Near Maximum Likelihood Multiuser Receivers for Direct Sequence Code Division Multiple Access

Hak Keong Sim

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
November 2000

Abstract

Wideband wireless access based on direct-sequence code-division multiple access (DS-CDMA) has been adopted for third-generation mobile communications systems. Hence, DS-CDMA downlink communications systems form the platform for the work in this thesis. The principles of the spread spectrum concept and DS-CDMA technology are first outlined, including a description of the system model and the conventional receiver. The two classes of codes used in this system, namely spreading codes and forward error correction codes (including Turbo codes), are discussed.

Due to the fact that practical communications channels are non-ideal, the performance of an individual user is interference limited. As a result, the capacity of the system is greatly restricted. Fortunately, multiuser detection is a scheme that can effectively counteract this multiple access interference. However, the optimum multiuser detection scheme is far too computationally intensive for practical use. Hence, the fundamental interest here is to retain the advantages of multiuser detection and simplify its implementation.

The objective of the thesis is to investigate the optimum multiuser receiver, regarded on a chip level sampling basis. The aim is to reduce the complexity of the optimum receiver to a practical and implementable level while retaining its good performance. The thesis first reviews various existing multiuser receivers. The chip-based maximum likelihood sequence estimation (CBMLSE) detector is formulated and implemented. However, the number of states in the state-transition trellis is still exponential in the number of users. Complexity cannot be reduced substantially without changing the structure of the trellis.

A new detector is proposed which folds up the original state-transition trellis such that the number of states involved is greatly reduced. The performance is close to that of the CBMLSE. The folded trellis detector (FTD) can also be used as a preselection stage for the CBMLSE. The FTD selects with high accuracy the few symbol vectors that are more likely to be transmitted. The CBMLSE is then used to determine the most likely symbol vector out of the small subset of vectors. The performance of this scheme is as good as the CBMLSE. The FTD is also applied in an iterative multiuser receiver that exploits the powerful iterative algorithm of Turbo codes.

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Department of Electronics and Electrical Engineering at The University of Edinburgh. The software program used to perform the simulations was written by myself with the following exceptions:

- The routines used to generate Gaussian distributed noise and uniform distributed deviates were obtained from *Numerical recipes in C* [1].
- The routines that perform convolutional encoding and decoding (using the Viterbi algorithm) were obtained from the World Wide Web, URL *http://people.qualcomm.com/karn/code/fec/*.

Hak Keong Sim
November 2000

Acknowledgements

I wish to express my gratitude to my supervisors Dr. David G. M Cruickshank and Prof. Peter M. Grant for their comments and guidance throughout the course of the work. I am grateful to Dr. David Cruickshank for his patience reading and correcting this manuscript. It has been a pleasure working with him and I sincerely hope that we can work together in the future.

I would like to thank all persons working in the Signals and Systems Group both in the past and present, especially Harald Haas, Mark Cowper, Ian Band, Rudolf Tanner, Sarat Patra and George Taylor, for their encouragement, support and assistance in one way or another. I am also grateful to all the Singapore students at the University of Edinburgh and all my housemates in David Horn House both in the past and present for giving me a fulfilling experience in Scotland.

The financial support provided by the National Science and Technology Board of Singapore and Infineon Technologies (Asia Pacific) Pte Ltd made this work possible and is thus gratefully acknowledged. I am grateful to Mr. Werner Luschnig, Mr. John Seow, Dr. Kwa and Mr. Babu Jain for making this scholarship possible.

I wish to express my deepest thanks to my parents for their help, support, and love they have provided for me throughout my life, to my sisters, Shereen and Jeslyn, for taking care of our parents during my absence, to my wife, Jessica, for her love, support and understanding she has shown to me, and to my parents-in-law for taking care of Jessica and our lovely daughter, Georgyann. I delicate this thesis to Jessica, who spent many lonely moments due to my work, as well as to Georgyann who brings joy to our lives.

To my family

Contents

List of figures

List of tables

Acronyms and abbreviations

3GPP	third generation partnership project
ACI	adjacent channel interference
AMPS	Advanced Mobile Phone Services
APP	*a posteriori* probabilities
AWGN	additive white Gaussian noise
BCJR	Bahl *et al.* [2]
BER	bit error rate
BPSK	binary phase shift keying
CBMLSE	chip-based MLSE
CCI	co-channel interference
CCL	cross-correlation level
CDMA	code division multiple access
CMFD	conventional matched filter detector
DD	decorrelating detector
DDFD	decorrelating decision-feedback detector
DFD	decision-feedback detector
DS	direct sequence
DSSS	direct sequence spread spectrum
FDD	frequency division duplex
FDMA	frequency division multiple access
FEC	forward error correction
FH	frequency-hopping
FIR	finite impulse response
FLOP	floating-point operations
FTD	folded trellis detector
GSM	Global System for Mobile communications
IC	interference canceller
IDDFD	improved DDFD
IIR	infinite impulse response

IMT-2000	International Mobile Telecommunications 2000
ISI	intersymbol interference
LMCBMLSE	linear metric CBMLSE
LMFTD	linear metric FTD
LMS	least mean square
MAI	multiple access interference
MAP	maximum *a posteriori*
MC	multicarrier
MLSE	maximum likelihood sequence estimator
MMSE	minimum mean-square error
MSD	multistage detector
MTSO	Mobile Telephone Switching Office
MUD	multiuser detection
NMT	Nordic Mobile Telephone
NSC	non-systematic convolutional code
NTT	Nippon Telephone and Telegraph
OFDM	orthogonal frequency-division multiplexing
PDC	Personal Digital Cellular
PDMA	polarisation-division multiple access
PN	pseudo-noise
PSTN	public switched telephone network
RBF	radial basis function
RSC	recursive systematic convolutional code
SB	sequence-based
SDMA	space-division multiple access
SISO	soft-input soft-output
SNR	signal-to-noise ratio
SSB	single-symbol based
TACS	Total Access Communications System
TDMA	time division multiple access
TH	time-hopping
VA	Viterbi algorithm

List of Symbols

$B_k(z)$	z-transform of the entire modulated symbols of the k-th user
$\mathbf{B}(z)$	z-transform of the entire modulated symbol sequence vector
E_b	energy per data bit
E_b/N_0	signal to Gaussian noise ratio
F	number of data bits in a frame
$\mathbf{F}$	Cholesky decomposition of Φ
G	processing gain of spreading code
$H(z)$	z-transform of the channel impulse response
$\mathbf{I}_G$	$G \times G$ identity matrix
J	total number of symbols transmitted including trailing zeros
K	number of active users in the system
L	number of multipath components in the channel
$L_1(d_t)$	log APP ratio of Turbo decoder 1 for data bit d_t
$L_2(d_t)$	log APP ratio of Turbo decoder 2 for data bit d_t
$L_{e1,t}$	t-th extrinsic information of Turbo decoder 1
$L_{e2,t}$	t-th extrinsic information of Turbo decoder 2
M	number of possible states in the trellis
$L_{s,t}$	t-th systematic information
$M(n, V_{n,s})$	Euclidean metric for each transition path entering state $V_{n,s}$ of group n in the AWGN folded trellis
M_l	l-th metric sum obtained at the end of the complete trellis
$M_m(n, V_{n,s}, V_{n+1,s})$	Euclidean metric for each transition path from state $V_{n,s}$ to state $V_{n+1,s}$ in a multipath folded trellis
$M_{\text{min}}^{\text{CBMLSE}}$	minimum metric sum of the CBMLSE
$M_{\text{min}}^{\text{FTD}}$	minimum metric sum of the FTD
$M_{j,t}^{P}$	path metric for state j at time t
$M_{i,j,t}^{T}$	transition metric for the transition from state i to state j at time t
N_c	number of paths in the complete trellis
N_f	number of paths in the folded trellis

$N_f(z)$	z-transform of the AWGN sequence at the f-th symbol interval
N_m	number of metric sums within the reliability sphere
$N(z)$	z-transform of the AWGN sequence
$\mathbf{P}$	diagonal matrix representing the received power of the users
$P_j(c)$	probability of making a right decision if $X^j(z)$ is selected
$P_j(e)$	probability of making a wrong decision if $X^j(z)$ is selected
P_k	transmitted power of the k-th user
$\mathbf{Q}$	matrix representation of the spreading codes
R	rate of the FEC code
$R_k(z)$	z-transform of the overall channel response of the k-th user due to multipath effect and spectrum spreading
$R_k^{RAKE}(z)$	transfer function of the k-th user's RAKE receiver
$\mathbf{R}(z)$	z-transform of the vector comprising of all the users' overall channel response
$\mathbf{R}(z^{-1})$	transfer function vector of the RAKE matched filter bank
$S_k(z)$	z-transform of $s_k(t)$
T	one symbol duration
$\mathbf{T}$	linear transformation matrix
T_c	one chip duration
$U_k(z)$	z-transform of the output of the k-th user's RAKE receiver
$\mathbf{U}(z)$	z-transform of the output vector of the RAKE matched filter bank
$V_f(z)$	z-transform of the transmitted signal in the f-th symbol interval
$V_{n,s}$	s-th state of group n in the folded trellis
$V(z)$	z-transform of the entire transmitted signal in a frame
$\mathbf{W}$	Wiener filter matrix of the MMSE detector
W_i	weight associated with $\mathbf{c}_i$
$X_f(z)$	z-transform of the noise-free received signal in the f-th symbol interval
$X^l(z)$	l-th possible noise-free received signal
$X(z)$	z-transform of the noise-free received signal
$Y_f(z)$	z-transform of the noisy received signal at the f-th symbol interval
$\mathbf{Y}_{f'}^{f}(z)$	a vector formed from $Y_{f'}(z)$ to $Y_f(z)$
$\mathbf{Y}_t$	noisy received codeword at time t

$\mathbf{Y}_{t'}^{t}$	vector form of the noisy received sequence of codewords from time t' to time t
$Y(z)$	z-transform of the noisy received signal
$\mathbf{b}$	input data symbol vector in a symbol duration of a synchronous system
$\mathbf{b}_f$	f-th input symbol vector
$\mathbf{b}_{f,j}$	j-th possible symbol vector in the f-th interval
b_k	encoded symbol of the k-th user of a synchronous system
$b_{k,f}$	f-th encoded symbol of the k-th user
$b_k(t)$	transmitted symbols of the k-th user
$\mathbf{c}_i$	i-th centre of the RBF detector
$c_{i,k}$	i-th chip value of the k-th user's spreading code
$d_k(t)$	data bits of the k-th user
d_t	input information bit at time t
h_l	path coefficient of the l-th multipath
$h(t)$	channel impulse response
h_{α}^{t}	normalising constant of $\alpha_t(m)$
h_{β}^{t}	normalising constant of $\beta_t(m)$
n	number of encoded bits per uncoded data bit
$\mathbf{n}$	spreaded noise vector in a symbol duration of a synchronous system
n_M	maximum number of metric sums allowed
n_c	number of chips per state in the folded trellis
n_m	number of metric sums selected by the FTD first stage
n_s	number of symbol combinations selected by the FTD first stage
$n(t)$	AWGN
$p(t)$	rectangular chip waveform
r	radius of the reliability sphere
$\mathbf{r}_i$	overall channel response at the i-th chip
$s_k(t)$	spreading code of the k-th user
$\mathbf{u}$	sampled output vector of the matched filter bank of a synchronous system
$\mathbf{v}$	vector representation of a path through the folded trellis
$v_i(\mathbf{b}_f)$	discrete signal of the i-th chip due to the transmission of $\mathbf{b}_f$

$v_{i,d}$	d-th discrete chip value of the i-th chip
$v(t)$	transmitted signal
$w_{1,t}$	code bit generated by the first Turbo encoder at time t
$w_{2,t}$	code bit generated by the second Turbo encoder at time t
w_f	f-th encoded bit
$\mathbf{w}_t(m^-, m)$	output codeword associated with the transition from state $S_{t-1} = m^-$ to state $S_t = m$ at time t
$\mathbf{x}$	noise-free received vector in a symbol interval of a synchronous system
$x_i(v_{i,d})$	noise-free received signal of the i-th chip due to the transmission of $v_{i,d}$
$x_{i,f}(\mathbf{b}_f)$	noiseless received signal of the i-th chip in the f-th symbol interval due to the transmission of $\mathbf{b}_f$
x_i^l	coefficient of z^{-i} in $X^l(z)$
$x(t)$	noise-free received signal
$\mathbf{y}$	noisy received vector in a symbol interval of a synchronous system
y_f	f-th noisy received encoded symbol
$\mathbf{y}_i$	sequence formed from the i-th noise-corrupted received symbol in each codeword
$y_{i,f}$	coefficient of z^{-i} in $Y_f(z)$
$y(t)$	noisy received signal
z^{-1}	delay of one chip
Δ_j	smallest $\Delta_{l,j}$ if $M_j(z)$ is smallest
$\Delta_{l,j}$	scaled difference between the metric sums, $M_l(z)$ and $M_j(z)$
Λ	$\mathbf{P}^2$
Φ	correlation matrix of a synchronous system
$\Phi^R(z)$	z-transform of the correlations between the users subjected to the effect of the channel
$\Phi^R_{k,j}(z)$	cross-correlation between $R_k(z^{-1})$ and $R_j(z)$
$\Phi^S(z)$	z-transform of the correlations between the users without the channel effect
$\Phi^S_{k,j}(z)$	cross-correlation between $S_k(z^{-1})$ and $S_j(z)$
$\alpha_t(m)$	forward state probability of state m at instant t

$\alpha_t^k(m)$	$\alpha_t(m)$ of the k-th user
$\beta_t(m)$	reverse state probability of state m at instant t
$\beta_t^k(m)$	$\beta_t(m)$ of the k-th user
$\delta(t)$	Dirac function
$\gamma_t(m^-, m)$	transition probability from state $S_{t-1} = m^-$ to state $S_t = m$ at instant t
$\gamma_t^k(m^-, m)$	$\gamma_t(m^-, m)$ of the k-th user
κ	constraint length of the convolutional code
$\lambda_f(\mathbf{b}_{f-1}, \mathbf{b}_f)$	transition metric of the MLSE detector
$\lambda_t(m)$	joint probability of state $S_t = m$ and $\mathbf{Y}_1^\tau$ at instant t
σ^2	variance of AWGN samples ($= N_0/2$)
$\sigma_t(m^-, m)$	joint probability of state $S_{t-1} = m^-$, state $S_t = m$ and $\mathbf{Y}_1^\tau$ at instant t
$\sigma_t^k(m^-, m)$	$\sigma_t(m^-, m)$ of the k-th user
τ	number of transmitted data bits including the $\kappa - 1$ trailing zeros
$\|\cdot\|$	Euclidean norm

Chapter 1
Introduction

In a world that appears ever-shrinking, the desire for freedom is ever-growing. Nowhere is this demand more evident than in telephony, one of the greatest engineering accomplishments of the last century. Fixed telephone service at home has paved the way for portable phones. People are discovering the advantages of being able to communicate on the move. This does not come as a surprise since wireless communication has become a reliable and affordable form of interaction, anywhere and anytime. However, in this new world of mobile users, the communications equipment and facilities are expected to meet the escalatingly stringent requirements of the consumers [3]. Over the last few decades, we have seen the evolution of the mobile phone from its basic function of voice communications to advanced data services such as email, fax, Internet access, even full-motion digital video or interactive multimedia communications in the future.

The next generation of mobile communications entails:

- Enhanced information and multimedia capabilities
- Subscriber flexibility and control
- Increased capacity to meet a more demanding environment
- Robust voice systems to operate in more dynamic environment
- Interoperability between various systems
- Ease of operations and maintenance
- Preservation of significant investment in present systems

In order to meet the needs and demands of the customers, new technologies have to be deployed since traditional methods of communications are nearing their technical or financial limits. This has sparked the search for the ultimate last mile solution and with it a hot debate on the “right technology.” Industry pundits looked to the hope of a unified worldwide standard that would facilitate global roaming anytime, anywhere. Work in this direction [4–7], which has been ongoing for a few years, is known as the third generation mobile standard International Mobile Telecommunications 2000 (IMT-2000). The IMT-2000 global standard provides the inherent flexibility necessary for today's operators to seamlessly evolve their networks towards the needs of the information age. To enforce global harmonisation of the standards, the third generation partnership project (3GPP) has been set up to co-operate for the production of technical specifications for IMT-2000 based on the radio access technologies the organisational partners support. In May 1999, the code division multiple access (CDMA) radio access technology harmonisation agreement was reached, and in November 1999 CDMA emerged as the winning radio access with the endorsement of the CDMA standards for IMT-2000.

The following section gives a general overview of cellular communications. The various multiple access techniques used by the subscribers to access the network are briefly discussed in Section 1.2. The shortcomings of each technique are also stated. Section 1.3 addresses the motivation for the work carried out in this thesis and gives the organisation of the thesis.

1.1 Cellular communications

The traditional approach to mobile radio involved setting up a high-power transmitter located on top of the highest point in the coverage area. A line of sight is needed between the mobile telephones and the base station for adequate radio coverage [8]. This implies that the few available radio channels were locked up over a large area by a small number of users.

1.1.1 Cellular concept

A different approach is to handle the coverage problem using the cellular concept instead of the broadcast method. This new method uses a large number of low-power transmitters designed to serve only a small area. Each small area is called a cell. By reducing the total coverage area into small cells, it became possible to reuse the same frequencies in different cells.

In this cellular concept, frequencies allocated to the service are reused in a regular pattern of cells, each covered by one radio station or base station. In mobile network these cells are theoretically hexagonal, but in reality the cell coverage is an irregular shape. To ensure that the mutual interference between users remains below a harmful level, adjacent cells use different frequencies[1]. Figure 1.1 shows that a set of three different frequencies is used for each cluster of three adjacent cells. Hence, this system has a frequency reuse factor of 1/3. Cluster patterns and the corresponding frequencies are reused in a regular pattern over the entire service area.

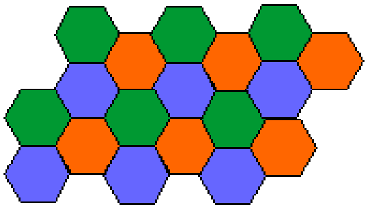

Figure 1.1: *A cluster of cells using three different frequencies*

However, the problem with small cells was that not all mobile calls would now be completed within a single cell. A ‘handover’ or ‘handoff’ to another base station is required to ensure sufficient quality of reception, including acceptable interference power levels. Handover is the process used to allow a call in progress to continue as the mobile terminal moves between cells. Handovers can be classified as soft or hard. Soft handover occurs when the mobile terminal communication is switched to the target base station without interrupting communications with the current serving base station. In a soft handover, the mobile terminal communicates with two or more base stations simultaneously, with the signals received from the base stations coherently combined. On the other hand, hard handover occurs when the communication to the mobile terminal is passed between disjointed radio systems, different frequency assignments, or different air interface characteristics or technologies. This form of handover is a “break-before-make” process at the air interface. Whether a soft or hard handover can occur is dependent on the multiple access technology. (See Section 1.2.3).

1.1.2 Mobile cellular network

Figure 1.2 depicts a simple scheme of relationship between mobile terminals, base stations and Mobile Telephone Switching Office (MTSO) in a mobile cellular terrestrial network. Mobile users communicate with one another via the public switched telephone network (PSTN). The MTSO acts as the interface between the PSTN and the cellular network. Each MTSO controls a group of base stations, to which the mobile terminals are linked. The radio transmission from the base station to the mobile is called the downlink and the reverse transmission path is hence

[1] Not always the case in CDMA and is discussed in Section 1.2.3

the uplink.

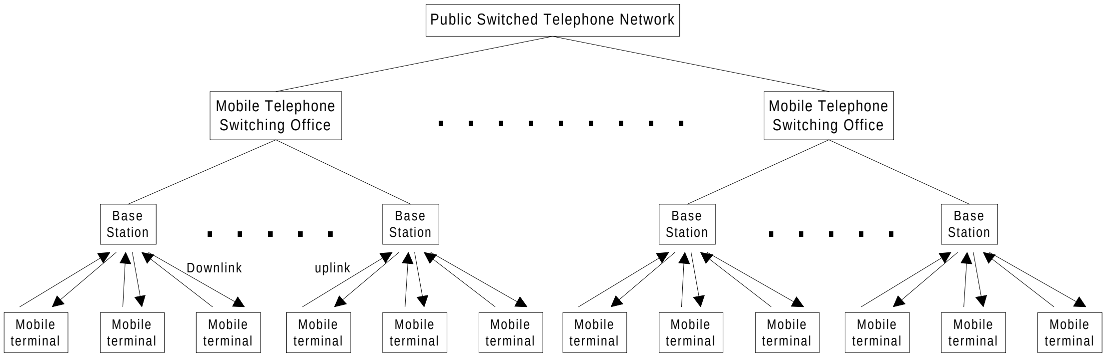
Public Switched Telephone Network
Mobile Telephone Switching Office
Mobile Telephone Switching Office
Base Station
Base Station
Base Station
Base Station
Downlink
uplink
Mobile terminal

Figure 1.2: *Basic mobile cellular network*

In the downlink, all the users' individual signals in a cell originate from one source, the base station. Hence, the signals received at each mobile terminal are synchronous. On the other hand, the uplink is asynchronous, since the signals from different mobile terminals arrive at the base station with different delays. Moreover, all these signals travel through different paths and experience different forms and degrees of impairments. The current status is that the capacity of the uplink is approximately twice that of the downlink's due to array processing and multiuser detection at the base station where higher complexity can be tolerated[2]. However, improving only the capacity of the uplink does not obviously improve the overall capacity, the more limiting link will determine the overall capacity. Hence, only the downlink is considered in this work as attention should now be focused on the downlink as well.

Impairments in cellular radio communications include:

- Additive white Gaussian noise
- Inter-symbol interference
- Co-channel interference
- Adjacent channel interference
- Fading - flat and selective

Additive white Gaussian noise (AWGN) is the inevitable thermal noise in the receiver and is assumed to follow a Gaussian distribution. The mobile radio channel is characterised by 'mul-

[2] Cost, size and weight are critical concerns in mobiles

tipath reception': The signal offered to the receiver contains not only a direct line-of-sight radio wave, but also a large number of reflected radio waves [9]. These reflected waves when added to the direct wave, cause significant distortion to the received signal. This form of interference is known as intersymbol interference (ISI). A wireless network has to be designed in such a way that the adverse effect of these reflections is minimised. ISI is usually combatted by using an equaliser in the receiver. Co-channel interference (CCI) is caused by the interference from neighbouring cells due to frequency reuse. This is the interference[3] arriving at a desired cell from the cell using the same frequency channel. Two channels with different frequencies can also cause interference if they are next to each other in the frequency spectrum. This form of interference is known as adjacent channel interference (ACI) which is the result of signals from one channel 'spilling' into the adjacent channel due to imperfect filters. To avoid this cross-talk between channels, guard bands are inserted between channels. In addition to this impairments, the received signal exhibits fluctuations in signal level called fading. Two forms of fading are flat and frequency-selective fadings. When ISI is low and the received signal cannot be "resolved" into its multiple paths components, flat fading is experienced. On the other hand, severe ISI with several time-shifted and scaled versions of the transmitted signal arriving at the receiver, results in frequency-selective fading.

Hence, sophisticated signal processing techniques are needed in the receiver to combat these impairments and designing receivers for the mobile communications systems has always been a challenge.

1.2 Multiple access technologies

Multiple access refers to a technique to share a common communications channel between multiple users. A multiple access platform is needed for multiple users to exist harmoniously in a network. The freedoms in use when designing multiuser communications systems include space, time and frequency. In the space domain, users can be separated by making their distance large enough. In wireless communications, an example is to separate transmitters geographically to have enough distance attenuating the signals so that they do not interfere significantly. More advanced techniques include polarisation-division multiple access (PDMA) and space-division multiple access (SDMA). In PDMA, two users can be separated by using

[3] Multipath is not always detrimental and is even beneficial in CDMA with the use of RAKE receiver

electromagnetic waves with different polarisation. In SDMA, sectorised antennas are usually applied to separate users at the same frequency [10].

In time-frequency domain multiple-access, each user's transmitted data signal is modulated by a signature waveform. The receiver can demodulate each user's data, if the signature waveforms of the users are different enough. Various signature waveform designs result in different multiple access technologies, namely, frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA).

1.2.1 Frequency division multiple access

Frequency division multiple access (FDMA) is the oldest multiple access technique. The signature waveform of each user in FDMA occupies its own frequency band and by simply filtering the receiver can separate the users' signals. Although FDMA is applicable to both analog and digital modulations, the maximum bit rate per channel is fixed and small, inhibiting the flexibility in bit-rate capability which is essential in the future communications services. Making the bit rate higher requires more frequency channels to be allocated for a user. This implies a need for several bandpass filters. Furthermore, FDMA does not use the frequency spectrum efficiently because each frequency channel requires guard bands to minimise cross-talk between channels and the channel is occupied even if no information is transmitted.

The first generation analog cellular FDMA systems [11] include the North America's Advanced Mobile Phone Services (AMPS), United Kingdom's Total Access Communications System (TACS), Scandinavia's Nordic Mobile Telephone (NMT), Germany's C-450 and Japan's Nippon Telephone and Telegraph (NTT).

1.2.2 Time division multiple access

The introduction of digital modulation made it possible to deploy digital mobile communications systems which are known as the second generation of mobile communications. A narrowband second generation system known as time division multiple access (TDMA) appeared. In TDMA, the signature waveform of each user is limited to a predetermined time interval. Relatively simple to implement, TDMA permits very flexible bit rates. By using multiple transmission intervals, a wide range of user bit rates is supported. However, the transmission of all users must be exactly synchronised to each other and substantial amount of signal processing is needed for synchronisation. As there is no frequency guard band required between channels, TDMA utilises bandwidth more efficiently. Nevertheless, ACI is still present in this system and exists in the time domain instead of the frequency spectrum. Sufficient guard time between time intervals is needed to accommodate timing discrepancies and delay spread. Another disadvantage is that high peak power in the transmit mode of the handset shortens the battery life.

The second generation digital cellular systems [11] based on TDMA are the pan-European Global System for Mobile communications (GSM) [12], IS-54 in the United States and the Personal Digital Cellular (PDC) in Japan.

1.2.3 Code division multiple access

The invention of spread-spectrum techniques for communications systems with anti-jamming and low probability of undesired interception capabilities lead to the idea of CDMA. There are numerous ways of implementing CDMA, four methods are frequency-hopping (FH), time-hopping (TH), direct sequence (DS) and multicarrier (MC). Hybrid CDMA systems based on the combination of some of the techniques are also possible. In FH-CDMA, users' signature waveforms are centred on generating different carrier frequencies at different time intervals. The signal hops from a frequency to another according to a pseudo-random spreading sequence. In TH-CDMA, bursts of signal are initiated at pseudo-random times. In DS-CDMA systems, each user's signature waveform is continuous in the time domain and has a relatively flat spectrum. Hence, all signature waveforms occupy the entire frequency band allocated for transmission at all times and the users are separated neither in time nor in frequency domains. The data of users can be separated in the receivers because the signature waveforms of DS-CDMA are formed by spreading sequences which are unique to each user. MC-CDMA is based on a combination of code division and orthogonal frequency-division multiplexing (OFDM). More details can be found in [13].

Traditional FDMA and TDMA are designed to be orthogonal in the sense that the waveforms are mutually orthogonal. DS-CDMA, on the other hand, can be either orthogonal or non-orthogonal depending on the orthogonality of the spreading sequences [14]. Provided that there is no time delay and the transmission does not cause time dispersion, the received signals of the users appear as orthogonal if orthogonal signature waveforms are used. The spreading sequences may also be designed to be non-orthogonal. Non-orthogonality is attractive in the sense that there is no hard limit on the number of users since the number of codes is unconstrained.

The primary advantage of CDMA [15, 16] is its ability to tolerate a fair amount of interfering signals compared to FDMA and TDMA. As a result of the interference tolerance of CDMA, frequency planning is simplified. Moreover, flexibility in system design and deployment are significantly improved since interference with others is not a problem and it is less susceptible to ACI. On the other hand, sophisticated filtering and guard-band protection is needed with FDMA and TDMA to ensure no ACI with similar assumptions. The CDMA waveform occupies a significantly larger bandwidth than FDMA or TDMA waveforms, thus CDMA is more likely to undergo frequency-selective fading. However, the effect of this fading is less detrimental in CDMA because the energy is spread over a much larger spectrum and fading at certain frequencies has little effect on the overall spectrum. For a DS-CDMA system, capacity improvement is possible with a frequency reuse factor of one, as compared to the other two technologies which have reuse factors of less than one[4]. Since adjacent cells use the same frequency band, soft handover is possible and users no longer experience break in communications when transiting between cells. With a reuse factor of one, FH-CDMA cannot avoid frequency hits between users at the same frequencies in adjacent cells causing a severe performance degradation. Furthermore, coherent demodulation is not practical in FH systems, which causes a performance penalty in comparison to DS systems with coherent demodulation. Another advantage of DS-CDMA is that the transmission bandwidth exceeds the coherence bandwidth and hence provides an inherent delay diversity receiver. This implies that the received signal after despreading can be resolved into multiple signals with different time delays, and subsequently recovered and combined into an enhanced signal. However, DS-CDMA has its limitations and additional constraints such as stringent power control is required.

Not only is DS-CDMA used in the present second generation systems, for example IS-95 [18] initially in the United States and the later cdmaOne in other parts of the world as well, it is also the chosen technology for IMT-2000 in the future. Deployment is scheduled to commence in 2001 and 2002. Hence, DS-CDMA will form the main area of study of this work.

[4] Further capacity gains can also result from antenna technology advancement by using directional antennas [17], which allow the microcell area to be divided into sectors

1.3 Aim and organisation of the thesis

With the standardisation of the third generation mobile communications systems, CDMA is going to have a direct impact in our life in the near future. One difference between the present CDMA systems (generally known as cdmaOne) and those in the future is that advanced receiver concepts have been incorporated into the new standards. This new receiver approach, known as multiuser detection, is able to increase capacity of the wireless network by exploiting powerful digital signal processing. In spite of the major research effort invested in multiuser demodulation techniques, several practical as well as theoretical open issues still exist in the field of multiuser receivers. The objective of the work presented is to develop practical multiuser demodulation algorithm with reasonable performance and implementable complexity. More specifically, the work aims to simplify the prohibitive complexity of the optimum multiuser receiver.

The organisation of the thesis structure is shown in Figure 1.3. This introductory chapter gives a general overview of cellular communications. The motivation for the work carried out in this thesis, as well as the organisation of the thesis, is addressed. The next chapter presents the models of the forward error correction (FEC) coded and uncoded DS-CDMA communications systems, on which all discussions in the subsequent chapters are based. Two decoding algorithms of the FEC codes are detailed. A powerful FEC coding scheme known as Turbo codes is also presented. The spreading codes used for spectrum spreading are introduced and the conventional receiver structure is also discussed.

Chapter 3 reviews some multiuser receivers, starting with a discussion on the problems of the conventional receiver. These established receivers are compared with the proposed detectors in the subsequent chapters. A different implementation of the optimum multiuser detector is investigated in Chapter 4. A new multiuser detector is introduced in Chapter 5. In Chapter 6, the favourable features of this new detector such as low complexity is incorporated into a near-optimum multiuser receiver to yield a sub-optimum multiuser detector with good performance. Finally, the powerful iterative algorithm of the Turbo code is exploited in a new iterative multiuser detector which is based on the proposed sub-optimum multiuser detector. This is discussed in Chapter 7.

The concluding chapter gives a summary of the thesis and lists the contributions to knowledge as well as the works for future development.

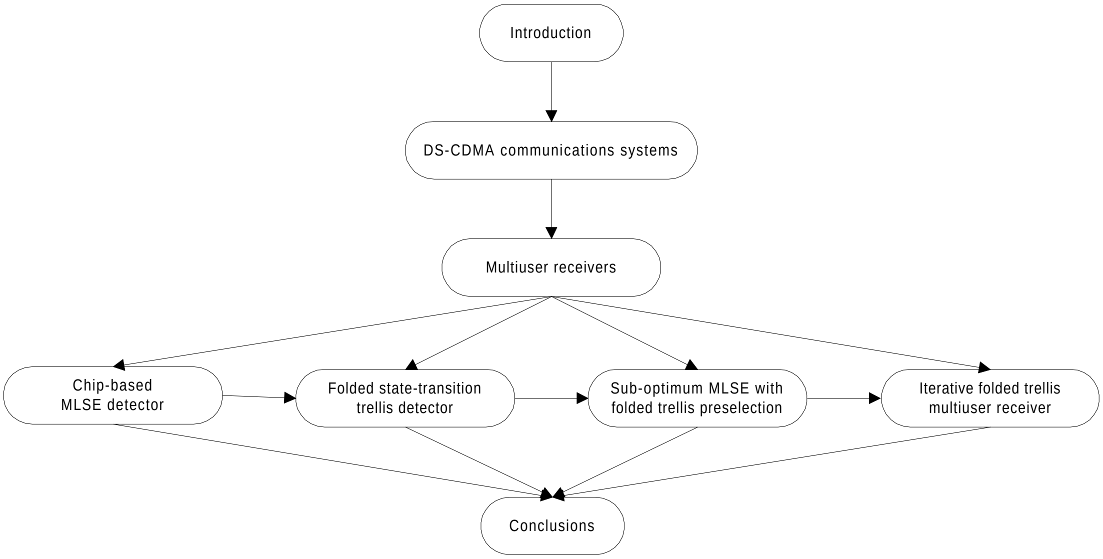
Introduction
DS-CDMA communications systems
Multiuser receivers
Chip-based MLSE detector
Folded state-transition trellis detector
Sub-optimum MLSE with folded trellis preselection
Iterative folded trellis multiuser receiver
Conclusions

Figure 1.3: *The organisation of the thesis structure*

Chapter 2
DS-CDMA communications systems

This chapter concentrates on the DS-CDMA communications systems. The principle of spread spectrum technology is first presented in Section 2.1. A simple illustration of its application in DS-CDMA communications systems is shown in Section 2.2. The following section (2.3) describes the downlink of the DS-CDMA system model in detail including the transmitter and the multipath channel. The mathematical models for the signals at different stages are also derived. Section 2.4 covers the spreading codes used in the system. The conventional detector is introduced in the next section (2.5), with emphasis on its shortcomings. Channel coding in the form of forward error correction is discussed in Section 2.6 which includes convolutional coding, two convolutional decoding methods (Viterbi algorithm (VA) and Bahl *et al.* [2] (BCJR) algorithm) and also a new class of coding scheme known as Turbo codes. The final section gives a summary of the chapter.

2.1 Spread spectrum principles

The most widely recognised form of spread spectrum [19, 20] is probably direct sequence, on which all discussions in this chapter are based. The direct sequence spread spectrum (DSSS) process is performed by effectively multiplying an RF carrier, $\cos(w_c t)$, and a pseudo-noise (PN) digital signal, $s(t)$. Figure 2.1 shows a basic SS system model for both the transmitter and the receiver. First the PN code is modulated onto the information signal, $b(t)$, using one of several modulation techniques (eg. BPSK, QPSK, etc). Then, a mixer is used to multiply the RF carrier and PN modulated information signal. This process causes the RF signal to be replaced with a very wide bandwidth signal with the spectral equivalent of a noise signal. On reception, the demodulation process (for the BPSK case) begins with the mixing of the incoming RF signal with a similar RF carrier, $2\cos(w_c t)$, to give $r(t) = b(t)s(t) + b(t)s(t)\cos(2w_c t)$[1]. After which, the filtered output, $b(t)s(t)$, is multiplied with the same PN code. The output is

[1] Delays in the channel have been ignored here for simplicity

a signal that is a maximum, $b(t)s^2(t)$, when the two signals exactly equal one another or are "correlated". The correlated signal is then filtered and sent to a BPSK demodulator.

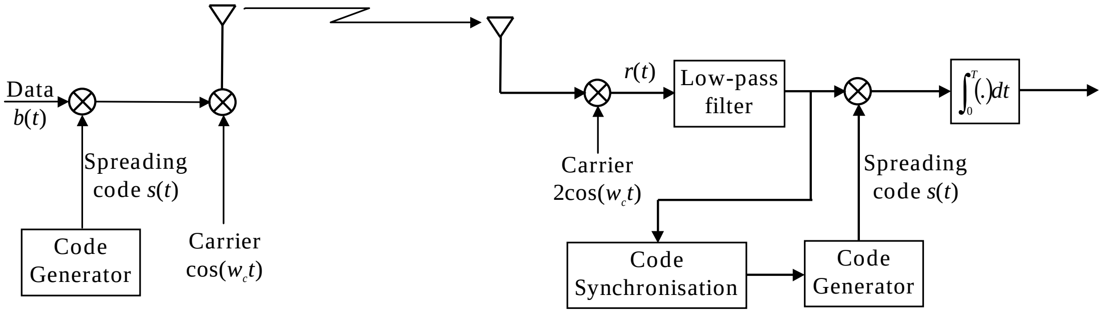

Figure 2.1: *Direct sequence spread spectrum communications systems model*

To illustrate the spectrum spreading concepts, consider a rectangular pulse of amplitude $+1$ and duration T as shown in Figure 2.2. Its amplitude spectrum is a sinc function[2] with the first zero crossing point at $1/T$. If the duration of the rectangular pulse is compressed to T_c, the corresponding effect is an expansion in the frequency domain. The amplitude spectrum still retains its original shape, but its first zero crossing is now at $1/T_c$, and the amplitude is reduced from T to T_c [21]. In DSSS, the PN spreading code is a continuous train of rectangular or chip pulses, each with duration T_c. The envelope of the frequency spectrum of the entire spreading code follows the shape of the frequency spectrum of a single chip pulse. The spreading of the signal is achieved by multiplying the spreading code with the data signal, such that the spreading code "chops" the data up to give an output which has a spectrum similar to the code. From another perspective, the bit rate has increased in such a way that it still contains the original information. The bandwidth in DSSS systems is often taken as the null-to-null bandwidth of the main lobe of the power spectral density plot. Therefore, the bandwidth is spread by a factor of $G = T/T_c$, which is also known as the processing gain of the system. In this simplified scenario, G is also the length or number of chips of the spreading sequence. The signals generated with this technique appear as noise in the frequency domain. The wide bandwidth provided by the PN code allows the signal power to drop below the noise threshold without loss of information.

