has been successfully applied to many detection/decoding problems such as serial concatenation, equalization, coded modulation, multiuser detection, joint source and channel decoding and others [69].

The problem of iterative decoding for serially concatenated codes (consisting of inner code and outer code) has been addressed e.g., in [70] for serially concatenated convolutional codes and in [23, 71] for serially concatenated block code and convolutional code. Analogous to the decoding of turbo codes, the inner decoder extracts the soft information from the outer decoder to update and improve its soft decision on code bits. The inner decoder also provide the outer decoder with soft unquantized decisions to improve performance. The process of passing soft information between two SISO stages proceeds, and after a few iterations, the information data are decoded with a hard decision at the output of the outer decoder. In [71, 72], a MAP demodulator and a SOVA (soft-output VA) decoder were applied to a similar system using M -ary modulation and FEC. A performance gain of about 0.6 dB at a bit error rate (BER) of 10^{-3} was noticed for a single user system in AWGN channel when compared against the conventional non-SISO demodulator and decoder. It was indicated in [72] that the interleaver design has significant impact on the system performance. However, some important issues e.g., channel estimation and MAI mitigation were

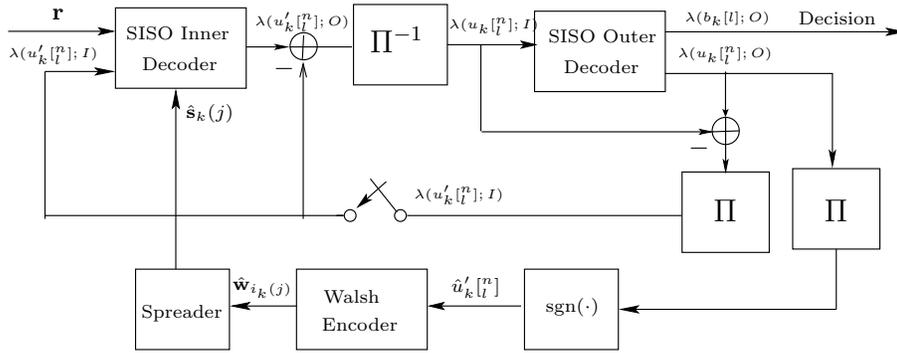


Figure 10.1: Block diagram of iterative decoding for user k .

not addressed in the above references.

In addition to FEC, multiuser detection is another effective tool to increase the capacity of interference limited CDMA systems. The algorithms developed for joint multiuser detection and decoding in most literature are confined to uncascaded systems with single convolutional code, and the issue of joint detection/decoding and channel estimation is not investigated. In this chapter, we study iterative decoding of a serially cascaded asynchronous CDMA system which involves convolutional coding and orthogonal modulation. The contribution of this work is the treatment of joint multiuser detection, decoding and channel estimation by utilizing turbo processing principle for the systems in question. The iterative decoding is assisted by hard or soft decision directed channel estimation and interference cancellation to effectively combat interference. Some correction and adaptation algorithms are proposed to better utilize the extrinsic information in bad channels (severe multiuser interference environment).

The iterative receiver structure for decoding the data transmitted by user k is illustrated in Fig. 10.1. It consists of two stages: a SISO inner block decoder, followed by a SISO outer convolutional decoder. The two stages are separated by the deinterleaver Π^{-1} and the interleaver Π . The k^{th} user's outer convolutional decoder takes $\lambda(u_k[l]^n; I)$, the extrinsic values of the code bits, as input. It delivers as output an update of the LLRs of the code bits $\lambda(u_k[l]^n; O)$, as well as the LLRs of the information bits $\lambda(b_k[l]; O)$, based on the code constraints. The latter are used for making hard decisions on transmitted information bits at the final iteration; while the former are used for two purposes: deriving extrinsic information $\lambda(u'_k[l]^n; I)$ for inner decoding and deriving estimate of transmitted Walsh sequence $\hat{s}_k(j)$ for channel estimation and multiuser detection. The inner decoder accepts a priori information $\lambda(u'_k[l]^n; I)$ and channel values and delivers soft output value $\lambda(u'_k[l]^n; O)$. Decoding is based on alternately decoding the two component codes and passing the updated extrinsic information which is part

of the soft output of the SISO decoder to the next decoding stage. The process is repeated until no further refinement is noticed and ended by making a hard decision on the LLR values of the information bits in the last iteration.

To avoid statistical dependencies between the soft values of several iteration steps, it is necessary to feed back only the extrinsic value

$$\lambda(u_k[l]; I) = \Pi^{-1} \{ \lambda(u_k[l]; O) - \lambda(u_k[l]; I) \}$$

to the outer decoder and

$$\lambda(u_k[l]; I) = \Pi \{ \lambda(u_k[l]; O) - \lambda(u_k[l]; I) \}$$

to the inner decoder as shown in Fig. 10.1. These two decoder modules are discussed in detail next.

10.2 SISO Outer Convolutional Decoder

In case the code rate of the convolutional code is $R_c = 1/r$, the information bit $b_k[l]$ is encoded into code bits $u_k[l]$, $n = 1, 2, \dots, r$. To facilitate understanding of the principle of outer decoding, we take a simple example and assume a rate 1/2 convolutional code with generator polynomial (7, 5) in octal form as demonstrated in Fig. 10.2. In this case, the code trellis is binary, one branch corresponds to a single information bit $b_k[l]$ and two corresponding code bits $u_k[l]^0, u_k[l]^1$. Let us denote the state metrics at the $(l-1)^{th}$ node $\alpha_{l-1}(s')$ and those at the l^{th} node $\alpha_l(s)$, where $s' \in \{s'_0, s'_1, s'_2, s'_3\}$ and $s \in \{s_0, s_1, s_2, s_3\}$ are the generic states at the $(l-1)^{th}$ and l^{th} nodes, respectively.

Since Log-MAP algorithm is equivalent to (true) symbol-by-symbol MAP, while without the numerical representation problems posed in MAP [67], it is adopted here for outer convolutional decoding. With Log-MAP algorithm, $\alpha_l(s)$ is computed recursively as

$$\alpha_l(s) = \max_{s'}^* [\alpha_{l-1}(s') + \gamma_l(s', s)]; \quad (10.1)$$

with initial conditions $\alpha_0(0) = 0$, $\alpha_0(s \neq 0) = -\infty$, and $\gamma_l(s', s)$ is the branch metric for the branch connecting state s' at node $(l-1)$ to state s at node l . In (10.1), the summation is over all the states s' where the transition (s', s) is possible. The function $\max^*(\cdot)$ is defined as

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|}) \quad (10.2)$$

which is max operation compensated with a correction term $\ln(1 + e^{-|x-y|})$. The state metrics for the portion of the trellis beyond the l^{th} node can be

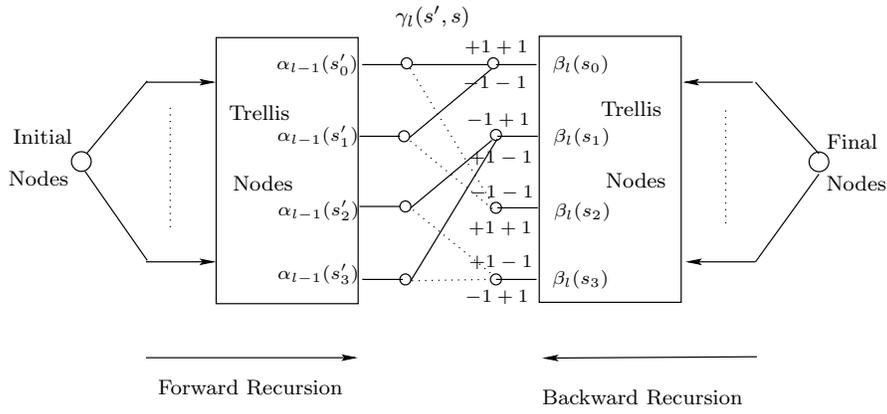


Figure 10.2: Trellis of the code with rate 1/2 and generator polynomial (111,101). The state transition represented by a solid line is due to input bit +1, the state transition represented by a dashed line is due to input bit -1. The output bits are written on top/bottom of each ending node. The trellis starts from all-zero state and ends at all-zero state with zero tail-bits.

computed similarly by a backward recursion starting at the last node

$$\beta_{l-1}(s') = \max_s^* [\beta_l(s) + \gamma_l(s', s)]; \quad (10.3)$$

with initial conditions $\beta_{L_b}(0) = 0$, $\beta_{L_b}(s \neq 0) = -\infty$, and L_b is the block length. The summation is over all the states s where the transition (s', s) is possible.