The result of modulating an RF carrier with such a code sequence is to produce a direct sequence modulated spread spectrum with a sinc frequency spectrum centred at the carrier fre-

[2] $sinc(x) = sin(\pi x)/(\pi x)$

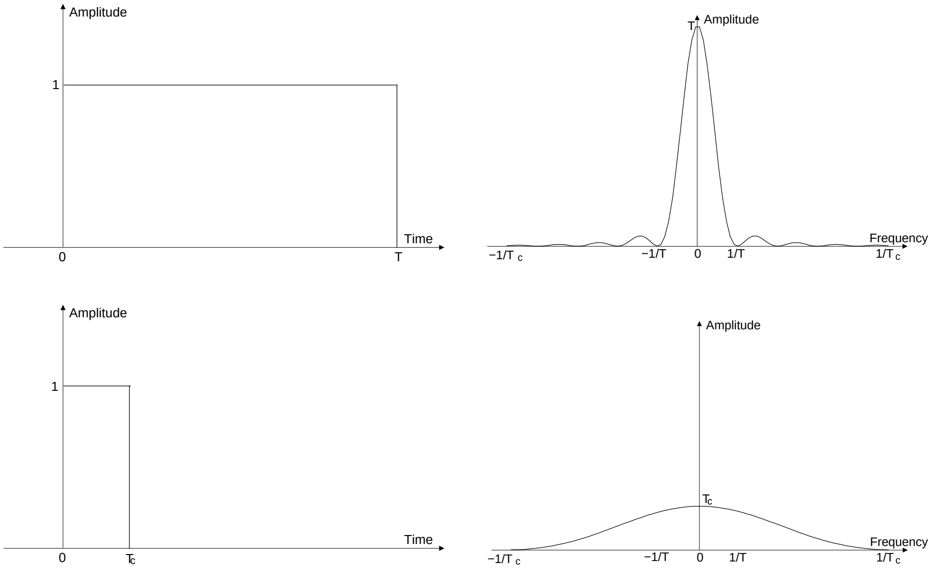

Figure 2.2: *Spectrum spreading*

quency. The main lobe of this spectrum has a bandwidth twice the clock rate of the modulating code, from null to null. The sidelobes have a null to null bandwidth equal to the code's clock rate. For simplicity, all further discussions ignore the carrier and focus only on the baseband equivalent system.

2.2 DS-CDMA principles

In direct-sequence CDMA (DS-CDMA), users can share the same channel because they are assigned unique spreading codes to minimise mutual interference. To illustrate the application of spread spectrum technique in this multiple access scheme, consider two users in a system as shown in Figure 2.3. The spreading codes of the two users, $s_1(t)$ and $s_2(t)$, are distinct from each other. Assuming a noise free channel, the output signals of both transmitters arrive at both receivers as a single signal, which is given as $b_1(t)s_1(t) + b_2(t)s_2(t)$. Each receiver correlates the received signal with their respective spreading code. From the normalised autocorrelation, $s_1^2(t) = s_2^2(t) = 1$, the resulting outputs are $b_1(t) + b_2(t)s_1(t)s_2(t)$ and $b_2(t) + b_1(t)s_1(t)s_2(t)$. If orthogonal spreading codes are used, the ideal cross correlation is $s_1(t)s_2(t) = 0$, and the idealised outputs are simply $b_1(t)$ and $b_2(t)$ respectively. However, this is usually not the case (the reasons for this will be discussed in Section 2.5). The residual, non-zero term translates to CCI for other users in the CDMA system and this interference is called multiple access interference (MAI). Fortunately, the residual cross-correlation term, which appears like background noise to a simple receiver, is small and the desired information can still be recovered successfully, as seen in the spectrum of the output. As the number of users increases, the power of the MAI also increases and the system performance degrades. The work here focuses only on the intra-cell interference (interference from users within the same cell), and inter-cell interference (leakage from adjoining cells) is modelled as a contribution to the background Gaussian noise. Hence, there is a limitation to the number of users in the CDMA system sharing the same spectrum. This limitation is affected by the processing gain, the correlation of the spreading codes, the relative power of transmission, noise and many other factors. Therefore, the capacity of a CDMA system is softly limited, i.e., the maximum number of users is not clear-cut, unlike the other two schemes (FDMA and TDMA), where the number of users cannot go beyond the number of frequency bands or time slots.

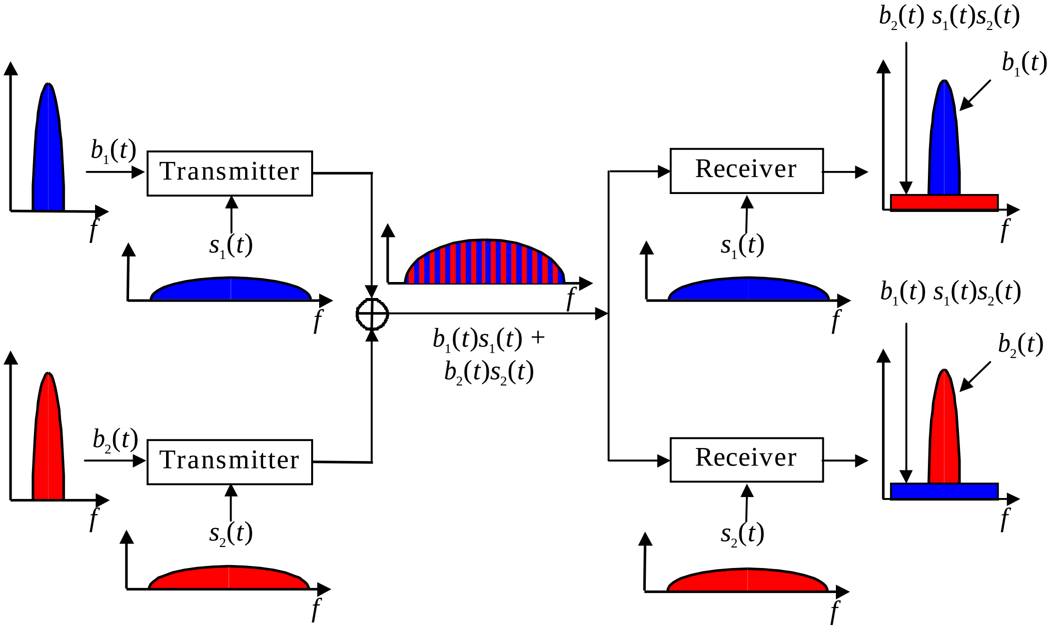

Figure 2.3: *Two-user DS-CDMA system*

2.3 System model

A general downlink forward error correction (FEC) coded DS-CDMA system is illustrated in Figure 2.4. In this model, K users share the same communication media. Prior to spectrum spreading, the input data, $d_k(t) \in \{0, 1\}$ $(k = 1, \ldots, K)$, is first pre-processed by incorporating FEC coding and binary phase shift keying (BPSK) modulation. The encoded and modulated signals, $b_k(t) \in \{-1, 1\}$ $(k = 1, \ldots, K)$ are then bandwidth-expanded by the user-specific spreading sequences, $s_k(t)$ $(k = 1, \ldots, K)$. In the case of the downlink, the signals transmitted by the users are combined at the transmitter to give a joint signal, $v(t)$, which is then passed synchronously through the same channel. The synchronisation of the signals considerably simplifies analysis and often permits the derivation of some closed-form expressions for desired performance measures. This is useful since, from the analytical results of a synchronous system, many useful conclusions can be drawn for the more complex asynchronous cases [22, 23]. As well as the channel effect, additive white Gaussian noise (AWGN), $n(t)$, is also present in the system. The received signal, $y(t) = x(t) + n(t)$, is the sum of the multipath-corrupted transmitted signal, $x(t)$, and the Gaussian noise. The receiver designs which include signal detection and decoding are discussed in detail in the next chapter. This work is focused on the downlink and thereby the receivers for mobile handsets. This is because the challenge lies in designing a near-optimum receiver that is simple enough to be implemented in the power, cost and size limited handset.

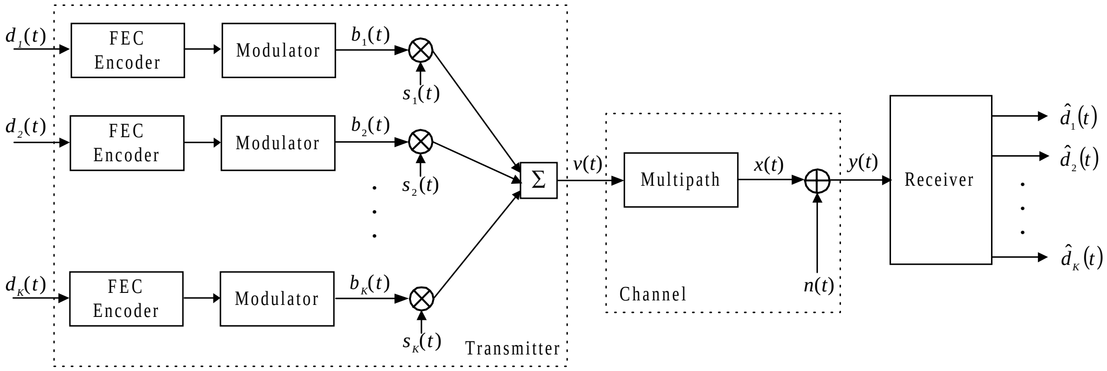

Figure 2.4: *Downlink DS-CDMA system model*

2.3.1 Transmitter

The pre-assigned signature waveforms are defined as,

$$s_k(t) = \sum_{i=0}^{G-1} c_{i,k} p(t - iT_c), \qquad (k = 1, \ldots, K) \tag{2.1}$$

where G is the processing gain of the spreading waveform, $c_{i,k} \in \{-1, +1\}$ is the i-th chip value of the k-th user, T_c is the chip duration and $p(t)$ is the rectangular chip waveform which is zero outside $[0, T_c)$. Each waveform, $s_k(t)$, is thus restricted to a symbol duration, $T = GT_c$, i.e., zero outside $[0, T)$. This type of spreading codes is known as short codes, and codes that span more than one symbol duration are hence the long codes. Only short spreading codes are considered in the work presented here. Since signal processing is carried out on sampled signals, the received continuous-time signal is assumed to be sampled after front-end filtering with G samples per symbol interval. Hence, the equivalent discrete-time mathematical model, which is more suitable for algorithm derivations, is used instead. The spreading waveform can be rewritten in discrete time as a z-transform spreading code polynomial of length G,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}, \qquad (k = 1, \ldots, K) \tag{2.2}$$

where z^{-1} represents a delay of one chip.

The input data bit sequences for different users are identically independently distributed (i.i.d) and equiprobable. During the f-th signalling interval, the joint transmitted signal can be expressed as

$$V_f(z) = \sum_{k=1}^{K} \sqrt{P_k} b_{k,f} S_k(z), \qquad (f = 0, \ldots, F-1) \tag{2.3}$$

where P_k and $b_{k,f} = \{-1, +1\}$ represent the transmitted power and the f-th modulated symbol of the k-th user respectively. Each user transmits F bits.

2.3.2 Channel

For slowly fading channels, the channel impulse response can be estimated precisely and thus the channel impulse response can be assumed to be known. In cdmaone and IMT-2000 CDMA systems, channel impulse responses are monitored using pilot channels. The effect of multipath on downlink CDMA system has been studied in [24, 25]. A more complete discussion of multipath fading effect can be found in [9].

In addition to the Gaussian noise, the channel has multipath effect which is modelled by an L-tap finite impulse response (FIR) filter [26] as shown in Figure 2.5.

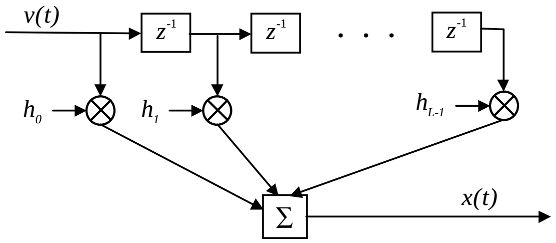

Figure 2.5: *The multipath channel model*

This model represents a multipath channel with L resolvable paths. The channel impulse response can be written as

$$h(t) = \sum_{l=0}^{L-1} h_l \delta(t - lT_c), \tag{2.4}$$

where h_l is the path coefficient of the l-th path and $\delta(t)$ is the Dirac function. Only real signals are considered in this thesis, extension to complex signals is straightforward. Similarly, equation (2.4) can be rewritten in z-transform as

$$H(z) = \sum_{l=0}^{L-1} h_l z^{-l}. \tag{2.5}$$

Due to the multipath effect, the bit intervals are no longer independent of each other. Hence, the entire data transmission has to be considered,

$$V(z) = \sum_{f=0}^{F-1} V_f(z) z^{-Gf}, \tag{2.6}$$

where z^{-G} is a delay of one symbol interval. Let

$$R_k(z) = H(z)S_k(z), \tag{2.7}$$

which is a polynomial of length $G + L - 1$, represents the overall channel response of the k-th user due to multipath effect and spectrum spreading. The convolution of the multipath channel response, $h(t)$, and the transmitted signal, $v(t)$, results in the noise-free received signal, $x(t)$. In the z domain, convolution translates to multiplication, and $X(z) = H(z)V(z)$ can be expanded as

$$\begin{aligned} X(z) &= \sum_{k=1}^{K} R_k(z)\sqrt{P_k}\sum_{f=0}^{F-1} b_{k,f} z^{-Gf} \\ &= \sum_{k=1}^{K} R_k(z)\sqrt{P_k} B_k(z) \\ &= \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z), \end{aligned} \tag{2.8}$$

where:

$$B_k(z) = \sum_{f=0}^{F-1} b_{k,f} z^{-Gf} \qquad (k = 1, \ldots, K) \tag{2.9}$$

represents the entire transmission of the modulated symbols for the k-th user,

$\mathbf{R}(z) = [R_1(z), \ldots, R_K(z)]^T$ is a vector comprising of all the users' overall channel response due to spreading and multipath effect,

$\mathbf{P} = diag\{\sqrt{P_1}, \ldots, \sqrt{P_K}\}$ is the $K \times K$ diagonal matrix representing the received power of the users,

$\mathbf{B}(z) = [B_1(z), \ldots, B_K(z)]^T$ is the entire modulated symbol sequence vector.

The received signal is thus $Y(z) = X(z) + N(z)$, where $N(z)$ is the z-transform of the AWGN sequence, which has zero mean and two-sided power spectral density equals to $N_0/2$ or σ^2.

If multipath effect is not present or negligible in the channel and only Gaussian noise exists, the channel is a non-dispersive AWGN channel. Although this channel is less practical, it is useful for simplified analysis. Each symbol in a symbol duration does not interfer with symbols in adjacent symbol intervals for a AWGN channel, hence it is sufficient to focus on just a single symbol duration. As $R_k(z) = S_k(z)$ now, the transmitted sequence becomes

$$X(z) = \sum_{k=1}^{K} S_k(z)\sqrt{P_k} b_k, \tag{2.10}$$

where b_k $(k = 1, \ldots, K)$ are the transmitted symbols of the symbol interval in question.

2.4 Spreading sequences

Recall that the receiver despreading operation is a correlation operation with the spreading code of the desired transmitter. Ideally, a received signal that has been spread using a different spreading code will not be despreaded and will cause minimal interference to the desired signal. The specific amount of interference from a user employing a different spreading code is related to the cross-correlation between the two spreading codes. Orthogonal codes have zero cross-correlation levels, thus MAI can be totally eliminated. However, it is usually not the case in the sense that perfect orthogonality is hard to achieve in a practical DS-CDMA system. Even if the channel is synchronous, orthogonality still cannot be ensured when multipath effect exists or long codes[3] are used. Hence, low cross-correlation spreading codes, which also exhibit other more desired features, are sought.

Noise-like wideband spread-spectrum signals are generated using pseudo-noise or pseudo-random spreading code sequences [27], which are so called because they exhibit the properties of random numbers. However, if the code sequences were truly random, then nobody, including the intended receiver, could access the channel. Thus, PN sequences should be relatively easy to generate and should appear as random noise to everyone else, except to the transmitter and the intended receiver. Other properties of a PN sequence include the ease of generation, long periods and difficulty of reconstruction from a short segment. In DS-CDMA systems, a PN spreading waveform is a time function of a PN sequence. Examples of PN spreading sequences are Gold codes and Kasami codes used in IMT-2000 [28].

2.4.1 Orthogonal codes

One of the most common type of orthogonal codes is Walsh codes. This set of codes is mutually orthogonal at zero phase offset. However, at other phase offsets, the cross-correlations are very high. Even if phase synchronisation is maintained in the downlink, orthogonality of the codes is still lost under multipath channel effect. Another limitation on Walsh codes is that the number of available codes for a certain degree is relatively small as compared to its period. The implication is that to increase the number of codes, the degree and consequently the period of the code have to be increased, resulting in a larger storage.

[3] Long codes are desirable because it is difficult for parties to eavesdrop

2.4.2 Random codes

Random codes here refer to randomly selected sequences that are completely chosen at random while preserving the condition of uniqueness. An ideal spreading code would be an infinite sequence of equally likely random binary digits. Unfortunately, the use of an infinite random sequence implies infinite storage in both the transmitter and receiver. Hence, periodic PN codes are always employed. The disadvantage of using finite length random codes is that they have poor cross-correlation properties. However, these codes are able to mimic the use of long sequences.

Random codes are chosen for each symbol interval because it models the use of long spreading codes [18] or when properties of the sequences have been corrupted due to inter-chip interference arising from the presence of a multipath channel. Furthermore, the performance of a synchronous system using random codes is on average the same as that of an asynchronous system [29] and it allow for interferer diversity [30] such as different processing gains. Hence, random codes will be used as the main spreading sequences in the investigations reported here.

2.5 Conventional detector

The conventional detector for the multipath system, also known as the RAKE receiver [31, 32], consists of L matched filters[4] or correlators (called fingers) for each user, as shown in Figure 2.6. Although multipath introduces interpath interference similar to intersymbol interference (ISI) in TDMA, it is also a valuable source of time diversity here. The output of each finger is combined into a single output to maximise the signal-to-noise ratio (SNR). Based on the sampled value of this single output, a decision can be made by a decision device which is represented by the signum function for BPSK.

Each of the L fingers is matched to one path, i.e., matched to the spreading code but with a delay of one chip from each other. Assume that the interference from the other users is Gaussian, the best combining weights are the path coefficients of the channel. The transfer function of the

[4] Note that the correlation operation can be performed using a matched filter for sampled signals

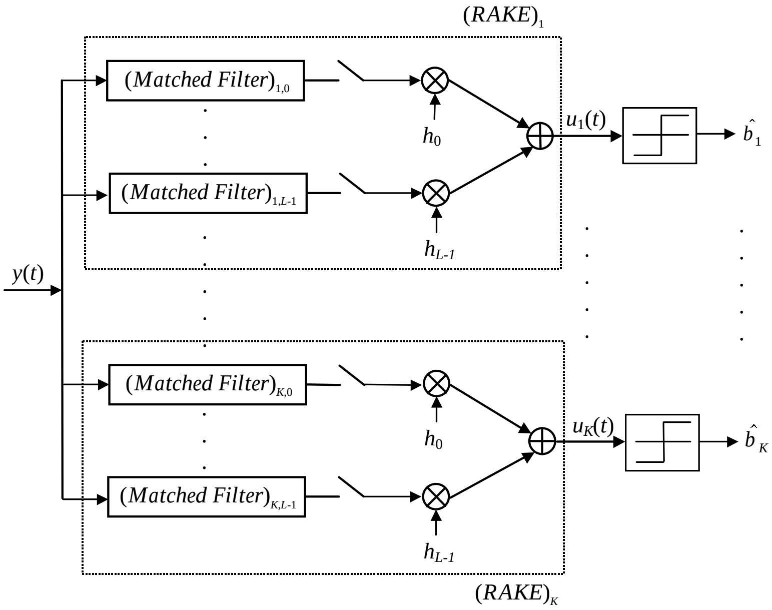

Figure 2.6: *The RAKE receiver*

RAKE receiver for the k-th user is thus

$$\begin{aligned} R_k^{RAKE}(z) &= \sum_{l=0}^{L-1} h_l S_k(z^{-1}) z^l \\ &= H(z^{-1}) S_k(z^{-1}) = R_k(z^{-1}), \qquad (k = 1, \ldots, K) \end{aligned} \tag{2.11}$$

which is identical to a filter matched to $R_k(z)$ defined in equation (2.7). Note that $R_k^{RAKE}(z)$ is non-causal, hence a delay of z^{-G-L+2} is required in the actual implementation. Consequently, the output of the RAKE is

$$\begin{aligned} U_k(z) &= R_k^{RAKE}(z) Y(z) \\ &= \Phi_{k,k}^R(z) \sqrt{P_k} B_k(z) + R_k(z^{-1}) N(z) \\ &\quad + \sum_{\substack{j=1, \\ j \neq k}}^{K} \Phi_{k,j}^R(z) \sqrt{P_j} B_j(z), \qquad (k = 1, \ldots, K) \end{aligned} \tag{2.12}$$

where $\Phi_{k,j}^R(z)$ represents the cross-correlation between $R_k(z^{-1})$ and $R_j(z)$. The sampled value at time $(G(f+1) + L - 2)T_c$ $(f = 0, \ldots, F-1)$ is equivalent to the coefficient of z^{-Gf} in $U_k(z)$, and based on this statistic a decision for the f-th bit of the k-th user can be made.

In equation (2.12), the first term represents the desired signal, the last term is the MAI and the second term represents spread coloured noise with zero mean and covariance $\sigma^2\Phi^R_{k,k}(z)$. If the coefficient of z^0 in $\Phi^R_{k,j}(z)\,\forall(j=1,\dots,K, j\neq k)$ is equal to 0, then there is no MAI. This is possible if the channel has only a single path and the codes used are orthogonal. This can be easily shown by observing that the cross-correlation of $S_k(z)$ and $S_j(z)$ $(j \neq k)$ for orthogonal codes always gives a z^0 with zero-valued coefficient and that the channel response $H(z)$ now has only one non-zero term (z^0). However, practical channels are non-ideal, and even with orthogonal codes, MAI is usually inevitable. From equation (2.12), it is observed that the MAI term involves the powers of the users. Hence, if the powers of the interfering users are high, the MAI would also be increased. This is known as "near-far" effect. Usually, stringent power control is required to suppress this effect.

The output of all RAKE receivers matched to all the users before sampling can be expressed neatly in matrix form as

$$\mathbf{U}(z) = \Phi^R(z)\mathbf{P}\mathbf{B}(z) + \mathbf{R}(z^{-1})N(z), \tag{2.13}$$

where: $\mathbf{U}(z) = [U_1(z),\dots,U_K(z)]^T$ is the RAKE matched filter bank output vector,

$\mathbf{R}(z^{-1}) = [R_1(z^{-1}),\dots,R_K(z^{-1})]^T$ is the transfer function vector of the RAKE matched filter bank,

$\Phi^R(z) = \mathbf{R}(z^{-1})\mathbf{R}^T(z)$ is the $K\times K$ matrix representing correlations between the users subjected to the effect of the channel, and the (i,j)-th element is $\Phi^R_{i,j}(z) = R_i(z^{-1})R_j(z)$.

Two assumptions are made here: the receiver knows exactly the received power of the users and the channel attenuation is simplified to be real for analysing coherent methods. Note that equations (2.8) and (2.10) are still valid if P_k is the received power (instead of transmitted power) of the k-th user and $H(z)$ is the normalised channel response.

In the case of an AWGN channel, a number of simplifications can be made. Firstly, only one matched filter is required for each user. Again, attention is focused only on one symbol interval, the output of the conventional detector becomes

$$\begin{aligned} U_k(z) &= \Phi^S_{k,k}(z)\sqrt{P_k}b_k + S_k(z^{-1})N(z) \\ &\quad + \sum_{\substack{j=1,\\ j\neq k}}^{K} \Phi^S_{k,j}(z)\sqrt{P_j}b_j, \qquad (k=1,\dots,K) \end{aligned} \tag{2.14}$$

where $\Phi_{k,j}^{S}(z)$ represents the cross-correlation between $S_k(z^{-1})$ and $S_j(z)$ since the transfer functions of the matched filters are $S_k(z^{-1})$ $(k = 1, \ldots, K)$. In matrix form, it can be expressed as

$$\mathbf{U}(z) = \Phi^S(z)\mathbf{Pb} + \mathbf{S}(z^{-1})N(z), \tag{2.15}$$

where: $\mathbf{S}(z^{-1}) = [S_1(z^{-1}), \ldots, S_K(z^{-1})]^T$ is the transfer function vector of the matched filter bank,

$\Phi^S(z) = \mathbf{S}(z^{-1})\mathbf{S}^T(z)$ is the $K \times K$ matrix representing purely the correlations between the users,

$\mathbf{b} = [b_1, \ldots, b_K]^T$ is the input data symbol vector in the symbol duration of interest.

2.6 Forward error correction coding

As shown in Section 2.3.2, the communications system suffers from noise and multipath effect in the channel. In order to provide reliable transmission of digital information over the channel, an essential signal processing operation, namely, channel coding, is exploited. Another motivation for the use of channel coding is to reduce the required SNR for a fixed bit error rate (BER). This reduction in SNR translates to a reduction in transmitted power or a reduction in hardware costs [33] and an increase in capacity.

Channel coding may be implemented by means of forward error correction (FEC). The channel encoder adds redundancy to the data bits according to a prescribed rule. The amount of redundancy introduced by the encoding of the data in this manner is measured by the ratio (R) of the number of input data bits to the number of encoded symbols. This redundancy is then exploited by the decoder to estimate the original data bits, thereby correcting for channel disturbances. However, the addition of redundancy in the coded message implies the need for increased transmission bandwidth and increased complexity. Hence, the value of $R = m/n$ (where m and n are integers) is usually kept high to achieve an appropriate trade-off.

FEC coding has been classified into block codes and convolutional codes. The difference is the presence or absence of memory in the encoders for the two codes. The channel encoder for a block code accepts information in successive m-bit blocks; for each block, it adds $n - m$ redundant bits that are algebraically dependent on the m message bits. Each block is treated independently, hence the encoder does not have memory. A convolutional code, as its name implies, is a discrete-time convolution of the input sequence with the impulse response of the encoder. The duration of the impulse response equals the memory of the encoder. Unlike a block code, the encoder for a convolutional code accepts message bits as a continuous sequence. Convolutional coding is attractive for continuous transmission and is thus more widely accepted in wireless communications [34]. Only convolutional codes are of interest in this thesis.

The convolutional code is described in Section 2.6.1. Decoding of this code using the VA and the BCJR algorithm is outlined in Section 2.6.2 and Section 2.6.3 respectively. Concatenation of two convolutional codes yields a powerful code known as a Turbo code, which is introduced briefly in Section 2.6.4.

2.6.1 Convolutional coding

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register. The encoder of a binary convolutional code with rate $R = 1/n$ may be viewed as a finite-state machine that consists of a κ-stage shift register, n modulo-2 adders and a multiplexer that serialises the outputs of the adders. The parameter κ is called the constraint length of the convolutional code. The input data to the encoder, assumed to be binary, is shifted into the shift register one bit at a time. For every shift, the number of output bits is n.

The concept of convolutional encoding is best illustrated with an example. Figure 2.7 shows the encoder for a convolutional code of constraint length $\kappa = 3$, code rate $R = 1/2$ and linear algebraic function generators $(7, 5)$ in octal form. Each generator describes the connections of the shift register to the corresponding modulo-2 adder.

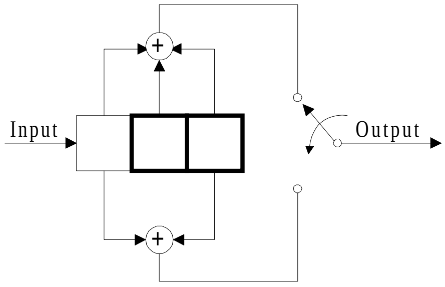

Figure 2.7: *The encoder for the convolutional code of rate $R = 1/2$, constraint length $\kappa = 3$ and generators (7,5)*

The structural properties of a convolutional encoder can be portrayed in graphical form by using a code trellis[5] as shown in Figure 2.8. The state of a convolutional encoder of rate $1/n$ is defined as the most recent $(\kappa - 1)$ message bits moved into the shift register before the current message bit. In this case where $\kappa = 3$, each state contains two bits and is the content of the last 2 stages (see Figure 2.7) in the shift register. There are $2^{\kappa-1}$ possible states. Transitions between states are governed by the incoming (uncoded) data bits (0 or 1). The label pertaining to each branch in the code trellis represents the current input message bit and the output codeword of the encoder. For example, a binary digit 0 input to the encoder in state 10, will output the codeword 10 and move to state 01. If however a 1 had been present at the input while in state 10, the bit sequence 01 would have appeared at the output, and the system would have moved to state 11 instead.

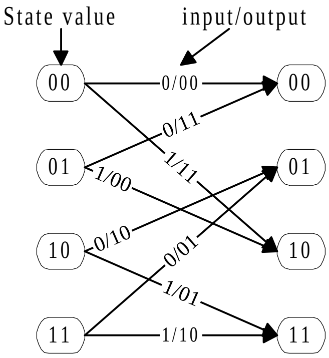

Figure 2.8: *Code trellis for the convolutional code of code rate $R = 1/2$, constraint length $\kappa = 3$ and generators (7,5)*

2.6.2 Decoding – Viterbi algorithm

The coded bit stream produced by the convolutional coder from the previous section is transmitted over a noisy, power limited channel to a receiver containing a convolutional decoder. The noise can cause some of the bits to be decoded incorrectly, causing one or more bit errors. If the noise is AWGN, the maximum likelihood sequence estimator (MLSE) implemented using the VA [36, 37] is the optimum method of finding the most-likely original noise free sequence, given the sequence of noisy symbols. The VA has been frequently used commercially in satellites and space communications and is fairly effective for short constraint length $(\kappa <= 7)$ codes. The optimum decoding involves searching through the trellis for the most probable se-

[5] A more detailed discussion on trellis can be found in [35]

quence. In other words, the sequence found is the one that is closest in distance to the received sequence of noisy symbols. Depending on whether hard or soft decisions are made by the decoder, Hamming distance or Euclidean distance respectively is taken as the distance measure. As the latter improves performance by 2 dB, it is the scheme adopted throughout this thesis.

The first step taken in Viterbi decoding is to form the Viterbi trellis which is simply a time-indexed concatenation of many code trellis. See Figure 2.9 for a simplified Viterbi trellis that is used to decode the above example with 4 message bits or 4 transmitted codewords. Each node corresponds to a distinct state at a given time, and each branch represents a transition to some new state at the next instant of time. The trellis starts and ends at state 00. From time $t = 0$ to $t = 2$ and from $t = 4$ to $t = 6$, the trellis is different from the code trellis because the shift register is assumed to be initialised with zeros and the encoder is flushed with zeros to drive the shift register back to the zero state. All transitions that are not possible have been eliminated.

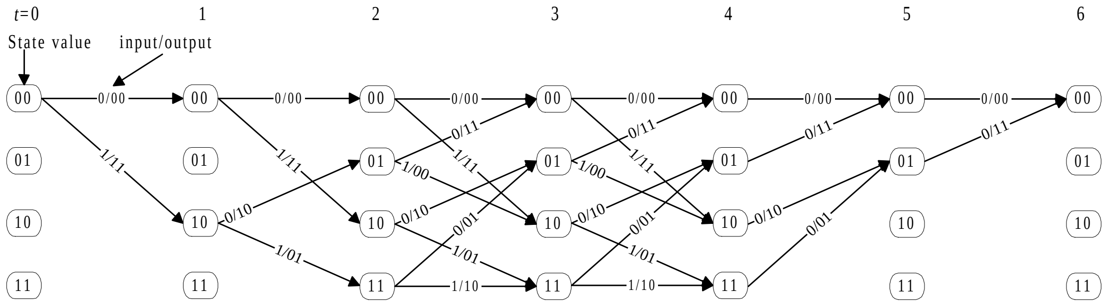

Figure 2.9: *Viterbi trellis for the convolutional code of code rate $R = 1/2$, constraint length $\kappa = 3$ and generators (7,5)*

To compute the most probable sequence, the VA first recursively computes the survivor path entering each state. The survivor path for a given state is the sequence of symbols, entering the given state, that is closest in distance to the received sequence of noisy symbols. The distance between the survivor path and the sequence of noisy symbols is called the path metric for that state. At the end of the trellis or after sufficient delay, the path metrics pertaining to all the survivor paths entering all the states are compared and the path with the minimum path metric is selected as the global survivor path.

Let the received sequence of noisy symbols be $(y_1, \ldots, y_{n\tau})$, where τ is the number of transmitted data bits including the $\kappa - 1$ trailing zeros. The transition metric is defined as the Euclidean distance between the sequence of received noisy symbols and the sequence of noiseless output symbols corresponding to that transition. That is, assuming BPSK modulation, the transition metric for the transition from state i to state j at time t is

$$M_{i,j,t}^{T} = \sum_{l=1}^{n}(y_{(t-1)n+l} - \sqrt{E_s}b_{i,j,l})^2, \quad (t = 1, \ldots, \tau; i, j = 1, \ldots, 2^{\kappa-1}) \tag{2.16}$$

where $b_{i,j,l} \in \{-1, +1\}$ is the l-th modulated output symbol of the transition from state i to state j, and E_s is the energy of each symbol. If $M_{j,t}^{P}$ is defined as the path metric for state j at time t, and I as the set of states that have transitions to state j at time t, then

$$M_{j,t}^{P} = \min_{i \in I}(M_{i,t-1}^{P} + M_{i,j,t}^{T}). \quad (t = 1, \ldots, \tau; j = 1, \ldots, 2^{\kappa-1}) \tag{2.17}$$

Let $i_{min} \in I$ be the state that results in the minimum value. Then the transition from state i_{min} to state j at time t is appended to the survivor path of state i_{min} in order to form the updated survivor path of state j at time t. The path metrics at time $t = 0$ are initialised to zeros, i.e., $M_{j,0}^{P} = 0\,(j = 1, \ldots, 2^{\kappa-1})$. Hence, by recursively calculating the path metrics and forming the survivor path until the end of the trellis, the global survivor path can be traced out. For the example in Figure 2.9, at time $t = 6$, the most likely path is traced from the state 00. If the shift register is not flushed with zeros, then at time $t = 4$, the path metrics of all the states are compared, and the global survivor path is traced out from the state with the minimum path metric. Performance is not affected by the flushing as shown in [38].

For long information sequences, the decoding delay is intolerable for practical applications. Furthermore, the memory storage required to store the long survivor paths is large and expensive. In practice, the length of the survivor paths is fixed at a reasonable value, say t_s. That is, at time t, the state with the minimum path metric is traced back until time $t - t_s$, and a final decision is made on the uncoded bit. If t_s is chosen large enough, all survivor paths at time t stem from the same node at time $t - t_s$ and thereby the same decision is made if they are all traced back. It is shown in [38] that $t_s \geq 5\kappa$ results in negligible degradation in performance due to the truncation of the survivor paths.

The VA is also applicable to detection problems in digital communications. It is widely used in equalisation to detect signals in channels with memory or ISI channels [39, 40]. See Appendix A. In GSM, the VA is used for both demodulation and decoding [41].

2.6.3 Decoding – BCJR algorithm

The VA is a maximum likelihood decoding method which minimises the codeword error rate for convolutional codes. However, it does not necessarily minimise the probability of symbol error. Another method of decoding that results in minimum symbol error rate is proposed by Bahl *et al.* [2]. This BCJR algorithm is an optimum decoding method for convolutional codes which minimises the symbol error probability by making decisions based on the maximum *a posteriori* (MAP) probability [42].

Consider the convolutional codes of rate $1/n$ and constraint length κ. Define S_t as the state at time $t \in (0, \ldots, \tau)$ and index m as one of the $M = 2^{\kappa-1}$ possible states. Assume that F data bits in a frame are transmitted, followed by $\kappa - 1$ zeros to flush the encoder, thus $\tau = F + \kappa - 1$. Let $d_t \in \{0, 1\}$ $(t = 1, \ldots, \tau)$ be the t-th uncoded bit in the data stream including the trailing zeros, $w_f \in \{0, 1\}$ $(f = 1, \ldots, n\tau)$ be the f-th encoded bit, and y_f $(f = 1, \ldots, n\tau)$ be the f-th noisy received encoded symbol. Also let $\mathbf{w}_t(m^-, m) = [w_{(t-1)n+1}, \ldots, w_{tn}]^T$ $(t = 1, \ldots, \tau)$ represents the output codeword associated with the transition from state $S_{t-1} = m^-$ to state $S_t = m$, $\mathbf{Y}_t = [y_{(t-1)n+1}, \ldots, y_{tn}]^T$ $(t = 1, \ldots, \tau)$ the noisy received codeword at time t, and $\mathbf{Y}_{t'}^t = [\mathbf{Y}_{t'}, \ldots, \mathbf{Y}_t]^T$ the vector form of the noisy received sequence of codewords from time t' to time t in a frame. In order to obtain the *a posteriori* probabilities (APP), first define the following probability functions

$$\alpha_t(m) = \Pr(S_t = m|\mathbf{Y}_1^t) \qquad (t = 1, \ldots, \tau) \tag{2.18}$$

$$\beta_t(m) = \Pr(S_t = m|\mathbf{Y}_{t+1}^\tau) \qquad (t = 1, \ldots, \tau - 1) \tag{2.19}$$

$$\gamma_t(m^-, m) = p(S_t = m; \mathbf{Y}_t|S_{t-1} = m^-) \qquad (t = 1, \ldots, \tau). \tag{2.20}$$

The forward state probability, $\alpha_t(m)$, can be obtained by forward recursion,

$$\begin{aligned}
\alpha_t(m) &= \frac{p(S_t = m; \mathbf{Y}_1^t)}{p(\mathbf{Y}_1^t)} \\
&= \frac{1}{p(\mathbf{Y}_1^t)} \sum_{m^-=0}^{M-1} p(S_t = m; S_{t-1} = m^-; \mathbf{Y}_1^{t-1}; \mathbf{Y}_t) \\
&= \frac{1}{p(\mathbf{Y}_1^t)} \sum_{m^-=0}^{M-1} p(S_{t-1} = m^-; \mathbf{Y}_1^{t-1}) p(S_t = m; \mathbf{Y}_t | S_{t-1} = m^-; \mathbf{Y}_1^{t-1}) \\
&= \frac{p(\mathbf{Y}_1^{t-1})}{p(\mathbf{Y}_1^t)} \sum_{m^-=0}^{M-1} p(S_{t-1} = m^- | \mathbf{Y}_1^{t-1}) p(S_t = m; \mathbf{Y}_t | S_{t-1} = m^-) \\
&= h_\alpha^t \sum_{m^-=0}^{M-1} \alpha_{t-1}(m^-)\gamma_t(m^-, m), \; (t = 1, \ldots, \tau; m = 0, \ldots, M-1) \qquad (2.21)
\end{aligned}$$

where $h_\alpha^t = p(\mathbf{Y}_1^{t-1})/p(\mathbf{Y}_1^t)$ is a constant and can be determined using the condition that $\alpha_t(m)$ is normalised as

$$\sum_{m=0}^{M-1} \alpha_t(m) = \sum_{m=0}^{M-1} \Pr(S_t = m | \mathbf{Y}_1^t) = 1. \qquad (t = 1, \ldots, \tau) \qquad (2.22)$$

The normalisation is required to ensure stability of the algorithm. The following Markov property has been applied in the fourth equality of equation (2.21): if S_{t-1} is a known condition, events after time $t-1$ do not depend on $\mathbf{Y}_1^{t-1}$.

Similarly, the reverse state probability can be obtained by backward recursion,

$$\begin{aligned}
\beta_t(m) &= \frac{p(S_t = m; \mathbf{Y}_{t+1}^{\tau})}{p(\mathbf{Y}_{t+1}^{\tau})} \\
&= \frac{1}{p(\mathbf{Y}_{t+1}^{\tau})} \sum_{m^+=0}^{M-1} p(S_t = m; S_{t+1} = m^+; \mathbf{Y}_{t+2}^{\tau}; \mathbf{Y}_{t+1}) \\
&= \frac{1}{p(\mathbf{Y}_{t+1}^{\tau})} \sum_{m^+=0}^{M-1} p(\mathbf{Y}_{t+2}^{\tau} | S_t = m; S_{t+1} = m^+; \mathbf{Y}_{t+1}) p(S_t = m; S_{t+1} = m^+; \mathbf{Y}_{t+1}) \\
&= \frac{1}{p(\mathbf{Y}_{t+1}^{\tau})} \sum_{m^+=0}^{M-1} \Pr(S_t = m) p(\mathbf{Y}_{t+2}^{\tau} | S_{t+1} = m^+) p(S_{t+1} = m^+; \mathbf{Y}_{t+1} | S_t = m) \\
&= \frac{p(\mathbf{Y}_{t+2}^{\tau})}{p(\mathbf{Y}_{t+1}^{\tau})} \sum_{m^+=0}^{M-1} \frac{\Pr(S_t = m) p(S_{t+1} = m^+ | \mathbf{Y}_{t+2}^{\tau}) p(S_{t+1} = m^+; \mathbf{Y}_{t+1} | S_t = m)}{\Pr(S_{t+1} = m^+)} \\
&= h_{\beta}^{t} \sum_{m^+=0}^{M-1} \beta_{t+1}(m^+) \gamma_{t+1}(m, m^+), \ (t = 1, \ldots, t-1; m = 0, \ldots, M-1) \qquad (2.23)
\end{aligned}$$

where $h_{\beta}^{t} = p(\mathbf{Y}_{t+2}^{\tau})/p(\mathbf{Y}_{t+1}^{\tau})$ is a constant, and $\Pr(S_t = m) = \Pr(S_{t+1} = m^+) = 2^{-M}$ for convolutional codes except at the beginning and the ending stages of the trellis. For these special cases, if transitions to the states are possible, then the probabilities of these states are still a constant and can be incorporated into h_{β}^{t}. Note again that $\beta_t(m)$ is also normalised as

$$\sum_{m=0}^{M-1} \beta_t(m) = \sum_{m=0}^{M-1} \Pr(S_t = m | \mathbf{Y}_{t+1}^{\tau}) = 1. \qquad (t = 1, \ldots, \tau - 1) \qquad (2.24)$$

The same Markov property has been applied again in the fourth equality of equation (2.23). The boundary conditions are $\alpha_0(0) = 1$, $\alpha_0(m) = 0$ for $m \neq 0$, $\beta_{\tau}(0) = 1$ and $\beta_{\tau}(m) = 0$ for $m \neq 0$.

For a valid transition from state m^- to state m, the transition probability is given as

$$\begin{aligned}
\gamma_t(m^-, m) &= p(\mathbf{Y}_t | S_t = m; S_{t-1} = m^-) \Pr(S_t = m | S_{t-1} = m^-) \\
&= \Pr(S_t = m | S_{t-1} = m^-) p(\mathbf{Y}_t | \mathbf{w}_t(m^-, m)) \\
&= \Pr(S_t = m | S_{t-1} = m^-) \prod_{i=(t-1)n+1}^{tn} p(y_i | w_i(m^-, m)), \\
&\qquad (t = 1, \ldots, \tau) \qquad (2.25)
\end{aligned}$$

where $\Pr(S_t = m | S_{t-1} = m^-) = 0.5$ for $t \leq F$ and equals to 1 for $t > F$. For $t \leq F$, there are 2 possible transitions out of each state while for $t > F$, only one transition is possible out of each valid state.

Next derive the following joint probabilities at time t:

$$\begin{aligned}
\lambda_t(m) &= p(S_t = m; \mathbf{Y}_1^\tau) \\
&= p(S_t = m; \mathbf{Y}_1^t) p(\mathbf{Y}_{t+1}^\tau | S_t = m; \mathbf{Y}_1^t) \\
&= p(\mathbf{Y}_1^t) \Pr(S_t = m | \mathbf{Y}_1^t) p(\mathbf{Y}_{t+1}^\tau | S_t = m) \\
&= \frac{p(\mathbf{Y}_1^t) p(\mathbf{Y}_{t+1}^\tau)}{\Pr(S_t = m)} \alpha_t(m) p(S_t = m | \mathbf{Y}_{t+1}^\tau) \\
&= h_\lambda^t \alpha_t(m) \beta_t(m), \qquad (t = 1, \ldots, \tau)
\end{aligned} \tag{2.26}$$

where h_λ^t is a constant and

$$\begin{aligned}
\sigma_t(m^-, m) &= p(S_{t-1} = m^-; S_t = m; \mathbf{Y}_1^\tau) \\
&= p(S_{t-1} = m^-; S_t = m; \mathbf{Y}_1^{t-1}; \mathbf{Y}_t; \mathbf{Y}_{t+1}^\tau) \\
&= p(\mathbf{Y}_{t+1}^\tau | S_{t-1} = m^-; S_t = m; \mathbf{Y}_1^{t-1}; \mathbf{Y}_t) p(S_{t-1} = m^-; S_t = m; \mathbf{Y}_1^{t-1}; \mathbf{Y}_t) \\
&= p(\mathbf{Y}_{t+1}^\tau | S_t = m) p(S_t = m; \mathbf{Y}_t | S_{t-1} = m^-; \mathbf{Y}_1^{t-1}) p(S_{t-1} = m^-; \mathbf{Y}_1^{t-1}) \\
&= \frac{p(\mathbf{Y}_{t+1}^\tau) p(\mathbf{Y}_1^{t-1})}{p(S_t = m)} p(S_t = m | \mathbf{Y}_{t+1}^\tau) \\
&\qquad p(S_t = m; \mathbf{Y}_t | S_{t-1} = m^-) p(S_{t-1} = m^- | \mathbf{Y}_1^{t-1}) \\
&= h_\sigma^t \alpha_{t-1}(m^-) \gamma_t(m^-, m) \beta_t(m), \qquad (t = 1, \ldots, \tau)
\end{aligned} \tag{2.27}$$

where h_σ^t is a constant. The same Markov property has been applied again to the second equality in equation (2.26), and the third and fourth equalities in equation (2.27).

The decoding procedure is then:

1. The boundary conditions of $\alpha_0(m)$ and $\beta_\tau(m)$ are determined for $m = 0, \ldots, M-1$.

2. Upon receiving $\mathbf{Y}_t$, the decoder computes $\gamma_t(m^-, m)$ and $\alpha_t(m)$ using equation (2.25) and equation (2.21) respectively. The calculated values of $\alpha_t(m)$ are stored for all t and m. All the calculated values of $\gamma_t(m^-, m)$ can be stored or recomputed again when necessary since it is not computationally intensive.

3. After the complete sequence $\mathbf{Y}_1^\tau$ has been received, the decoder can then recursively calculates $\beta_t(m)$ using equation (2.23). When $\beta_t(m)$ have been computed for all t and

m, the joint probabilities $\lambda_t(m)$ and $\sigma_t(m^-, m)$ can be obtained using equation (2.26) and equation (2.27) respectively.

Recall that each state, $S_t = (d_t, \ldots, d_{t-\kappa+2})$ is made up of the $\kappa - 1$ most recent input bits. Let D_t^d be the set of states, S_t, such that $d_t = d \in \{0, 1\}$. Now,

$$p(d_t = 0; \mathbf{Y}_1^\tau) = \sum_{m \in D_t^0} \lambda_t(m) = h_\lambda^t \sum_{m \in D_t^0} \alpha_t(m)\beta_t(m), \quad (t = 1, \ldots, F) \qquad (2.28)$$

thus the decision rule is given as

$$\sum_{m \in D_t^0} \alpha_t(m)\beta_t(m) \underset{d_t=1}{\overset{d_t=0}{\gtrless}} \sum_{m \in D_t^1} \alpha_t(m)\beta_t(m). \qquad (t = 1, \ldots, F) \qquad (2.29)$$

Similarly, the following can be deduced for the encoded output symbols,

$$\begin{aligned} p(w_i = 0; \mathbf{Y}_1^\tau) &= \sum_{(m^-, m) \in W_t^0} \sigma_t(m^-, m) \\ &= h_\sigma^t \sum_{(m^-, m) \in W_t^0} \alpha_{t-1}(m^-)\gamma_t(m^-, m)\beta_t(m), \\ &\qquad (t = 1, \ldots, \tau; i = n(t-1)+1, \ldots, nt) \end{aligned} \qquad (2.30)$$

where W_t^w is the set of transitions, $S_{t-1} = m^- \rightarrow S_t = m$, such that $w_i = w \in \{0, 1\}$ where the value of i lies between $n(t-1)+1$ and nt. Hence, the APP of the encoded symbol can be determined as

$$\begin{aligned} p(w_i = 0 | \mathbf{Y}_1^\tau) &= \frac{p(w_i = 0; \mathbf{Y}_1^\tau)}{p(\mathbf{Y}_1^\tau)} \\ &= \frac{\sum_{(m^-, m) \in W_t^0} \sigma_t(m^-, m)}{p(w_i = 0; \mathbf{Y}_1^\tau) + p(w_i = 1; \mathbf{Y}_1^\tau)} \\ &= \frac{\sum_{(m^-, m) \in W_t^0} \sigma_t(m^-, m)}{\sum_{(m^-, m) \in W_t^0} \sigma_t(m^-, m) + \sum_{(m^-, m) \in W_t^1} \sigma_t(m^-, m)}. \\ &\qquad (t = 1, \ldots, \tau; i = n(t-1)+1, \ldots, nt) \end{aligned} \qquad (2.31)$$

Note that the scaling factor, h_σ^t, in equation (2.27) does not have to be computed since it is cancelled away in the ratio of equation (2.31).

The performance of the BCJR algorithm in comparison with that of the VA is presented in

Figure 2.10. The simulation was carried on the convolutional code in the earlier example as shown in Figure 2.7. The size of each frame is $F = 128$ data information bits, giving $\tau = 130$. A total of 10,000 frames have been transmitted. The simulation results show that the performance of both algorithms is almost identical.

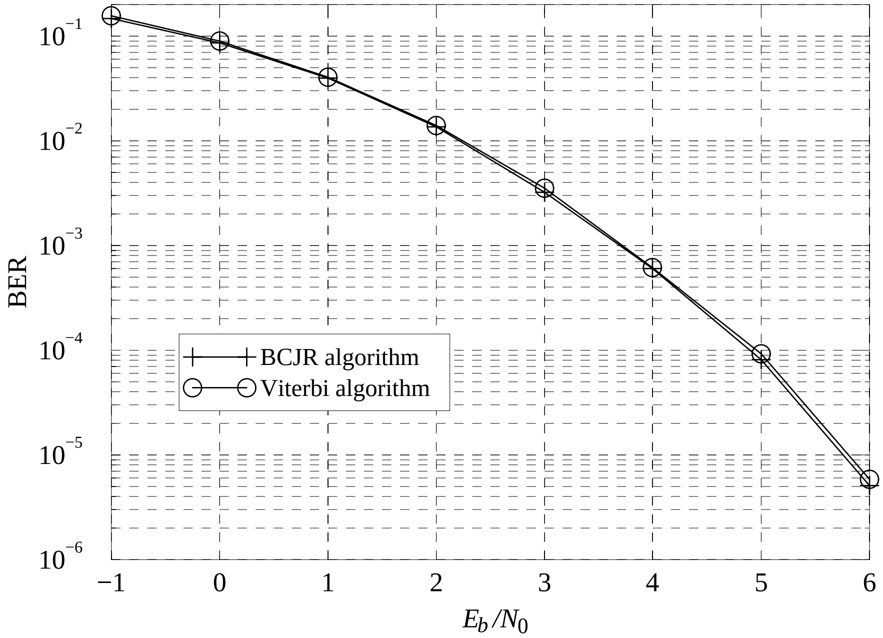

Figure 2.10: *Bit error rate (BER) as a function of E_b/N_0 for the convolutional code of code rate $R = 1/2$, constraint length $\kappa = 3$ and generators (7,5)*

The BCJR algorithm is more complex than the VA, even though the gain in performance is negligible. In most applications, the performance of both algorithms would be identical. Hence, until 1993, not only was the BCJR algorithm a less popular alternative in the decoding of convolutional codes, it was almost never applied in practical systems. However, recently, interest in the BCJR algorithm has been revived with the introduction of Turbo codes, in which the BCJR algorithm is essential for finding the APP of the data digits.

2.6.4 Turbo codes

Turbo codes were first proposed in 1993 by C. Berrou *et al.* [43]. This new coding scheme [44–48] achieves near capacity performance in the power limited region of the AWGN channel. The first Turbo code used a parallel concatenation of rate 1/2 convolutional encoders combined with iterative MAP decoding to achieve a BER of 10^{-5} at a signal-to-noise ratio (SNR) of only $E_b/N_0 = 0.7$ dB. The Turbo coding scheme comes within 0.5 dB of capacity at a BER of 10^{-5} as the Shannon limit is 0.2 dB. The codes are constructed by applying two or more component codes to different interleaved versions of the same information sequence. Due to its superior performance, this new coding technique has already been considered in IMT-2000. In order for this concatenated coding scheme to work properly, the decoding algorithm should not limit itself to passing hard decisions from the inner decoder to the outer decoder. Soft decisions need to be exchanged between the decoders, which is the main concept behind Turbo decoding, that is to iterate the soft decisions between decoders several times to produce better decisions [42, 49].

2.6.4.1 Turbo encoder

The component codes used in Turbo codes are referred to as recursive systematic convolutional (RSC) codes. This type of codes form the class of infinite impulse response (IIR) convolutional codes, which feed back previously encoded information bits continually to the encoder's input. The original form of convolutional codes is known as non-systematic convolutional code (NSC) and is generated by an encoder with a finite impulse response (FIR) as seen in Figure 2.7. At small E_b/N_0, the performance of the RSC is better than the NSC [43]. An example of a Turbo encoder is shown in Figure 2.11. The encoder consists of two identical rate 1/2 RSC encoders with constraint length 3. The RSC code is generated from an NSC encoder with a feedback loop and setting one of the outputs to the input information bit, d_t. If each RSC encoder is punctured to rate 2/3, the overall rate of the Turbo encoder is 1/2, otherwise the overall rate is 1/3. The interleaver between the two RSC encoders permutes the data in a pseudorandom fashion to ensure with high probability that the parity sequence, $w_{1,t}$, generated by the first encoder in response to the input sequence, d_t, is different from the parity sequence, $w_{2,t}$ generated by the second encoder in response to the interleaved input sequence.

2.6.4.2 Turbo decoder

This section discusses the MAP symbol estimator decoding, other methods of decoding can be found in [50]. Assume a Turbo code of rate 1/3, with $(d_t, w_{1,t}, w_{2,t})$ as the transmitted codeword at time t. Let $\mathbf{y}_i$ $(i = 0, 1, 2)$ be the sequence formed from the i-th noise-corrupted received symbol in each codeword. The sequences $\mathbf{y}_0$ and $\mathbf{y}_1$ are fed to decoder 1 since the noise free versions of these two sequences are the output of RSC encoder 1. Similarly, the

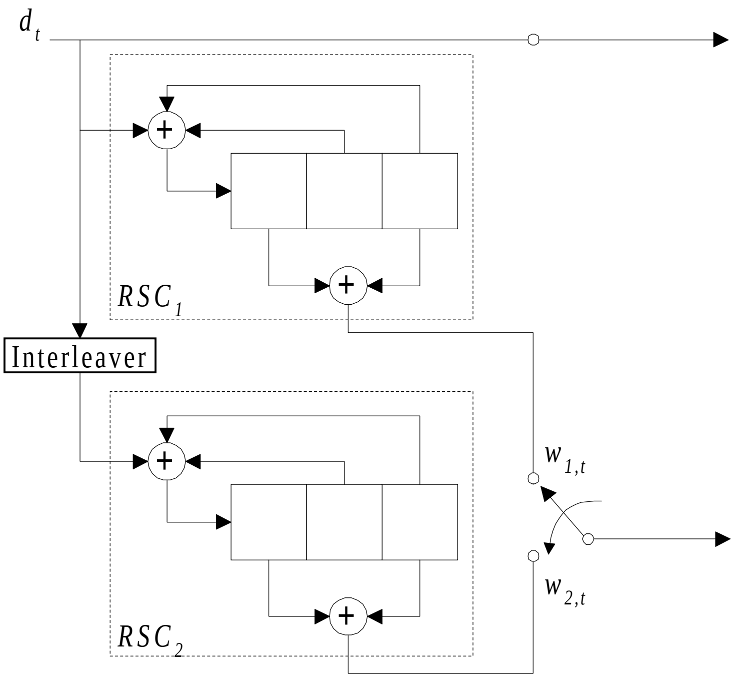

Figure 2.11: *The encoder of the Turbo code with rate $R = 1/2$ and constraint length $\kappa = 3$*

sequences $\mathbf{y}_0$ and $\mathbf{y}_2$ are passed to decoder 2 as shown in Figure 2.12.

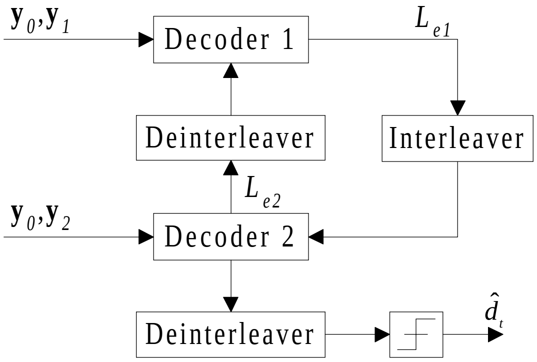

Figure 2.12: *The Turbo decoder with iterative feedback*

The optimal decision rule is the MAP rule:

$$\hat{d}_t = \arg \max_{d=\{0,1\}} \Pr(d_t = d|\mathbf{y}_0; \mathbf{y}_1; \mathbf{y}_2), \qquad (t = 1, \ldots, F) \tag{2.32}$$

The decision can also be based on the log APP ratio of decoder 1 and decoder 2, which can be written respectively as

$$\begin{aligned} L_1(d_t) &= \log\left(\frac{\Pr(d_t=1|\mathbf{y}_0;\mathbf{y}_1)}{\Pr(d_t=0|\mathbf{y}_0;\mathbf{y}_1)}\right) \\ &= \log\left(\frac{\Pr(\mathbf{y}_0;\mathbf{y}_1|d_t=1)}{\Pr(\mathbf{y}_0;\mathbf{y}_1|d_t=0)}\right)+\log\left(\frac{\Pr(d_t=1)}{\Pr(d_t=0)}\right) \quad (t=1,\ldots,F) \end{aligned} \tag{2.33}$$

and

$$L_2(d_t) = \log\left(\frac{\Pr(\mathbf{y}_0;\mathbf{y}_2|d_t=1)}{\Pr(\mathbf{y}_0;\mathbf{y}_2|d_t=0)}\right)+\log\left(\frac{\Pr(d_t=1)}{\Pr(d_t=0)}\right). \quad (t=1,\ldots,F) \tag{2.34}$$

The second term in equation (2.33) and equation (2.34) represents *a priori* information. Since $P(d_t=1)=P(d_t=0)$ generally, the *a priori* term is usually zero for conventional decoders. However, for iterative decoding scheme, decoder 1 receives extrinsic or soft information for each d_t from decoder 2 which serves as *a priori* information. Similarly, extrinsic information is also passed as *a priori* information from decoder 1 to decoder 2. This exchange of soft information improves performance provided that the information is not already available to the other decoder. This is achieved with the aid of the interleaver which permutes the data bits to the encoders such that they appear uncorrelated.

Although these ratios in equations (2.33) and (2.34), contain soft information regarding the uncoded information bit, d_t, they cannot be used as the *a priori* information because they contain information already available to the decoder. To extract the extrinsic information, the first term in equation (2.33) is split up as

$$\begin{aligned} \log\left(\frac{\Pr(\mathbf{y}_0;\mathbf{y}_1|d_t=1)}{\Pr(\mathbf{y}_0;\mathbf{y}_1|d_t=0)}\right) &= \log\left(\frac{\Pr(\mathbf{y}_0|d_t=1)}{\Pr(\mathbf{y}_0|d_t=0)}\right)+\log\left(\frac{\Pr(\mathbf{y}_1|d_t=1)}{\Pr(\mathbf{y}_1|d_t=0)}\right), \\ &\quad (t=1,\ldots,F) \end{aligned} \tag{2.35}$$

since $\mathbf{y}_0$ and $\mathbf{y}_1$ are uncorrelated. As the channel is memoryless, d_t only affects $y_{0,t}$ which is the t-th term in $\mathbf{y}_0$, and the first term in equation (2.35) simplifies to

$$L_{s,t} = \log\left(\frac{\Pr(\mathbf{y}_0|d_t=1)}{\Pr(\mathbf{y}_0|d_t=0)}\right) = \log\left(\frac{\Pr(y_{0,t}|d_t=1)}{\Pr(y_{0,t}|d_t=0)}\right). \quad (t=1,\ldots,F) \tag{2.36}$$

This term contains the systematic information since it does not depend on the encoded sequence, only on the systematic information bits. The second term in equation (2.35) is the sought extrinsic information of decoder 1, $L_{e1,t}$, since it involves $\mathbf{y}_1$ which is not available to decoder 2. Similarly, the extrinsic information, $L_{e2,t}$, of the decoder 2 can be derived. The extrinsic information of both decoders is therefore

$$L_{e1,t} = L_1(d_t) - L_{s,t} - L_{e2,t} \quad (t = 1, \ldots, F) \tag{2.37}$$

and

$$L_{e2,t} = L_2(d_t) - L_{s,t} - L_{e1,t}. \quad (t = 1, \ldots, F) \tag{2.38}$$

The log-APP ratios, $L_1(d_t)$ and $L_2(d_t)$, can be computed from the BCJR algorithm in equation (2.28) after some modifications[6]. One crucial change is that the *a priori* information obtained from the extrinsic information of the other decoder is used to bias the conditional probability, $\Pr(S_t = m|S_{t-1} = m^-)$, in equation (2.25).

At each iteration, the extrinsic information is recursively computed and exchanged between the decoders. The most recently computed values of the extrinsic information are applied in equations (2.37) and (2.38). When sufficient iterations have been completed, the decision rule is then

$$L_2(d_t) \underset{d_t=0}{\overset{d_t=1}{\gtrless}} 0. \quad (t = 1, \ldots, F) \tag{2.39}$$

[6] Soft output Viterbi algorithm can also be used for this purpose [51]

2.7 Summary

In this chapter, the basic principles of the spread spectrum technology have been covered. The application of the spread spectrum concepts to the multiple access system has been shown. The model of the downlink of the resulting DS-CDMA system, together with the mathematical models of the signals involved, has been defined. The transmitter structure and the multipath communication channel have also been described. The spreading codes used for spectrum spreading have been discussed with emphasis on random codes and orthogonal codes. The conventional detector for the DS-CDMA system has been detailed, and shown that performance is limited by MAI. The FEC channel coding has also been reviewed, and attention has been focused on convolutional codes including encoding and comparison between two methods of decoding, namely, the VA and the BCJR algorithm. It is shown that both techniques have similar performance although the latter is more complex. Finally, Turbo code which comprises two recursive systematic convolutional codes in parallel or serial concatenation, have been introduced.

Chapter 3
Multiuser receivers

This chapter discusses the various established multiuser receivers proposed for the DS-CDMA communications systems. The first section (3.1), describes the detrimental effect of multiple access interference (MAI) that exists in the systems due to the conventional signal detection method. One solution is to use multiuser detector (MUD) which is covered in Section 3.2. The next section (3.3) looks at the optimum multiuser detectors (MAP and MLSE) based on sequence detection and single-symbol detection. Due to the high complexity of the optimum MUD, sub-optimum MUDs are needed and Section 3.4 reviews some of these detectors which are divided into two classes, namely linear and non-linear. Section 3.5 introduces a new type of multiuser receiver for FEC coded DS-CDMA systems, known as iterative multiuser receivers. This receiver iterates between multiuser detection and channel decoding using soft inputs and soft outputs.

3.1 Multiple access interference

Multiple access interference is an impairment factor that limits the system's capacity and performance in DS-CDMA communications systems. Two main parameters contribute to MAI:

1. Cross-correlation levels (CCLs)

   The cross-correlation function between users j and k is given by $\Phi^R_{j,k}(z)$ in equation (2.12). For a synchronous system, the CCL of interest is represented by the coefficient of z^0. Usually, the CCLs between the users are not zero, even if orthogonal spreading codes are used, unless the system is perfectly synchronous and there is negligible multipath effect. The other terms in the cross-correlation function are the partial CCLs and are useful in the investigation of the asynchronous system. For the uplink where signals are received asynchronously, the situation is even more adverse. The random time offsets between signals make it impossible to design completely orthogonal code waveforms. Hence, even if the channel is ideal, all users still interfere with all other users due to the non-zero CCLs between the spreading codes of different users. With well-designed spreading codes, the CCLs are generally small. However, as the number of interferers increases, the contribution to MAI becomes substantial. One way of suppressing this interference is to increase the processing gain when the number of users is large. However, this increases bandwidth.

2. "Near-far" problem

   Due to the non-zero CCLs among the users, the power levels of the interferers directly affect the MAI. This is a secondary factor since it is only present if the CCLs are non-zero. In addition to the desired user's power, P_k, the power of the other users, P_j $(j \neq k)$, are also involved as observed in the MAI term of equation (2.12). If these power levels are much larger than the desired user's power, then the intended signal would be masked. For example, the signals of users nearer the receiver will be received at a higher power than those further away and hence these users tend to overshadow those weaker ones. This is known as the "near-far" effect. Even if they are at the same distance, "near-far" effect may still exist because some users may be received during a deep fade, or some users may be transmitting at higher or lower power levels than others (assuming no power control). Even if power control is employed, the overall power of the interference is still substantial when the number of users is large.

However, it should be emphasised that these problems are not inherent problems of the CDMA systems but of the conventional detector. In the next section, the multiuser detection is discussed in detail to show how they can mitigate the MAI.

If the negative effect that each user has on the others can be cancelled, significant capacity increase can be achieved. Various efforts have been made to mitigate the effect of MAI:

- Code Waveform Design

  As noted earlier that if the spreading codes are orthogonal to each other, their CCLs are all zeros and there would be no MAI terms in an ideal synchronous channel. This approach leads to the design of codes with very low cross-correlation properties. Unfortunately, most channels are not perfectly synchronous in practice, and thus it is impossible to design codes that can maintain orthogonality over all possible delays. The best that can be done is to look for codes that are have bounded CCLs for all possible delays such as Gold codes.

- Power Control

  The "near-far" effect can be counteracted by using strict power control where users are required to make adjustments to their transmitting power levels whenever necessary. When the receiver detects that the received signal is too strong or too weak, a feedback signal is sent to the transmitter to make appropriate adjustments. This method is known as closed loop power control. Another method, known as open loop power control, adjusts the power of the mobiles according to the inverse of the power levels received from the base station. This stringent power co-ordination between the transmitter and receiver adds to the complexity of both the transmitter and receiver. Currently, the power control is indispensable for a successful DS-CDMA system using the conventional receiver.

- Forward Error Correction (FEC) Codes

  Acceptable error rate performance at lower signal-to-interference ratio can be achieved with more powerful FEC coding.

- Sectored/Adaptive Antennas

  By directing the reception of the desired signal over a narrow angle range, the antenna enhances only the desired signal and attenuates the MAI from the other incident angles. The direction of the antenna can be fixed, as in the sectored antennas, or dynamically adjustable in the adaptive antennas. In the latter case, the antennas are adjusted to the direction of the desired user adaptively by utilising adaptive signal processing. This approach is not considered in this thesis.

- Multiuser detection

  This is another scheme proposed to overcome the MAI problem. There has been great interest in improving DS-CDMA detection through the use of multiuser detectors. In multiuser detection, code, timing and amplitude information of multiple users are jointly utilised to detect each individual user. Instead of treating the other users' signals as interference, all signals are collectively used for their mutual benefit by joint detection. It is believed that this scheme not only negates the need for a power control in the detector, but also offers significant additional benefits for the DS-CDMA systems. However, the drawback of the optimum MUD is its complexity so that sub-optimum approaches are being sought. There is a wide range of possible performance/complexity compromises. Presently, much research is aimed at finding an appropriate trade-off between complexity and performance.

3.2 Multiuser detection

In a cellular system, a number of mobiles communicate with their base station. Two frequency bands are needed in each cell because the uplink and downlink use different frequency bands if frequency division duplex (FDD) is used. However, this pair of frequency bands is reused in neighbouring cells, resulting in inter-cell interference which adds to the original intra-cell interference. If this inter-cell interference is removed, capacity would increase. To include this interference in the multiuser detection, the spreading codes of the users in the neighbouring cells have to be known by the receiver in the central cell as well. The capacity gain of a cellular system is bounded if the inter-cell MAI is not mitigated. In the work on multiuser receivers presented here, only intra-cell MAI is considered.

Multiuser detection has a number of potential benefits:

- Improvement in capacity

  With the removal of intra-cell MAI alone, the improvement in capacity is already significant [52]. If the inter-cell interference is removed as well, the capacity can be further improved. In satellite personal communications, where the inter-cell MAI is insignificant, multiuser detection for one cell alone is sufficient to have a large capacity gain.

- Efficient uplink spectrum utilisation

  The performance improvement, as a result of removing MAI from received signals by using the multiuser detection, allows the system to operate at a lower processing gain. This reduces the bandwidth required; the extra bandwidth could then be used to improve capacity or to support higher data rates.

- Relax the need for power control

  Stringent power control is usually believed to be indispensable in CDMA communications systems. However, with the introduction of multiuser detection, the impact of other users' power on the intended user is much reduced. Hence, less precision is needed in controlling the transmitted power levels of the mobiles, but a modicum of power control is still necessary for traditional method of acquisition and synchronisation. Although the implementation of multiuser detection adds to the complexity, it relaxes the need of power control for both the base station and the mobiles.

- More Efficient power utilisation

Reduction in MAI may result in a lower transmit power.

Although multiuser detection can solve this MAI problem and increase capacity of the system, the optimum multiuser detection scheme is far too computationally intensive and has a variable decoding delay which is unacceptable in many applications. Hence, there is a need for sub-optimum receivers which perform reliably under MAI and have a reasonable computational complexity to ensure practical implementation. The fundamental interest here is to maximise the advantages of the multiuser detection and simplify its implementation.

3.3 Optimum multiuser detector

The maximum-likelihood sequence estimation (MLSE) MUD proposed by Verdu [53] is the first MUD, and it is often claimed in literature to be the optimum MUD. However, it is not truly optimum, even though its performance is near-optimum. [54] has divided MUDs into sequence detection and single-symbol detection, MAP and MLSE, constrained and unconstrained detections. The sequence detection based MUD finds that particular data sequence which was most probably transmitted according to some hypothesis conditions. Single symbol detection based MUD, on the other hand, finds the particular symbol which was most probably transmitted at a certain time according to some hypothesis criteria. The constrained approach restricts the hypothesis to the set of alphabet of the symbol while the unconstrained approach makes an estimate without any restriction and then quantises the estimate to conform to the alphabet. Only the constrained single-symbol detection MUD that minimises symbol error probability is the optimum MUD. The MLSE MUD proposed by Verdu is the optimum constrained sequence-based MUD under certain conditions. Unconstrained detection forms the class of linear MUDs, from which sprouts many non-linear MUDs. This section focuses on constrained detection, starting with sequence-based detection in Section 3.3.1 and followed by single-symbol based detection in Section 3.3.2. The next section focuses on unconstrained sub-optimal MUDs, which includes linear MUDs and non-linear MUDs in Section 3.4.1 and Section 3.4.2 respectively.

3.3.1 Sequence-based constrained detection

The optimum sequence-based (SB) detection finds the sequence that has minimal sequence error probability. Minimising error probability implies maximising the probability of correct decisions, and the probability of a correct decision is

$$\Pr(\text{correct decision}) = \sum_{\mathbf{Y}(z)\in\Omega} \Pr(\text{correct decision}|\mathbf{Y}(z))p(\mathbf{Y}(z)), \tag{3.1}$$

where Ω is the set consisting of all possible observation $\mathbf{Y}(z)$. Since $p(\mathbf{Y}(z)) \geq 0$, the probability of a correct decision is maximised if the decision for each observation maximises $\Pr(\text{correct decision}|\mathbf{Y}(z))$. This is equivalent to the maximum *a-posteriori* probability (MAP) detector which maximises the *a-posteriori* probability, $\Pr(\mathbf{B}(z)|\mathbf{Y}(z))$, since

$$\Pr(\text{correct decision}|\mathbf{Y}(z)) = \Pr(\mathbf{B}(z)|\mathbf{Y}(z)). \tag{3.2}$$

The MAP decision is thus

$$\hat{\mathbf{B}}(z) \quad = \quad \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{FK}} \Pr(\mathbf{B}(z)|\mathbf{Y}(z)), \tag{3.3}$$

and can be expressed in terms of likelihood using Bayes' rule

$$\begin{aligned} \hat{\mathbf{B}}(z) &= \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{FK}} \frac{p(\mathbf{Y}(z)|\mathbf{B}(z)) \Pr(\mathbf{B}(z))}{p(\mathbf{Y}(z))} \\ &= \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{FK}} p(\mathbf{Y}(z)|\mathbf{B}(z)) \Pr(\mathbf{B}(z)). \end{aligned} \tag{3.4}$$

The denominator, $p(\mathbf{Y}(z))$, is independent of the decision, hence it can be dropped. If all $\mathbf{B}(z)$ are equiprobable prior ($p(\mathbf{B}(z)) = 1/(FK)$), then only the likelihood, $p(\mathbf{Y}(z)|\mathbf{B}(z))$, needs to be considered. Consequently, the maximum likelihood decision criterion is obtained as

$$\hat{\mathbf{B}}(z) \quad = \quad \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{FK}} p(\mathbf{Y}(z)|\mathbf{B}(z)). \tag{3.5}$$

The likelihood function can be computed as

$$p(\mathbf{Y}(z)|\mathbf{B}(z)) \quad = \quad \frac{1}{(\pi N_0)^{(FG+L-1)/2}} \exp\left(-\frac{||Y(z)-X(z)||^2}{N_0}\right), \tag{3.6}$$

where

$$||Y(z)-X(z)||^2 = \sum_{i=0}^{FG+L-2} (y_i - x_i)^2, \tag{3.7}$$

given that y_i and x_i are the coefficients of the z^{-i} terms in $Y(z)$ and $X(z)$ respectively. Maximising the exponent is equivalent to minimising the Euclidean distance, $\|Y(z) - X(z)\|^2$. In other words, the MLSE MUD searches through all the possible input sequences and selects the one closest to the received sequence based on the Euclidean distance [55]. Mathematically, it is

$$\begin{aligned}
\hat{\mathbf{B}}(z) &= \arg \min_{\mathbf{B}(z) \in \{-1,+1\}^{FK}} \|Y(z) - X(z)\|^2 & (3.8)\\
&= \arg \min_{\mathbf{B}(z) \in \{-1,+1\}^{FK}} \left(\|Y(z)\|^2 + \|X(z)\|^2 - 2\langle Y(z) \centerdot X(z)\rangle\right) \\
&= \arg \max_{\mathbf{B}(z) \in \{-1,+1\}^{FK}} \left(2\langle Y(z) \centerdot X(z)\rangle - \|X(z)\|^2\right), & (3.9)
\end{aligned}$$

where

$$\langle Y(z) \centerdot X(z)\rangle = \sum_{i=0}^{FG+L-2} y_i x_i. \tag{3.10}$$

The term $\|Y(z)\|^2$ is common to all $\mathbf{B}(z)$, so it is not considered. The resultant MLSE MUD is the one proposed by Verdu.

Since the receiver must solve a nondeterministic polynomial-time complete combinatorial optimum algorithm [56], no algorithm polynomial in K is known for optimum multiuser detection. A brute-force exhaustive search over the 2^{FK} possible sequences is clearly impractical for typical message sizes and numbers of users. Fortunately, $\hat{\mathbf{B}}(z)$ has the right structure to employ the VA that results in significantly better complexity. This method is similar to the use of the VA to implement equalisation in intersymbol interference channel [26, 39, 57, 58]. Implementing the MLSE using the VA in the asynchronous uplink, for both the AWGN channel and the multipath channel[1] have been shown in [53, 60] and [61] respectively. Although the complexity is reduced to $O(2^K)$ for an AWGN asynchronous channel, it is still exponential in the number of users. For the AWGN channel, the front-end is the matched filter bank, and the back-end is the VA that performs the likelihood ratio comparisons. The output of the matched filter bank provides sufficient statistics to perform the VA at the symbol-level. In the multipath uplink, the front-end is replaced by the RAKE receiver which despreads and multipath-combines the signal. In the downlink, where the system is synchronous, the VA can also be employed when there is multipath in the channel. This has not been shown in literature, hence the proof is given here which shows how the MLSE can be implemented with a RAKE receiver front-end without

[1] single-path fading is discussed in [59]

any degradation in performance. It should be emphasised here that the output of the RAKE receiver should not be viewed as a set of pre-processed signals but a set sufficient statistics.

Consider the first term in equation (3.9)

$$\begin{aligned}\langle Y(z)\,\mathbf{.}\,X(z)\rangle &= \langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)\mathbf{P}\mathbf{b}(z)\rangle \\ &= \sum_{f=0}^{F-1}\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)\mathbf{P}\mathbf{b}_f z^{-Gf}\rangle \\ &= \sum_{f=0}^{F-1}\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle\mathbf{P}\mathbf{b}_f, \end{aligned} \tag{3.11}$$

where $\mathbf{b}_f = [b_{1,f},\ldots,b_{K,f}]^T$ is the f-th input symbol vector and $\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle$ is actually the output of the RAKE matched filter bank in the f-th message interval. Even though each value of $\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle$ is not a sufficient statistics for the detection of $\mathbf{b}_f$, the entire sequence of outputs of the RAKE receiver is a sufficient statistics for the selection of the most likely sequence, $\mathbf{B}(z)$.

Next, the second term is expanded as,

$$\begin{aligned}\|X(z)\|^2 &= \langle X(z)\,\mathbf{.}\,X(z)\rangle \\ &= \langle \mathbf{B}^T(z)\mathbf{P}\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z)\rangle \\ &= \left\langle \sum_{j=0}^{F-1}\mathbf{b}_j^T z^{-Gj}\mathbf{P}\mathbf{R}(z)\,\mathbf{.}\,\sum_{f=0}^{F-1}\mathbf{R}^T(z)\mathbf{P}\mathbf{b}_f z^{-Gf}\right\rangle \\ &= \sum_{f=0}^{F-1}\left[\mathbf{b}_f^T\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}_f\right. \\ &\qquad \left. +2\mathbf{b}_{f-1}^T\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}\mathbf{b}_f\right], \end{aligned} \tag{3.12}$$

making use of the fact that $\mathbf{b}_f = \mathbf{0}$ for $f < 0$ and

$$\langle \mathbf{R}(z)z^{-Gj}\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle = \begin{cases}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)\rangle & \text{if } j = f \\ 0 & \text{if } |j-f| > 1 \\ \langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle & \text{if } |j-f| = 1\end{cases}\ . \tag{3.13}$$

The second condition is due to the fact that all the components of $\mathbf{R}(z)$ are of length $G+L-1 < 2G$, or simply, the symbols that are more than one symbol interval away from each other do not interfere with each other.