Let \mathcal{B}_l^+ be the set of state pairs (s', s) such that the information bit is +1 at time l . Similarly define \mathcal{B}_l^- . In the case shown in Fig. 10.2,

$$\begin{aligned} \mathcal{B}_l^+ &= \{(s'_0, s_0), (s'_1, s_0), (s'_2, s_1), (s'_3, s_1)\} \\ \mathcal{B}_l^- &= \{(s'_0, s_2), (s'_1, s_2), (s'_2, s_3), (s'_3, s_3)\} \end{aligned} \quad (10.4)$$

The LLR value of the information bit $b_k[l]$ at the output of the channel decoder is given by

$$\begin{aligned} \lambda(b_k[l]; O) &= \ln \frac{\sum_{\mathcal{B}_l^+} \exp[\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)]}{\sum_{\mathcal{B}_l^-} \exp[\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)]} \\ &= \max_{\mathcal{B}_l^+}^* [\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)] \\ &\quad - \max_{\mathcal{B}_l^-}^* [\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)] \end{aligned} \quad (10.5)$$

Let \mathcal{U}_{l0}^+ be the set of state pairs (s', s) such that the first code bits $u_k[l]^0$ is +1. Similarly define \mathcal{U}_{l0}^- . In the case shown in Fig. 10.2,

$$\begin{aligned}\mathcal{U}_{l0}^+ &= \{(s'_0, s_0), (s'_3, s_1), (s'_1, s_2), (s'_2, s_3)\} \\ \mathcal{U}_{l0}^- &= \{(s'_1, s_0), (s'_2, s_1), (s'_0, s_2), (s'_3, s_3)\}\end{aligned}$$

The LLR value of the first code bit $u_k[l]^0$ at the output of the channel decoder is given by

$$\begin{aligned}\lambda(u_k[l]^0; O) &= \ln \frac{\sum_{\mathcal{U}_{l0}^+} \exp[\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)]}{\sum_{\mathcal{U}_{l0}^-} \exp[\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)]} \\ &= \max_{\mathcal{U}_{l0}^+}^* [\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)] \\ &\quad - \max_{\mathcal{U}_{l0}^-}^* [\alpha_{l-1}(s') + \gamma_l(s', s) + \beta_l(s)]\end{aligned}\quad (10.6)$$

The soft value of the second code bit $u_k[l]^1$ can be computed in a similar way. To reduce the complexity, the $\max^*(\cdot)$ operation can be approximated by $\max(\cdot)$, i.e., ignoring the compensation term in (10.2), which leads to the so-called max-log-MAP algorithm.

According to the definition $\lambda(u_k[l]^n; I) = \ln \frac{P(u_k[l]^n=+1)}{P(u_k[l]^n=-1)}$, we can readily derive [23]

$$\begin{aligned}P(u_k[l]^n = \pm 1) &= \frac{\exp[\pm \lambda(u_k[l]^n; I)]}{1 + \exp[\pm \lambda(u_k[l]^n; I)]} \\ &= \frac{\exp[-\lambda(u_k[l]^n; I)/2]}{1 + \exp[-\lambda(u_k[l]^n; I)]} \cdot \exp[u_k[l]^n \lambda(u_k[l]^n; I)/2] \\ &= A_l \exp[u_k[l]^n \lambda(u_k[l]^n; I)/2]\end{aligned}\quad (10.7)$$

For a rate $R_c = 1/r$ code, the branch metric, $\gamma_l(s', s)$ can be computed as [22]

$$\begin{aligned}\gamma_l(s', s) &= \ln \prod_{n=1}^r P(u_k[l]^n) = \sum_{n=1}^r \ln P(u_k[l]^n) \\ &= \sum_{n=1}^r \ln A_l + \sum_{n=1}^r \ln \{\exp[u_k[l]^n \lambda(u_k[l]^n; I)/2]\} \\ &= a_l + \sum_{n=1}^r u_k[l]^n \lambda(u_k[l]^n; I)/2\end{aligned}\quad (10.8)$$

The constant a_l is independent of $u_k[l]^n$ and will be canceled since $\gamma_l(s', s)$

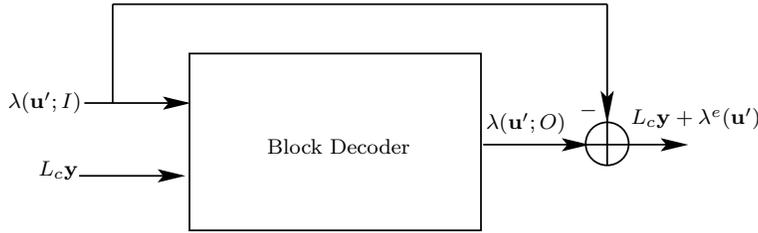


Figure 10.3: Inner decoding with extrinsic feedback.

appears both in the numerator (where $u_k[l^n] = +1$) and denominator (where $u_k[l^n] = -1$) of equations (10.5) and (10.6). In the case shown in Fig. 10.2,

$$\gamma_l(s', s) = \frac{1}{2} \{u_k[l^0] \lambda(u_k[l^0]; I) + u_k[l^1] \lambda(u_k[l^1]; I)\}$$

10.3 SISO Inner Block Decoder

Fig. 10.3 illustrates the principle for inner decoding when the extrinsic values are used. The decoder accepts a priori information $\lambda(\mathbf{u}'; I)$ and channel values $L_c \mathbf{y}$. The vector $\mathbf{u}' \in \{u'_k[l^n]\}$ is of length $\log_2 M$, it consists of the $\log_2 M$ systematic bits of the Walsh codeword $\mathbf{w}_{i_k(j)}$. The soft output $\lambda(\mathbf{u}'; O)$ can be expressed in the form

$$\lambda(\mathbf{u}'; O) = L_c \mathbf{y} + \lambda(\mathbf{u}'; I) + \lambda^e(\mathbf{u}') \quad (10.9)$$

which consists of three parts: the channel value $L_c \mathbf{y}$; a priori information $\lambda(\mathbf{u}'; I)$ about \mathbf{u}' provided from the outer decoder; extrinsic value $\lambda^e(\mathbf{u}')$ which is the updated information about the current bit obtained through the decoding process from all the other bits through constraints of the code. The output is subtracted by $\lambda(\mathbf{u}'; I)$ to eliminate the statistical dependency before feeding to the outer decoder.

The LLR of a transmitted $+1$ and -1 for every coded and interleaved bit $u'_k[l^n]$ from each user $k = 1, 2, \dots, K$ is given according to [69, 71] by

$$\begin{aligned} \lambda(u'_k[l^n]; O) &= \ln \frac{P[u'_k[l^n] = +1 | \mathbf{y}]}{P[u'_k[l^n] = -1 | \mathbf{y}]} = \ln \frac{\sum_{m: u'_k[l^n] = +1} P(\mathbf{w}_m | \mathbf{y})}{\sum_{m: u'_k[l^n] = -1} P(\mathbf{w}_m | \mathbf{y})} \\ &= \ln \frac{\sum_{m: u'_k[l^n] = +1} \exp\left(\frac{1}{2} \sum_{i=1}^N L(w_i; y_i) \cdot w_i\right)}{\sum_{m: u'_k[l^n] = -1} \exp\left(\frac{1}{2} \sum_{i=1}^N L(w_i; y_i) \cdot w_i\right)} \\ &= \ln \frac{\sum_{m: u'_k[l^n] = +1} \exp\left(\frac{1}{2} \mathbf{L}^T \mathbf{w}_m\right)}{\sum_{m: u'_k[l^n] = -1} \exp\left(\frac{1}{2} \mathbf{L}^T \mathbf{w}_m\right)} \end{aligned} \quad (10.10)$$

where we use the notation $m : u'_{[l]}^{[n]} = \pm 1$ to denote the set of Walsh codes $\{\mathbf{w}_m\}$ that correspond to the code bit $u'_k[l] = \pm 1$, and assume $u'_k[l]$ is one of the $\log_2 M$ systematic bits of the inner Walsh codeword. The i^{th} bit of the Walsh codeword \mathbf{w}_m is denoted as $w_i \in \{+1, -1\}$. The vector \mathbf{y} is of length M , and is due to the k^{th} user's j^{th} transmitted Walsh symbol, and is obtained by despreading and Rake combining of the received vector $\mathbf{r}(k, j)$ or its interference canceled version $\mathbf{r}'(k, j)$. The vectors \mathbf{u}' and \mathbf{y} change from one processing window to the next, which is not explicitly shown in the figure and equations to simplify notation. The process of despreading and multipath combining will be elaborated shortly in the next subsection when different approaches of inner decoding are discussed. Each element of the vector $\mathbf{L} = [L(w_1; y_1), L(w_2; y_2), \dots, L(w_M; y_M)]^T$ is defined as