Hence, the decision rule can be expressed as a sum of metrics,

$$\hat{\mathbf{B}}(z) = \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{FK}} \sum_{f=0}^{F-1} \lambda_f(\mathbf{b}_{f-1}, \mathbf{b}_f), \tag{3.14}$$

with the transition metric given as

$$\begin{aligned} \lambda_f(\mathbf{b}_{f-1}, \mathbf{b}_f) \quad = \quad & 2\langle Y(z) \bullet \mathbf{R}^T(z) z^{-Gf} \rangle \mathbf{P}\mathbf{b}_f \\ & -\mathbf{b}_f^T \mathbf{P} \langle \mathbf{R}(z) \bullet \mathbf{R}^T(z) \rangle \mathbf{P}\mathbf{b}_f \\ & -2\mathbf{b}_{f-1}^T \mathbf{P} \langle \mathbf{R}(z) \bullet \mathbf{R}^T(z) z^{-G} \rangle \mathbf{P}\mathbf{b}_f. \end{aligned} \tag{3.15}$$

Each metric can be computed using the current RAKE receiver output, a subset of control sequences, $\mathbf{b}_f$, and the set of transmitted sequences, $\mathbf{b}_{f-1}$. Therefore, this is a discrete time deterministic control problem with additive cost and finite input and state spaces, so the VA can be readily applied. It is observed that the transition metric in equation (3.15) involves two symbol sequences, $\mathbf{b}_{f-1}$ and $\mathbf{b}_f$, hence the complexity is $O(2^{2K})$. The block diagram of the MLSE MUD is given in Figure 3.1.

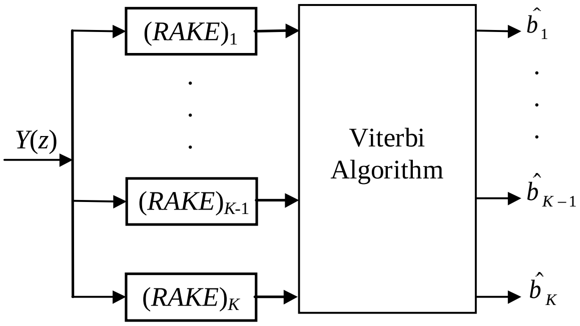

Figure 3.1: *The MLSE MUD for a downlink multipath channel*

If there is no multipath in the channel, then $R_k(z) = S_k(z)\,\forall(k = 1, \ldots, K)$ and the RAKE receiver reduces to the conventional matched filter bank. Since $S_k(z)$ is of length G, there is no interference from the other symbol intervals. The last term in equation (3.15) becomes 0 since $\langle \mathbf{R}(z) \bullet \mathbf{R}^T(z) z^{-G} \rangle = 0$. Hence, the transition metric, $\lambda_f(\bullet)$, does not involve any other symbol interval and a decision can be made for the symbol interval in question based on these metrics. Consequently, the complexity becomes $O(2^K)$ and the VA is not necessary for signal detection.

3.3.2 Single-symbol based constrained detection

The single-symbol based (SSB) detector makes decision on a symbol-by-symbol basis. The transmitted symbols are not taken jointly as a single entity anymore. The optimum single-symbol based constrained MUD aims to find the symbol alphabet that has minimum symbol error probability. Again, minimising error probability is equivalent to maximising the *a posteriori* probability which is the hypothesis criterion of the SSB MAP MUD. Mathematically, the decision criterion is

$$\begin{aligned}\hat{b}_{k,f} &= \arg\max_{b\in\{-1,+1\}} \Pr(b_{k,f}=b|\mathbf{Y}(z)) \\ &= \arg\max_{b\in\{-1,+1\}} \sum_{B^b_{k,f}} \Pr(\mathbf{B}(z)|\mathbf{Y}(z)) \\ &= \arg\max_{b\in\{-1,+1\}} \sum_{B^b_{k,f}} \exp\left(-\frac{\|Y(z)-X(z)\|^2}{N_0}\right)\Pr(\mathbf{B}(z)), \\ &\qquad (k=1,\ldots,K; f=0,\ldots,F-1) \end{aligned} \tag{3.16}$$

where $B^b_{k,f}$ is the set consisting of all possible values of sequence, $\mathbf{B}(z)$, with $b_{k,f} = b$. The Bayes rule and equation (3.6) have been applied in the above derivation. The MAP criterion exploits the *a priori* probability of $\mathbf{B}(z)$, $\Pr(\mathbf{B}(z))$, hence it is well suited for iterative detection/decoding receivers which will be discussed in more detail in Section 3.5 and Chapter 7. If the symbols are all equiprobable, the SSB MAP decision criterion simplifies to a SSB maximum likelihood decision criterion, and is obtained as

$$\begin{aligned}\hat{b}_{k,f} &= \arg\max_{b\in\{-1,+1\}} \sum_{B^b_{k,f}} \exp\left(-\frac{\|Y(z)-X(z)\|^2}{N_0}\right). \\ &\qquad (k=1,\ldots,K; f=0,\ldots,F-1)\end{aligned} \tag{3.17}$$

For simplicity, now consider the synchronous and single-path case where each symbol interval can be considered separately, the decision rule of the k-th user in each interval can be rewritten as

$$\sum_{B_k^{-1}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{N_0}\right) \underset{b_k=1}{\overset{b_k=-1}{\gtrless}} \sum_{B_k^1} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{N_0}\right), \quad (k=1,\ldots,K) \tag{3.18}$$

or

$$\hat{b}_k = \operatorname{sgn}\left[\sum_{B_k^1} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{N_0}\right) - \sum_{B_k^{-1}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{N_0}\right)\right], \quad (k = 1, \ldots, K) \tag{3.19}$$

where $\mathbf{y}$ and $\mathbf{x}$ are the noisy and noise free received vectors in the symbol interval respectively, and the signum function returns the sign of its operand. This gives the single-user Bayesian decision rule since the SSB considers each symbol and thereby each user individually. This detector is the optimum one-shot detector for decentralised detection of a single-user in an uncoded DS-CDMA system as shown in [62]. A neural network MUD based on the SSB technique is the radial basis function (RBF) detector [63]. An adaptive approach to the RBF has been shown in [64]. The maximum likelihood RBF decision rule is given as

$$\hat{b}_k = \operatorname{sgn}\left[\sum_{i=1}^{2^K} W_i \exp\left(-\frac{\|\mathbf{y}-\mathbf{c}_i\|^2}{N_0}\right)\right], \qquad (k = 1, \ldots, K) \tag{3.20}$$

where: $\mathbf{c}_i$ is the i-th centre, which is one of the 2^K possible noiseless transmitted symbol sequence in an interval,

W_i is the weight with a value equal to b_k associated with $\mathbf{c}_i$ for independent data sequence and equiprobable. Otherwise, the probability of this centre has to be incorporated into the weight which yields a MAP RBF.

For the AWGN channel scenario, the complexity of the RBF is 2^K per symbol as shown above. However, for the multipath channel, the complexity of the detector grows with a factor of 2^{3K} per symbol as shown in [63]. A simplified RBF has been proposed in [65, 66].

Despite the excellent performance of the constrained detector, it is prohibitively complex and impossible to implement practically in large user populations. A large amount of storage and an intolerable delay may be needed since the convergence of the survival paths in the VA might be slow. A centralised structure which demodulates all signals is used, so single-user implementation is difficult. However, this does not mean that this detector is not worth analysing. Various approximations can be made to reduce its complexity such as [67] where only a fixed number of interferers is considered such that complexity is maintained at a practical level. Furthermore, the VA can also be implemented with reduced states [68] and yet achieves near-optimum performance, for example sequential decoders [69] such as the stack algorithm [58] and the Fano algorithm [33, 58].

3.4 Sub-optimum multiuser detectors

Due to the poor performance of the conventional detector and the high complexity of the optimum multiuser detector, there is a need for sub-optimum schemes to bridge the gap. Several sub-optimum detectors have been proposed, and they can be classified into two major categories, linear and non-linear detectors. For a concise and yet detailed summary of the various detectors it is best to limit the discussion to synchronous single-path channel. In this special case, each symbol duration can be analysed separately by considering just the z^0 coefficients of all terms, so equation (2.15) reduces to

$$\mathbf{u} = \Phi\mathbf{P}\mathbf{b} + \mathbf{n}, \tag{3.21}$$

where: $\mathbf{u}$ is the sampled output vector of the matched filter bank,

Φ is the correlation matrix,

$\mathbf{n}$ is the vector of the noise which has been spreaded by the matched filter.

3.4.1 Linear multiuser detectors

In the linear detector, a linear transformation, usually in the form of a matrix filter, is applied to the output of a matched filter bank. In other words, a decision for each user can be made based on the vector $\mathbf{Tu}$, where $\mathbf{T}$ is the linear transformation. The transformation operates without restriction on the symbols, the symbol alphabet is obtained by passing the transformed signal through a decision-making device such as the signum function. Hence, it is classified as an unconstrained detection. Two popular linear detectors [70] are the decorrelating detector (DD) [71, 72] and the minimum mean-square error (MMSE) detector [73]. It is shown in [54] that the DD and MMSE are the unconstrained MLSE [74] and MAP detectors respectively. Other linear MUDs can be found in [75–77].

3.4.1.1 Decorrelator Detector

The DD transforms the soft output of the conventional detector to decorrelate the users such that each user becomes independent of the other users. Considering equation (3.21), the objective of multiuser detection is to recover the input data vector $\mathbf{b}$ given the output vector $\mathbf{u}$. Inspecting this equation immediately suggests a method to solve for $\mathbf{b}$. It is obvious that the transformation matrix is $\mathbf{T} = \Phi^{-1}$, which can be obtained by inverting Φ. This is analogous to a zero-forcing equaliser for the ISI channel.

The resulting output vector becomes

$$\Phi^{-1}\mathbf{u} = \mathbf{P}\mathbf{b} + \Phi^{-1}\mathbf{n}. \tag{3.22}$$

Since $\mathbf{P}\mathbf{b}$ does not contain any cross correlation terms, MAI is completely eliminated. Moreover, as $\mathbf{P}$ is a matrix with all non-diagonal elements equal to zero, each element of vector $\mathbf{P}\mathbf{b}$ does not involve the other users' power. Hence, the "near-far" problem is also solved and the detector performance is independent of the power levels of the interfering users, resulting in the optimum "near-far" resistance. Furthermore, the DD does not require the knowledge of the users' power levels. Its computational complexity, $O(K)$, is significantly lower than the optimum detector. The per-symbol complexity is linear in the number of users, excluding the costs of recomputation of the inverse mapping. When the users' power levels are unknown, the DD is the maximum likelihood approach for the worst case MAI scenario. Moreover, its transformation matrix can be split up into rows, allowing decentralised implementation for single-user detector.

However, the new Gaussian noise vector, $\Phi^{-1}\mathbf{n}$, which has autocorrelation matrix of $\sigma^2\Phi^{-1}$, contains larger variances than the original AWGN. From equation (3.22), the error probability of the k-th user can be easily deduced as

$$P_k^{DD}(E) = Q\left(\sqrt{\frac{P_k}{N_o\Phi_{k,k}^{-1}}}\right), \tag{3.23}$$

where $\Phi_{k,k}^{-1}$ is the (k,k)-th element of matrix Φ^{-1}. From equation (3.23), it is noted that the error probability increases and performance degrades as the CCL increases. Even though the DD is independent of other users' power, it is directly affected by the CCLs. This creates a gap between the single-user bound and the DD's performance. In fact, if all the interferers are very weak, the conventional receiver can even outperform it [71]. This is because the noise enhancement is greater than the weak MAI. Its most significant disadvantage is in fact the computations involved in inverting matrix Φ. This operation is difficult to perform in real time. For synchronous system, the dimensions of matrix Φ is dependent on the number of users. Thus, computation is somewhat simplified and the inversion is only performed on a $K \times K$ matrix. However, in an asynchronous system, the size of this matrix is also dependent on the message length which could be very large. Hence, the matrix inversion becomes very difficult to perform in real time especially when the active user population varies rapidly. In addition, signals arrived at the receiver in multiple paths and if each path is treated as a separate user, then the dimension of the correlation matrix increases.

A windowing method is used in [78] to reduce long delay of the DD due to the large message length. [79] suggests an adaptive approach to the DD. The DD for multipath fading channel has been addressed in [80].

3.4.1.2 Minimum mean-square error detector

Similar to temporal equalisation, where the zero-forcing equaliser enhances noise and the MMSE equaliser is used instead, the MMSE MUD [81, 82] also has the analogous feature of suppressing noise enhancement. The matrix $\mathbf{T}$ is chosen to minimise the mean-square error between the transmitted and the training signal or the received signal (decision-based),

$$\mathbf{T} = \arg \min_{\mathbf{T} \in \Re^{K \times K}} E[\|\mathbf{Pb} - \mathbf{Tu}\|^2]. \tag{3.24}$$

The transformation matrix of the MMSE is [26, 54] $\mathbf{T} = (\Phi + \sigma^2 \mathbf{I}_K)^{-1}$ where $\mathbf{I}_K$ is a $K \times K$ identity matrix. Notice that it differs from the DD by the term $\sigma^2 \mathbf{I}_K$, which takes the noise term into account such that if MAI is small, the error is minimised by putting more emphasis on the noise. Hence, noise enhancement is reduced. Note that if noise is very small ($\sigma^2 \to 0$), the MMSE detector approaches the DD and thus it also achieves optimum “near-far” resistance. On the other hand, if MAI is small as compared to noise, MMSE detector approaches the conventional detector. Two papers [83, 84] show that there are other linear detectors that have performance between that of the DD and MMSE. However, noise power is required for the MMSE which means that the received power must be tracked. Fortunately, this detector lends itself easily to adaptive implementation[2] and can be implemented with single-user FIR filter [87] without the need of the matched filter front-end. The bank of optimum FIR filters or Wiener filters for all the users, if the spreading codes and noise variance are known, is given by the $K \times G$ matrix,

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^T[\mathbf{Q}\Lambda\mathbf{Q}^T + \sigma^2\mathbf{I}_G]^{-1}, \tag{3.25}$$

where: $\mathbf{Q}$ is a $G \times K$ matrix with each column representing the spreading codes of the users,

$\Lambda = \mathbf{P}^2$ is a $K \times K$ diagonal matrix.

In a multipath environment, the length of the FIR filter is $G + L - 1$ instead of G. This is to take into account the residual effect of the symbol spilling into the next symbol interval. The dimension of matrix $\mathbf{Q}$ becomes $(G + L - 1) \times K$ with the addition of the first $L - 1$ chips of the spreading codes, and the identity matrix is $\mathbf{I}_{G+L-1}$. With adaptive implementation, most information such as received power, interferers' spreading codes and noise variance, needs not be known, but some form of training is required [73]. A blind MUD has been suggested in [23] which minimises the output energy instead of the mean-square error. However, simple adaptive algorithms such as least mean square (LMS) are too slow to keep up with the rapid variation of the channel.

3.4.2 Non-linear multiuser detectors

Although linear detectors have lower complexity than the optimum detector, the linear structure often limits its performance. Hence, non-linear approaches have also been extensively investigated. Other than the MLSE and MAP, there are a number of other non-linear MUDs. The non-linear sub-optimum detectors usually utilise feedback of some decisions to reduce MAI in the received signals. They are also known as subtractive interference cancellers, since they subtract the interference away by regenerating the interference using known or decoded information. Three basic classes of the subtractive interference cancellers [22] are interference cancellers (ICs), multistage detectors (MSDs) and decision-feedback detectors (DFDs). Other non-linear detectors include [88, 89].

[2] Adaptive techniques for other MUDs are discussed in [85, 86]

3.4.2.1 Interference cancellers

From equation (2.8), it is observed that the noiseless received signal consists of $K-1$ interfering signals. If the power and symbols of all the interfering users are known, they can be cancelled away by regenerating the MAI term provided the interferers' spreading codes are known. The joint value of the power and symbol information can be obtained easily from the output of the conventional detector. This is a soft decision method [90] since this value is passed to the next stage without making any decision. A hard decision scheme can also be used where the power has to be estimated and the symbols information is deduced from the hard decisions of the correlator outputs [91–93]. The idea of this scheme is to first decode the strongest user's transmitted data and to estimate the received power. Using this user's spreading code, the transmitted signal of this particular user can be regenerated provided the decoded symbol and estimated power are accurate. This regenerated signal can then be cancelled away from the received signal resulting in a purer signal which can be used to decode the next strongest user. This procedure is repeated until all the users are decoded. For the best performance, the users are arranged in decreasing power order because the strongest user gives the most accurate estimate and the most negative effect can be cancelled away first.

However, this type of interference canceller (IC) only works best when the imbalance in power is large, and may even perform worse than the decorrelator under equal power condition. Reordering of the users is needed whenever the received power changes significantly as the wrong order results in drastic performance loss. Finally, the successive scheme must operate fast enough such that intolerable delay is not introduced.

Due to the large delay if each user's interference is cancelled successively, parallel interference cancellation is more suitable. Figure 3.2 shows a two-stage parallel interference cancellation which operates on all the users simultaneously. The figure shows that interference cancellation is performed on the signal before matched filtering. It is also possible to carry out interference cancellation on the matched filter bank outputs.

However, the decisions of the weak users are made in the first stage without cancelling the strong MAI first. These decisions will not be reliable and if feedback to the second stage, the result may not be beneficial. If the first stage is replaced with a more reliable detector such as the DD, then performance is expected to improve. In a power controlled channel, the IC performs the best since the decisions of the first stage are more reliable. The number of stages is usually

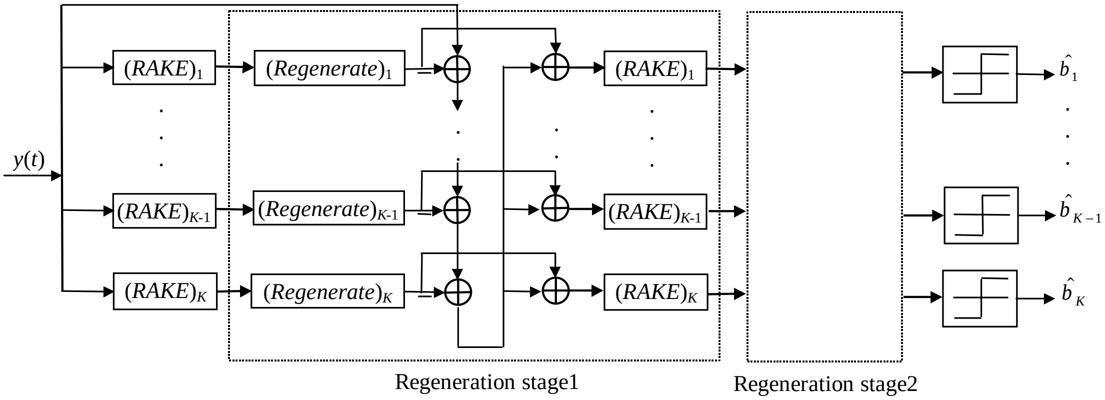

Figure 3.2: *A parallel interference canceller*

limited to two due to delay constraints. Different options for the first and subsequent stages are also possible, for example a wiener filtering first stage followed by parallel cancellation has been proposed in [94]. Other parallel ICs can be found in [95–98].

3.4.2.2 Multistage detector

This detector [99] is somewhat similar to the parallel IC in that the detection is in parallel instead of in series and it operates on the despreaded signal. Instead of using equation (3.9), this detector uses (for synchronous case) the following criterion to make decision at the n-th stage,

$$\hat{b}_k(n) \quad = \quad \arg \max_{\substack{b_k \in \{-1,+1\} \\ b_j = \hat{b}_j(n-1), j \neq k}} (2\mathbf{u}^T\mathbf{P}\mathbf{b} - \mathbf{b}^T\mathbf{P}\Phi\mathbf{P}\mathbf{b}). \tag{3.26}$$

Note that the MSD makes use of the decisions of the interferers made at the $(n-1)$-th stage. For the case of BPSK, instead of calculating 2^K metrics, only $2K$ metrics are calculated at each stage here, hence the complexity becomes linear with the number of users. The initial decision for the first stage can be obtained using simply the conventional detector.

The shortcomings and advantages of the parallel IC also apply to the MSD since they have similar architectures.

3.4.2.3 Decision-feedback detector

This form of detectors is analogous to the decision-feedback equalisers in ISI channels [100]. It consists of two matrix transformations: a feedforward filter and a feedback filter. Certain decisions are made based on the output of the feedforward filter and these decisions are fed back via the feedback filter to carry out cancellation at the output of the feedforward filter. The users are decoded successively in the order similar to the successive IC.

The first DFD MUD proposed is the decorrelating (zero-forcing) decision-feedback detector (DDFD). The feedforward section of this detector is similar to the DD except that the matrix filter is $(\mathbf{F}^T)^{-1}$ instead of Φ^{-1} where $\Phi = \mathbf{F}^T\mathbf{F}$ and $\mathbf{F}$ is a lower triangle matrix obtained by Cholesky decomposition algorithm. After passing through the matrix filter, the output vector is $(\mathbf{F}^T)^{-1}\mathbf{u} = \mathbf{FPb} + \mathbf{n}$. As $\mathbf{n}$ is a white Gaussian vector, the transformation matrix is actually a whitening filter and noise enhancement is thus eliminated. As $\mathbf{F}$ is a lower triangle matrix, the first term of the output vector, $(\mathbf{F}^T)^{-1}\mathbf{u}$, contains only the first user's information and no other MAI terms are involved. Immediately a decision for the first user can be made. If this decision is fed to the second user, it can be used to subtract away the interference term of the first user. Thus, the remainder consists purely of the second user information plus noise, and a decision can then be made. Therefore, by making use of previous decisions all users can be decoded without MAI. As each user is dependent of previous users' decisions, these previous decisions must be reliable in order to obtain an accurate overall result. Hence, a sorter is usually placed before the matrix filter to sort the users in descending order of power, for the same reasons as described in the successive IC section.

The DDFD performance may be desirable but there are certain drawbacks, such as performance is good only under large imbalances of power, correct order is very crucial and the strongest user does not benefit from the scheme at all. An attempt has been made to improve the DDFD and the resultant detector is called improved DDFD or IDDFD [101]. This scheme gives performance between the DDFD and the MLSE detector depending on the complexity. The idea of the IDDFD is to perform a limited tree search [102] instead of making a hard decision at each stage. If decision making is deferred until the final stage, then the optimum performance is obtained. Instead of keeping all paths (MLSE) or 1 path (DDFD), a pre-assigned number of them is retained. However, complexity increases by a factor equivalent to the number of paths kept.

[103] looks into two other shortcomings of the DDFD, firstly, computation of the Cholesky decomposition and the inverse whitening filter is difficult and secondly, all parameters need to be updated when new users enter the system or existing users leave the system or the users are re-ordered, resulting in a complicated updating procedure. The suggested solution is to combine the matched filter bank and the whitening filter into a whitening matched filter bank. The new filter can be computed by applying a Gram-Schmidt orthogonalising procedure to the spreading codes. Note that this new filter is not merely multiplying the whitening filter with the matched filter bank. Each active user uses the same set of parameters regardless of the incoming and outgoing users.

3.4.3 Discussions

The DD, MMSE, MSD and IC are compared in [104]. All of them give significant performance improvement over the conventional detector and even close to the optimum detector under certain operating environment. Generally, all non-linear detectors require knowledge of the received power. MSDs have the highest complexity because they require several stages of decisions. Though the successive interference canceller is the simplest (less hardware), it is likely to introduce intolerable delay since this detector generates the precorrelated waveform which means that correlation for each user can only be done after the previous users have been decoded. This also results in a large memory requirement to store all the chip values. Although the DFD can correlate the received signal with all the users spreading codes first to produce the post-correlation vector, it requires more computation since it is necessary to regenerate a different MAI term for each of the other users.

So far most work on sub-optimum MUDs usually starts from the linear detectors and then try to improve performance by introducing non-linearity into the receivers in the form of feedback. For example, the IC, MSD, DFD all requires a linear front-end such as DD, MMSE or simply the matched filter bank. However, introducing non-linearity also increases complexity significantly. The approach taken in the thesis is to look at the problem from another perspective. The aim is to begin with the optimum detector and try to simplify the complexity while retaining the superior performance. There has not been much research in this area as the optimum detector with an exponential complexity in the number of users has always been regarded as useless and impracticable [105]. However, in system with small number of users, the optimum receiver is ideal. One of the IMT2000 standards, which is a hybrid combination of the CDMA and TDMA system, the spreading gain of the codes is only 16 and the average number of operating users is only 8. The optimum receiver may actually be feasible with some simplifications.

3.5 Iterative multiuser receivers

Most of the initial work on MUDs focused on uncoded CDMA systems while more recent work addresses FEC coded DS-CDMA systems.

The optimal multiuser sequence estimator for an asynchronous convolutionally coded DS-CDMA system [106] performs multiuser detection and the convolutional decoding jointly, resulting in a prohibitive computational complexity, $O(2^{K\kappa})$, that is exponential in the product of the number of users and the constraint length of the convolutional code. In [107], a suboptimal scheme that performs symbol detection and decoding separately is proposed, along with a number of receiver structures that are based on the partitioned method. By implementing MLSE to signal detection and decoding separately, which results in a serial concatenation of the sequence based MLSE (discussed in Section 3.3.1) and the VA decoding of the convolutional code, the complexity becomes $O(2^{K+\kappa})$, exponential in the sum of the number of users and the constraint length. Consequently, the complexity of this partitioned scheme can be further reduced if other sub-optimal MUDs are used for the signal detection portion.

Figure 3.3 shows the partitioned multiuser receiver which consists of a MUD first stage and a bank of single-user decoders second stage. The class of partitioned multiuser receivers is further divided into those which use a hard-decision MUD to supply hard-decisions to the Viterbi decoders and those which use a soft-decision MUD to supply soft-decisions to the decoders. The outputs of hard-decision MUD are $\hat{b}_{k,f} \in \{-1,+1\}\,(k = 1,\ldots,K)$ which are then passed to the hard-decision Viterbi decoder. For the soft-decision unconstrained MUD, the detector simply passes the outputs to the decoders without going through any decision-making device such as a signum function. Constrained MUD, on the other hand, output hard decisions and certain techniques have to be employed to supply soft decisions. One example of such techniques is used in the optimum iterative multiuser receiver, to be discussed in Section 3.5.1.

Recently, the powerful iterative technique of Turbo codes has been applied to the multiuser detection of a FEC coded CDMA system [29, 30, 108, 109]. This scheme is similar to the partitioned scheme whereby the MUD and the channel decoders are separated, and different in that the MUD and single-user decoders iterate between one another, each producing soft

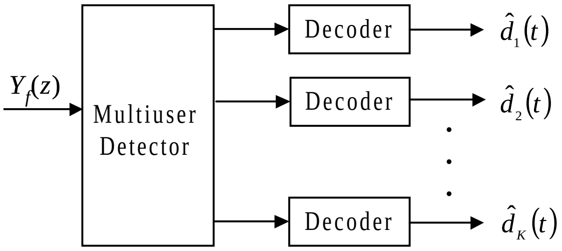

Figure 3.3: *The partitioned multiuser receiver*

outputs. The main difference between these works is in the multiuser detection portion since its complexity is the dominant factor in the overall complexity. [29] uses the M-algorithm to find a subset of the possible symbol vectors. The same algorithm is also used in [30], but a pre-whitening filter is added. This technique is similar to the IDDFD [101] which has a decorrelating or whitening filter before the tree search. However, [30] implements the whitening filter with the Gram-Schmidt orthogonal filter which is suggested in [103] (See Section 3.4.2.3 for a brief discussion on the IDDFD and the Gram-Schmidt orthogonal filter). The work in [109] is somewhat different as the soft outputs of the single-user decoders are used to modify the received signal by removing the estimated MAI. After a MMSE filtering, it is assumed that all MAI is removed and the channel becomes purely Gaussian. The extrinsic information needed by the single-user decoders can then be calculated.

3.5.1 Optimum iterative multiuser receiver

The optimum iterative multiuser receiver is shown in Figure 3.4. It comprises a CDMA Bayesian detector (also known as the SSB MAP detector described in 3.3.2) and single-user maximum *a posteriori* (MAP) decoders. The discrete time polynomial $Y_f(z)$ is the received chip signal at the f-th code symbol interval, $b_{k,f}$ is the f-th code symbol of the k-th user. Due to the fact that only the single-user FEC decoders are used for the decoding instead of joint decoding, the Bayesian detector needs to generate single-user information. This single-user information, $p(Y_f(z)|b_{k,f})$, can be viewed as the extrinsic information which is the knowledge about the symbol based on the structure of the multiuser signal. The single-user information, $\Pr(b_{k,f}|\mathbf{Y}_1^J(z))$, provided by the single-user decoder is the extrinsic information about the symbol based on the constraint of the FEC. The basic idea of the iterative scheme is to produce soft information both across the users and across the convolutional code trellis in a frame by using a soft-output multiuser detector and soft-input soft-output (SISO) single-user decoders respectively. The soft information across the two separate dimensions improves with diminishing returns at every iteration.

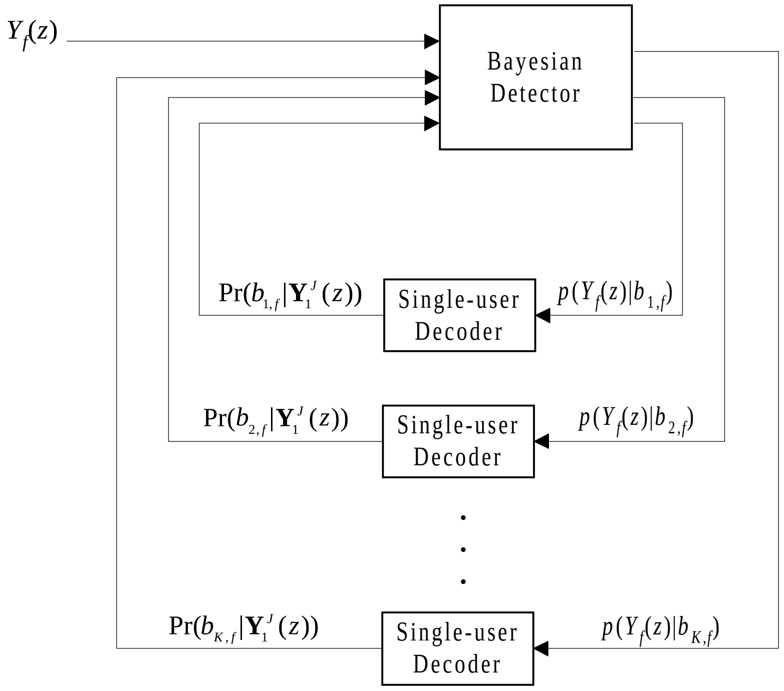

Figure 3.4: *The iterative multiuser receiver*

Consider the same channel coding as in Section 2.6.3, where τ n-symbol codeword including the $\kappa - 1$ trailing zeros are transmitted. Hence, there are $J = n\tau$ code symbols.

3.5.1.1 Bayesian detector

Based on the Bayes theorem, the observation probability density at the f-th symbol interval conditioned on the hypothesis that $b_{k,f} = b$ is

$$p(Y_f(z)|b_{k,f} = b) \;=\; \sum_{B_k^b} p(Y_f(z)|\mathbf{b}_f) \prod_{\substack{i=1 \\ (i \neq k)}}^{K} \Pr(b_{i,f}), \qquad (k = 1, \ldots, K; f = 1, \ldots, J) \tag{3.27}$$

where B_k^b is defined as a set that consists of all the symbol vectors, $\mathbf{b}_f$, with $b_{k,f} = b$ [29]. This is the extrinsic information or the extra knowledge based on the structure of the spreading sequences. The *a priori* information, $\Pr(b_{i,f})$, on the other hand, is derived from the trellis of the convolutional codes which will be discussed in Section 3.5.1.2.

Note that if the symbols are equally likely, then the single-user Bayesian decision rule,

$$\begin{aligned}\hat{b}_{k,f} &= \operatorname{sgn}(\Pr(b_{k,f}=+1|Y_f(z)) - \Pr(b_{k,f}=-1|Y_f(z))) \\ &= \operatorname{sgn}\left(\sum_{B_k^{+1}} p(Y_f(z)|\mathbf{b}_f) - \sum_{B_k^{-1}} p(Y_f(z)|\mathbf{b}_f)\right), \\ &\qquad (k=1,\ldots,K; f=1,\ldots,J)\end{aligned} \tag{3.28}$$

is in fact the SSB MAP constrained MUD discussed in Section 3.3.2 as shown by equation (3.16). If hard-decisions are made based on the initial (first iteration) output of the Bayesian detector, the result would be the same as the optimum single-user detector for an uncoded system. This is because the initial values of the *a priori* probabilities, $\Pr(b_{i,f})$, are 0.5 for all users' symbols. The subsequent values of the *a priori* probabilities are determined by the single-user FEC decoders.

The conditional probability in the summation of equation (3.27) is the likelihood of the received sequence, $Y_f(z)$, based on the hypothesis that $\mathbf{b}_f$ is transmitted. As the hypothesis is on the entire f-th symbol vector instead of just $b_{k,f}$, it can be viewed as the joint likelihood at instant f and can be computed as

$$\begin{aligned}p(Y_f(z)|\mathbf{b}_f) &= \frac{1}{(\pi N_0)^{G/2}} \exp\left(-\frac{\sum_{i=0}^{G-1}(y_{i,f}-x_{i,f}(\mathbf{b}_f))^2}{N_0}\right), \\ &\qquad (f=1,\ldots,J)\end{aligned} \tag{3.29}$$

where $y_{i,f}$ is the coefficient of z^{-i} in $Y_f(z)$ and $x_{i,f}(\mathbf{b}_f)$ is the i-th noiseless received chip signal due to the transmission of $\mathbf{b}_f$. Note that this joint likelihood is considered in the optimum MLSE [53]. The numerator in the exponential term is the Euclidean distance. Since $1/(\pi N_0)^{G/2}$ in equation (3.29) is common to all $\mathbf{b}_f$, it can be ignored.

However, the computational complexity of the optimum Bayesian detector is even higher than the optimum MLSE which is of the order of 2^K. The additional computations stem from the summation of 2^{K-1} terms in equation (3.27) which has to be carried out twice for the 2 possible symbols of $b_{k,f}$ and then repeated for the other $K-1$ users. Thus, even though the partitioned detector/decoder reduces complexity substantially, the complexity in the multiuser detector is still the bottleneck.

3.5.1.2 Single-user MAP decoder

The single-user MAP decoders operate identically to the method proposed in [49] for second stage decoders in a serial Turbo decoder. This method is equivalent to the BCJR algorithm discussed in Section 2.6.3 that generates the *a posteriori* probabilities of the states and state transitions at each time interval for a finite-state machine based on noisy observation.

As each user has exactly the same MAP decoder, the following discussion is based solely on the k-th user. The input of the k-th single-user decoder is a sequence of the k-th single-user likelihoods, $p(Y_f(z)|b_{k,f} = b)$ $(b = -1, +1; f = 1, \ldots, J)$, in a frame. The output of the decoder is a sequence of *a posteriori* probabilities, $\Pr(b_{k,f}|\mathbf{Y}_1^J(z))$, where $\mathbf{Y}_{f'}^f(z) = [Y_{f'}(z), \ldots, Y_f(z)]^T$ represents a vector form of the received sequence from time f' to time f in a frame. The *a posteriori* probability is conditioned on the hypothesis of the joint received signal and not just the received signal of the individual user since we are unable to distinctly separate them.

The single-user decoder calculates the *a posteriori* probabilities using the algorithm proposed in [2]. This soft-output algorithm has been used in the iterative decoding of Turbo codes [42] and has been discussed in detail in Section 2.6.3. The various terms in this section are analogous to those defined in Section 2.6.3, the superscript k has been added to limit the discussion to the k-th user. Define the k-th user's joint probability at time t from state $S_{t-1}^k = m^-$ to state $S_t^k = m$ as

$$\begin{aligned} \sigma_t^k(m^-, m) &= p(S_{t-1}^k = m^-; S_t^k = m; \mathbf{Y}_1^J(z)) \\ &= (h_\sigma)_t^k \alpha_{t-1}^k(m^-)\gamma_t^k(m^-, m)\beta_t^k(m), \\ &\qquad (t = 1, \ldots, \tau; k = 1, \ldots, K) \end{aligned} \tag{3.30}$$

where $(h_\sigma)_t^k$ is a constant. The forward state probability and reverse state probability can be computed as shown in equations (2.21) and (2.23) respectively, with the same boundary conditions. The forward state probability is

$$\begin{aligned} \alpha_t^k(m) &= \Pr(S_t^k = m|\mathbf{Y}_1^{tn}(z)) \\ &= (h_\alpha)_t^k \sum_{m^-=0}^{M-1} \alpha_{t-1}^k(m^-)\gamma_t^k(m^-, m), \quad (t = 1, \ldots, \tau; k = 1, \ldots, K) \end{aligned} \tag{3.31}$$

where $(h_\alpha)_t^k$ is such that

$$\sum_{m=0}^{M-1} \alpha_t^k(m) = 1. \qquad (t = 1, \ldots, \tau; k = 1, \ldots, K) \tag{3.32}$$

The reverse state probability is

$$\begin{aligned} \beta_t^k(m) &= \Pr(S_t^k = m | \mathbf{Y}_{tn+1}^J(z)) \\ &= (h_\beta)_t^k \sum_{m^+=0}^{M-1} \beta_{t+1}^k(m^+)\gamma_{t+1}^k(m, m^+), \\ &\qquad (t = 1, \ldots, \tau - 1; k = 1, \ldots, K) \end{aligned} \tag{3.33}$$

where $(h_\beta)_t^k$ is such that

$$\sum_{m=0}^{M-1} \beta_t^k(m) = 1. \qquad (t = 1, \ldots, \tau - 1; k = 1, \ldots, K) \tag{3.34}$$

The boundary conditions are $\alpha_0^k(0) = 1$, $\alpha_0^k(m) = 0$ for $m \neq 0$, $\beta_\tau^k(0) = 1$ and $\beta_\tau^k(m) = 0$ for $m \neq 0$.

For a valid transition from state m^- to state m, the transition probability is given as

$$\begin{aligned} \gamma_t^k(m^-, m) &= \frac{1}{2} \prod_{f=(t-1)n+1}^{tn} p(Y_f(z)|b_{k,f} = b), \\ &\qquad (t = 1, \ldots, \tau; k = 1, \ldots, K) \end{aligned} \tag{3.35}$$

where $[b_{k,(t-1)n+1}, \ldots, b_{k,tn}]$ is the BPSK-modulated codeword associated with the transition from state m^- to state m. The *a priori* information, $p(Y_f(z)|b_{k,f} = b)$, needed in equation (3.35), is the single-user likelihood information determined by the Bayesian detector.

Let $W_{t,f,k}^b$ be the set of transitions, $S_{t-1}^k = m^- \rightarrow S_t^k = m$, such that $b_{k,f} = b$ where the value of f lies between $(t-1)n + 1$ and tn. Then the joint probability of the state transition is computed as

$$\begin{aligned} p(b_{k,f} = b; \mathbf{Y}_1^J(z)) &= \sum_{W_{t,f,k}^b} \sigma_t^k(m^-, m). \\ &\qquad (t = 1, \ldots, \tau; k = 1, \ldots, K; f = 1, \ldots, J) \end{aligned} \tag{3.36}$$

The *a posteriori* probabilities are therefore

$$\Pr(b_{k,f}=b|\mathbf{Y}_1^J(z)) \;=\; \frac{p(b_{k,f}=b;\mathbf{Y}_1^J(z))}{p(b_{k,f}=+1;\mathbf{Y}_1^J(z))+p(b_{k,f}=-1;\mathbf{Y}_1^J(z))}. \quad (k=1,\ldots,K; f=1,\ldots,J) \qquad (3.37)$$

Note that the scaling factor, $(h_\sigma)_t^k$, in equation (3.30) does not have to be computed since it is cancelled away in the ratio of equation (3.37).