$$L(w_i; y_i) = \begin{cases} L_c \cdot y_i + \lambda(u'_k[l]; I), & \text{for } i = \frac{M}{2^{s+1}}, s = 0, \dots, \log_2 M - 1 \\ L_c \cdot y_i, & \text{otherwise} \end{cases}$$

which is the channel value y_i multiplied with channel reliability L_c supplemented with a priori information $\lambda(u'_k[l]; I)$ for the $\log_2 M$ systematic bits of each codeword \mathbf{w}_m , and L_c is defined such that $L_c y_i = \ln \frac{p(y_i | w_i = +1)}{p(y_i | w_i = -1)}$. Typically, one term will dominate each sum in (10.10), which suggests the “dual-maxima” approximation [65, 71]

$$\lambda(u'_k[l]; O) \approx \frac{1}{2} \max_{m: u'_k[l] = +1} \mathbf{L}^T \mathbf{w}_m - \frac{1}{2} \max_{m: u'_k[l] = -1} \mathbf{L}^T \mathbf{w}_m \quad (10.11)$$

The vectors \mathbf{y} and \mathbf{L} should be formed and L_c computed according to the chosen strategy for the inner decoding, which can be a traditional single user approach or a MUD-aided approach as discussed below.

10.3.1 Conventional single user approach

The conventional inner decoding scheme is illustrated in Fig. 10.4. For simplicity of notation we will suppress the index k and/or j from $\mathbf{s}_k(j)$, $\mathbf{C}_k(j)$, $\mathbf{r}(k, j)$, $\mathbf{A}(k, j)$, $\mathbf{n}(k, j)$, $\mathbf{X}_k(j)$ and $\mathbf{h}_k(j)$, etc., whenever no ambiguity arises.

Let $\mathbf{r}_{k,l}$, ($l = 1, 2, \dots, L_k$) denote the delay aligned version of the received vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path. The vector $\tilde{\mathbf{r}}_{k,l} = [\tilde{r}_{k,l}[1] \ \tilde{r}_{k,l}[2] \ \dots \ \tilde{r}_{k,l}[N]]^T \in \mathbb{C}^N$ and $\tilde{\mathbf{n}}_{k,l} \in \mathbb{C}^N$ are $\mathbf{r}_{k,l}$ and the original noise vector \mathbf{n} descrambled with the scrambling sequence \mathbf{C}_k . Ideally, different user's scrambling sequences are orthogonal to each other (their crosscorrelations are approximately zero) and their autocorrelations approximate delta function. Hence, $\mathbf{r}_{k,l}$ after descrambling and despreading will hopefully only contain the contribution from k^{th} user's l^{th} path plus additive noise.

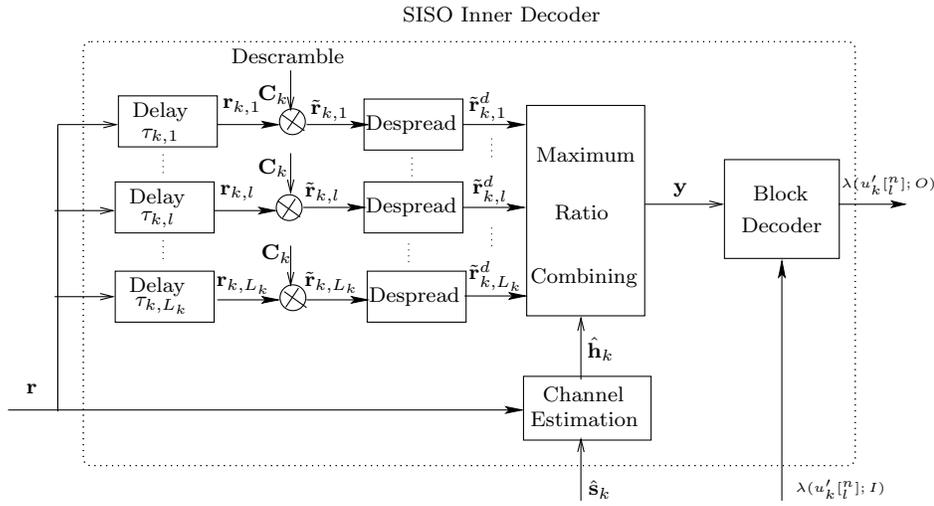


Figure 10.4: Internal structure of the conventional SISO inner decoder.

Let us assume unit chip energy and define

$$\tilde{\mathbf{r}}_{k,l}^d = [\tilde{r}_{k,l}^d[1] \cdots \tilde{r}_{k,l}^d[M]] \in \mathbb{C}^M$$

The i^{th} element $\tilde{r}_{k,l}^d[i]$ is the output of the l^{th} path's despreader corresponding to w_i , it is formed simply by

$$\begin{aligned} \tilde{r}_{k,l}^d[i] &= \sum_{n=1}^{N_c} \tilde{r}_{k,l}[(i-1)N_c + n] \\ &= h_{k,l} \cdot N_c \cdot w_i + \sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \end{aligned} \quad (10.12)$$

where $N_c = N/M$ is the number of chips for each bit w_i . Let us define $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_M]$ the output of Maximum Ratio Combiner (MRC) with element y_i computed as

$$y_i = \text{Re} \left\{ \sum_{l=1}^{L_k} \hat{h}_{k,l}^* \cdot \tilde{r}_{k,l}^d[i] \right\} = \text{Re} \left\{ \sum_{l=1}^{L_k} \hat{h}_{k,l}^* \sum_{n=1}^{N_c} \tilde{r}_{k,l}[(i-1)N_c + n] \right\} \quad (10.13)$$

where $\hat{h}_{k,l}$ is an estimate of the channel gain $h_{k,l}$. Substituting (10.12) into (10.13) and assume perfect channel estimation, i.e., $\hat{h}_{k,l} = h_{k,l}$, we

derive

$$\begin{aligned}
y_i &= \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \left(h_{k,l} \cdot N_c \cdot w_i + \sum_{n=1}^{N_c} \tilde{n}_{k,l} [(i-1)N_c + n] \right) \right\} \\
&= N_c \cdot w_i \sum_{l=1}^{L_k} |h_{k,l}|^2 + \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \sum_{n=1}^{N_c} \tilde{n}_{k,l} [(i-1)N_c + n] \right\} \\
&= N_c \cdot w_i \cdot \check{P}_k + n_{yi}
\end{aligned} \tag{10.14}$$

where $\check{P}_k = \sum_{l=1}^{L_k} \check{P}_{k,l} = \sum_{l=1}^{L_k} |h_{k,l}|^2$ is the received power from all paths of user k .

Since the descrambling operation does not change the noise statistic, we have the complex noise sample $\tilde{n}_{k,l}[i] \sim \mathcal{CN}(0, N_0)$, and

$$\sum_{n=1}^{N_c} \tilde{n}_{k,l} [(i-1)N_c + n] \sim \mathcal{CN}(0, N_c N_0)$$

Thereby,

$$\begin{aligned}
n_{yi} &= \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \sum_{n=1}^{N_c} \tilde{n}_{k,l} [(i-1)N_c + n] \right\} \sim \mathcal{N}(0, N'_0/2) \\
N'_0 &= N_c \cdot \check{P}_k \cdot N_0
\end{aligned} \tag{10.15}$$

Recall that $y_i = N_c \cdot \check{P}_k \cdot w_i + n_{yi}$, thus

$$\begin{aligned}
p(y_i | w_i = \pm 1) &= \frac{1}{\sqrt{\pi N'_0}} \exp \left[\frac{-(y_i \mp N_c \check{P}_k)^2}{N'_0} \right] \\
\ln \frac{p(y_i | w_i = +1)}{p(y_i | w_i = -1)} &= \frac{-(y_i - N_c \check{P}_k)^2 + (y_i + N_c \check{P}_k)^2}{N'_0} \\
&= \frac{4N_c \check{P}_k y_i}{N_c \check{P}_k N_0} = \frac{4}{N_0} \cdot y_i
\end{aligned} \tag{10.16}$$

From (10.16), we obtain the channel reliability value $L_c = 4/N_0$. In reality, the code orthogonality condition is hardly fulfilled. Also there will be errors in channel estimation, the condition $\hat{h}_{k,l} = h_{k,l}$ can not be guaranteed. The algorithm derived based on these assumptions is therefore quite suboptimal. Especially, the presence of MAI and ISI will deteriorate the system performance. One way to work around this problem would be to increase the value of N_0 to capture the MAI and ISI. A more effective measure to alleviate their effect is the use of multiuser detection (MUD) techniques, which will be discussed next.

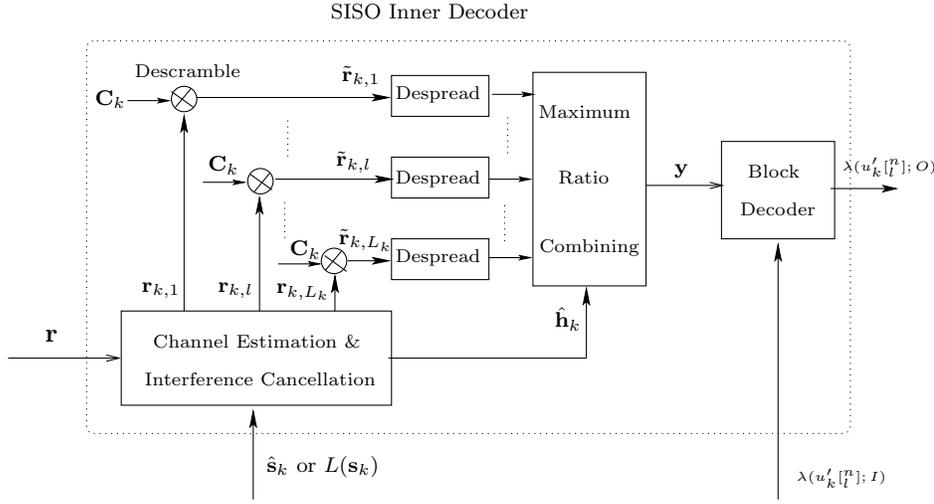


Figure 10.5: Internal structure of MUD-aided SISO inner decoder.

10.3.2 MUD approach

The aperiodic nature of the long codes employed in this work usually precludes the use of linear multiuser detection schemes like MMSE detector and decorrelator, etc. due to high computational complexity. Therefore, only the nonlinear parallel interference cancellation scheme is considered here. The inner decoding scheme combined with interference cancellation is depicted in Fig. 10.5. Interference cancellation with hard and soft decision feedback will be discussed next.

10.3.2.1 Hard decision interference cancellation

Hard decision interference cancellation (HDIC) is performed by estimating the transmitted signals in parallel for all the users, and then subtracting the estimated signals of the interfering users from the received signal \mathbf{r} to form a new signal vector $\mathbf{r}_{k,l}^{\text{HDIC}}$ for demodulation of the signal transmitted from the l^{th} path of user k . Mathematically, it is expressed as

$$\mathbf{r}_{k,l}^{\text{HDIC}} = \mathbf{r} - \hat{\mathbf{A}}\hat{\mathbf{h}} + \hat{\mathbf{a}}_k\hat{h}_{k,l} \quad (10.17)$$

where $\mathbf{r} \in \mathbb{C}^{N_k}$ denote the received signal vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path, it contains N_k chips (usually $N_k > N$ due to multipath delay spread). The vector $\mathbf{r}_{k,l}^{\text{HDIC}} \in \mathbb{C}^{N_k}$ is its interference canceled version after subtracting the contributions from all the other users and the same user's other paths using hard decision feedback. The vector $\hat{\mathbf{A}}\hat{\mathbf{h}}$ represents the estimated contribution from all the users calculated by using the data matrix $\hat{\mathbf{A}}$ and channel vector $\hat{\mathbf{h}}$ estimated at

the previous iteration. The vector $\hat{\mathbf{a}}_k \hat{h}_{k,l}$ is the estimated contribution from the l^{th} path of user k .

The derivation of L_c is the same as in Section 10.3.1 except $\mathbf{r}_{k,l}$ is replaced by $\mathbf{r}_{k,l}^{\text{HDIC}}$, the delay compensated and MAI and ISI canceled version of the received vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path. The MAI and ISI are estimated by making tentative hard decisions on the output from the outer decoder, i.e., $\hat{u}_k[l^n] = \text{sgn}\{\Pi(\lambda(u_k[l^n]; O))\}$ (see Fig. 10.1) for all k . Then we go through block encoding and spreading to produce an estimate of the transmitted chip sequence $\hat{\mathbf{s}}_k$, which is used for both interference cancellation and channel estimation. Channel estimation was treated in Chapter 5.

In the ideal situation, the MAI from other users and ISI from the same user's other paths are canceled. Going through the same procedure as shown in (10.12) – (10.16), we come up with the same channel reliability value $L_c = 4/N_0$. However, the mechanisms for deriving $\mathbf{r}_{k,l}$ and $\mathbf{r}_{k,l}^{\text{HDIC}}$ are different (single user and MUD approach, respectively) which result in different \mathbf{y} and \mathbf{L} vectors used in equations (10.10) – (10.11) for computing LLR values.

It should be noted that the inner decoding can be accomplished without extrinsic information. In this case, $L(w_i; y_i) = L_c y$ for all i in equation (10.10) and (10.11). The switch in Fig. 10.1 is turned off. The performance can still be improved at each iteration without extrinsic information because we get better estimate of the channel $\hat{\mathbf{h}}_k$ and transmitted sequence $\hat{\mathbf{s}}_k$ (better cancellation) as the iteration proceeds.

10.3.2.2 Soft decision interference cancellation

To reduce the likelihood of error propagation, we can use soft information $L(\mathbf{s}_k)$ instead of hard decision on \mathbf{s}_k for interference cancellation and channel estimation. When soft IC is used, the iterative decoding scheme illustrated in Fig. 10.1 should be modified accordingly. Fig. 10.6 shows the revised version.

In [21, 22], interference cancellation and channel estimation using soft information were proposed, which is, however, not directly applicable in our scenario, because we do not have the soft estimates for all the inner code bits, but only for the systematic bits. To derive the LLR value of \mathbf{s}_k , we feed $\lambda(u_k[l^n]) = \Pi\{\lambda(u_k[l^n]; O)\}$ into a soft inner encoder which computes $\lambda(\mathbf{w}_{i_k(j)}^p)$, the LLRs for codeword bits $\mathbf{w}_{i_k(j)}^p$, then spread them to derive $L(\mathbf{s}_k)$. The design of the soft encoder (modulator) was introduced in Section 9.2.4. With soft estimate of \mathbf{s}_k , we can derive the cancellation residual after soft cancellation as

$$\mathbf{r}_{k,l}^{\text{SDIC}} = \mathbf{r} - \mathbf{E}[\mathbf{A}|\mathbf{r}]\hat{\mathbf{h}} + \mathbf{E}[\mathbf{a}_k|\mathbf{r}]\hat{h}_{k,l} \quad (10.18)$$

where $\mathbf{E}[\mathbf{a}_k|\mathbf{r}]$, and the columns of $\mathbf{E}[\mathbf{A}|\mathbf{r}]$, are derived by scrambling $\mathbf{E}[\mathbf{s}_k|\mathbf{r}]$

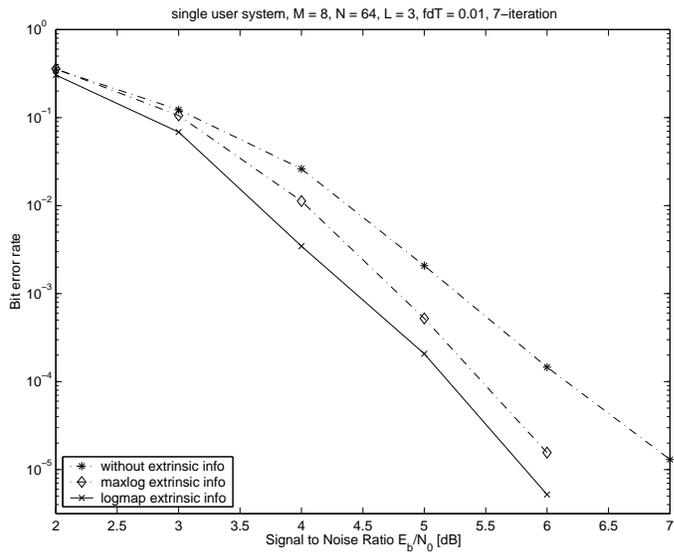


Figure 10.7: Performance of iterative decoding for single user system.