By making the following assignment,

$$\Pr(b_{k,f}=b) \;=\; \Pr(b_{k,f}=b|\mathbf{Y}_1^J(z)), \qquad (k=1,\ldots,K; f=1,\ldots,J) \qquad (3.38)$$

the *a posteriori* probabilities generated by the K single-user decoders can be used as the *a priori* information in the Bayesian detector. This is valid since the correlation between the spreading codes and the convolutional codes is small [29].

To decode the signal after the appropriate number of iterations, let $V_{t,k}^d$ be the set of states, S_t^k, such that $d_k(t)=d\in\{0,1\}$, then the decision rule is thus given as

$$\sum_{V_{t,k}^0}\alpha_t^k(m)\beta_t^k(m) \;\overset{d_k(t)=0}{\underset{d_k(t)=1}{\gtrless}}\; \sum_{V_{t,k}^1}\alpha_t^k(m)\beta_t^k(m), \qquad (t=1,\ldots,F; k=1,\ldots,K) \qquad (3.39)$$

since

$$\alpha_t^k(m)\beta_t^k(m)=h_t^k\Pr(S_t^k=m|\mathbf{Y}_1^J(z)), \qquad (t=1,\ldots,F; k=1,\ldots,K)(3.40)$$

where h_t^k is a constant that will be cancelled away in equation (3.39).

3.6 Summary

In this chapter, the MAI caused by the conventional method of signal detection in DS-CDMA communications systems has been investigated. MAI is dependent on two factors, namely the CCLs and the “near-far” effect. Multiuser detection seems to be the most promising candidate to resolve the MAI since it improves capacity and relaxes power control. The various existing multiuser detectors are introduced, starting with the optimum MUDs based on sequence detection (MLSE) and single-symbol detection (RBF). Sub-optimum MUDs are also discussed which include linear detector (DD and MMSE) and non-linear detectors (IC, MSD and DFD). Optimum MUD is too complex to be implemented, hence sub-optimum MUDs are sought. Most non-linear detectors introduce non-linearity into linear MUDs, and improves performance from the simple linear structure. The work here takes a different approach, that is to start from the optimum MUD with superior performance and then reduce the complexity. A recently developed class of multiuser receivers for FEC coded DS-CDMA systems known as iterative multiuser receivers has also been discussed. This class of receivers exploits the powerful iterative techniques used in Turbo codes to iterate soft information between signal detection and channel decoders.

Chapter 4
Chip-based MLSE detector

Verdu's MLSE MUD has very good performance but at the expense of high computational complexity. This detector searches through all the possible input symbol sequences and selects the one closest to the received sequence based on the Euclidean distance. It can be implemented using a matched filter bank front-end and based on the sufficient statistics of these matched filters' output, a maximum likelihood ratio comparison is then carried out at the symbol level.

The initial idea of the detector in this chapter is to operate the MLSE at the chip-level instead of the symbol-level. This negates the need for a matched filter front-end, consequently, despreading, symbol-detection and multipath-combination are performed jointly. Immediately, the absolute linear distance instead of the Euclidean distance can be considered. As squaring is not needed, no multiplication is involved at run-time. It is important to note that the focus of the work here is only on the downlink.

The first section of this chapter formulates the chip-based MLSE (CBMLSE) detector such that the VA can be applied. The trellis for the AWGN channel and the multipath channel will be constructed in Sections 4.2 and 4.3 respectively. The application of the VA to the CBMLSE is also shown in the latter section. Section 4.4 considers the absolute linear distance instead of the Euclidean (squared) distance. The absolute value can be obtained simply by dropping the signed bit in the DSP, no multiplication is involved during run-time. Section 4.5 considers another simplification, that is to use the Hamming distance. The complexity of the linear distance detector is then compared with the MLSE in the next section. The number of addition and multiplication operations needed both at run-time and pre-computation are given in detail for both the AWGN channel and the multipath channel in Section 4.6.1 and Section 4.6.2 respectively. Finally, in Section 4.7, simulation results under AWGN and multipath channels are presented in Section 4.7.1 and Section 4.7.2 respectively. The truncation length of the VA is also investigated and presented in Section 4.7.3.

4.1 Chip-based MLSE

The MLSE can also be implemented at chip-level, that is without first matched filtering the signal with the spreading codes of the users. This detector will be referred to as the chip-based MLSE (CBMLSE) while the symbol-based MLSE is still referred to as MLSE. Instead of implementing equation (3.9), equation (3.8) can be used. A point to note here is that this implementation yields the same result as that implemented at symbol-level since no assumption or approximation is made, though it has to run at a higher speed because processing is carried out at the chip level. At each chip, the squared of the differences between the received chip value and all the possible noiseless chip values are computed. Each of these values will be referred to as the Euclidean distance metric. Note that by implementing the receiver at the chip level, the matched filter bank is no longer required, despreading and detection is processed jointly.

Although the noiseless chip values, x_i, can be calculated using equation (2.8), it not only has to be based on the entire transmitted symbol sequence but is also exponential in the message length. A discrete-time system can be found such that it can be solved efficiently by the forward dynamic programming algorithm (the Viterbi algorithm). Starting from equation (2.8),

$$\begin{aligned}
X(z) &= \sum_{f=0}^{F-1}\left[\sum_{i=0}^{G+L-2}\mathbf{r}_i^T z^{-i}\right]\mathbf{P}\mathbf{b}_f z^{-Gf} \\
&= \sum_{f=0}^{F-1}\left[\sum_{i=0}^{G-1}\mathbf{r}_i^T z^{-i} + \sum_{i=G}^{G+L-2}\mathbf{r}_i^T z^{-i}\right]\mathbf{P}\mathbf{b}_f z^{-Gf} \\
&= \sum_{f=0}^{F-1}\left[\sum_{i=0}^{G-1}\mathbf{r}_i^T z^{-i} + \sum_{j=0}^{L-2}\mathbf{r}_{G+j}^T z^{-G-j}\right]\mathbf{P}\mathbf{b}_f z^{-Gf} \\
&= \sum_{f=0}^{F}\left[\sum_{i=0}^{G-1}\mathbf{r}_i^T z^{-i}\mathbf{P}\mathbf{b}_f + \sum_{j=0}^{L-2}\mathbf{r}_{G+j}^T z^{-j}\mathbf{P}\mathbf{b}_{f-1}\right] z^{-Gf} \\
&= \sum_{f=0}^{F} z^{-Gf}\sum_{i=0}^{G-1} x_i(\mathbf{b}_{f-1},\mathbf{b}_f) z^{-i}, \qquad (4.1)
\end{aligned}$$

where $\mathbf{r}_i = [r_{1,i},\ldots,r_{K,i}]^T$ $(i = 0,\ldots,G+L-2)$, and $r_{k,i}$ is the coefficient of z^{-i} in $R_k(z)$ $(k = 1,\ldots,K)$. The fact that $\mathbf{b}_f = \mathbf{0}$ for $f = -1$ and $f = F$ has been utilised in the above equation. The inner summation consists of all the noiseless chip values, $x_i(\mathbf{b}_{f-1},\mathbf{b}_f)$ $(i = 0,\ldots,G-1)$, in the f-th message symbol interval. The i-th chip value of

$x_i(.)$ is independent of the previous symbols if $i > L-2$. Only the first $L-1$ chips are interfered with by the previous symbols. Equation (4.1) shows that the noiseless chip values can be calculated just based on the current symbols and (if necessary) the previous symbols. These chip values also repeat every symbol interval, so they can be calculated easily regardless of the message length. Decoding can also begin immediately once the signal is received and does not have to wait until transmission ends. Hence, the Viterbi algorithm can be readily applied to calculate the Euclidean distance.

4.2 AWGN state-transition trellis

For the synchronous AWGN channel, each symbol interval is independent of the other symbol intervals without intersymbol interference. This can be seen clearly from equation (4.1) where the summation term that is dependent on the previous symbol vector equates to zero since $L = 1$ here. The noiseless chip values become $x_i(\mathbf{b}_f) = \mathbf{r}_i^T \mathbf{P} \mathbf{b}_f$ and $\mathbf{r}_i = [c_{i,1}, \ldots, c_{i,K}]^T$ $(i = 0, \ldots, G-1)$ (see equations (2.2) and (2.10)). Hence, a decision can be made once the complete sequence in a symbol duration is received. In a symbol interval, there are 2^K possible transmitted symbol vectors. The corresponding 2^K possible metrics can be easily calculated at each chip, provided the spreading codes and received power are known. The decoding procedure is best illustrated with the state-transition trellis diagram in Figure 4.1, assuming a 2-user system. Each state, ($\mathbf{b}_{f,j}, i$), is defined by a possible symbol vector, $\mathbf{b}_{f,j}$ $(j = 1, \ldots, 2^K)$ and the chip position, i $(i = 0, \ldots, G-1)$. The initial state is set to be an arbitrary state, (00,-1), since it does not have any effect on the remaining states. In the diagram, each transition path is labelled with the value of $x_i(\mathbf{b}_{f,j})$, which is independent of the previous symbols. In this case, there is only one possible transition path from one state to the next, since the transmitted symbols remain unchanged in a symbol duration. Hence, state ($\mathbf{b}_{f,j}, i$) can only transit to state ($\mathbf{b}_{f,j}, i+1$). The sum of all the metrics in each path (metric sum) determines the best path (path with the smallest metric sum) which corresponds to the best symbol vector. The trellis diagram here is very simple and is not even necessary for decoding, but it gives an insight into decoding at chip level and is especially for the Viterbi algorithm, which is not necessary in an AWGN channel but essential in a multipath channel.

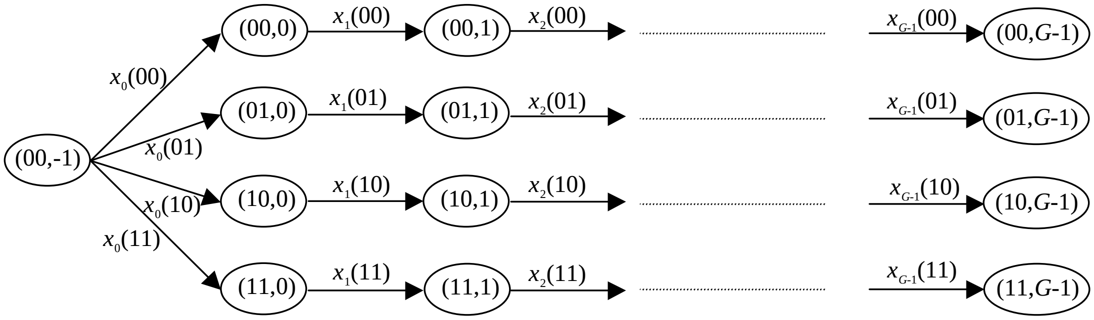

Figure 4.1: *State-transition trellis for AWGN channel*

4.3 Multipath state-transition trellis

If the channel exhibits multipath properties, then the structure of the trellis is not so straightforward, it has to be expanded between symbols whenever multipath effect arises, and then collapsed when the interference from the previous symbols dies away. For illustration, consider the same system above but now with a multipath of length $L = 4$. The state-transition trellis is given in Figure 4.2 which shows the transition between 2 adjacent symbol-intervals, the $(f-1)$-th and f-th symbol-intervals. The trellis is expanded when transiting from the last chip of the $(f - 1)$-th interval to the first chip of the f-th interval. The state of the expanded trellis, which has to take the previous symbol vector into consideration as well as the current symbol vector, is described as ($\mathbf{b}_{f-1,j}, \mathbf{b}_{f,p}, i$), where $\mathbf{b}_{f-1,j}$ $(j = 1, \ldots, 2^K)$ and $\mathbf{b}_{f,p}$ $(p = 1, \ldots, 2^K)$ are possible symbol vectors in the $(f - 1)$-th and f-th intervals respectively. Each state in the $(f - 1)$-th interval is expanded into 2^K states, resulting in a total of 4^K states and transition paths each. Each transition path is labelled with the value of $x_i(\mathbf{b}_{f-1,j}, \mathbf{b}_{f,p})$. For the second chip, it is still affected by the previous symbols, so only horizontal transition paths are possible. This ensures that at each state, there is only 1 incoming transition path and thus preventing any decision making. Even though the third chip is still interfered by the previous symbols, the trellis can be collapsed since the next chip is independent of the previous symbols and so some form of decision can be made here. There are still 4^K transition paths, only that 2^K of them converge into a state. At each terminating state of this transition path, a decision can be made to select the survival path at this stage, which is the one with the smallest metric sum. Therefore, there will be a total of 2^K survival paths. The fourth chip ($i = L - 1$) is only dependent on the current symbols, hence only 2^K noiseless received chip values or transition paths are possible. The states, ($\mathbf{b}_{f,j}, i$), and transition paths, $x_i(\mathbf{b}_{f,j})$, in the collapsed trellis, do not involve the previous symbol vector.

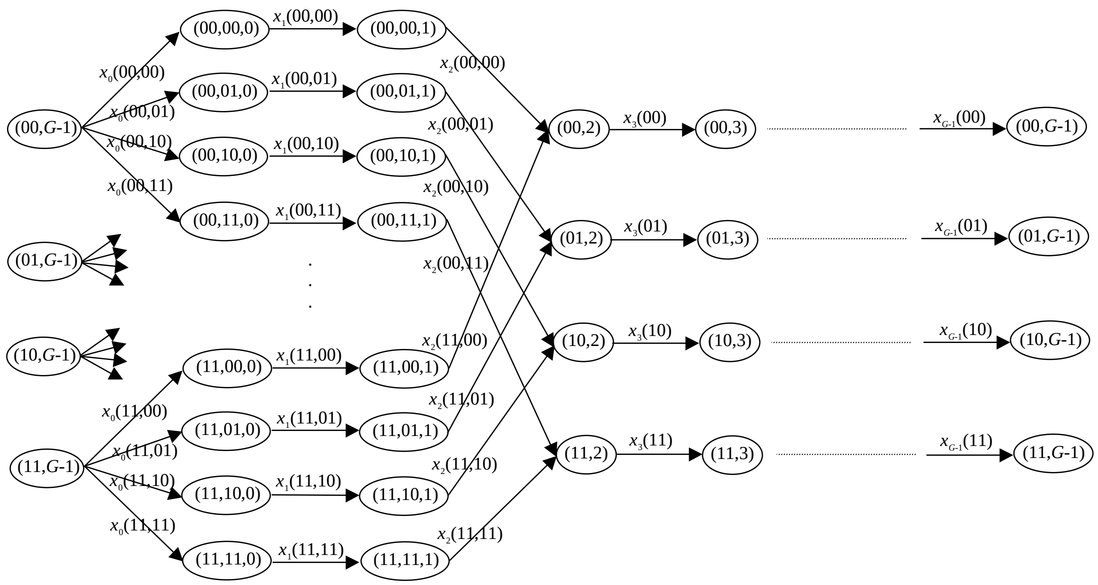

Figure 4.2: *State-transition trellis for multipath channel*

Assuming that the fading channel parameters are known, the noiseless received chip value associated with each transition path can be computed using equation (4.1). These values, $x_i(\cdot)\ (i = 1, \ldots, G-1)$, repeat after each symbol interval, hence the trellis of 1 symbol duration is sufficient to compute the metric values for the entire transmission. The metric associated with a transition path from a state will be added to the metric sums of all incoming paths into that state. If any other path to each state replaces the survival path in any overall path, then the path metric would be larger. This is the basis of the Viterbi algorithm. Once the survival paths have been selected, then they can be traced back. If all the survival paths originate from a particular state, then the symbol vector associated with this state is the decoded symbol vector.

4.4 Linear distance metrics

The CBMLSE detector yields the optimum performance for the sequence-based detection, however the cost of such good performance is the multiplication (squaring) that has to be performed at each chip. Multiplications are very computational intensive for a DSP. If the absolute value is taken instead of the Euclidean distance, then no multiplication is needed at all. However, this also means an inevitable trade-off in performance. The work here investigates whether the trade-off in complexity and performance is appropriate for practical implementation. Mathematically, instead of making decisions using equation (3.8), the decision rule is now

$$\begin{aligned}\hat{\mathbf{B}}(z) &= \arg \min_{\mathbf{b}(z)\in\{-1,+1\}^{FK}} \|Y(z) - X(z)\| \\ &= \arg \min_{\mathbf{b}(z)\in\{-1,+1\}^{FK}} \sum_{i=0}^{G(F-1)+L-2} |y_i - x_i|. \end{aligned} \tag{4.2}$$

Hence, each linear distance metric value, $|y_i - x_i|$, is simply taking the absolute value of the difference between the received value and one of the possible noiseless received values at each chip. The absolute value can be taken simply by discarding the sign bit. This detector will be referred to as the linear metric CBMLSE (LMCBMLSE).

4.5 Hamming distance

Another simplification is to considered the Hamming distance. Hard decision is made at each chip level, i.e., the received chip signal is passed through a theshold device such that the output conforms to only 1 out of all the possible chip values. For equal power case, each chip can only take certain discrete values, for example if there are 3 users, then the possible chip values are only $\{-3, -1, 1, 3\}$, assuming unit power. This feature is exploited further in Chapter 5. The Hamming distance between two adjacent values is 1, the next nearest 2, and so on. For example, the Hamming distance between -1 and 3 is taken to be 2. At the end of the symbol duration, the sums of the Hamming distances between the selected sequence and all the possible transmitted sequences are compared. The symbol vector associated with the sequence with the smallest sum of the Hamming distances is the decoded symbol vector. If two or more transmitted sequences result in the same overall Hamming distance sum, then the linear distance or Euclidean distance method is resorted to. In fact, for practical implementation, the received signal is quantised and the true real value is not passed to the detector. The Hamming distance method is in fact the worst case of quantisation for the LMCBMLSE. Different degrees of quantisation would lie between the Hamming distance graph and the LMCBMLSE.

4.6 Complexity analysis

The complexity of the LMCBMLSE MUD is now compared with the MLSE under the AWGN and multipath downlink conditions where all the users have equal power. The complexity analysis is divided into run-time and pre-calculation.

4.6.1 AWGN channel

The complexity comparison for the AWGN case is summarised in Table 4.1. The LMCBMLSE does not involve any multiplication operation during run-time as the values of $x_{\,i}$ $(i = 0, \ldots, G-1)$ are pre-calculated, only 2^K addition operations corresponding to the 2^K possible states are carried out at each chip. In one symbol interval, total number of arithmetic operations is $(2G-1)2^K$ additions for the 2^K metric sums.

The complexity of the MLSE at run-time is to be determined next. From equation (3.15), the third term equates to zero for AWGN channel ($\langle \mathbf{R}(z) \mathbin{\textbf{.}} \mathbf{R}^T(z) z^{-G} \rangle = 0$) and only the first term needs to be computed at run-time since it involves the observation, the second term can be pre-computed. Assume that the matched filter bank does not require any multiplication since the matched filters only involve +1 and -1, and multiplication with -1 can be carried out by a simple inversion. However, $G-1$ additions are required for each matched filter and the matched filter bank, $\langle Y(z) \mathbin{\textbf{.}} \mathbf{R}^T(z) z^{-Gf} \rangle$, requires a total of $K(G-1)$ additions. For maximum likelihood ratio comparisons, equation (3.15) is considered. Since $2\langle Y(z) \mathbin{\textbf{.}} \mathbf{R}^T(z) z^{-Gf} \rangle \mathbf{P}$ is common to all the possible states, only KG additions are required for this partial term if normalised power is considered. Note that an addition has been used to compute the scaling by a factor of 2. Hence, a total of $(K-1)2^K + KG$ additions are needed for the calculation of all the 2^K possible values of $2\langle Y(z) \mathbin{\textbf{.}} \mathbf{R}^T(z) z^{-Gf} \rangle \mathbf{P}\mathbf{b}_f$. During run-time, the computation of the 2^K possible metrics, $\lambda_f(\mathbf{b}_f)$, needs $K(2^K + G)$ additions. The extra 2^K additions are required for the summation of the first and second terms in equation (3.15). The dominant complexity factors in the MLSE and the LMCBMLSE are respectively $K2^K$ and $2G2^K$. As G is usually greater than K, the number of additions in the chip-based scheme is expected to be higher.

For pre-calculation, the LMCBMLSE needs $K-1$ additions per chip per state using equation (4.1) and taking note that the second term is zero since $L = 1$. The value of $\mathbf{r}_i^T \mathbf{P}$ is the same for all states, hence giving a total of $G2^K(K-1)$ additions for all the 2^K possible states in every symbol interval.

	Multiplications	Additions
Run-time		
LMCBMLSE		
Each transition, $\lvert y_i - x_i(\mathbf{b}_f)\rvert$	0	1
Total for all possible states	0	$(2G-1)2^K$
MLSE		
$\langle Y(z)\boldsymbol{.}\mathbf{R}^T(z)z^{-Gf}\rangle$ for all users	0	$K(G-1)$
$2\langle Y(z)\boldsymbol{.}\mathbf{R}^T(z)z^{-Gf}\rangle\mathbf{P}\mathbf{b}_f$		
for all states	0	$KG+(K-1)2^K$
$\lambda_f(\mathbf{b}_f)$ for all states	0	$K(2^K+G)$
Pre-calculation		
LMCBMLSE		
$\mathbf{r}_i^T\mathbf{P}\mathbf{b}_f$ for all chips and states	0	$(K-1)2^KG$
MLSE		
$\mathbf{b}_f^T\mathbf{P}\langle\mathbf{R}(z)\boldsymbol{.}\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}_f$	0	$(G-1)K(K+1)/2$
for all states		$+2^K(K^2-1)$

Table 4.1: *Complexity comparison between the LMCBMLSE and the MLSE in a AWGN channel*

From equation (3.15), the MLSE requires to pre-compute $\mathbf{b}_f^T\mathbf{P}\langle\mathbf{R}(z)\boldsymbol{.}\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}_f$. The last term in equation (3.15) does not have to be computed for the AWGN channel. Assuming no multiplication for the calculation of the correlation matrix, $\langle\mathbf{R}(z)\boldsymbol{.}\mathbf{R}^T(z)\rangle$, and whenever multiplying with +1 and -1, $(G-1)K(K+1)/2$ additions are needed for the calculation of $\mathbf{P}\langle\mathbf{R}(z)\boldsymbol{.}\mathbf{R}^T(z)\rangle\mathbf{P}$, which remains unchanged for all the states. Note that the correlation matrix is symmetrical, so only $K(K+1)/2$ out of the K^2 elements need to be computed since the rest are repeated. The total number of arithmetic operations for the MLSE is thus $2^K(K^2-1)+(G-1)K(K+1)/2$ additions. Clearly, the pre-calculation of the MLSE is of the same order as the LMCBMLSE.

4.6.2 Multipath channel

In this section, the complexities of these 2 detectors under multipath conditions are compared and Table 4.2 gives a summary of the complexity comparison in the multipath scenario. For the MLSE, the front-end is the RAKE receiver with L fingers if the number of multipaths is L. Hence, the RAKE receiver for each user requires L multiplications and $GL-1$ additions where they include L multiplications and $L-1$ additions for maximum ratio combining. From equation (3.15), the second and third terms can be pre-computed jointly to yield the 4^K possibilities. The total number of arithmetic operations for the 2^K possible first terms is KL multiplications and $KGL+2^K(K-1)$ additions. Hence, the overall complexity in computing the symbol metric given by equation (3.15) is KL multiplications and $KGL + 2^K(K-1) + 4^K$ additions where the additional 4^K additions are due to the summation of the first term to the 4^K possible sums of the second and third terms.

For the LMCBMLSE, the calculation of the linear distance metrics once again does not require any multiplication. The number of additions needed for all the metrics calculation in the expanded and the rest of the trellis are $(L-1)4^K$ and $(G-L+1)2^K$ respectively. Hence, for the metric sums in a symbol interval, total number of additions is $2(L-1)4^K + (G-L+1)2^{K+1}$.

Number of multiplications in the MLSE is now dependent not only on the number of users but also the number of multipaths. On the other hand, the LMCBMLSE still does not require any multiplication at all. Although both detectors have the same order of complexity ($O(4^K)$) in terms of additions, the LMCBMLSE needs more additions due to the scaling factor $2(L-1)$ in the dominant term.

	Multiplications	Additions
Run-time		
LMCBMLSE		
Each transition, $\lvert y_i - x_i(\mathbf{b}_{f-1}, \mathbf{b}_f)\rvert$	0	1
Total for all possible states	0	$2(L-1)4^K$ $+(G-L+1)2^{K+1}$
MLSE		
$\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle$ for all users	KL	$K(GL-1)$
$2\langle Y(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-Gf}\rangle\mathbf{P}\mathbf{b}_f$ for all states	KL	$KGL+(K-1)2^K$
$\lambda_f(\mathbf{b}_{f-1}, \mathbf{b}_f)$ for all states	KL	$KGL+(K-1)2^K+4^K$
Pre-calculation		
LMCBMLSE		
$\mathbf{r}_i^T\mathbf{P}\mathbf{b}_f$ for all chips and states	0	$(K-1)2^K G$
$\mathbf{r}_{G+j}^T\mathbf{P}\mathbf{b}_f$ for all chips and states	0	$(K-1)2^K(L-1)$
Total for all states	0	$(K-1)2^K(G+L-1)$ $+4^K(L-1)$
MLSE		
$\mathbf{b}_f^T\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}_f$ for all states	$(G+L-1)K(K+1)/2$	$(G+L-2)K(K+1)/2$ $+2^K(K^2-1)$
$2\mathbf{b}_{f-1}^T\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}\mathbf{b}_f$ for all states	$(L-1)K^2$	$2^K(K-1)K$ $+(L-1)K^2+4^K(K-1)$
Total for all states	$(G+L-1)K(K+1)/2$ $+(L-1)K^2$	$(G+L-2)K(K+1)/2$ $+2^K(2K+1)(K-1)$ $+(L-1)K^2+4^K K$

Table 4.2: *Complexity comparison between the LMCBMLSE and the MLSE in a multipath channel*

Next, the complexity of the pre-computation is compared. Equation (3.15) is again used for the MLSE. The matrix, $\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)\rangle\mathbf{P}$, just have to be computed once. Similarly, $2\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}$ in the last term needs to be computed only once. Hence, the complexity of the common matrices in the second and third terms are respectively $(G+L-1)K(K+1)/2$ multiplications and $(G+L-2)K(K+1)/2$ additions, and $(L-1)K^2$ multiplications and $(L-2)K^2+K^2$ additions. In matrix $\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle$, all K^2 elements need to be computed as they are not repeated. Total complexity of the second term is thus $(G+L-1)K(K+1)/2$ multiplications and $(G+L-2)K(K+1)/2+2^K(K^2-1)$ additions for the 2^K possible states. The last term is harder to compute as it involves 2 symbol-intervals, resulting in 4^K possible states. Making use of the fact that each of the 2^K possible $2\mathbf{b}_{f-1}^T\mathbf{P}\langle\mathbf{R}(z)\,\mathbf{.}\,\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}$ can be reused by all the 2^K possible $\mathbf{b}_f$, the last term requires $(L-1)K^2$ multiplications and $(L-1)K^2+2^KK(K-1)+4^K(K-1)$ additions. Therefore, the total pre-computation is $(G+L-1)K(K+1)/2+(L-1)K^2$ multiplications and $(G+L-2)K(K+1)/2+(L-1)K^2+2^K(2K^2-K-1)+4^KK$ additions where extra 4^K additions are needed for the summation of the second and third terms to yield the 4^K possible joint terms for run-time computation.

For the LMCBMLSE, equation (4.1) is used to determine the pre-calculation complexity. Now the second term has to be considered as $L>1$ and it needs $(L-1)2^K(K-1)$ additions every symbol. The complexity of the first term is the same as that computed in the AWGN channel scenario. Thus, overall pre-computation is $(G+L-1)2^K(K-1)+4^K(L-1)$ additions every symbol interval.

In the multipath case, the number of multiplications in the MLSE is an order higher than the LMCBMLSE. Although both detectors have the same order of complexity, ($O(4^K)$), in terms of additions, the MLSE still has a marginally higher complexity due to the fact that K^2 is usually greater than $L-1$.

To illustrate the complexity gain, two examples are considered. Both with 3 paths ($L=3$). One has 8 users ($K=8$) and a processing gain of $G=16$, the other example has 4 users and $G=8$. The complexity is tabulated in Table 4.3.

AWGN	Multiplications	Additions	Multiplications	Additions
Run-time	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
LMCBMLSE	0	7936	0	240
MLSE	0	2176	0	96
Pre-calculation	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
LMCBMLSE	0	28672	0	384
MLSE	0	16668	0	310

Multipath	Multiplications	Additions	Multiplications	Additions
Run-time	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
LMCBMLSE	0	269312	0	1216
MLSE	24	67712	12	400
Pre-calculation	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
LMCBMLSE	0	163328	0	992
MLSE	776	555492	132	1578

Table 4.3: *Complexity comparison between the LMCBMLSE and the MLSE for AWGN channel and multipath ($L = 3$) channel*

4.7 Simulation results

In order to find the performance of the LMCBMLSE, Monte-Carlo simulations [110] have been used to obtain the bit error rate (BER).

4.7.1 AWGN channel

Initially, only the synchronous case in AWGN channel has been considered. The simulation has been carried out for a system with equal-power users and random spreading codes of length, $G = 7$. Figure 4.3 shows the BER averaged over all users in the system plotted against the number of active users for $E_b/N_0 = 9$ dB. The simulation has been carried out for 1 million trials to ensure reliable data for low BER. The spreading codes of the users are chosen from a preselected set of random codes and remain the same thoughout the simulations. The BER depends largely on the cross-correlation levels of the codes and it is these high cross-correlation values of the code of the sixth user that cause the jump in BER. Other MUDs such as the conventional matched filter detector (CMFD), the MMSE, and both the CBMLSE and the MLSE, have also been simulated so as to compare the performance. The MLSE simulated makes use of equation (3.9) which performs likelihood comparisons only on the output of the matched filter bank. The graph labelled CBMLSE uses equation (3.8) which computes the Euclidean distance between the received and transmitted sequences. This detector can use exactly the same decoding algorithm as suggested in the LMCBMLSE except that instead of taking the absolute value, the squared value is computed. Theoretically, this detector and the MLSE give the same result. This is verified as observed from the plot. The purpose of this comparison is to ensure that the simulation for the proposed algorithm has been carried out correctly. The conventional detector performs poorly especially when the number of users (MAI) increases. The MMSE detector shows substantial improvement in performance over the conventional matched filter detector but still far from optimum. The LMCBMLSE only degrades slightly from optimum and is even indistinguishable from the MLSE and CBMLSE for large MAI. This is probably because the Gaussian noise becomes insignificant as compared to the large MAI. The metrics that do not correspond to the actual transmitted symbols are very large as compared to the near zero values of the metrics that are associated with the actual transmitted symbols. Since these metrics are already very large, squaring them does not improve performance significantly. The Hamming distance estimation with linear and Euclidean distances options have also been simulated and plotted. The graphs follow the trend of the LMCBMLSE and CBMLSE. At low MAI, the graphs are indistinguishable because Hamming distance estimation is sufficient and resorting to linear or Euclidean distance is unnecessary.

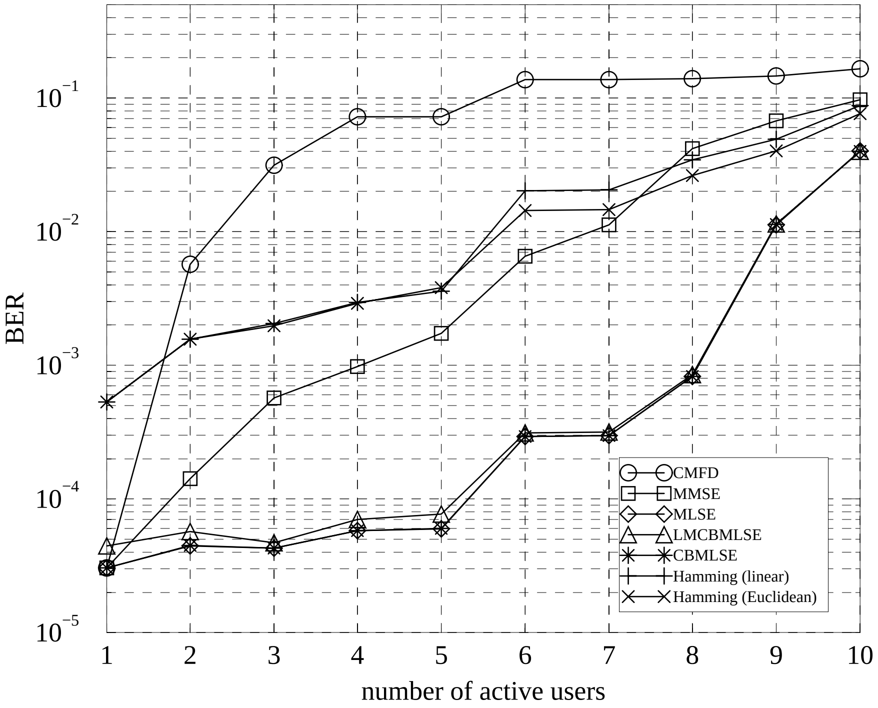

Figure 4.3: *BER against number of active users for $E_b/N_0 = 9$ dB using random codes of processing gain $G = 8$ in AWGN channel*

4.7.2 Multipath channel

A stationary multipath channel with channel impulse response,

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}, \tag{4.3}$$

is considered next. As would be the case in a downlink, all signals are assumed to pass through the same channel. Walsh codes are used in the simulations, and to ensure the codes selected are unbiased, the codes selected for the active users are randomised. The simulations with 100000 trials have been repeated 200 times, and each time, a new set of codes has been randomly selected for the users. Walsh codes are orthogonal codes that are ideal for a perfect channel, but degrade in a multipath channel. The simulation results have been plotted in Figure 4.4 where the performance of the CMFD (or RAKE), LMCBMLSE and CBMLSE are shown for the AWGN and multipath channels. As expected the Walsh codes have every good performance in the AWGN channel since there is no MAI. As the users do not interfere with each other, performance is constant even as the number of users increases. Whereas in a multipath channel, the performance degrades significantly. Using likelihood-based detectors, the performance can be maintained at a relatively constant level as MAI increases.

The same multipath channel is used in the next simulation. Both the LMCBMLSE and CBMLSE detectors have been simulated for random codes and Walsh codes, both of length $G = 8$ and $E_b/N_0 = 7$ dB. Results are shown in Figure 4.5. The simulation has been carried out under the same conditions. This is observed in the simulation plot. Even though the two types of codes are of the same processing gain, the random codes show much better performance especially at low MAI. The RAKE receiver has also been simulated for the two kinds of codes and the BER results are also plotted in Figure 4.5. It is observed that here the Walsh codes perform slightly better when the number of active users is more than 3. This verifies that Walsh codes are better for the RAKE receiver at large MAI [7]. Under no MAI condition ($K = 1$), the RAKE receiver performs close to the LMCBMLSE and the MLSE, but degrades rapidly when MAI increases, regardless of the codes used.

The graph of the BER against E_b/N_0 for the same multipath channel has also been plotted as shown in Figure 4.6. A 4-user system using spreading code of processing gain 8 is considered here. The performances of the LMCBMLSE and CBMLSE are much better than the RAKE receiver. The random codes again perform better than the Walsh codes except for the RAKE

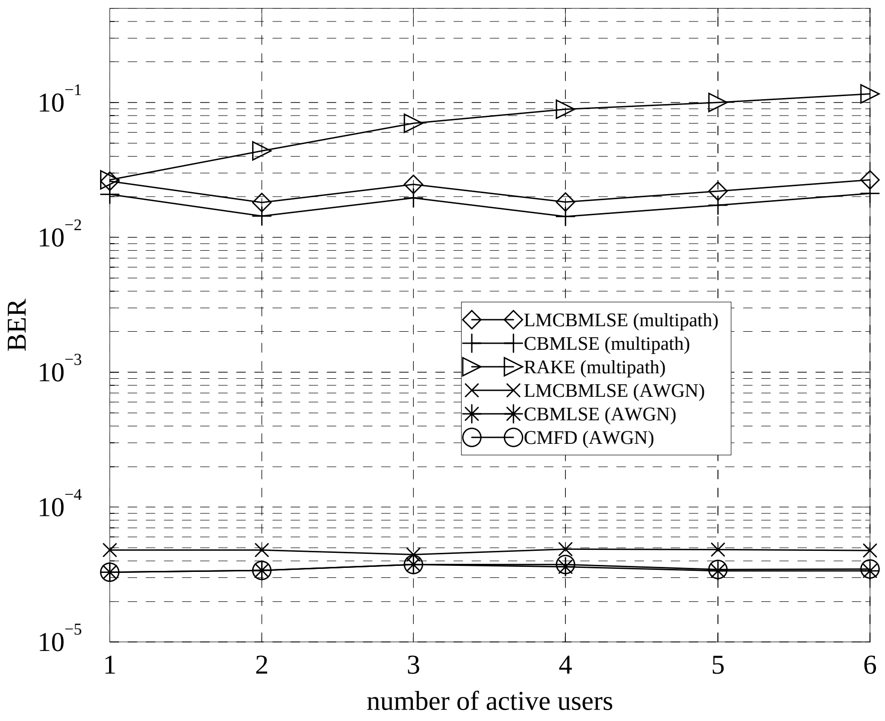

Figure 4.4: *BER against number of active users for $E_b/N_0 = 9$ dB using Walsh codes of length 8*

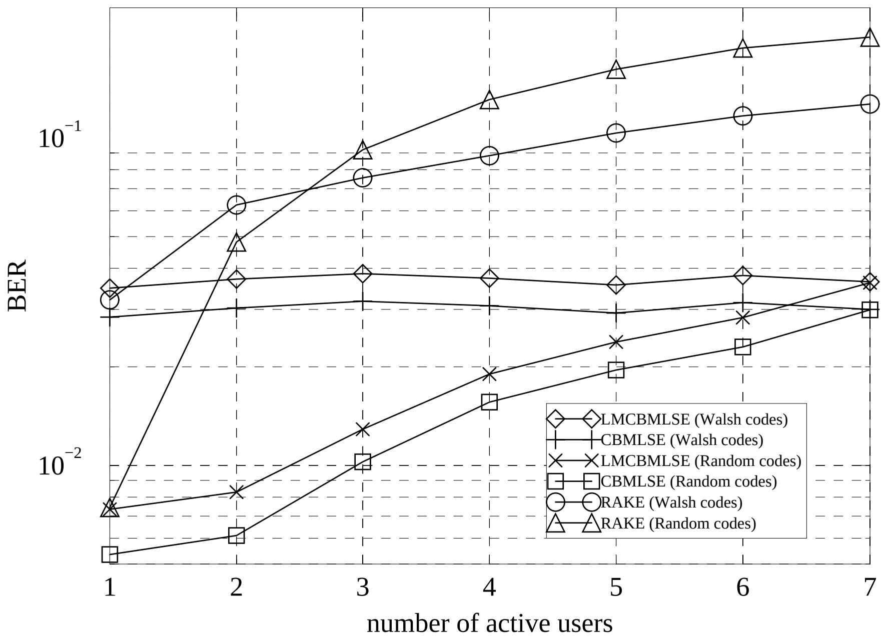

Figure 4.5: *BER against number of active users for $E_b/N_0 = 7$ dB using spreading codes of processing gain $G = 8$ in multipath channel*

receiver. For a BER of 0.025, it is found that the LMCBMLSE requires a E_b/N_0 of just about 0.5 dB more to attain the same performance as the MLSE for both the codes. Note that this performance is unachievable by the RAKE receiver.