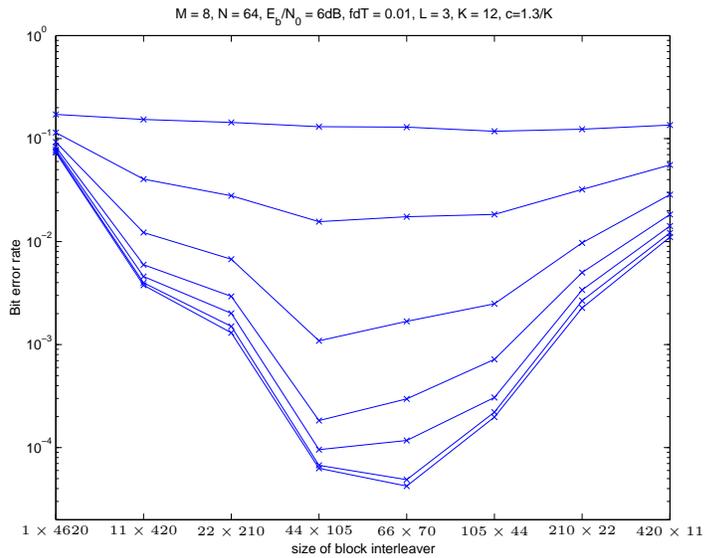


Figure 10.8: Performance of different block interleavers on 7-stage HDIC aided iterative decoding with extrinsic information.

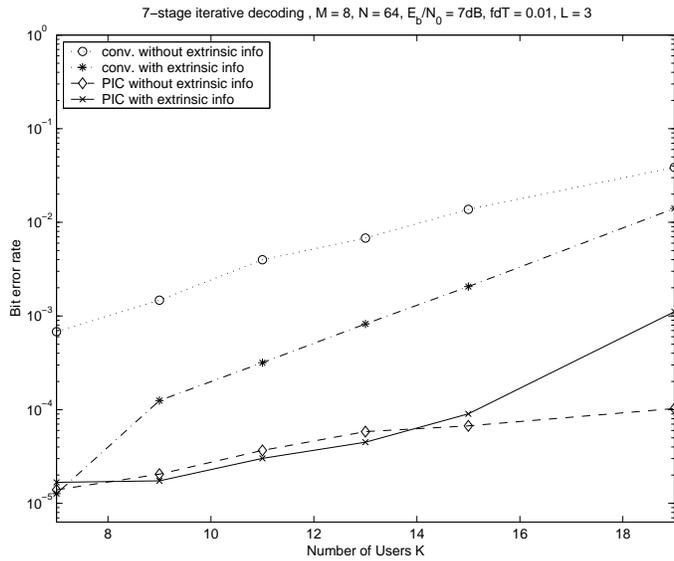


Figure 10.9: Comparison of different iterative decoding schemes.

extrinsic information, a similar trend is also noticed for other algorithms. The interleaver size is therefore set to be 66×70 for all the conducted experiments.

Different schemes discussed above are compared in Fig. 10.9. As expected, HDIC aided iterative decoding outperforms the conventional single user approach in a multiuser environment. The results also show that the reliability of the extrinsic information goes down as the number of user increases. It is attributed to the fact that these algorithms assume perfect channel estimation and perfect cancellation. However, this is not the case in a system with high level of interference. When the number of user goes beyond 14, the HDIC aided decoding without extrinsic information performs better than the one with extrinsic information. A similar trend is also observed with conventional approach, not as drastic though (the gain by applying extrinsic information gradually diminishes as the number of user increases). In the next subsection, we propose some schemes to improve the reliability of the extrinsic information, and make it useful in heavily loaded systems.

10.5 Correction/Adaptation of Extrinsic Information

It is stated in [73] that for bad channels the reliability information of soft decoder output is too optimistic. The output can be considered as being

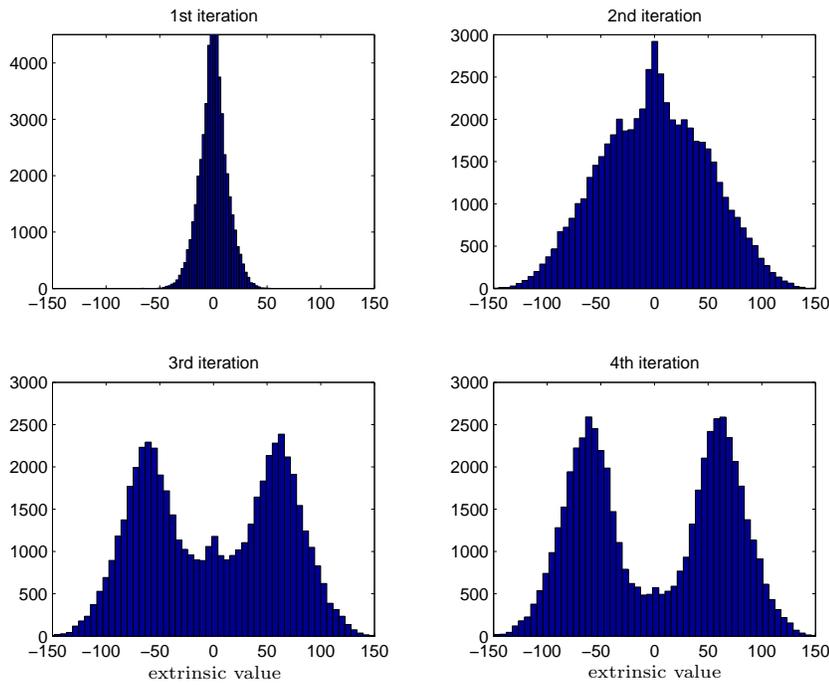


Figure 10.10: Histograms of the extrinsic value (12-user system, $E_b/N_0 = 7$ dB) at different iteration stages.

multiplied by a factor, that depends on the current BER. To become closer to the true LLR, the output has to be normalized (corrected). Although the authors drew the conclusion for the soft-output-Viterbi-decoder (SOVA) in bad channels (low SNR), we discovered similar problem also with our decoding scheme in severe interference environment. As indicated in Fig. 10.9, when the level of interference increases, the extrinsic information becomes unreliable and leads to worse performance than decoding without extrinsic feedback. That implies the need for normalization or correction of the extrinsic values.

Fig. 10.10 shows the histogram of the output (λ) of the SISO outer decoder at different iterations. Apparently, λ can be approximated as Gaussian distributed variable with mean value m_λ (or $-m_\lambda$) and variance σ_λ^2 . The pdf of λ conditioned on the bit $u = \pm 1$ being transmitted can be expressed as

$$p(\lambda|u = \pm 1) = \frac{1}{\sqrt{2\pi}\sigma_\lambda} \cdot \exp \left[-\frac{1}{2\sigma_\lambda^2} (\lambda \mp m_\lambda)^2 \right] \quad (10.19)$$

The conditional LLR, given the observation of the decoder output is

calculated as [73]

$$\begin{aligned}\lambda(u) &= \ln \frac{P(\lambda|u = +1)}{P(\lambda|u = -1)} = \ln \left[e^{-\frac{1}{2\sigma_\lambda^2}((\lambda - m_\lambda)^2 - (\lambda + m_\lambda)^2)} \right] \\ &= (2m_\lambda/\sigma_\lambda^2) \cdot \lambda\end{aligned}\quad (10.20)$$

which means the output λ has to be multiplied with the factor $c = 2m_\lambda/\sigma_\lambda^2$ to obtain the real LLR. Since the value of c depends on the current BER of the decoder output, which can vary from block to block, c has to be calculated for each block individually. From our experiments, we also notice that slightly better results can be achieved when modifying the normalization factor as

$$c = \begin{cases} 2m_\lambda/\sigma_\lambda^2, & \text{if } 2m_\lambda/\sigma_\lambda^2 < 1 \\ 1, & \text{otherwise} \end{cases}$$

The performance of this extrinsic normalization scheme is shown by the dashed curve in Fig. 10.11. It works rather well for a moderate number of users (up to 20 users). However, it gradually becomes ineffective as the system becomes more heavily loaded. We tend to think that the correction factor should be proportional to the reciprocal of the total number of user K to combat the detrimental effect of the interference. The solid line in Fig. 10.11 shows this correction method ($c = 1.3/K$) yields better performance than the extrinsic normalization scheme introduced above in severe MAI situation.

We compared different correction factors $c = 1/K, 1.3/K, 1.5/K$ in Fig. 10.12. In HDIC case, decoding with $c = 1.3/K$ and $c = 1.5/K$ give almost identical result, $c = 1/K$ is slightly worse. All of them perform better than decoding without extrinsic feedback, the gain is 0.4 – 1.1 dB in 15 user case. It proves that extrinsic information really helps improve the decoding performance if properly manipulated. With the conventional approach, the gain by introducing extrinsic correction is not noticeable: $c = 1.3/K$ and $c = 1$ (meaning no extrinsic correction) yield almost the same result. It is worth noticing the significant gain achieved by incorporating HDIC into inner decoding compared to the conventional scheme, the difference can be as large as 2.4 dB. However, the price to pay for the performance improvement is the added complexity due to the interference cancellation process.

Another work around method to mitigate the deteriorative effect of the interference and to exploit extrinsic information more efficiently is to use some adaptive scheme. The basic idea is to do decoding without extrinsic feedback for a few iterations, the channel becomes cleaner (MAI and ISI are more effectively suppressed) and closer to single user channel as the iteration goes on. Then we turn on the extrinsic feedback and let it run for a few more iterations. The results of this adaptive decoding scheme are shown Fig. 10.13 and Fig. 10.14. Seven iterations (4 stages without

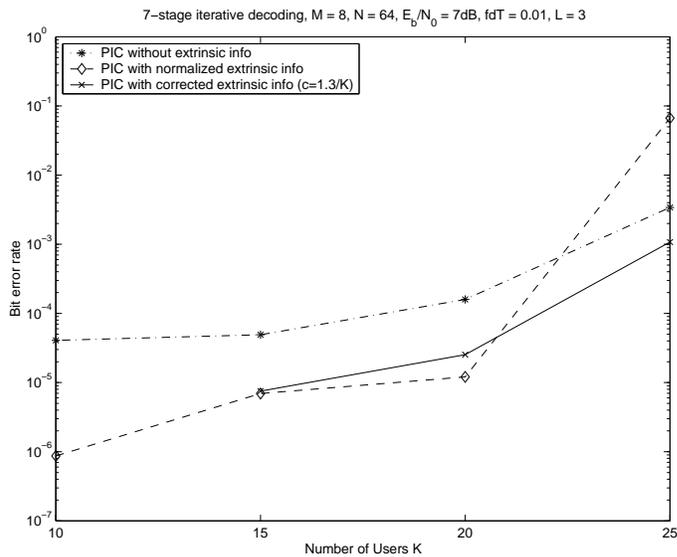


Figure 10.11: Performance of extrinsic correction/normalization schemes.

extrinsic feedback and 3 stages with unprocessed extrinsic feedback) are carried out in the test. It can be observed from Fig. 10.13 that adaptive decoding always performs better than decoding with unprocessed extrinsic information, it, however, converges to decoding without extrinsic feedback when the system becomes more heavily loaded (even performs slightly worse when the number of users goes beyond 19). But as indicated in Fig. 10.14 the gain achieved by the adaptive scheme tends to become bigger as SNR increases.

The initial LLRs are statistically independent in the iterative decoding process, however, since the decoders use indirectly the same information, the improvement through the iterative process becomes marginal, as the LLRs become more and more correlated. The convergence property of the iterative decoding algorithms is examined in Fig. 10.15. One can observe from the figure that iterative decoding without extrinsic information converges faster than the one with extrinsic information. Clearly, when exploited properly, extrinsic information improves the system performance, especially when SNR and the number of iterations increases. In both cases, 6 or 7 stages would suffice for maximum achievable performance.

Fig. 10.16 demonstrates the promising results when soft information rather than the hard decision feedback is used for interference cancellation and channel estimation. From the experiment results, an improvement of up to 0.5 dB gain is observed at BER around 10^{-5} for a 21-user system when the algorithm converges. As discussed earlier, the added complexity for deriving soft information rather than using hard decision feedback is

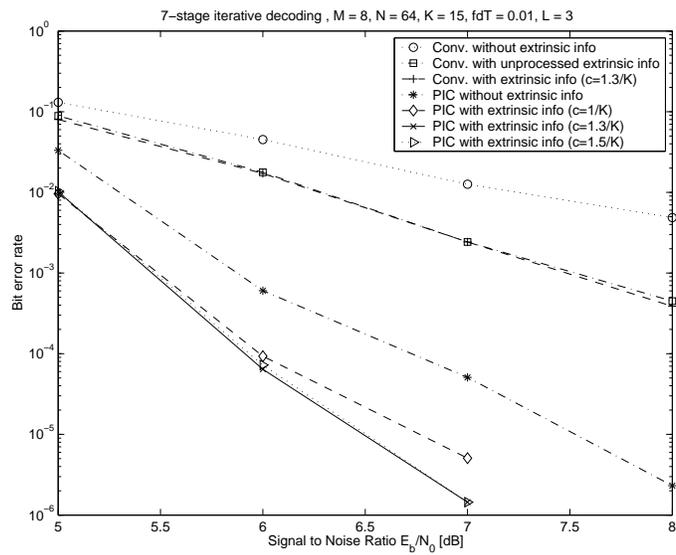


Figure 10.12: Performance of iterative decoding as function of SNR.

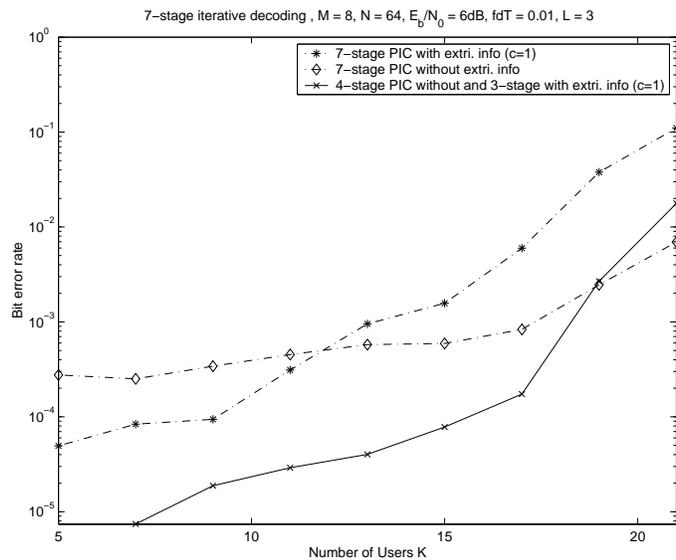


Figure 10.13: Adaptive iterative decoding: performance as function of K .

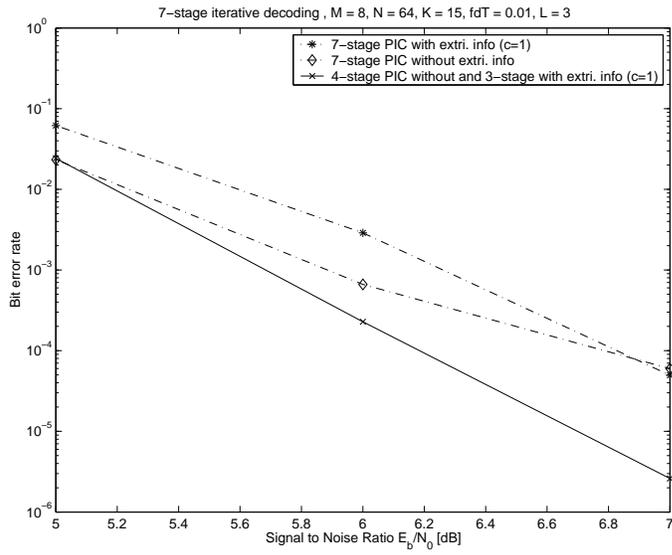


Figure 10.14: Adaptive iterative decoding: performance as function of SNR.