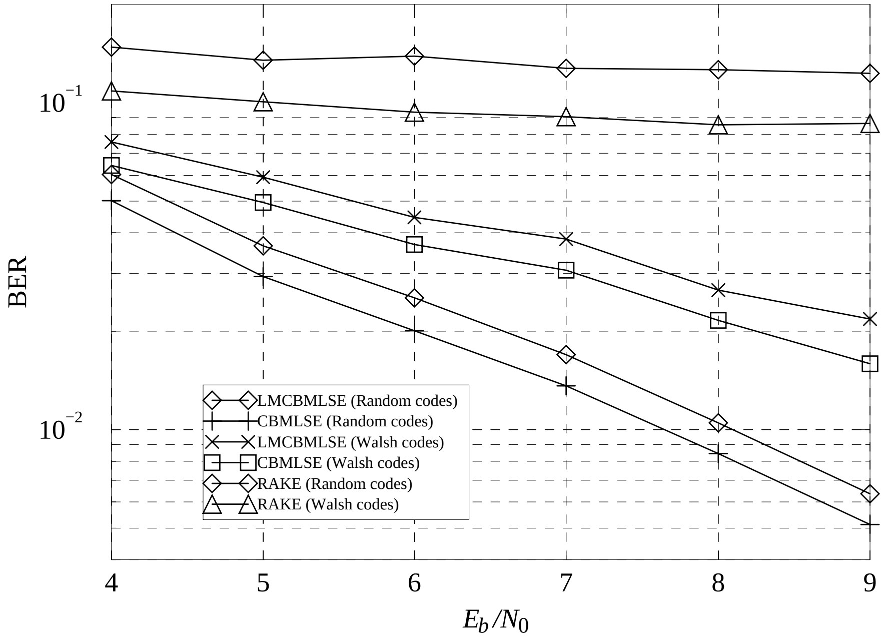

Figure 4.6: *BER against E_b/N_0 for 4 users using codes of processing gain $G = 8$ in multipath channel*

4.7.3 Truncation length of the VA

The length of the survival paths in the VA considered so far is unlimited. If the survival paths are long, then a large amount of storage is needed and the delay may be intolerable. For practical implementation, the length is preset to a sufficiently small finite value, which means that if the survival path lengths are equal to this preset value, the symbol vector corresponding to the best survival path is selected. Simulations have again been used to determine the performance when the path length varies from 1 symbol to 5 symbols. A path length of 1 symbol means that decision for the f-th symbol interval is made immediately at the next interval ($f + 1$). Since the transitions are all horizontal after the trellis converges in the $(f+1)$-th interval, there is no point delaying decision-making until the end of the interval. A decision made immediately once the trellis converges (in this example where $L = 3$, it is the second chip as shown in Figure 4.2) would give the same result. Generally, a delay of j symbols implies making a decision at chip $i = L - 2$ of the $(f + j)$-th symbol-interval. Figure 4.7 shows the BER as the path length varies for a 4-user system. It is found that for a delay of just 2 symbols, the performance is already very close to that of the unlimited path length for both the LMCBMLSE and the CBMLSE. The same trend is observed in both codes although the Walsh codes with the same processing gain perform much worse. The maximum survival path lengths under unlimited delay condition are also indicated in parenthesis in the figure and they are very much larger than the fixed delay. Hence, the well-known advantage that little degradation of performance occurs when the Viterbi algorithm uses a sufficient fixed finite decision delay in real-time implementation has been verified. The RAKE receiver has also been simulated for the two kinds of codes in the multipath channel as shown in the plot. Notice the wide gap between the RAKE performance and the LMCBMLSE.

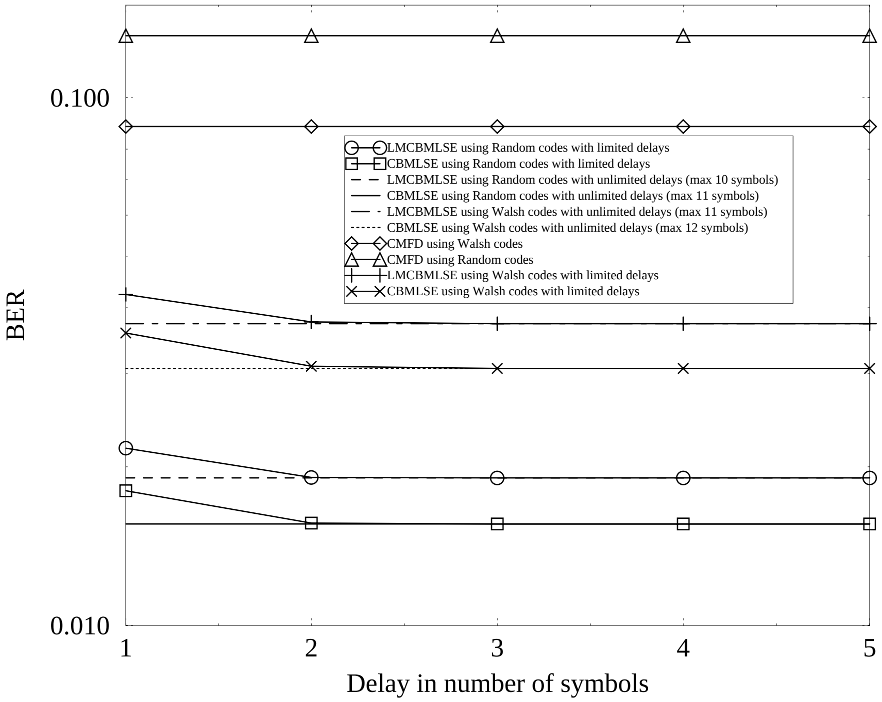

Figure 4.7: *BER against number of delays for $E_b/N_0 = 7$ dB and 4 users using codes of processing gain $G = 8$ in stationary multipath channels*

4.8 Summary

A MUD for the downlink of the DS-CDMA communications systems has been suggested. This detector operates at chip-level and uses the Euclidean distance, linear distance and Hamming distance for its decision rule. This chapter shows that the MLSE detector can be implemented at both the symbol and chip level. Simulation results of the LMCBMLSE have been compared with the CBMLSE, while the complexity comparison has been carried out on the MLSE since it has a simpler implementation. It has been found that the LMCBMLSE does not require any multiplication operation at run-time for the AWGN channel as well as the multipath channel, while multiplications are needed by the MLSE for the multipath channel. Although the number of addition operations is still exponential in the number of active users, the number of multiplications has been reduced. However, the implementation is still impractical due to the exponential complexity. It has been observed in the simulation results that the LMCBMLSE degrades only slightly from the MLSE, under the AWGN channel and multipath channels scenarios. The favourable truncation length of the VA has also been verified by simulations, which shows that for just a delay of 2 symbol-intervals, the performance has already approached that of the unconstrained VA.

Chapter 5
Folded state-transition trellis detector

It is found that the numbers of states of the MLSE and CBMLSE still increase exponentially with the number of active users. Even if linear distance is used, the complexity is still impractical for implementation. In this chapter, a new low-complexity scheme [111, 112] is suggested that also operates at chip level but uses a folded state-transition trellis instead of the original complete trellis shown in Sections 4.2 and 4.3. This new scheme exploits the feature of power-equality in the downlink to reduce the number of states involved. This detector will be referred to as the folded trellis detector (FTD). The various symbol-combinations in the MLSE are taken to be the different states in the decoding trellis. The folded trellis exploits the assumption of equal power for all the users ($P = P_k \ (k = 1, \ldots, K)$) to reduce the number of states involved. Due to the fact that the combined chip values of the spreading codes can only take discrete values, the number of states can be reduced by using these discrete values as the states in a folded state-transition trellis diagram and operating the VA at the chip level.

Section 5.1 introduces the FTD using single-chip state transition trellis in an AWGN environment and then extends it to the multi-chip trellis in Section 5.2. After which, the trellis for the multipath scenario is constructed in Section 5.3. The next section gives a detailed analysis of the complexity of the FTD in comparison to the MLSE detector. Simulation results for AWGN and multipath channels are presented in Section 5.5. Finally, in Section 5.6 some conclusions are drawn.

5.1 AWGN folded state-transition trellis

The transmitted signal in equation (2.6) can be rewritten as

$$V(z) \quad = \quad \sqrt{P} \sum_{f=0}^{F-1} z^{-Gf} \sum_{i=0}^{G-1} v_i(\mathbf{b}_f) z^{-i}, \tag{5.1}$$

where

$$v_i(\mathbf{b}_f) \quad = \quad \sum_{k=1}^{K} c_{i,k} b_{k,f} \qquad (i = 0, \dots, G-1) \tag{5.2}$$

takes discrete values from $-K$ to K in steps of 2 since $c_{i,k}$ and $b_{k,f}$ only take the two possible values, -1 or +1. Total number of discrete values is $K+1$. Let $v_{i,d}\,(d = 0, \dots, K)$ be all the $K+1$ possible values of $v_i(\mathbf{b}_f)$ and if all these values are used as the states in the trellis, the trellis is folded up. The folded trellis is generated based on the transmitted signal given by equation (5.2) and not the noiseless received signals even though they are scaled version of each other in an AWGN channel.

For the AWGN channel and considering only one symbol interval, the noiseless received signal in equation (2.10) becomes

$$X(z) = \sqrt{P} \sum_{i=0}^{G-1} v_i(\mathbf{b}) z^{-i}, \tag{5.3}$$

where $\mathbf{b}$ is the transmitted symbol vector in the symbol interval of interest. To illustrate the folding of the trellis, consider a 2-user case with spreading codes $\{$-1,-1,1,-1,-1,-1,1$\}$ and $\{$-1,1,1,-1,-1,1,1$\}$. Hence the resulting complete AWGN trellis is shown in Figure 5.1. Using the $K+1=3$ discrete values of -2, 0 and 2 as the states in the transition trellis diagram, the trellis can be folded up as shown in Figure 5.2.

From equation (5.3), the transition value pertaining to the transition path entering state $v_{i,d}$ is $x_i(\mathbf{b}) = \sqrt{P} v_{i,d} = x_i(v_{i,d})$ since $v_{i,d}$ and $\mathbf{b}$ are related. As more than 1 value of $\mathbf{b}$ can result in the same $v_{i,d}$, more than 1 symbol combination can be associated with each state. The symbol combinations associated with each state are given in the transition paths of Figure 5.2. For example, node (0,2) in Figure 5.2 corresponds to nodes (01,2) and (10,2) in Figure 5.1, hence the associated symbol combinations are 01 and 10. The number of states is reduced from 2^K (exponential) to $K+1$ (linear). Hence, the number of transition paths is reduced from $N_c = 2^K$ to $N_f \leq$ min$(2^K, (K+1)^2)$. The number of transition paths is obviously limited by 2^K (complete trellis case), and under the worst scenario, there might be $K+1$ transition paths emerging from each of the $K+1$ states, resulting in a total of $(K+1)^2$ transitions. However, in most cases, paths coincide and the actual number of transition paths is usually less than the upper bound, for example there are only 3 transition paths from chip 2 to chip 3 in Figure 5.2.

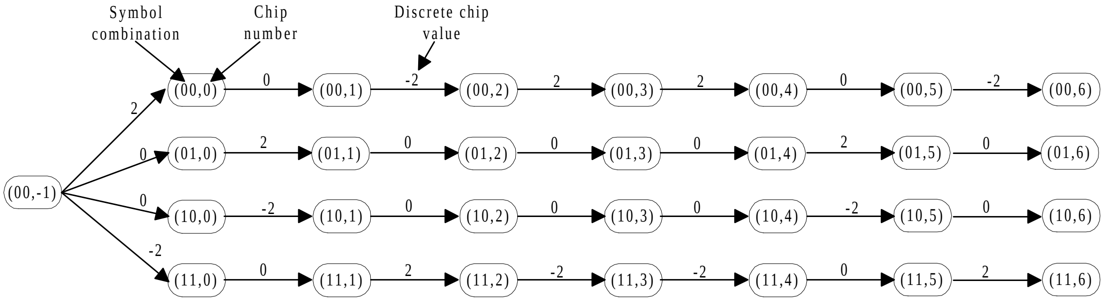

Figure 5.1: *Complete state-transition trellis*

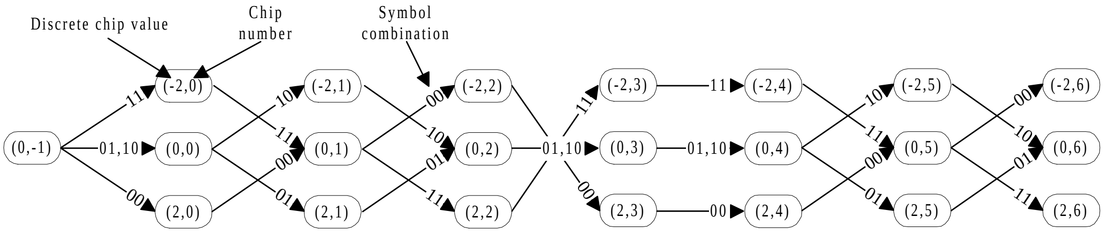

Figure 5.2: *Single-chip folded state-transition trellis*

Despite the reduction in the number of states, a one-to-one correspondence still exists between the two trellis and no information is lost as all the possible chip sequences can still be traced out. The Euclidean metric pertaining to each transition path in the trellis can be calculated in the same way as in the complete trellis case. Although one-to-one mapping between the combinations of transmitted bits and combinations of transmitted chips exists along the trellis, it is lost when the Viterbi algorithm is applied. The reason is that there are not enough paths between stages to represent all possible combinations of the transmitted bits. As a result of the crossings in the folded trellis, a decision is made based on the metric sums, whenever two or more transition paths converge at a state (the VA). The path associated with the smallest metric sum is selected and the other paths discarded. The metric values pertaining to the paths emerging from this node add on to this selected metric sum. At the end of the symbol interval, the survival path which corresponds to the smallest metric sum can be traced out. For example, if the survival path is $\{$(0,-1),(-2,0),(0,1),(-2,2),(2,3),(2,4),(0,5),(2,6)$\}$, then the symbol combinations associated with the survival path and the respective states are given in Table 5.1. As each state is associated with 1 or more symbol combinations, the total number of occurrences of each symbol combination is determined for the entire survival path. The symbol combination with the highest frequency of occurrences at the end of the interval is selected to be the decoded symbols. In the example, symbol combination 00 occurs 5 times while symbol combination 11 has 4 occurrences. Hence, 00 is the decoded symbol combination. If 2 or more symbol combinations have the same highest frequency of occurrences, then the complete trellis search is performed on this subset.

state	(-2,0)	(0,1)	(-2,2)	(2,3)	(2,4)	(0,5)	(2,6)
symbol combination	11	00,11	00	00	00	00,11	11

Table 5.1: *Symbol combinations associated with the states of the survival path*

The folded trellis is sub-optimum because a decision has to be made whenever two or more paths converge towards a state (the VA) within the folded trellis. For example, at node (0,1) in Figure 5.2, the metric sums corresponding to path $\{$(0,-1),(-2,0),(0,1)$\}$ and path $\{$(0,-1),(2,0),(0,1)$\}$ are compared, and the path with the smaller metric sum is selected while the other path is discarded. Path $\{$(0,-1),(-2,0),(0,1)$\}$ is selected in the survival path shown in Table 5.1. A wrong decision at this point due to large noise power causes the wrong transition path to be selected and thus the correct symbol-combination associated with the correct transition path is lost at this stage. It is obvious that no such decision has to be made for the complete trellis case and thus optimum performance is obtained. A wrong decision at a node does not necessary imply the wrong symbol-combination is decoded eventually. Consider the survival path given in Table 5.1, the actual transmitted symbol combination is 00, hence a wrong decision has been made at node (0,1) since the survival path contains node (-2,0) instead of (2,0). However, the right decision is still made eventually.

5.2 Multi-chip folded state-transition trellis

Multiple chips can be grouped together and taken as the states too. Consider the same 2-user example, Figure 5.3 illustrates how this can be achieved. The states used are all the possible combinations of n_c of the $K+1$ discrete values $v_{i,d}$ $(d = 0, \ldots, K)$ if there are n_c chips per state. Using multiple chips per state reduces the number of discarded paths and hence improves the performance since information is lost whenever a decision is made. Obviously, if all the chips are grouped together as one state, then the performance would approach that of the MLSE since there would be no decision-making until the end of the symbol-interval. Therefore, by varying the number of chips used per state, the performance lies between that of the MLSE and the single-chip folded trellis.

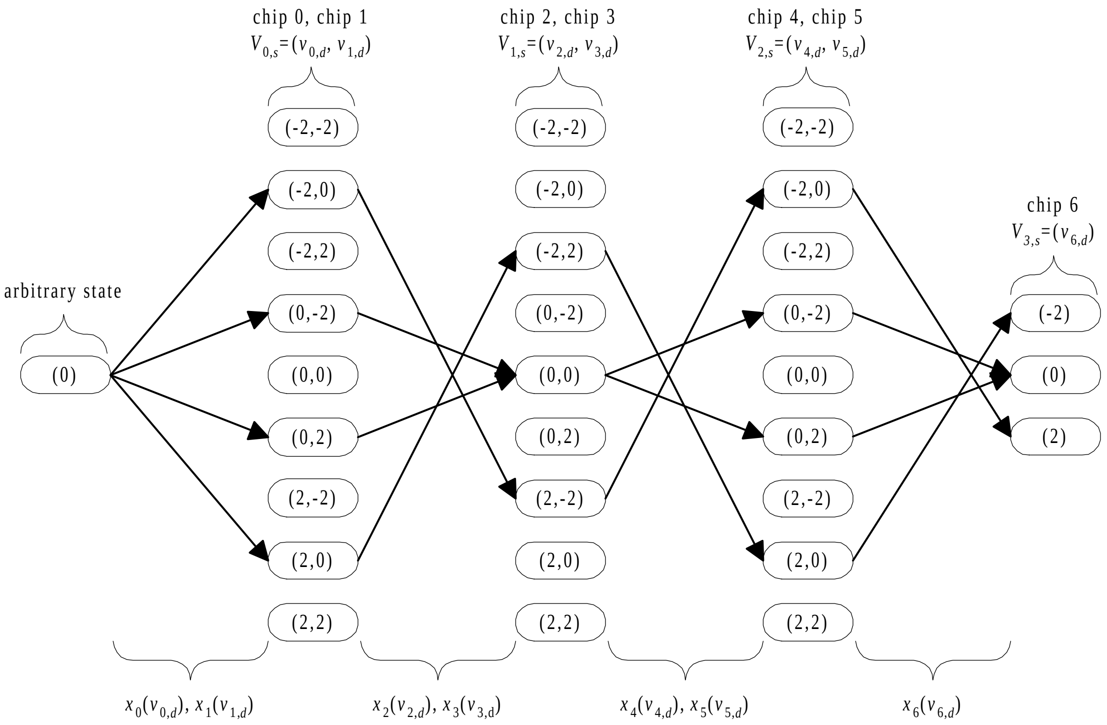
chip 0, chip 1
$V_{0,s}=(v_{0,d}, v_{1,d})$
chip 2, chip 3
$V_{1,s}=(v_{2,d}, v_{3,d})$
chip 4, chip 5
$V_{2,s}=(v_{4,d}, v_{5,d})$
chip 6
$V_{3,s}=(v_{6,d})$
arbitrary state
(0)
(-2,-2)
(-2,0)
(-2,2)
(0,-2)
(0,0)
(0,2)
(2,-2)
(2,0)
(2,2)
(-2)
(0)
(2)
$x_0(v_{0,d}), x_1(v_{1,d})$
$x_2(v_{2,d}), x_3(v_{3,d})$
$x_4(v_{4,d}), x_5(v_{5,d})$
$x_6(v_{6,d})$

Figure 5.3: *Folded state-transition trellis for AWGN channel with 2 chips per state*

Let $V_{n,s}$ $(s = 0, \ldots, (K+1)^{n_c} - 1)$ be the s-th state of group n. For the first $g = [G/n_c]$ groups where $\lfloor x \rfloor$ denotes the largest integer not greater than x, each state in these groups has n_c chips, i.e., $V_{n,s} = \{v_{n_c n,d}, \ldots, v_{(n+1)n_c-1,d}\}$ $(n = 0, \ldots, g-1)$. The last group $(n = g)$ uses the remaining chips, i.e. $V_{g,s} = \{v_{gn_c,d}, \ldots, v_{G-1,d}\}$ and has $(K+1)^{G-gn_c}$ different states. Except for the transition paths in the last stage, all other transition paths in the previous stages of the trellis comprise n_c noiseless received chip values, $x_i(v_{i,d}), \ldots, x_{i+n_c-1}(v_{i+n_c-1,d})$ $(i = n_c n; n = 0, \ldots, g-1)$. The transition values in the last stage are $x_{gn_c}(v_{gn_c,d}), \ldots, x_{G-1}(v_{G-1,d})$. The Euclidean metric for each transition path is calculated as

$$M(n, V_{n,s}) = \begin{cases} \sum_{j=n_c n}^{n_c(n+1)-1} (y_j - x_j(v_{j,d}))^2 & n = 0, \ldots, g-1 \\ \sum_{j=n_c n}^{G-1} (y_j - x_j(v_{j,d}))^2 & n = g \end{cases} . \tag{5.4}$$

The number of transition paths per symbol interval is reduced while the number of states is increased. The number of transition paths is $N_f \leq \min(2^K, (K+1)^{2n_c})$ and the number of states is $(K+1)^{n_c}$. The performance/complexity tradeoff of the multi-chip folded trellis scheme can easily be adjusted by varying the number of chips per state.

5.3 Multipath folded state-transition trellis

The folded trellis is also applicable to the multipath environment. From equations (2.5) and (5.1), the noiseless received signal, $X(z) = H(z)V(z)$, is given as

$$\begin{aligned} X(z) &= \sqrt{P}\sum_{l=0}^{L-1} h_l z^{-l} \sum_{f=0}^{F-1} z^{-Gf} \sum_{i=0}^{G-1} v_i(\mathbf{b}_f) z^{-i} \\ &= \sqrt{P}\sum_{f=0}^{F-1} z^{-Gf} \sum_{l=0}^{L-1} \sum_{i=l}^{G-1+l} h_l v_{i-l}(\mathbf{b}_f) z^{-i} \\ &= \sqrt{P}\sum_{f=0}^{F} z^{-Gf} \sum_{i=0}^{G-1} \sum_{l=0}^{L-1} h_l v_{\text{rem}(i-l+G,G)}(\mathbf{b}_{f+\lfloor (i-l)/G \rfloor}) z^{-i} \\ &= \sum_{f=0}^{F} z^{-Gf} \sum_{i=0}^{G-1} x_i(v_{\text{rem}(i-L+1+G,G),d}, \ldots, v_{\text{rem}(i+G,G),d}) z^{-i}, \end{aligned} \tag{5.5}$$

where rem(x, y) is the remainder of x divided by y. Again, the assumption that $\mathbf{b}_f = \mathbf{0}$ for $f = -1$ and $f = F$ has been made. Equation (5.5) and equation (4.1) are similar except that one expresses $x_i(\centerdot)$ in terms of the transmitted discrete chip values while the other expresses it in terms of the symbol combinations.

Due to the multipath effect, each received chip signal is affected by L consecutive chips if there are L paths as shown in equation (5.5). Hence, the number of states in the trellis is affected by the number of paths in the channel. For example, if $L = 3$, then the number of chips used must be greater than 1. This is because each noiseless received chip signal is dependent on 3 chips but each transition path only involves 2 successive chips if there is only 1 chip per state. Hence, the number of chips per state must be greater or equal to $L - 1$. L chips per state has been chosen in all examples so as not to increase complexity too much. The number of paths is about 5 in a 1 Mchips/s IS-95 downlink environment with a delay spread of 5μs, so the complexity is maintained at a reasonable level in practise.

Considering the same example, Figure 5.4 shows the folded state-transition trellis in a multipath case, assuming that the received signal comes from 2 different paths, i.e. $L = 2$. In general, the G chips are first divided into $g = \lfloor G/L \rfloor$ groups with L non-repeating chips each. Let $V_{n,s}$ $(n = 0, \ldots, G-(g-1)(L-1)-1; s = 0, \ldots, (K+1)^L - 1)$ be the s-th state of group

n. The states in group n $(n = 0, \dots, g-1)$ are hence $V_{n,s} = (v_{nL,d}, \dots, v_{(n+1)L-1,d})$. The remaining chips are grouped such that each time there is only 1 new chip, the other $L-1$ chips come from the previous stage, i.e., the states in group n $(n = g, \dots, G-(g-1)(L-1)-1)$ are $V_{n,s} = (v_{\text{rem}((g-1)(L-1)+n,G),d}, \dots, v_{\text{rem}(g(L-1)+n,G),d})$. The trellis extends to the next symbol interval and the groups repeat. The noise-free received chip signals or transition values in equation (5.5), $x_i(v_{\text{rem}(i-L+1+G,G),d}, \dots, v_{\text{rem}(i+G,G),d}) = x_i(V_{n,s}, V_{n+1,s})$ $(i = 0, \dots, G-1)$, are related to the discrete transmitted chip values, and in turn dependent on the states where each transition path originates and/or terminates. Note that if all the related chip values can be found in one state, i.e., $V_{n,s} = (v_{\text{rem}(i-L+1+G,G),d}, \dots, v_{\text{rem}(i+G,G),d})$, then the transition value is only dependent on one state, $x_i(V_{n,s})$. For the transition paths between the first g groups, there are L noise-free received chip signals associated with each path. These L associated transition values for each transition path emerging from state $V_{n,s}$ $(n = 0, \dots, g-2)$ are $x_i(V_{n,s}, V_{n+1,s}), \dots, x_{i+L-1}(V_{n,s}, V_{n+1,s})$ where $i = (n+1)L-1$. As for the transition paths between the remaining groups, only one noise-free received chip signal is associated with each path. Hence, for transition paths emerging from state $V_{n,s}$ $(n = g-1, \dots, G-(g-1)(L-1)-1)$, the noise-free received chip value for each path is $x_i(V_{n,s})$ where $i = \text{rem}(g(L-1)+n, G)$. The Euclidean metric for each transition path is calculated as

$$M_m(n, V_{n,s}, V_{n+1,s}) =$$
$$\begin{cases} (y_j - x_j(V_{n,s}))^2 & (n = g-1, \dots, G-(g-1)(L-1)-1; \\ & j = \text{rem}(g(L-1)+n, G)). \\ \sum_{j=(n+1)L-1}^{(n+2)L-2} (y_j - x_j(V_{n,s}, V_{n+1,s}))^2 & (n = 0, \dots, g-2) \end{cases} \tag{5.6}$$

Again, each state is associated with 1 or more symbol combinations. The occurrences of the symbol combinations determine the final decision on the decoded symbol vector as shown in the decoding procedure (Section 5.1). The repetition of the chips during the multipath effected transition stages repeats the occurrences of the symbol combinations, and since the metric sums of the paths are more reliable at the end of the interval due to the accumulation of the metrics, decisions made here should be more reliable than those made early in the symbol interval and thus put more weight on the correct symbol-combination, in a sense biasing the occurrences of the symbol-combinations. The right decisions for the last $L-1$ chips of the current interval imply the right decisions for the first $L-1$ chips in the next interval. Another reason for repeating the chips is to reduce the complexity which is discussed in Section 5.4.

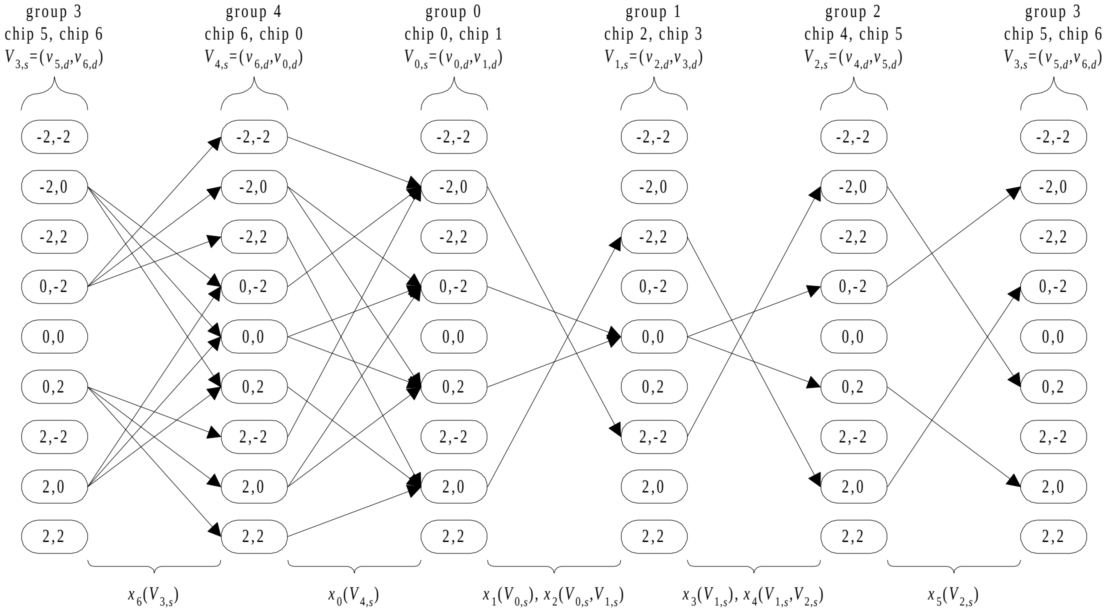
group 3
chip 5, chip 6
$V_{3,s}=(v_{5,d},v_{6,d})$
group 4
chip 6, chip 0
$V_{4,s}=(v_{6,d},v_{0,d})$
group 0
chip 0, chip 1
$V_{0,s}=(v_{0,d},v_{1,d})$
group 1
chip 2, chip 3
$V_{1,s}=(v_{2,d},v_{3,d})$
group 2
chip 4, chip 5
$V_{2,s}=(v_{4,d},v_{5,d})$
group 3
chip 5, chip 6
$V_{3,s}=(v_{5,d},v_{6,d})$
-2,-2
-2,0
-2,2
0,-2
0,0
0,2
2,-2
2,0
2,2
$x_6(V_{3,s})$
$x_0(V_{4,s})$
$x_1(V_{0,s})$, $x_2(V_{0,s},V_{1,s})$
$x_3(V_{1,s})$, $x_4(V_{1,s},V_{2,s})$
$x_5(V_{2,s})$

Figure 5.4: *Folded state-transition trellis for multipath channel with 2 chips per state*

As illustrated, the folded trellis for the multipath channel is still based on the transmitted discrete chip values. Another important point to note here is that the trellis remains exactly the same even if the coefficients of the multipath channel vary. The multipath effect only changes the transition values and not the number of states or transition paths. The transmitted signals are scaled version of the chip values in the state and remain the same regardless of the channel. Each transition path in the trellis represents one and only one transmitted sequence of L chips, even though it may arise from more than one symbol-combination. For example, the transition path from $(-2, 0)$ in group 3 to $(0, -2)$ in group 4 corresponds to a transmitted sequence with the fifth and sixth chip values equal to $-2\sqrt{P}$ and 0 respectively. This gives rise to one and only one noiseless received chip signal regardless of the coefficients of the multipath channel. This noiseless multipath-corrupted received chip value formulated by equation (5.5), $x_6(V_{3,s})$, is the transition value pertaining to this transition path.

5.4 Complexity analysis

5.4.1 AWGN channel

In this section, the complexity of the FTD and the MLSE are compared. The number of floating-point operations (FLOP) required per information symbol transmitted is determined.

First, consider the complexity of the FTD. As the nature of the transitions in the folded trellis depends on the spreading codes of the users, the exact structure of the folded trellis differs with different code sets. However, the computational complexity can still be determined without the knowledge of the actual transitions in each stage. For each transition path, the Euclidean metric, $(y_i - \sqrt{P}v_{i,d})^2$ $(i = 0, \ldots, G-1)$, is dependent on $v_{i,d}$ which is the value of the state where the path terminates. Hence, it is clear that this metric is the same regardless of the state where the transition path emerges. Therefore, all transitions entering a particular state have the same Euclidean metric and recomputation is unnecessary. In other words, only one metric computation is required for each state.

For every metric, the square of the difference between the observation and the hypothesis is calculated. Hence, the complexity of the metric sums calculation is $(3G - 1)(K + 1)$. Comparing the $K + 1$ metric sums at the end of the trellis and selecting the smallest sum requires K FLOP. Tracing back the survival paths and selecting the most frequently occurring symbol-combinations only requires fixed-point operations. Hence, complexity of the FTD is of the order of K.

Now, consider the complexity of the MLSE. The total run-time complexity is $K(2^K + G)$ FLOP, which includes the K matched filters' $K(G - 1)$ FLOP as shown in Section 4.6.1. The remaining operations are the calculation of the metric sums.

The complexity is summarised in Table 5.2. It is observed that the dominating factors in the FTD and the MLSE are $3GK$ and $K2^K$ respectively. Hence, there is significant reduction in complexity.

5.4.2 Multipath channel

In Figure 4.2, the number of states increases by a factor of 2^K at the junction of 2 symbol intervals, while in Figure 5.4, the number of states does not increase, even though the number

	Complexity (FLOP per symbol)
FTD	
metric sum computations	$(3G-1)(K+1)$
metric sum selection	K
MLSE	$K(2^K+G)$

Table 5.2: *Complexity comparison between the FTD and the MLSE detector in a AWGN channel*

of transition paths increases. Hence, in a multipath situation, the number of states per chip is 4^K (exponential in K) and $(K+1)^L$ for the complete trellis and the folded trellis respectively. The complexity of the complete trellis chip-based MLSE is hence 4^K as a metric has to be calculated for each transition path out of each state. Similarly, the complexity of the symbol-based MLSE is also 4^K as shown in equation (3.15) where the transition metric is dependent on the 4^K possible symbol combinations of the previous and current symbol intervals.

As the structure of the folded trellis depends on the nature of the spreading codes, the exact complexity varies with the code set. However, the lower and upper complexity bounds can be determined. The complexity depends on the number of metric computations. Since each transition path corresponds to a metric, the computational complexity is equivalent to the number of transition paths if the metrics in these paths are different. However, there are many transition paths with the same metric and recomputation is unnecessary. Consider the first multipath transition stage (from group 3 to group 4 in Figure 5.4 or from group $G-g(L-1)-1$ to group $G-g(L-1)$ in general). The $K+1$ transition values pertaining to the $K+1$ transition paths emerging out of each state are the same since each transition value, $x_6(V_{3,s})$, is only affected by $L=2$ chips, namely chip 5 and chip 6 (state value in group 3), which are the same for all these transition paths. Hence, the resulting Euclidean metrics of these transition paths, $(y_6 - x_6(V_{3,s}))^2$, are also the same. The computational complexity for this particular stage thus ranges from $K+1$ to $\min(2^K,(K+1)^L)$. The lower complexity bound of $K+1$ occurs if there are only $K+1$ active states (states from which transition paths enter and exit) in group 3. This is possible if the last 2 chip values of the 2 spreading codes are for example $\{$-1,-1$\}$ and $\{$1,1$\}$. This gives rise to only $K+1=3$ possible states, (-2,-2), (0,0) and (2,2). However, in other situations, the number of active states can be as high as 2^K (which is the case shown in Figure 5.4) or $(K+1)^L$ (which occurs if $2^K > (K+1)^L$ and paths originate from all the $(K+1)^L$ possible states) which gives rise to the complexity of 2^K or $(K+1)^L$ respectively. Thus, the number of states ($(K+1)^L$) serves as a constraint on the complexity when K becomes large.

Similarly, for transition paths in the next stage (between groups 4 and 0 in Figure 5.4 or all the transition stages from group $G - g(L-1)$ to group 0 in general), only the metric computation for one transition path emerging from each active state has to be performed, the remaining paths out of the same state have the same metric. The computational complexity of this stage ranges from $(K+1)^2$ to $\min(4^K, (K+1)^L)$. The lower bound corresponds to the case when there are $(K+1)^2$ active states in group 4 due to $(K+1)^2$ transition paths in the previous stage terminating in different states of group 4 assuming that each of the $K+1$ (minimum) active states in group 3 has $K+1$ transition paths. For this stage, the transition values, $x_0(V_{4,s})$, are affected by one chip in the previous interval (chip 6) and one chip in the current interval (chip 0), hence there is a maximum of 4^K possible transition paths since there are 4^K possible symbol-combinations, as shown in Figure 4.2. However, most of the symbol-combinations have similar chip 6 and chip 0 values, resulting in a reduction in the number of transition paths and also all paths with similar transition values emerge from the same state. As only one metric needs to be calculated for all transition paths out of each state, the upper complexity bound is thus the number of available states which is again $(K+1)^L$. Hence, if $4^K > (K+1)^L$, the complexity is constrained by the number of states. For example in Figure 5.4, complexity is $(K+1)^L = 9$ since $4^K > 9$, this is a reduction in complexity compared with the optimum MLSE (4^K). For a value of $K = 6$ and $L = 3$, the optimum MLSE is about 12 times more complex than the upper complexity bound of the folded trellis. For larger values of K, the complexity reduction is even greater. The second term of the upper complexity bound, $\min(4^K, (K+1)^L)$, can be viewed as the factor that limits the complexity to a feasible level.

The remaining stages in the trellis are not affected by the symbols in the adjacent interval. As discussed earlier the minimum possible number of active states is $K+1$, thus the minimum number of transition paths possible is also $K+1$ with the assumption of one path per state. In this portion of the trellis, the chip values are not repeated in the adjacent stage and the transition values are not only dependent on the state values in the group where the transition paths originate but also on the state values in the group where the paths terminate. Hence, even if the transition paths emerge from the same state, they give different metrics and so the computational complexity is equivalent to the number of transition paths. As the chip values are not repeated in the adjacent stage, there is no limitation to the transition paths here (Note that due to the fact that there is only 1 new chip value per state in groups 3, 4 and 0, each state can only transit to a maximum of $K+1$ other states in the next group). Hence, each state can transit to a maximum of $(K+1)^L$ states, resulting in a total of $(K+1)^{2L}$ possible transition paths. However, the number of possible transition paths cannot be greater than the number of possible symbol-combinations which is only 2^K since it is independent of symbols in the adjacent interval. Thus the upper complexity bound here is $\min(2^K, (K+1)^{2L})$. In Figure 5.4, the number of users is only $K = 2$, therefore the maximum number of transition paths is 4 as shown in the figure. Note that the chip values for groups 2 and 3 of Figure 5.4 have been repeated, as in the first portion of the trellis, the upper complexity bound here becomes $\min(2^K, (K+1)^L)$ which has reduced by a factor of $(K+1)^L$ from the former case. However, the repetition is not carried out for all the chips because the bottleneck of the complexity lies in the first portion of the trellis, since it can be as high as 4^K if it is not suppressed by the folded trellis. When $K = 11$ and $L = 3$, complexity of the multipath layers is $(K+1)^L$ which is then less than 2^K, hence in this situation it is a good idea to use repeated chip values throughout which is analogous to a folded trellis with a single chip per state.

Therefore, the dominating computational complexity lies in the expanded portion of the trellis, the lower and upper complexity bounds of the folded trellis for the multipath channel are thus $(K+1)^2$ and $\min(4^K, (K+1)^L)$ respectively. The complexity comparison between the FTD and the optimum MLSE detector is plotted in Figure 5.5. The figure shows that this scheme is best for a system with a small number of multipath components and a large number of users since its complexity grows exponentially with the number of multipath components and not with the number of users under these conditions. The complexity of the folded trellis is never greater than that of the original complete trellis MLSE according to the upper complexity bound. Furthermore, in most cases complexity is near the lower bound. For example, the complexity is usually $(K+1)^2 = 9$ (lower complexity bound) or $12 < \min(4^2, 3^3)$ for $K = 2$ and $L = 3$ as observed in the folded trellis of the same example as shown in Figure 5.6 except that now $L = 3$. Table 5.3 gives some codes with $G = 7$ that resulted in these two complexity values.

If $L = 5$ and $K = 8$, $(K+1)^L < 4^K$, thus upper complexity bound is $(K+1)^L = 59049$, transition paths emerging from all the possible states are expected if the actual complexity is equal to the maximum complexity. By randomly generating a set of spreading codes, it has been found that the folded trellis actually has a complexity of only 1424.

The folded trellis concept is not limited to the BPSK case. It can be easily adapted for M-ary PAM and M-ary PSK systems. For the M-ary PAM, where there are M possible symbol levels, the number of possible transmitted signals is $(M-1)K+1$ where K is the number of

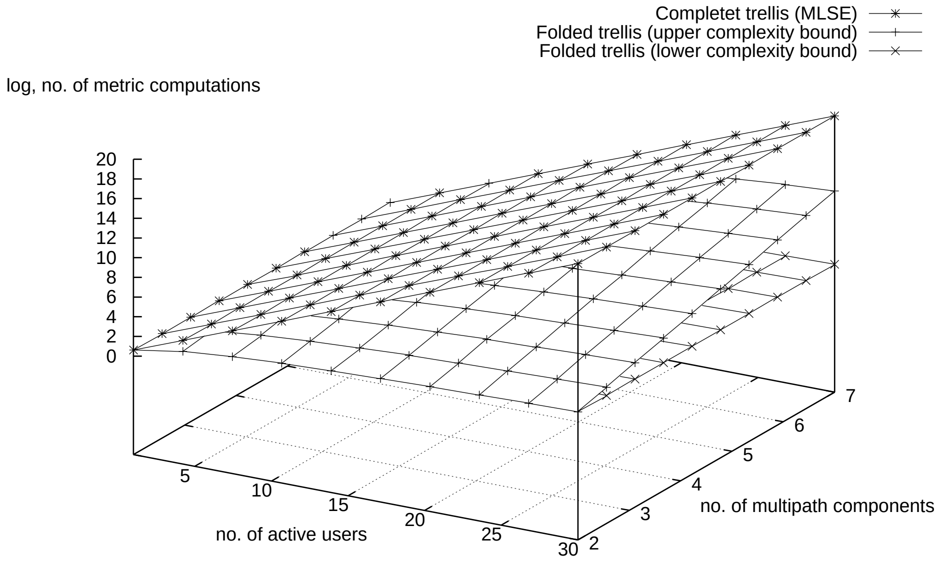

Figure 5.5: *Complexity comparison between the folded trellis detector and the optimum MLSE detector*

Complexity in multipath effected section (metric value calculations/chip)	Spreading codes
9	{-1 1 -1 -1 1 1 1} & {1 -1 -1 -1 1 -1 -1}
	{1 1 1 1 -1 -1 1} & {-1 -1 1 -1 1 -1 1}
	{-1 1 -1 -1 -1 1 1} & {-1 1 -1 1 1 1 1}
	{1 1 -1 -1 -1 1 -1} & {1 1 -1 -1 -1 1 1}
12	{1 -1 -1 -1 -1 -1 1} & {-1 -1 1 -1 1 1 -1}
	{1 -1 1 1 -1 -1 1} & {-1 -1 1 1 1 -1 -1}
	{-1 1 1 1 1 1 1} & {-1 -1 -1 1 1 1 1}
	{-1 -1 -1 1 -1 1 -1} & {-1 1 1 -1 -1 -1 1}

Table 5.3: *Table of actual folded trellis complexity for some pairs of spreading codes with $G = 7$, $K = 2$ and $L = 3$*

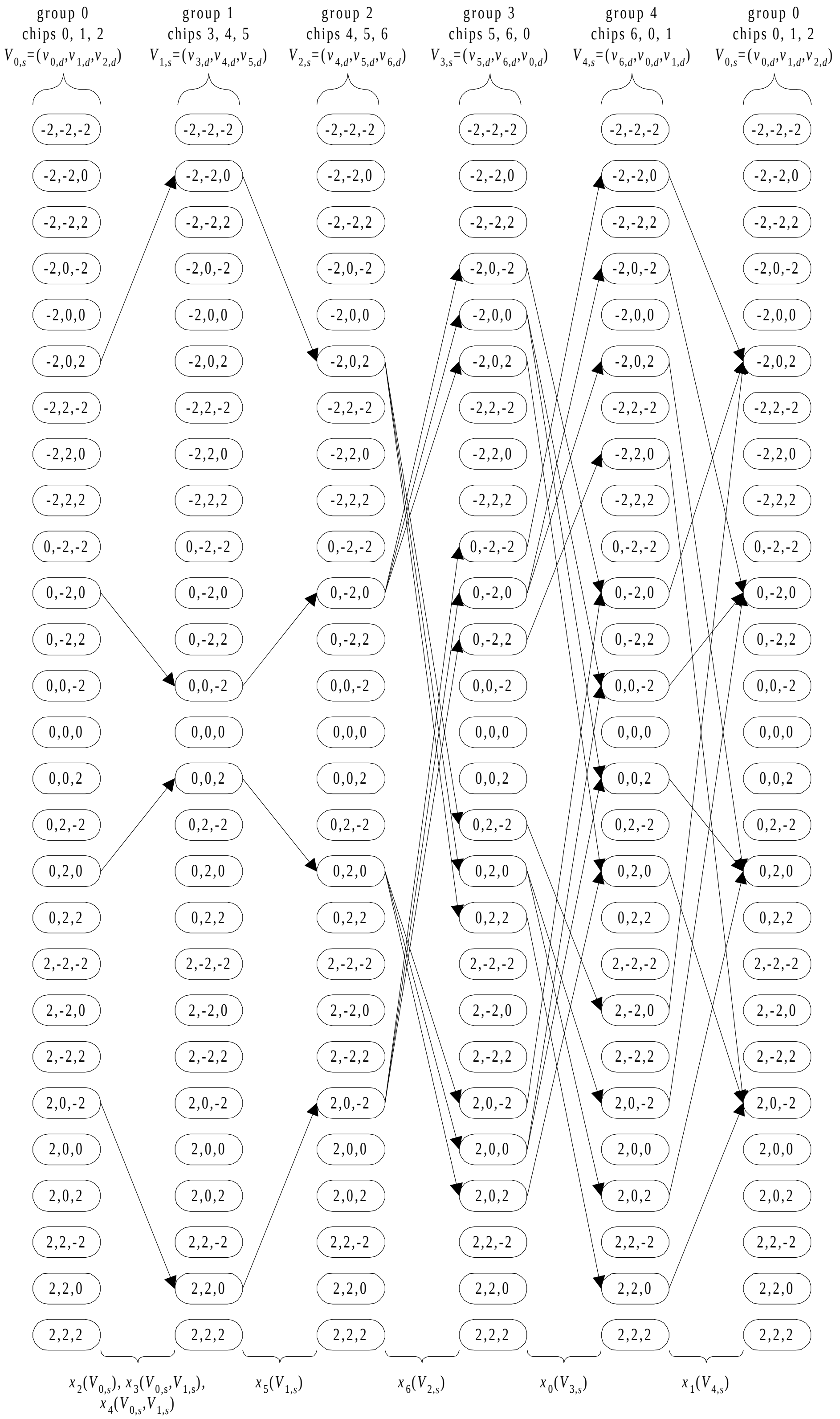

Figure 5.6: *Folded state-transition trellis for multipath channel with 3 chips per state*

users. This implies that there are $(M-1)K+1$ possible states (1 chip per state). Hence the complexity is $((M-1)K+1)^L$. There is no doubt an increase in complexity, but the penalty is definitely not as large as the MLSE which would have a complexity of M^{2K}.

For the M-ary PSK case, the number of states also increases. Consider the QPSK case, where the 4 possible symbols are $\{$(1,1),(-1,1),(-1,-1),(1,-1)$\}$. This can be viewed as BPSK in two dimensions. The trellis is the same as in the BPSK case, except that now we have 2 similar but independent trellis (if both dimensions use the same codes), one in the real dimension and the other in the imaginary dimension for the AWGN channel. As for the multipath channel, the structures of the trellis are the same if the numbers of multipath components are the same in both dimensions. Only the transition values are different if the multipath coefficients are different for the 2 dimensions. For higher order M-ary PSK, the symbol values in both dimensions can take more values but the folded trellis is still applicable. For instance, consider 8-PSK as an example, here the possible symbols are $\{(1,0)$, $(1/\sqrt{2})$, $(1/\sqrt{2})$, $(0,1)$, $(-1/\sqrt{2},1/\sqrt{2})$, $(-1,0)$, $(-1/\sqrt{2},-1/\sqrt{2})$, $(0,-1)$, $(1/\sqrt{2},-1/\sqrt{2})\}$. The 2 dimensions can still be considered with 13 states each. Hence, the complexity of each dimension becomes 13^L, which is again more favourable than the MLSE which has a complexity of 5^{2K} each dimension.

5.5 Simulation results

Monte-Carlo simulations have been used to obtain the bit error rate (BER).

5.5.1 AWGN channel

The FTD has been simulated for a system with equal-power users using random spreading codes with processing gain, $G = 7$. The FTD using the linear distance metric instead of the Euclidean distance metric is also simulated and is referred to as the linear metric FTD (LMFTD). Figure 5.7 shows the average BER for $E_b/N_0 = 9$ dB. Other receivers such as the CMFD, MMSE, LMCBMLSE, CBMLSE and MLSE, have also been simulated. The folded-trellis scheme (for both LMFTD and FTD) performs better than the MMSE. The LMFTD only degrades slightly from the FTD and is even indistinguishable from the MLSE for large MAI. Notice that for the single-user case, the FTD has the same performance as the MLSE because the 2 possible paths never crossed. Note that not only is each state in the trellis associated only with the symbol combinations, each transition path is also associated with the symbol combinations. For example, from Figure 5.2, state (0,6) is associated with symbol combinations, $\{01, 10\}$, while transition path from state (-2,5) to state (0,6) is associated with symbol combination $\{10\}$. The graphs labelled with "transition" in the plot of Figure 5.7 are the BERs if the occurrences of the symbol combinations associated with the transition paths of the survival path are considered instead. It is found that considering symbol combinations associated with the states performs better than considering the symbol combinations associated with the transition paths. Also plotted in Figure 5.7 are the graphs labelled "correlations". These correspond to detectors that perform the Viterbi algorithm to the folded trellis based on the linear distance metric and the Euclidean distance metric, and then pass the sequence of discrete chip values from the survival path through the respective matched filter of each user. In other words, it is similar to the CMFD except that the received signal is first processed by the folded trellis. As observed from the plot, the performance is better than the CMFD for small number of users. However, as the number of users increases, the performance becomes close to that of the CMFD. The reason is that at low MAI, the noise dominates and by applying the folded trellis, the noise is 'eliminated' since the resulting sequence takes only the discrete chip values. With the low MAI and the dominating noise removed, the matched filter improves performance. However, at high MAI, where MAI dominates, the performance cannot improve without considering the likelihood information which is to consider the likelihood of each symbol combination in terms of frequency of occurrence.

If G is now increased to 16 and E_b/N_0 reduced to 7 dB, the resulting performances of the FTD using different number of chips per state are plotted in Figure 5.8. The performance of the single-chip FTD is worse than the MMSE. This is because G has increased by more than twice its original value, and this has greatly benefited the MMSE detector. On the other hand, the FTD suffers because a hard decision is made whenever 2 or more paths converge at a state regardless of the length of the trellis which is directly proportional to the processing gain. If a wrong decision is made in the early stages of the trellis, the detrimental effect will propagate down the trellis. A wrong decision is made when the noise value in a chip is so large that it causes the noisy received signal vector to fall in the decision region of an incorrect noiseless received vector. Thus a wrong decision is mainly due to the noise and not the MAI. The erroneous effect is most detrimental for small number of users because the small MAI (greatly reduced due to the high G) is very beneficial to both the conventional receiver and the MMSE but not the FTD. However, for large MAI, the latter degrades less.

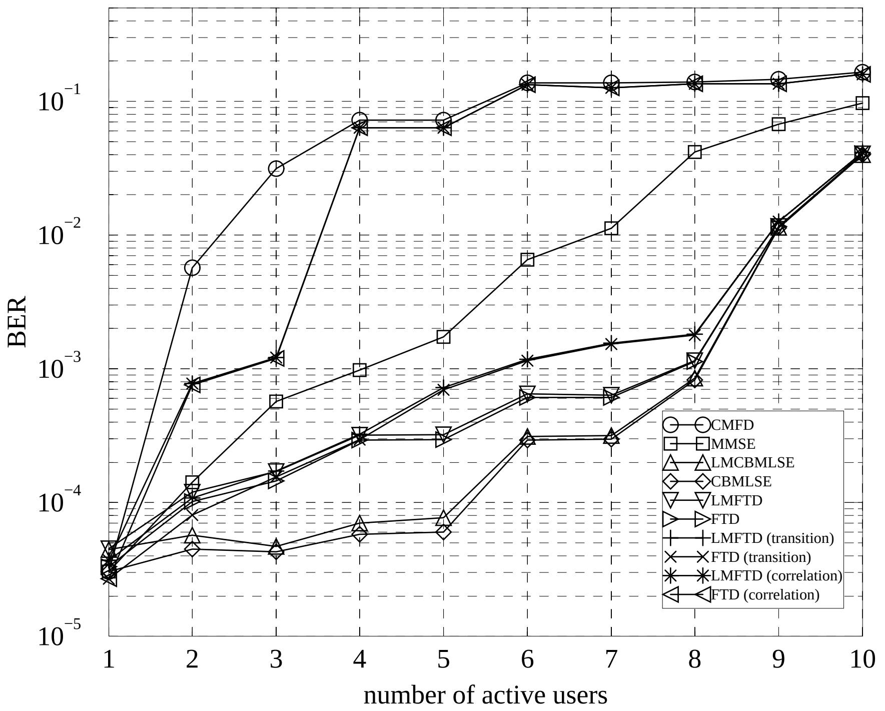

Figure 5.7: *BER against number of active users for $E_b/N_0 = 9$ dB and $G = 7$ in AWGN channel*

It is also observed that the performance improves by increasing the number of chips per state. If the number of chips is 4, the FTD has outperformed the MMSE. If 8 chips per state are considered then it has reached the performance of the MLSE. The high BER in the simulation results is due to the randomness of the spreading codes, low E_b/N_0 of the signals and that no channel coding has been used.

Next, the performance of the receiver using linear and Euclidean metrics is investigated. Figure 5.9 compares the performance of the linear and Euclidean metrics. It is found that as MAI increases, the degradation due to the linear metrics minimises. This is probably because the errors due to MAI are much more significant than that caused by the approximation.

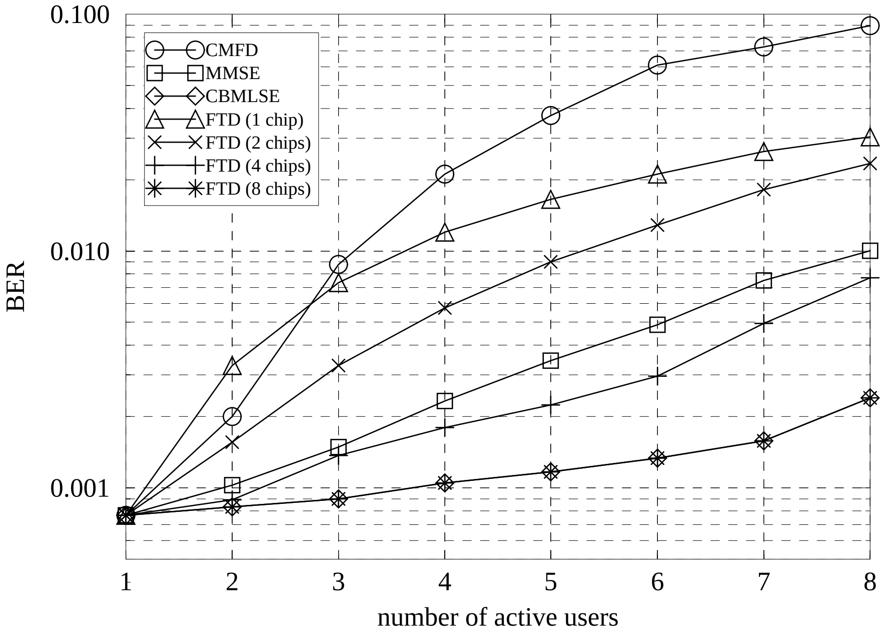

Figure 5.8: *BER against number of active users for $E_b/N_0 = 7$ dB and $G = 16$ using multichip folded trellis in AWGN channel*

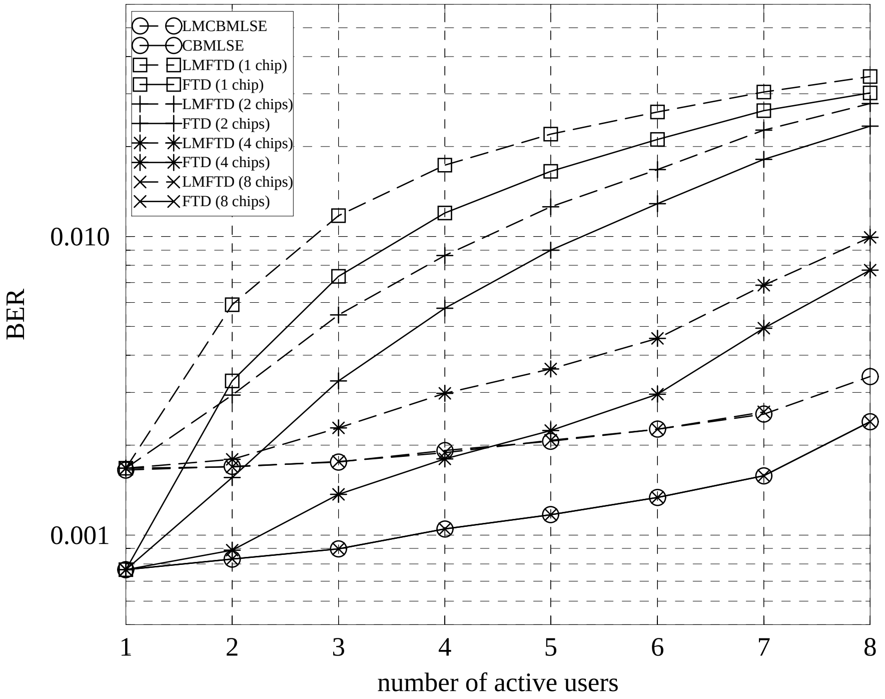

Figure 5.9: *Comparison between linear and Euclidean metrics for $E_b/N_0 = 7$ dB and $G = 16$ in AWGN channel*

5.5.2 Multipath channel

The FTD has been simulated for multipath channels with equal-power users using random spreading codes. First, a system with a processing gain of $G = 16$, E_b/N_0 of 7 dB and a channel impulse response of

$$H(z) = 0.227 + 0.466z^{-1} + 0.688z^{-2} + 0.466z^{-3} + 0.227z^{-4} \tag{5.7}$$

is considered. The simulation has been repeated using 200 frames with a frame size of 100000 symbols and each frame uses a different set of random codes. Figure 5.10 shows the simulation results. Other MUDs such as the RAKE receiver and the MMSE have also been simulated. The FTD uses $L = 5$ chips per state. Note that the MMSE detector in the multipath case has a filter length equal to $G + L - 1$ in order to capture all the signal energy that originated from the desired data symbol. The CBMLSE for the multipath case has also been plotted. The FTD performs very close to the CBMLSE. Note that for the single-user case, the FTD has the same performance as the MLSE because the two possible paths never crossed. Note that the FTD and the CBMLSE simulations have been carried out for a Viterbi truncation length of 2 symbol intervals in order to keep the length of the trellis at a feasible value.

A multipath environment with impulse response given in equation (4.3) has been simulated. This system uses a lower processing gain of $G = 7$. The simulation has been repeated using 5000 frames with a frame size of 100 symbols. Similar trends as those observed in Figure 5.10 are found in Figure 5.11.

Next, a non-symmetric multipath environment with impulse response ($L = 3$)

$$H(z) = 0.802 + 0.535z^{-1} + 0.267z^{-2} \tag{5.8}$$

is considered. Similar simulation conditions as those in Figure 5.11 have been used. This system uses a higher signal-to-noise ratio (SNR) of $E_b/N_0 = 9$ dB. The simulation results are plotted in Figure 5.12.

The figure shows that by increasing the SNR, the performance of the FTD is even closer to the MLSE detector. This is because the errors in the folded trellis are mainly due to noise and not due to MAI. The folding up of the original complete trellis does not remove any transition path from the trellis but merely combines paths that have similar transition values or noise-

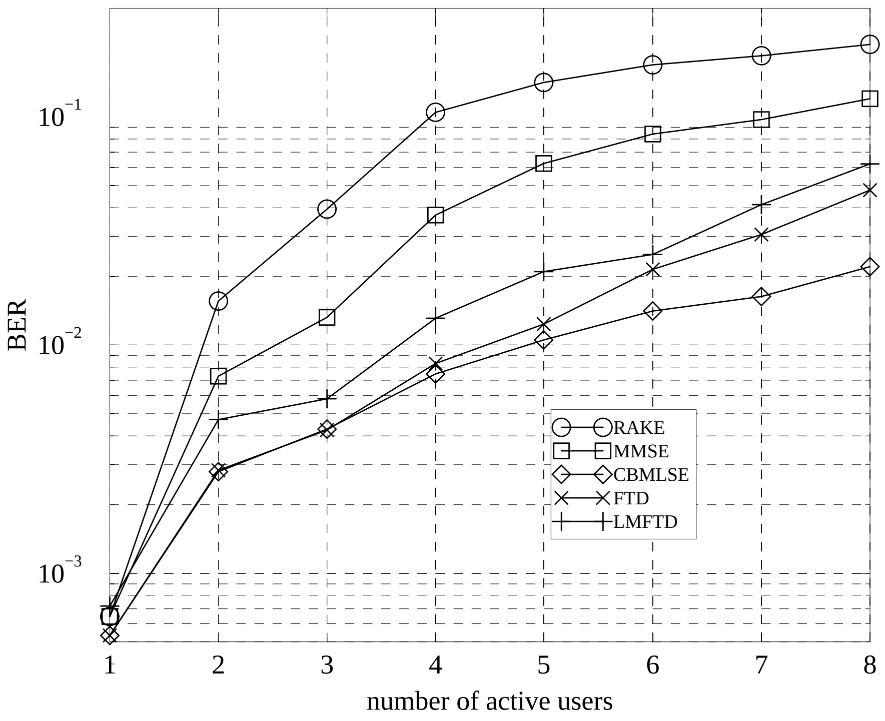

Figure 5.10: *BER against number of active users for $E_b/N_0 = 7$ dB and $G = 16$ in multipath channel*

less received values (even in multipath scenario) which happen frequently due to equal-power transmission. The folded trellis also reduces the number of states and results in convergence and divergence of transition paths. Due to the fact that no transition paths have been removed from the original trellis, all the possible hypothesis in each layer (or chip) are tested by the calculation of the square-law metrics, and errors are hence mainly due to noise.

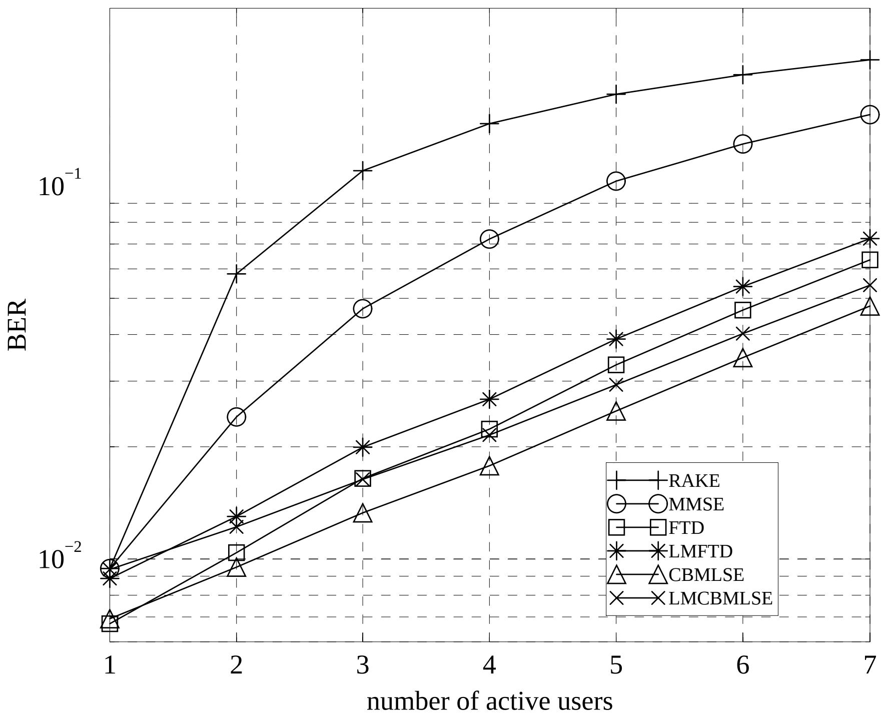

Figure 5.11: *BER against number of active users for $E_b/N_0 = 7$ dB and $G = 7$ in multipath channel*

5.6 Summary

The FTD folds up the original trellis and uses all the possible transmitted discrete chip values as the states instead of all the possible symbol combinations. The number of states is reduced from 2^K to $K+1$ for the AWGN channel and from 4^K to $(K+1)^L$ for the multipath channel. Consequently reducing the computational complexity from $O(2^K)$ to $O(K)$ for the AWGN channel and from 4^K to at most $\min(4^K, (K+1)^L)$ for the multipath channel. The receiver can also use the linear metrics for its decision rule instead of the Euclidean metrics without significant degradation in the performance, but with significant reduction in complexity.

The folded trellis has been illustrated for both the AWGN channel and the multipath scenario. Simulation results of the FTD have been compared with the MLSE. It has been observed in the simulations that the FTD does not gain very much from the processing gain due to the hard decision makings along the trellis. Fortunately, by using more chips per state, performance can be improved and eventually approaches that of the MLSE. It is also found that the performance of the FTD is close to the MLSE especially for large SNR, since the FTD is mainly limited by

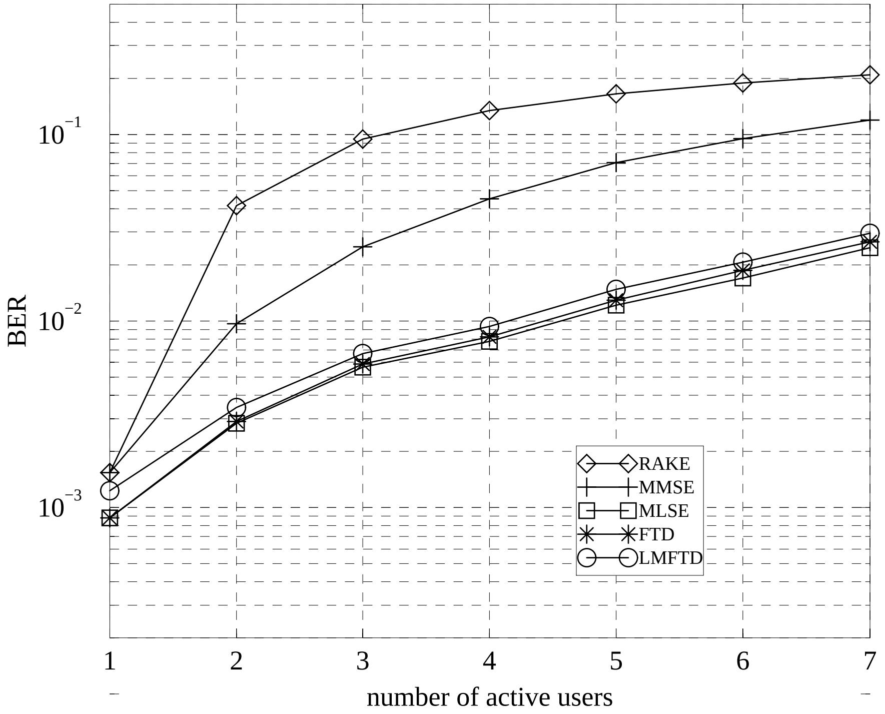

Figure 5.12: *BER against number of active users for $E_b/N_0 = 9$ dB and $G = 7$ in multipath channel*

noise and not MAI.

Chapter 6
Sub-optimum MLSE with folded trellis preselection

The maximum likelihood sequence estimator (MLSE) (discussed in Section 3.3.1) which finds the optimum data sequence that results in minimum joint probability, has exponential complexity growth in the number of users. All the possible symbol combinations are taken to be the different states in the decoding trellis of the MLSE. The folded trellis detector (FTD) that exploits the feature of power-equality in the downlink to reduce the number of states involved has been discussed in Chapter 5. There is of course a tradeoff between the performance and the complexity. The complexity of the FTD depends on the number of chips used per state. A small number of chips per state is favourable due to its low complexity but at the expense of performance. In this chapter, a better utilisation of the resources is suggested, since similar performance to the multi-chip FTD can be achieved with lower complexity. The basic idea here is to use the original folded trellis which has one chip per state to select those symbol combinations that are more likely to be transmitted, after which the MLSE criteria is performed on this much smaller subset of symbol combinations [113, 114]. If the transmitted symbol combination is always among those selected symbol combinations, the performance would be that of the MLSE. As the folded trellis has discarded most of the unlikely symbol combinations, the MLSE is only applied to a few symbol combinations, thereby limiting the complexity to a reasonable level.

The next section sets up the downlink AWGN channel model, and recaps the MLSE and the FTD in Section 6.1.1 and Sections 6.1.2 respectively. Section 6.2 looks at the new detector which comprises a FTD and a MLSE detector. The complexity of the new detector is compared with the optimum MLSE in Section 6.3. Simulation results are presented in Section 6.4. Finally, in Section 6.5 some conclusions are drawn.

6.1 The MLSE and the FTD

In this chapter, the focus is only on AWGN channel. In the synchronous AWGN downlink channel, each symbol interval is independent of the other symbol intervals without intersymbol interference. Hence, simply considering a symbol interval is sufficient. The assumption of an AWGN channel is for the ease of explanation and is not a limitation of the new detector as both the CBMLSE and the FTD can be extended to the multipath case as shown in Section 4.3 and Section 5.3 respectively.

From equation (2.10), the noiseless received signal in a symbol interval can be expressed as

$$X(z) \;=\; \sum_{i=0}^{G-1} x_i(\mathbf{b}) z^{-i}, \tag{6.1}$$

where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the input symbol vector in the interval of interest, $b_k \in \{-1, +1\}$ is the input symbol of the k-th user, P represents the equal received power of the user, and

$$x_i(\mathbf{b}) \;=\; \sum_{k=1}^{K} c_{i,k}\sqrt{P} b_k \tag{6.2}$$

is the i-th noiseless received chip signal in a symbol interval as a result of transmitting the data vector $\mathbf{b}$. The received signal is again $Y(z) = X(z) + N(z)$.

6.1.1 The MLSE detector

The MLSE detector in the multipath environment has been developed in Section 3.3.1. Here the equivalent MLSE for the AWGN channel is defined. Mathematically, the input symbol vector for one interval is decoded as

$$\begin{aligned} \hat{\mathbf{b}} &= \arg \min_{\mathbf{b}\in\{-1,+1\}^K} \sum_{i=0}^{G-1} [y_i - x_i(\mathbf{b})]^2 && (6.3) \\ &= \arg \max_{\mathbf{b}\in\{-1,+1\}^K} \left\{ 2\sum_{k=1}^{K} \sqrt{P_k} b_k \sum_{i=0}^{G-1} y_i c_{i,k} - \sum_{i=0}^{G-1} x_i^2(\mathbf{b}) \right\}, && (6.4) \end{aligned}$$

where y_i is the coefficient of z^{-i} in $Y(z)$ and $\sum_{i=0}^{G-1} y_i c_{i,k}$ is the output of the k-th user's matched filter [53]. Although it is independent of the message length, it still implies a brute-force exhaustive search over the 2^K possible sequences.

The CBMLSE for the AWGN channel is obtained if equation (6.3) is used while equation (6.4) yields the MLSE that operates at the symbol level in the AWGN channel. As stated earlier, both implementations have exactly the same performance theoretically, since no approximation has been made.

6.1.2 The FTD

In this section, the folded trellis detector (FTD) which has been introduced in Chapter 5, will be discussed briefly from another perspective. The folded trellis operates at the chip level using all the possible unit-power transmitted chip values as the states in the state-transition trellis, instead of using the symbol combinations. This reduces the number of states from 2^K (exponential) to $K+1$ (linear) and folds up the state-transition trellis, resulting in a reduction in complexity.

Let

$$v_{i,d} = \sum_{k=1}^{K} c_{i,k} b_k = 2d - K \qquad (d = 0, \ldots, K) \tag{6.5}$$

represents the d-th discrete value at the i-th chip and $\mathbf{v} = [v_{0,d}, \ldots, v_{G-1,d}]^T$ be the vector that represents a path in the folded state-transition trellis. Therefore,

$$x_i(\mathbf{b}) \quad = \quad \sqrt{P} v_{i,d}. \qquad (i = 0, \ldots, G-1) \tag{6.6}$$

If $\mathbb{V}_1$ is the set that consists of all the 2^K possible paths that correspond to the 2^K symbol combinations, then the minimum metric sum is given as

$$M_{\min}^{\mathrm{CBMLSE}} \quad = \quad \min_{\mathbf{v} \in \mathbb{V}_1} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} v_{i,d} \right]^2 . \tag{6.7}$$

The metric sum measures the distance between the received signal vector, $Y(z)$, and one of the 2^K possible noise-free received signal vectors, $X(z)$. The signal vector, $X(z)$ or $\mathbf{v}$, closest (minimum metric sum) to the observation $Y(z)$ has the highest likelihood of being the transmitted signal.

Up to this point, the folded trellis operates exactly as the CBMLSE. However, as a result of the crossings in the folded trellis, there are more possible paths (not limited to 2^K), as observed in Figure 5.2. For example, path {-2,0,-2,2,2,0,2} is one of the possible paths that is not included in the set $\mathbb{V}_1$. Let $\mathbb{V}_2$ be the set that consists of all the other paths, and $\mathbb{V} = \mathbb{V}_1 \cup \mathbb{V}_2$ be the joint set. If $\mathbb{V}_2$ is included into the metric sums calculation, then the minimum metric sum is obtained as

$$M_{\min}^{\mathrm{FTD}} = \min_{\mathbf{v}\in\mathbb{V}} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} v_{i,d} \right]^2 . \tag{6.8}$$

Direct implementation of equation (6.8) requires more computation. Fortunately, the Viterbi algorithm [36] can be used to carry out equation (6.8). Consequently, this reduces complexity from $O(2^K)$ to $O(K+1)$ (to be discussed in Section 6.3).

After the VA is applied to the trellis, the survival path with the minimum metric sum is selected. Since set $\mathbb{V}$ is used instead of set $\mathbb{V}_1$, it is possible that the trellis ends up with a minimum cost chip sequence which corresponds to none of the symbol combinations. For example, the minimum cost chip sequence, $\{$-2,0,-2,2,2,0,2$\}$, in the earlier example does not correspond to $\{$2,0,-2,2,2,0,-2$\}$, $\{$0,2,0,0,0,2,0$\}$, $\{$0,-2,0,0,0,-2,0$\}$ or $\{$-2,0,2,-2,-2,0,2$\}$ of symbol combinations 00, 01, 10 or 11 respectively. However, the situation is resolved with a voting system where the symbol combination that occurs most often in the minimum cost chip sequence (or survival path) is selected as the decoded symbol combination. If 2 or more symbol combinations have the same highest frequency of occurrence, then the complete trellis search (MLSE) is performed on this subset.

Multiple chips can be grouped together and taken as the states as discussed in Section 5.2. By varying the number of chips used per state, the performance lies between that of the MLSE and the single-chip folded trellis. However, the increase in complexity from $K+1$ (single-chip) to $\min(2^K, (K+1)^{n_c})$ (multi-chip) is exponential in n_c where n_c is the number of chips per state. The complexity approaches that of the complete trellis or the MLSE very rapidly.

6.2 The sub-optimum MLSE

In this section, the sub-optimum MLSE detector is introduced. It consists of a folded trellis first stage which uses the reliability information inherent in the trellis, followed by the CBMLSE with reduced number of symbol combinations. The idea here is to use the folded trellis to preselect a number of symbol combinations that are more likely to be transmitted based on the received signals. If the maximum likelihood rule is then applied to this reduced subset of symbol combinations, the complexity is greatly reduced.

Figure 6.1 shows the block diagram of the new receiver. The first stage consists of the FTD and the second stage the CBMLSE. The preselected n_s symbol combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$, are passed to the CBMLSE which makes used of this information to decode the received signal.

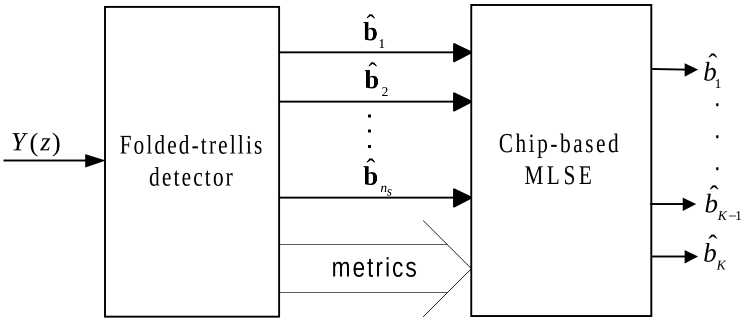

Figure 6.1: *Block diagram of the sub-optimum MLSE detector*

6.2.1 Reliability information

The probability density of the observation $Y(z)$ under the hypothesis that $X^{\,l}(z)$ is the noise free received signal is denoted as $p_l(Y(z))$. Assuming that each noise free received signal, $X^l(z), l = 1, \ldots, 2^K$, is independent and equally likely to be received, the likelihood function is

$$p_l(Y(z)) = \frac{1}{(\pi N_0)^{G/2}} \exp\left(-\frac{\sum_{i=0}^{G-1}(y_i - x_i^l)^2}{N_0}\right), \qquad (l = 1, \ldots, 2^K) \tag{6.9}$$

where x_i^l is the coefficient of z^{-i} in $X^l(z)$.