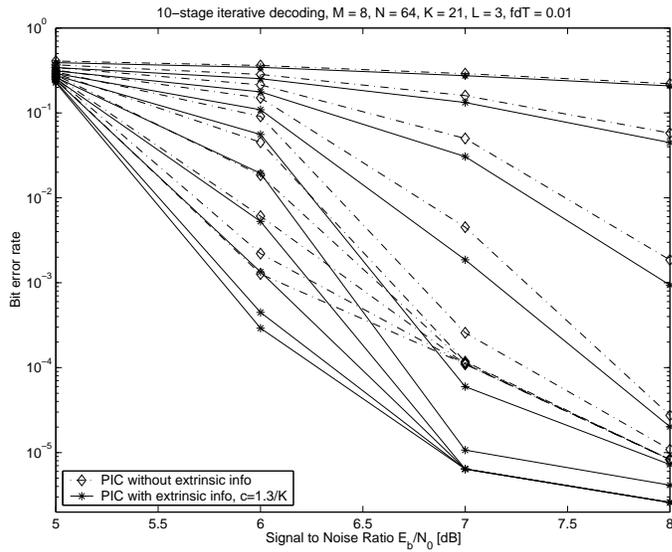


Figure 10.15: Performance of iterative decoding as function of SNR (21-user case).

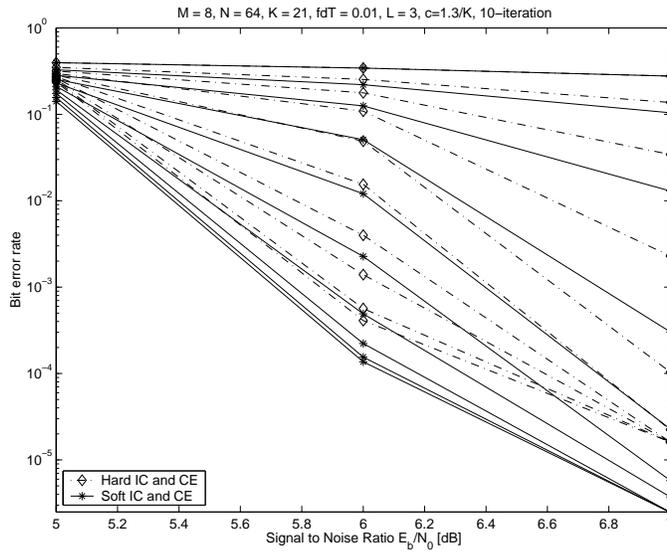


Figure 10.16: Hard vs. soft IC & CE in iterative decoding.

minor, which makes iterative decoding with soft IC and CE an attractive solution.

10.6 Conclusions

In this chapter, we presented an integrated approach to iterative multiuser detection, decoding and channel estimation for convolutionally coded and orthogonally modulated asynchronous CDMA systems in multipath Rayleigh fading channels. In addition to be used as extrinsic information for the inner decoder, tentative hard/soft decisions can be made on the output of the outer decoder for MAI and ISI cancellation to improve the performance of the inner soft decoder. Hard/soft decision directed channel estimation was also proposed for multipath Rake combining before decoding is done. Inner decoding can be done with or without extrinsic information. In the latter case, the performance improvement at each iteration is due to improved interference cancellation and channel estimation with decision feedback. The soft extrinsic information was found to have reduced reliability in bad channels (when the system is heavily loaded). Some extrinsic correction and non-extrinsic/extrinsic adaptation schemes were proposed to reduce the detrimental effect of the interference. The numerical results show that the inner decoding with corrected extrinsic feedback or with non-extrinsic/extrinsic adaptation outperforms the one without extrinsic feedback and that inner decoding with MAI and ISI cancellation is much

superior to the conventional single user decoding. Furthermore, soft information rather than hard decision feedback can be used for interference cancellation and channel estimation in order to achieve the best performance for the considered schemes.

- [1] J.G. Proakis. *Digital Communications*, 3rd edition, McGraw-Hill, 1995.
- [2] J.G. Proakis, M. Salehi. *Communication Systems Engineering*, Prentice Hall, NJ, 1994.
- [3] S. Glisic, B. Vucetic. *Spread spectrum CDMA systems for wireless communications*, Artech House, 1997.
- [4] B. Andy. *Digital Communications*, Addison-Wesley, 1999.
- [5] P.M. Shankar. *Introduction to wireless system*, John Wiley & Sons, 2001.
- [6] I. Glover, P. Grant. *Digital Communications*, Prentice Hall, 1998.
- [7] R.E. Ziemer, W.H. Tranter. *Principles of Communications*, John Wiley & Sons, 2002.
- [8] G.L. Stüber. *Principles of Mobile Communications*, Kluwer Academic Publishers, 1996.
- [9] T. Rappaport. *Wireless Communications, Principles and Practice*. Prentice-Hall, 1996.
- [10] W.C. Jakes. *Microwave Mobile Communications*, John Wiley & Sons, New York, 1974.
- [11] K. Pahlavan, M. Chase. "Spread-spectrum multiple-access performance of orthogonal codes for indoor radio communications". *IEEE Transactions on Communications*, vol. 38, no. 5, pp. 574-577, May 1990.

- [12] P. Enge, D. Sarwate. “Spread-spectrum multiple-access performance of orthogonal codes: linear receivers”. *IEEE Transactions on Communications*, vol. 35, no. 12, pp. 1309–1319, Dec. 1987.
- [13] T. Ojanperä and R. Prasad. An overview of air interface multiple access for IMT-2000/UMTS. *IEEE Communications Magazine*, vol. 36, pp. 82–95, Sept. 1998.
- [14] 3G TS 25.213. *3rd generation partnership project: technical specification group radio access networks; Spreading and modulation (FDD)*, V4.2.0, Dec. 2001.
- [15] S. Vembu, A. Viterbi. “Two different philosophies in CDMA - a comparison”. *Proc. Vehicular Technology Conference*, vol. 2, pp. 869–873, Jan. 1996.
- [16] S. Gollamudi, S. Nagaraj, Y.-F. Huang, and R. M. Buehrer. “Optimal multistage interference cancellation for CDMA systems using the non-linear MMSE criterion”. *Proc. 32nd Asilomar Conference on Signals, Systems, and Computers*, 1998.
- [17] M. Kawabe, T. Sato, T. Kato, A. Fukasawa, and R. Kohno. “M-ary CDMA scheme based on interference cancellation”. *Proc. Asia Pacific Conference on Circuit and Systems*, pp. 33–38, 1994.
- [18] J.I. Kim, S.H. Yoon, S.J. Kang and C.E. Kang. “Interference cancellation technique using channel parameter estimation in DS/CDMA system with M-ary orthogonal modulation”. *Electronics Letters*, vol. 34, no. 12, pp. 1194–1195, June 1998.
- [19] E. Ström, S. Miller. “Iterative demodulation and channel estimation of orthogonal signalling formats in asynchronous DS-CDMA systems”. *IEICE Transactions on Electronics*, vol. 85, no. 3, pp. 442–451, March, 2002.
- [20] S. Verdu. *Multuser Detection*, Cambridge Univ. Press, 1998.
- [21] H. Gamal, E. Geraniotis. “Iterative multuser detection for coded CDMA signals in AWGN and fading channels”. *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 1, pp. 30–41, Jan. 2000.
- [22] X. Wang, H. Poor. “Iterative (Turbo) soft interference cancellation and decoding for coded CDMA”. *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 1046–1061, July 1999.
- [23] J. Hagenauer, E. Offer, L. Papke. “Iterative decoding of binary block and convolutional codes”. *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, March 1996.

- [24] P. Robertson, E. Vilebrun, P. Hoeher. “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain”. *Proc. IEEE International Conference on Communications*, pp. 1009–1013, 1995.
- [25] R. Morrow, J. Lehnert. “Bit-to-bit error dependence in slotted DS/SSMA packet systems with random signature sequences”. *IEEE Transactions on Communications*, vol. 37, no. 10, pp. 1052–1061, Oct. 1989.
- [26] A. Naguib, A. Paulraj. “Performance of Wireless CDMA with M-ary orthogonal modulation and cell site antenna arrays”. *IEEE Journal on Selected Areas Communications*, vol. 14, no. 9, pp. 1770–1783, Dec. 1996.
- [27] L. Jalloul, J. Holtzman. “Performance analysis of DS/CDMA with noncoherent M-ary orthogonal modulation in multipath fading channels”. *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, pp. 862–870, June 1994.
- [28] J. Thompson, P. Grant, B. Mulgrew. “Algorithms for coherent diversity combining of M-ary orthogonal signals”. *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1886–1899, Nov. 1999.
- [29] A. Kaul, B. Woerner. “Analytic limits on performance of adaptive multistage interference cancellation for CDMA”. *Electronics Letters*, vol. 30, no. 25, pp. 2093–2095, Dec. 1994.
- [30] K. Ko, M. Joo, H. Lee, D. Hong. “Performance analysis for multistage interference cancellers in asynchronous DS-CDMA systems”. *IEEE Communications Letters*, vol. 6, no. 12, pp. 544–546, Dec. 2002.
- [31] R. Buehrer, B. Woerner. “Analysis of adaptive multistage interference cancellation for CDMA using an improved Gaussian approximation”. *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1308–1321, Oct. 1996.
- [32] G. Xue, J. Weng, T. Le-Ngoc, S. Tahar. “An analytical model for performance evaluation of parallel interference canceller in CDMA systems”. *IEEE Communications Letters*, vol. 4, no. 6, pp. 184–186, June 2000.
- [33] D. Torrieri. “Performance of direct-sequence systems with long pseudonoise sequences”. *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 4, pp. 770–781, May 1992.
- [34] J. Holtzman. “A simple, accurate method to calculate spread-spectrum multiple-access error probabilities”. *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 461–464, March 1992.

- [35] M. Clark, L. Greenstein, W. Kennedy, M. Shafi. “Matched filter performance bounds for diversity combining receivers in digital mobile radio”. *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 356–362, Nov. 1992.
- [36] P. Shan, T. Rappaport. “Parallel interference cancellation (PIC) improvements for CDMA multiuser receivers using partial cancellation of MAI estimates”. *Proc. IEEE Global Telecommunications Conference*, vol. 6, pp. 3282–3287, 1998.
- [37] S. Bensley, B. Aazhang. “Subspace-based channel estimation for code division multiple access communication systems”. *IEEE Transactions on Communications*, vol. 44, no. 8, pp. 1009–1020, Aug. 1996.
- [38] C. Sengupta, J. Cavallaro, B. Aazhang. “Maximum likelihood multipath channel parameter estimation in CDMA systems using antenna arrays”. *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 1406–1410, Sept. 1998.
- [39] A. Weiss, B. Friedlander. “Synchronous DS-CDMA downlink with frequency selective fading”. *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 158–167, Jan. 1999.
- [40] J. Kim, S. Yoon, S. Kang, C. Kang. “A channel estimation technique in a DS/CDMA system with noncoherent M-ary orthogonal modulation in multipath fading channels”. *Proc. Vehicular Technology Conference*, vol. 3, pp. 2383–2387, May 1998.
- [41] S. Kay. *Fundamentals of statistical signal processing*, Prentice Hall, NJ, 1998.
- [42] M. Tsatsanis, G. Giannakis, G. Zhou. “Estimation and equalization of fading channels with random coefficients”. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1093–1096, May 1996.
- [43] R. Iltis. “Joint estimation of PN code delay and multipath using the extended Kalman filter”. *IEEE Transactions on Communications*, vol. 38, no. 10, pp. 1677–1685, Oct. 1990.
- [44] R. Iltis, A. Fuxjaeger. “A digital DS spread-spectrum receiver with joint channel and doppler shift estimation”. *IEEE Transactions on Communications*, vol. 39, no. 8, pp. 1255–1267, August 1991.
- [45] K. Shanmugan, M. Sanchez, L. Haro, M. Calvo. “Channel estimation for 3G wideband CDMA systems using the Kalman filtering algorithm”. *IEEE International Conference on Personal Wireless Communications*, pp. 95–97, Dec. 2000.

- [46] P. Xiao, E. Ström. “Performance of iterative DS-CDMA M-ary demodulation in the presence of synchronization errors”. *Proc. Vehicular Technology Conference*, pp. 1703–1707, May, 2001.
- [47] S. Parkvall, E. Ström, B. Ottersten. “The impact of timing errors on the performance of linear DS-CDMA receivers”. *IEEE Journal on Selected Areas on Communications*, vol. 14, pp. 1660–1668, Oct. 1996.
- [48] P. Orten, T. Ottosson. “Robustness of DS-CDMA multiuser detectors”. *Proc. IEEE Communication Theory Mini-Conference*, pp. 144–148, Nov. 1997.
- [49] P. Xiao, E. Ström. “Synchronization algorithms for iteratively demodulated M-ary DS-CDMA systems”. *IEEE Global Telecommunications Conference*, pp. 1371–1375, Nov. 2001.
- [50] E. Ström, S. Parkvall, S. Miller, B. Ottersten. “Propagation delay estimation in asynchronous direct-sequence code-division multiple access systems”. *IEEE Transactions on Communications*, vol. 44, pp. 84–93, Jan. 1996.
- [51] E. Ström, F. Malmsten. “A maximum likelihood approach for estimating DS-CDMA multipath fading channels”. *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 132–140, Jan. 2000.
- [52] D. Zheng, J. Li, S. Miller, E. Ström. “An efficient code-timing estimator for DS-CDMA signals”. *IEEE Transactions on Signal Processing*, vol. 45, pp. 82–89, Jan. 1997.
- [53] S. Bensley, B. Aazhang. “Subspace-based estimation of multipath channel parameters for CDMA communication systems”. *Proc. IEEE Global Telecommunications Conference*, pp. 154–157, 1994.
- [54] J. Joutsensalo, J. Lilleberg, A. Hottinen, J. Karkunen. “A hierarchic maximum likelihood method for delay estimation in CDMA”. *Proc. IEEE Vehicular Technology Conference*, vol. 1, pp. 188–192, 1996.
- [55] M. Landolsi, W. Stark. “DS-CDMA chip waveform design for optimal power bandwidth performance”. *Proc. IEEE Symposium on Personal, Indoor and Mobile Radio Communication*, pp. 706–709, Toronto, Canada, Sept. 1995.
- [56] P. Dallas, F. Pavlidou. “Innovative chip waveforms in microcellular DS/CDMA packet mobile radio”. *IEEE Transactions on Communications*, vol. 44, pp. 1413–1416, Nov. 1996.
- [57] 3G TS 25.104. *3rd generation partnership project: technical specification group radio access networks; BS radio transmission and reception (FDD)*, V4.4.0, March 2002.

- [58] P. Orten, A. Svensson. "Some results on pulse shaping in DS-CDMA systems". *Technical report IT020/1997*, Chalmers University of Technology, Sept. 1997.
- [59] J. Hagenauer. "Forward error correcting for CDMA systems". *Proc. ISSSTA '96*, pp. 566–669, Sept. 1996.
- [60] M. Valenti, B. Woerner. "Iterative multiuser detection for convolutionally coded asynchronous DS-CDMA". *Proc. IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Communications*, pp. 213–217, Sept. 1998.
- [61] M. Moher. "An iterative multiuser decoder for near-capacity communications". *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 870–880, July 1998.
- [62] T. Giallorenzi, S. Wilson. "Suboptimum multiuser receivers for convolutionally coded asynchronous DS-CDMA systems". *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1183–1196, Sept. 1996.
- [63] M. Howlader, B. Woerner. "Iterative interference cancellation and decoding for DS-CDMA". *Proc. Vehicular Technology Conference - Fall*, vol. 3, pp. 1815–1819, 1999.
- [64] M. Valenti, B. Woerner. "Iterative multiuser detection for convolutionally coded asynchronous DS-CDMA". *Proc. IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Communications*, pp. 213–217, Sept. 1998.
- [65] A. Viterbi. "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes". *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 260–264, Feb. 1998.
- [66] L. Bahl, J. Cocke, F. Jelinek, J. Raviv. "Optimal decoding of linear codes for minimizing symbol error rate". *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, 1974.
- [67] P. Robertson, P. Hoeher, E. Villebrum. "Optimal and sub-optimal maximum a posteriori algorithms suitable for Turbo decoding". *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119–125, March-April, 1997.
- [68] M. Valenti. "Turbo codes and iterative processing". *IEEE New Zealand Wireless Commun. Symp.*, Nov. 1998, proceedings pages unnumbered.

- [69] J. Hagenauer. “The Turbo principle: tutorial introduction and state of the art”. *Proc. International Symposium on Turbo Codes*, pp. 1–11, Sept. 1997.
- [70] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara. “Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding”. *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 909–926, May 1998.
- [71] R. Herzog, J. Hagenauer, A. Schmidbauer. “Soft-in/soft-out Hadamard despreaders for iterative decoding in the IS-95(A) system”. *Proc. Vehicular Technology Conference*, vol. 2, pp. 1219–1222, May 1997.
- [72] L. Boulet, H. Leib. “Decoding of the IS-95 uplink code”. *IEEE Communication Letters*, vol. 5, no. 5, pp. 206–208, May 2001.
- [73] L. Papke, P. Robertson. “Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme”. *Proc. IEEE International Conference on Communication*, pp. 102–106, 1996.