Let

$$M_l = \sum_{i=0}^{G-1}(y_i - x_i^l)^2 \qquad (l = 1, \ldots, 2^K) \tag{6.10}$$

represents the l-th metric sum obtained at the end of the complete trellis. The probability of making a right decision if the MLSE selects $X^{\,j}(z)$ is thus

$$P_j(c) = \frac{\exp(-M_j/N_0)}{\sum_{l=1}^{2^K}\exp(-M_l/N_0)}, \qquad (j = 1, \ldots, 2^K) \tag{6.11}$$

where M_j is the smallest metric sum. Therefore, the probability of making a wrong choice if the MLSE selects $X^j(z)$ is

$$\begin{aligned} P_j(e) &= 1 - P_j(c) \\ &= 1 - \frac{1}{1 + \sum_{l=1, l \neq j}^{2^K} \exp(-\Delta_{l,j})}, \qquad (j = 1, \ldots, 2^K) \end{aligned} \tag{6.12}$$

where $\Delta_{l,j} = (M_l - M_j)/N_0$ denotes the scaled difference between the metric sums. The probability $P_j(e)$ can be used as the reliability information for selecting $X^j(z)$. Since $\exp(-\Delta_{l,j})$ decreases exponentially, the sum of the exponentials, $\sum_{l=1, l \neq j}^{2^K} \exp(-\Delta_{l,j})$, is dominated by the smallest $\Delta_{l,j}$. Hence, $\sum_{l=1, l \neq j}^{2^K} \exp(-\Delta_{l,j}) \approx \exp(-\Delta_j)$ where Δ_j is the smallest difference. The approximation,

$$P_j(e) \approx 1 - \frac{1}{1 + \exp(-\Delta_j)}, \qquad (j = 1, \ldots, 2^K) \tag{6.13}$$

can be used simply as a factor to determine how reliable the selected metric sum is. If $\Delta_j \rightarrow 0$, then $P_j(e) \rightarrow 0.5$, since a small difference in the metric sums is more likely to cause an error in the selection of the metric sum. On the other hand, if $\Delta_j \rightarrow \infty$, then $P_j(e) \rightarrow 0$, thus minimising the errors in the selection of the metric sum. Interpreting it in the signal space, as the other metric sums get further away from the minimum metric sum, the less likely it is that the minimum metric sum is incorrect.

A sphere can be defined with radius r and centre at the signal vector, V_0, which gives the minimum metric sum, M_j. The degree of reliability of V_0 is dependent on the number of other signal vectors that falls within this sphere. If there are no other vectors within this sphere, then V_0 is very reliable. On the other hand, if there are other vectors within this circle, V_0 is unreliable. The parameter r is thus a threshold determining whether to consider other signal vectors or not.

6.2.2 Folded trellis first stage

Instead of selecting the smallest metric sum at the end of the trellis and tracing back along the trellis to obtain the minimum cost chip sequence, the new scheme selects the n_m smallest metric sums and then trace back from all the nodes associated with these metric sums. For each trace-back, the frequencies of occurrences of the various symbol combinations are again determined. The frequencies of all these symbol combinations are then compared and ranked in a decreasing order. The first n_s unique symbol combinations are chosen and passed to the next stage which is the CBMLSE.

The following describes the method of determining n_m. First preassign a value to the radius r and the maximum number of metric sums to be n_M. Let N_m be the number of metric sums within the sphere of radius r including M_j. The value of n_m is defined as

$$n_m = \begin{cases} N_m & (1 \leq N_m < n_M) \\ n_M & (N_m \geq n_M) \end{cases} . \tag{6.14}$$

If $\Delta_{l,j} > r$ for all $l\,(l = 1, \ldots, 2^K, l \neq j)$ and M_j is smallest, then there are no other metric sums within the sphere ($N_m = 1$) and M_j is reliable. If $N_m < n_M$, then $n_m = N_m$ and all the vectors within the sphere of radius r are checked by the CBMLSE. However, if $N_M > n_M$, then n_m is restricted to n_M and only the $n_M - 1$ metric sums closest to M_j, as well as M_j, are investigated. By setting a small value for r, most of the other metric sums will be outside the sphere, hence complexity is low but performance is not favourable since most metric sums are not considered. On the other hand, by setting a large value for r, most metric sums fall within the sphere. If trace-back performs on all of these, complexity would increase, hence n_M is used to limit the number of metric sums considered. However, setting n_M to a small value and $r = \infty$, does not always result in the best performance (shown in Section 6.4).

There are $K + 1$ metric sums, $M_0, \ldots, M_K$, at the end of the trellis corresponding to the $K + 1$ chip states. Without loss of generality, assume that M_0 is the smallest metric sum and the other metric sums are ranked in the order of increasing distance. Figure 6.2 shows the vector representation of all the metric sums for an example with $K = 7$. $M_1, \ldots, M_4$ are the other metric sums within the sphere of radius r centred at M_0. Metric sums, $M_5, \ldots, M_7$, fall outside the sphere. Hence, $N_m = 5$ and if $n_M = 4$, then $n_m = n_M = 4$ and trace-back is carried out along the trellis starting from the states associated with $M_0, \ldots, M_{n_m-1}$. Assume that there are N_i symbol combinations associated with the trace-back path of metric sum $M_i\,(i = 0, \ldots, n_m - 1)$. Out of the $N_0 + \ldots + N_{n_m-1}$ symbol combinations, the n_s unique (note that some symbol combinations may be associated with more than one metric sum) and most frequently occurring ones are selected and passed to the CBMLSE stage. The value of n_s affects the complexity of the second stage CBMLSE and must be kept small. However, it should be greater than 1, otherwise the CBMLSE becomes redundant.

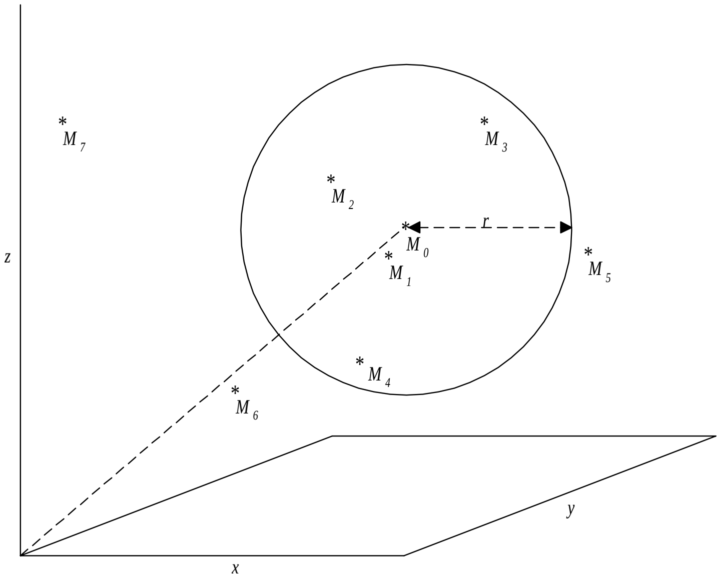

Figure 6.2: *A 3-dimensional signal vector space*

6.2.3 CBMLSE second stage

In the original FTD introduced in Chapter 5, the MLSE criteria is used to make a decision only when there are 2 or more symbol combinations with the same highest frequency of occurrence. This idea is extended here to include the n_s most frequently occurring symbol combinations from the n_m smallest metric sums.

The inputs of the CBMLSE are the metrics, $\left(y_i - \sqrt{P}v_{i,d}\right)^2$ $(i = 0, \ldots, G{-}1; d = 0, \ldots, K)$, and the n_s preselected symbol combinations from the previous stage. Let $\mathbb{B}$ be the set that consists of these n_s symbol combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$. Hence, the decoding rule becomes

$$\hat{\mathbf{b}} \quad = \quad \arg\min_{\mathbf{b}\in\mathbb{B}} \sum_{i=0}^{G-1} \left[y_i - x_i(\mathbf{b})\right]^2, \tag{6.15}$$

instead of equation (6.3). The CBMLSE selects from among the much smaller set of symbol combinations, the one with the smallest Euclidean distance. By limiting the values of n_m and n_s to reasonable values the complexity is maintained at a practical level.

Furthermore, it is recommended that the second stage is implemented with the CBMLSE instead of the MLSE since all the metrics, $\left[y_i - x_i(\mathbf{b})\right]^2$ $(i = 0, \ldots, G-1, \mathbf{b} \in \mathbb{B})$, have already been computed in the previous stage.

6.3 Complexity analysis

In this section, the complexities of the proposed sub-optimum detector and the optimum MLSE detector are compared. The number of floating-point operations (FLOP) required per information bit transmitted is determined.

First, the complexity of the sub-optimum scheme is investigated. As the nature of the transitions in the folded trellis depends on the spreading codes of the users, the exact structure of the folded trellis also differs from codes to codes. However, the computational complexity can still be determined without the knowledge of the actual transitions in each stage. For each transition path, the Euclidean metric, $\left(y_i - \sqrt{P}v_{i,d}\right)^2$ $(i = 0, \ldots, G-1)$, is dependent on $v_{i,d}$ which is the value of the state where the path terminates. Hence, it is clear that this metric is the same regardless of the state where the transition path emerges. Therefore, all transitions entering a particular state have the same Euclidean metric and recomputation is unnecessary. In other words, only one metric computation is required for each state.

For every metric, the square of the difference between the observation and the hypothesis is calculated. Hence, the complexity of the metric sums calculation is $(3G-1)(K+1)$. Comparing the $K+1$ metric sums at the end of the trellis and selecting the n_m smallest sums requires $K + \ldots + (K - n_m + 1) = n_m(2K - n_m + 1)/2$ FLOP. Tracing back the survival paths and selecting the n_s most frequently occurring symbol combinations only requires fixed-point operations.

The CBMLSE second stage only has to carry out the summation over the G chips for the n_s symbol combinations, since all the metrics have already been computed in the previous stage. This results in a complexity of $(G-1)n_s$ FLOP. The selection of the minimum metric sum out of the n_s metric sums requires $n_s - 1$ FLOP. Thus, the second stage complexity is small as compared with the first stage. Note that the optimum MLSE would require to sum up the metrics that correspond to all the 2^K symbol combinations. Thus the reduction in complexity due to the preselection of the small subset of symbol combinations is a factor of $2^K/n_s$.

Now, the complexity of the optimum MLSE is considered. The total run-time complexity is $K(2^K + G)$ FLOP as shown in Section 4.6.1, which includes the K matched filters' $K(G-1)$ FLOP. The remaining operations are the calculation of the metric sums. An additional $2^K - 1$ FLOP are needed for the selection of the minimum metric sum. This additional complexity has not been stated in Section 4.6.1 because it is the same for both the MLSE and the LMCBMLSE, hence the comparison is not necessary. If the symbol-based MLSE is used as the second stage in the proposed detector, then the additional complexity would be $(K+1)n_s + G(K-n_s)$ FLOP, even without considering the additional pre-calculations needed. The CBMLSE second stage does not require any precomputation while the MLSE needs $(G-1)K(K+1)/2 + n_s(K^2-1)$ FLOP as deduced from Table 4.1 if it is used to implement the second stage.

The complexity is summarised in Table 6.1. As n_m and n_s are small, the dominating factors in the sub-optimum detector and the MLSE are $3GK$ and $K2^K$ respectively. Hence, there is a significant reduction in complexity.

	Complexity (FLOP per information bit)
Folded trellis first stage:	
metric sums computations metric sums selection	$(3G-1)(K+1)$ $n_m(2K-n_m+1)/2$
CBMLSE second stage:	
metric sums computations metric sum selection	$(G-1)n_s$ n_s-1
Optimum MLSE:	
metric sums computations metric sum selection	$K(2^K+G)$ 2^K-1

Table 6.1: *Complexity comparison between the sub-optimum MLSE detector and the MLSE detector in a AWGN channel*

6.4 Simulation results

Monte-Carlo simulations are again used to obtain the bit error rate (BER). The sub-optimum scheme has been simulated for a system with equal-power users using random spreading codes and $E_b/N_0 = 7$ dB. These codes, which are randomly generated for every trial in the simulation, are considered because they allow for interferer diversity. Figure 6.3 shows the average BER against the number of active users for $G = 16$, $r = \infty$ and $n_M = 3$. Graphs of the sub-optimum MLSE are plotted for $n_s = 1, \ldots, 4$. Other MUDs such as the MMSE, the optimum MLSE, 1-chip FTD and 2-chip FTD have also been simulated. The sub-optimum MLSE scheme performs better than the MMSE and very close to the optimum MLSE for $n_s > 1$. As n_s increases from 1 to 3, the performance improves. Thereafter, no further improvement is observed. Notice the great improvement over the 1-chip FTD (which is in fact the sub-optimum MLSE with $n_M = 1$ and $n_s = 1$) and the 2-chip FTD, when the CBMLSE second stage is added. When $n_s = 1$, the undesirable performance shows that the most frequently occurring state is not necessarily the correct one, especially when metric sums, other than the smallest one are considered. As r is set to infinity, even highly unreliable metric sums are considered. This can be resolved by considering the reliability of the metric sums (shown in the last simulation results in this section). Furthermore, by selecting just 1 symbol combination, the second stage CBMLSE becomes redundant. It is simply the original FTD taking into consideration other unreliable metric sums. Therefore, the conclusion is that increasing n_M without increasing n_s has an adverse effect on the performance and n_s should be kept close to n_M.

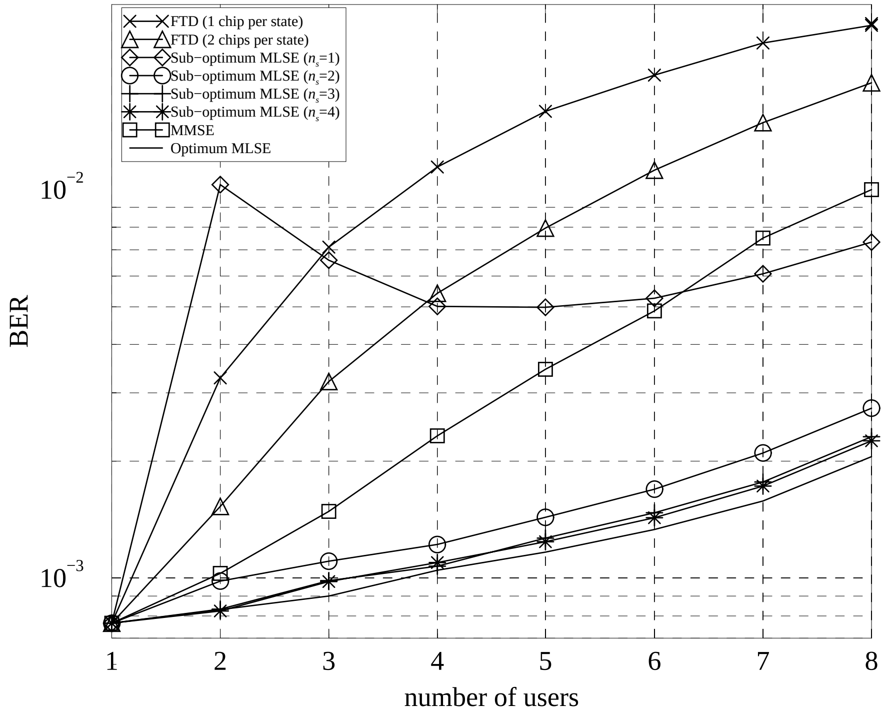

Figure 6.3: *BER against number of active users for $E_b/N_0 = 7$ dB, $G = 16$, $r = \infty$ and $n_M = 3$*

The conditions used in Figure 6.4 are the same as in Figure 6.3 except that here n_s is fixed at 2 and n_M varies from 1 to 4. It is observed that the performance improves as n_M is increased up to 3. For $n_M = 4$, there is no significant improvement. When $n_M = 2$, performance is already better than the 2-chip FTD which has a higher complexity of $\min(2^K, (K+1)^2)$. For this scenario, setting $n_M = 3$ and $n_s = 2$ is the optimal trade-off with respect to performance and complexity. Even for 8 users, when there are 256 symbol combinations, the sub-optimum MLSE only has to consider 2 symbol combinations to give near optimum performance.

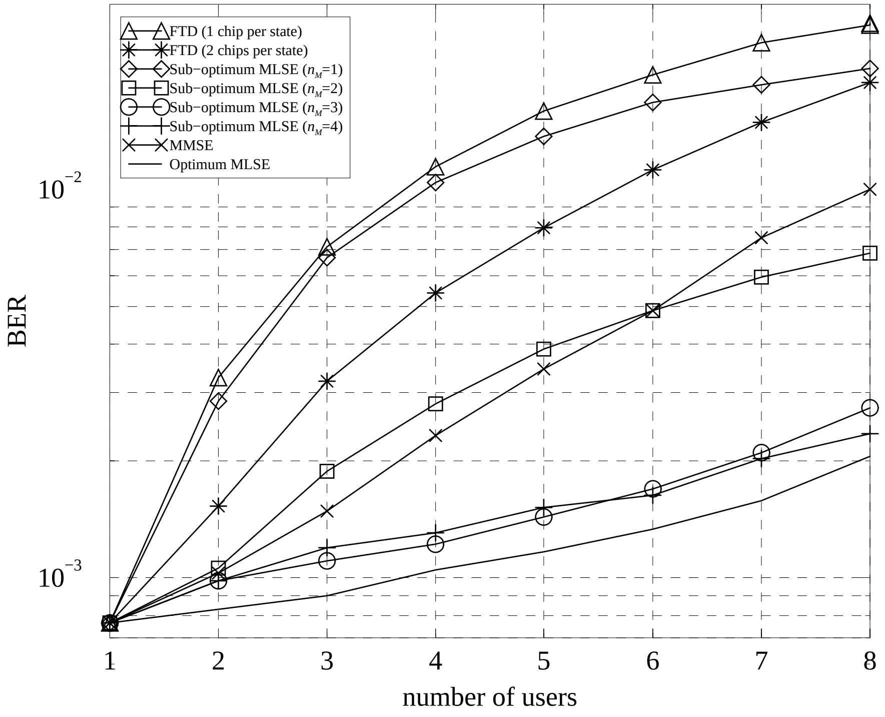

Figure 6.4: *BER against number of active users for $E_b/N_0 = 7$ dB, $G = 16$, $r = \infty$ and $n_s = 2$*

In Figure 6.5, the effect of "overloading" the system is investigated. Here the processing gain, G, is set to be 7 and the number of users is increased up to 10 which is greater than G. From the figure, it is clear that with just $n_M = 2$ and $n_s = 2$, the performance of the sub-optimum MLSE has already reached that of the MLSE even under excessive loading. For a BER of 0.05, the proposed detector can have 2.5 users more than the MMSE and 6 users more than the CMFD. Using the complexity table, it is found that for 10 users, the complexity of the MLSE is 45 times greater than the sub-optimum MLSE. The graph for $n_M = 1$ and $n_s = 1$, which is equivalent to the original FTD without the CBMLSE second stage, is also plotted. Here the great performance gain over the MMSE is observed especially for high loading, without even exploiting the CBMLSE second stage.

In Figure 6.6, the effect of varying r is investigated. The conditions are the same as in the simulation results shown in Figure 6.3. The difference is that here the BER against r is plotted for a 2-user system using $n_s = 1$ and $n_s = 2$. It is observed that for $n_s = 1$, as r increases, the BER decreases initially. However, on increasing r further performance degrades rapidly. This explains the unsatisfactory performance of the sub-optimum MLSE if $n_s = 1$ and $K = 2$

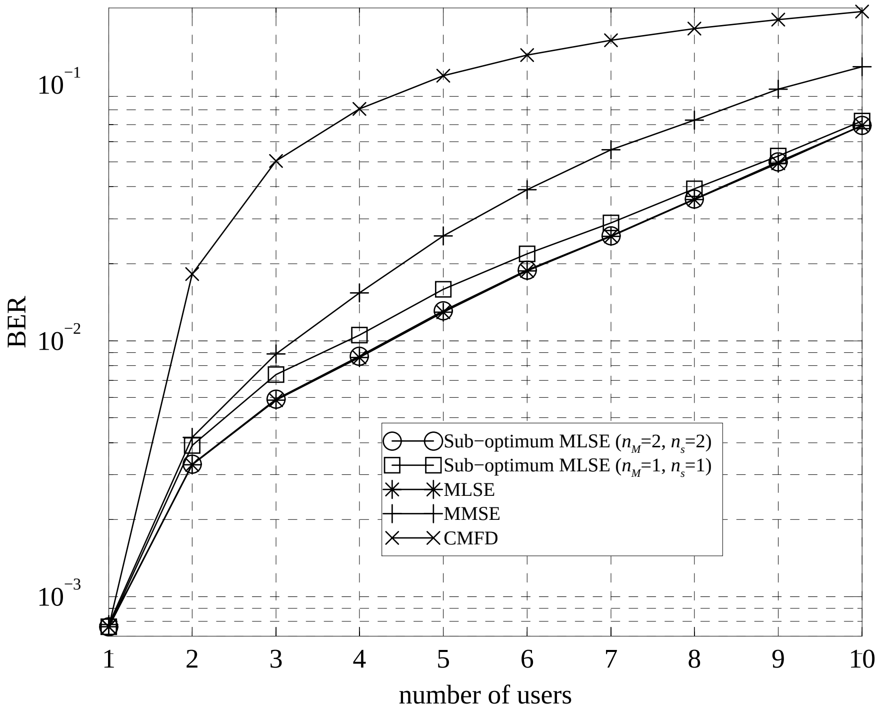

Figure 6.5: *BER against the number of active users for $E_b/N_0 = 7$ dB, $G = 7$ and $r = \infty$*

as seen in Figure 6.3. For $n_s = 2$, performance gradually improves as r increases. Note that if $n_s = 1$, the second stage is unnecessary, but simulation results show that if $n_m > 1$ and $n_s = 1$, performance can be worse than the original FTD. It has been observed that for $n_s = 1$, the best performance is obtained only for a particular value of the radius, r. As this value is difficult to derive theoretically due to the high non-linearity of the detector, it is recommended that $n_s = 1$ is avoided. As long as $n_s \neq 1$, $n_m \neq 1$, and n_s is kept closed to n_m, r does not have to be considered, or simply sets to infinity.

6.5 Summary

In this chapter, a MUD for the equal-power downlink of the DS-CDMA systems is investigated. This detector consists of two stages, namely the FTD stage and the CBMLSE stage. The MLSE and the FTD have been revisited. The FTD has been explained from another perspective, that is to consider the criteria for selecting the minimum metric sum. The main idea of the new detector is to use the folded trellis to preselect a subset of the symbol combinations and then apply the CBMLSE to this much smaller subset. Instead of considering the minimum metric

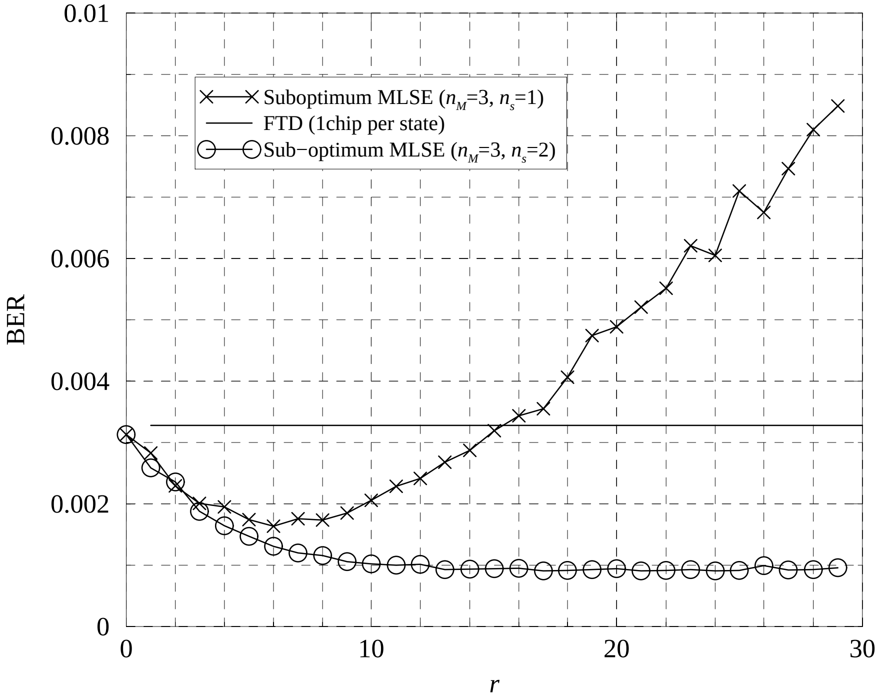

Figure 6.6: *BER against the radius of sphere for $E_b/N_0 = 7$ dB, $G = 16$ and 2 users*

sum only, a preassigned number of the smallest metric sums are considered, and a preassigned number of the most frequently occurring symbol combinations are selected instead of only the most frequently occurring one. Out of the set of preselected symbol combinations, the one with the minimum cost is selected based on the MLSE criteria.

Basically the preassigned number of metric sums, n_m, and preassigned number of symbol combinations, n_s, have to be kept small. The threshold r does not have to be considered if $n_s > 1$, $n_m > 1$ and n_s is kept near to n_m. It has been shown that by using the folded trellis as a pre-selection stage for the MLSE, the performance of the new sub-optimum MUD can be near that of the optimum MLSE. The complexity of the MLSE is greatly reduced from $O(2^K)$ to $O(K)$ due to the small subset of symbol combinations selected by the folded trellis. In short, the low complexity of the folded trellis and the high performance of the MLSE have been exploited, resulting in a better trade-off between complexity and performance. This new detector performs much better than the multi-chip folded trellis detector.

Chapter 7
Iterative folded trellis multiuser receiver

In this chapter, the powerful iterative algorithm used in Turbo codes is employed in the folded trellis multiuser detector to yield a sub-optimum iterative multiuser receiver for the downlink of a forward error correction (FEC) coded direct-sequence code-division multiple access (DS-CDMA) system [115]. The receiver iterates between the 2 coding dimensions, namely the spreading codes and the FEC codes. The optimum iterative multiuser receiver uses the optimum decentralised single-user detector to generate the single-user extrinsic information for the single-user decoders. This chapter suggests a sub-optimum detector using a folded trellis preselection stage for the multiuser signal detection, in order to reduce the dominating complexity of the optimum decentralised detector. The work in this chapter is similar to that of [29, 30], except that instead of the M-algorithm, the folded-trellis sub-optimum MLSE [111, 113] is employed.

The next section describes the FEC coded DS-CDMA downlink system model. Section 7.2 looks at the sub-optimum iterative multiuser receiver which is a partitioned receiver that iterates between multiuser signal detection and channel decoding using a sub-optimum Bayesian detector (Section 7.2.1) and single-user MAP decoders (Section 7.2.2) respectively. The sub-optimum scheme is suggested to reduce the high computational complexity of the signal detection portion. The complexity of the proposed sub-optimum structure with respect to the optimum architecture is analysed in Section 7.3. Simulation results are presented in Section 7.4. Finally, in Section 7.5 some conclusions are drawn.

7.1 System model

This section discusses the model of the downlink of a discrete-time FEC coded DS-CDMA system. The system model used is shown in Figure 2.4, except that only the AWGN channel is considered in this chapter for simplicity.

The binary information data, $d_k(t) \quad (k = 1, \ldots, K; t = 1, \ldots, \tau)$, are convolutionally encoded with constraint length κ and code rate R. Without loss of generality, it is assumed that $R = 1/n$, for every input data bit, each encoder produces n symbols. The τ data bits include $\kappa - 1$ trailing zeros, thus the actual number of transmitted data bits is $\tau - \kappa + 1$. The code-bits are BPSK modulated, yielding data symbols $b_{k,f} = \{-1, +1\} \quad (k = 1, \ldots, K; f = 1, \ldots, J)$, where $J = n\tau$ is the number of code symbols in one frame.

Although the channel is AWGN, the entire transmission sequence has to be considered instead of just one symbol interval. This is because the soft information obtained from the entire encoded signal is used in the iteration process. Hence, the symbol intervals are distinguished by adding the subscript f to represent the f-th symbol interval. From equation (6.1), the noiseless received signal can be expressed as

$$X_f(z) \quad = \quad \sum_{k=1}^{K} S_k(z)\sqrt{P} b_{k,f} = \sum_{i=0}^{G-1} x_{i,f}(\mathbf{b}_f) z^{-i}, \qquad (f = 1, \ldots, J) \tag{7.1}$$

where $\mathbf{b}_f = [b_{1,f}, \ldots, b_{K,f}]^T$ is the f-th input symbol vector, and $x_i(\mathbf{b}_f)$ is the i-th noiseless received chip signal in the f-th symbol interval as a result of transmitting the symbol vector $\mathbf{b}_f$. Equal-power condition ($P_k = P \forall k = 1, \ldots, K$) has been assumed. The received signal, is now $Y_f(z) = X_f(z) + N_f(z)$ where $N_f(z)$ is the AWGN sequence with mean 0 and two-sided power spectral density equals to $N_0/2$.

7.2 The iterative multiuser receiver

The optimum iterative multiuser receiver, shown in Figure 3.4 has been discussed in Section 3.5.1. It comprises a CDMA Bayesian detector and single-user maximum-*a posteriori* (MAP) decoders. In this chapter, a sub-optimum Bayesian detector is suggested to reduce the prohibitive computational complexity. Figure 7.1 shows the block diagram of the sub-optimum iterative multiuser receiver where the sub-optimality is due to the sub-optimum Bayesian detector. As shown in the diagram, the sub-optimum Bayesian detector consists of a folded trellis sub-optimum MLSE followed by a single-user likelihood calculation. The channel decoding portion of the receiver remains the same since the dominating complexity lies in the signal detection portion.

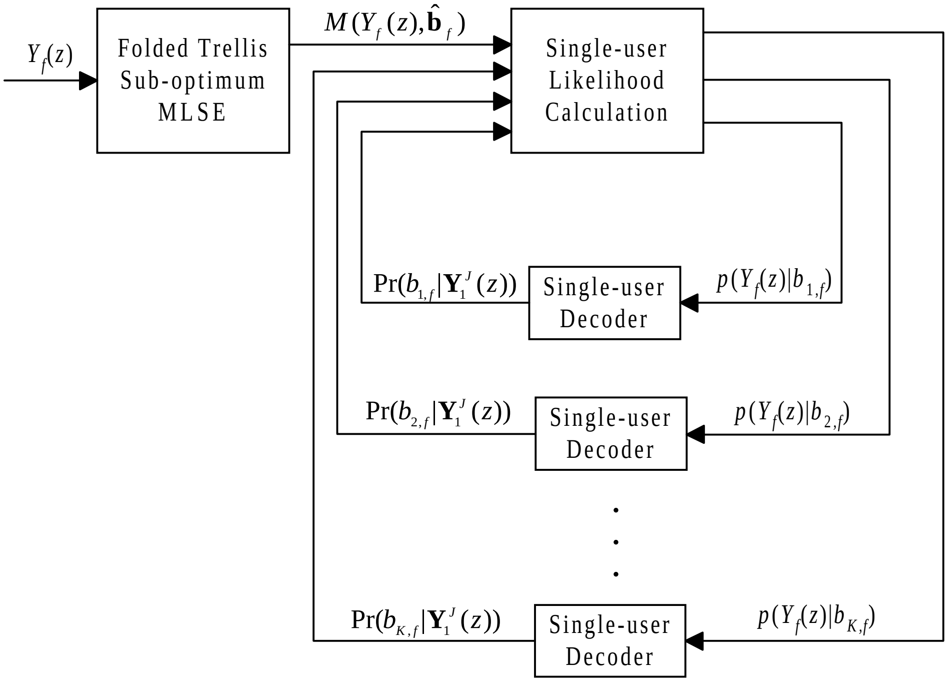
$Y_f(z)$
Folded Trellis Sub-optimum MLSE
$M(Y_f(z),\hat{\mathbf{b}}_f)$
Single-user Likelihood Calculation
$\Pr(b_{1,f}|\mathbf{Y}_1^J(z))$
Single-user Decoder
$p(Y_f(z)|b_{1,f})$
$\Pr(b_{2,f}|\mathbf{Y}_1^J(z))$
Single-user Decoder
$p(Y_f(z)|b_{2,f})$
$\Pr(b_{K,f}|\mathbf{Y}_1^J(z))$
Single-user Decoder
$p(Y_f(z)|b_{K,f})$

Figure 7.1: *An iterative multiuser receiver using the folded trellis sub-optimum MLSE for signal detection*

7.2.1 Sub-optimum Bayesian detector

To reduce complexity, the sub-optimum MLSE suggested in Chapter 6 is used to find a subset of the 2^K possible joint likelihoods. The sub-optimum MLSE uses the chip-based folded trellis preselection first stage to choose n_m of the $K+1$ metric sums, and then out of the associated symbol vectors, $\mathbf{b}_f$, the $n_s \ll 2^K$ ones that are more likely to be transmitted are selected. This piece of information is passed to the CBMLSE second stage which computes the Euclidean distances of these n_s preselected symbol vectors. The only difference between the folded trellis sub-optimum MLSE used here and that suggested in Chapter 6 is that the minimum Euclidean distance selection is not carried out here, all the n_s Euclidean distances in each symbol interval,

$$M(Y_f(z),\hat{\mathbf{b}}_f) \quad = \quad \sum_{i=0}^{G-1}(y_{i,f}-x_{i,f}(\hat{\mathbf{b}}_f))^2, \qquad (f=1,\ldots,J) \tag{7.2}$$

are passed to the single-user likelihood calculation stage. Utilising these calculated Euclidean distances, the joint likelihoods of these symbol vectors are then determined using equation (3.29). The numerator in the exponential term, is the Euclidean distance or metric sum determined by the second stage of the sub-optimum MLSE. Define $\mathbb{U}_k^b$ as the set that consists of all the preselected symbol vectors, $\hat{\mathbf{b}}_f$, in this subset that has $b_{k,f} = b$. Hence, using equation (3.27), the single-user likelihood is approximated as

$$p(Y_f(z)|b_{k,f} = b) \approx \sum_{\mathbb{U}_k^b} \exp\left(-\frac{M(Y_f(z), \hat{\mathbf{b}}_f)}{N_0}\right) \prod_{\substack{i=1 \\ (i \neq k)}}^{K} P(b_{i,f}). \quad (k = 1, \ldots, K; f = 1, \ldots, J) \tag{7.3}$$

This approximation is accurate if the actual transmitted symbol vector falls within this subset since it has the highest likelihood value if $N_0/2 \to 0$. The folded trellis has high probability of selecting the actual transmitted symbol vectors and other symbol vectors with high likelihood values. Moreover, the likelihood function is exponential, thus the less likely symbol vectors have very low likelihood values and can be ignored. This method is similar to that proposed in [29] but the latter uses the M-algorithm to find the subset.

7.2.2 Single-user MAP decoder

In Section 3.5.1.2, the *a posteriori* probabilities have been derived in equation (3.37). However, the information, $p(Y_f(z)|b_{k,f} = b)$, is included in $\Pr(b_{k,f} = b|\mathbf{Y}_1^J(z))$ and thus it does not constitute new information. In order to obtain the extra information of the symbol, $b_{k,f}$, gleaned from the prior information about the other symbols based on the structure of the spreading codes and independent of the symbol in question, this prior information of the symbol in question has to be removed. Hence, from equation (3.36) and following the similar method to [109], the extrinsic information obtained across the convolutional code for symbol $b_{k,f}$ becomes

$$p_{\text{ext}}(b_{k,f} = b; \mathbf{Y}_1^J(z)) = \frac{\sum_{W_{t,f,k}^b} \sigma_t^k(m^-, m)}{p(Y_f(z)|b_{k,f} = b)}. \quad (t = 1, \ldots, \tau; k = 1, \ldots, K; f = 1, \ldots, J) \tag{7.4}$$

The extrinsic *a posteriori* probabilities are therefore

$$\Pr{}_{\text{ext}}(b_{k,f} = b|\mathbf{Y}_1^J(z)) = \frac{p_{\text{ext}}(b_{k,f} = b; \mathbf{Y}_1^J(z))}{p_{\text{ext}}(b_{k,f} = +1; \mathbf{Y}_1^J(z)) + p_{\text{ext}}(b_{k,f} = -1; \mathbf{Y}_1^J(z))}. \quad (k = 1, \ldots, K; f = 1, \ldots, J) \tag{7.5}$$

By making the following assignment,

$$\Pr(b_{k,f}=b) \quad = \quad \Pr_{\mathrm{ext}}(b_{k,f}=b|\mathbf{Y}_1^J(z)), \qquad (k=1,\ldots,K; f=1,\ldots,J) \quad (7.6)$$

the *a posteriori* probabilities generated by the K single-user decoders can be used as the *a priori* information in the sub-optimum Bayesian detector, specifically equation (7.3).

7.3 Complexity analysis

Since single-user decoders are used, the complexity of the decoder is $O(2^{\kappa})$ which is low as κ is usually small. The optimum Bayesian detector has a complexity of $O(2^K)$ which dominates for a system with a large number of users. Hence, the optimum MLSE in the optimum Bayesian detector is substituted with the sub-optimum MLSE. The complexity reduction of the folded trellis sub-optimum MLSE has been discussed in Section 6.3. The complexity of the folded trellis, which has $K+1$ states, is in the order of $K+1$ since the transition paths converging into the same state have the same transition metric and recomputation is unnecessary. The complexity of the Euclidean distance ($M(Y_f(z),\hat{\mathbf{b}}_f)$) computation depends on the number of possible symbol vectors selected (n_s), and is hence of the order n_s. The computation of the joint likelihoods by the sub-optimum MLSE does not have to be repeated every iteration. For every iteration, equation (7.3) is recomputed, only the *a priori* information, $\Pr(b_{i,f})$, changes, the exponential joint likelihoods, remain the same throughout. The computational complexity is thus $O(n_s)$ per iteration. By keeping n_s to a small value such as $K+1$, the complexity is kept low, in the order of $K+1$.

7.4 Simulation results

Monte-Carlo simulations have been used to obtain the bit error rate (BER). A system with equal-power users using random spreading codes of processing gain, $G=7$, is considered. Random codes are used in a synchronous environment because the results can be generalised to a asynchronous system and/or a system with multipath effect using orthogonal codes. The same convolution codes with rate, $R=1/2$, constraint length, $\kappa=3$, and generators (7,5) in octal notation, are employed for all users. The size of each frame is 128 data information bits, giving $\tau=130$. A total of 10,000 frames are transmitted. The $J=260$ sets of encoded symbols of the K users in each frame are transmitted with different sets of randomly generated spreading codes.

In [29], as well as in Section 3.5.1.2, the single-user decoders do not extract the extrinsic information. Equation (3.37) is simply taken to be the extrinsic information. In [109], as well as in Section 7.2.2, equation (7.5) is used instead, to obtained the extrinsic information. By simulation, it is observed that if extrinsic information is used, the performance is better. The results in terms of BER for 3 iterations are plotted in Figure 7.2. In the first iteration, the receivers show the same performance since the extrinsic information only affects the subsequent iterations. The simulation results verify that the use of extrinsic information improves performance. Note the huge improvement in performance at the second iteration for both cases. As the soft outputs become more and more correlated after every iteration, the improvement drops rapidly. Further iterations show similar trend. The FEC coded single-user bound has been included to show how close to single-user performance the iterative multiuser detector/decoder can achieve.

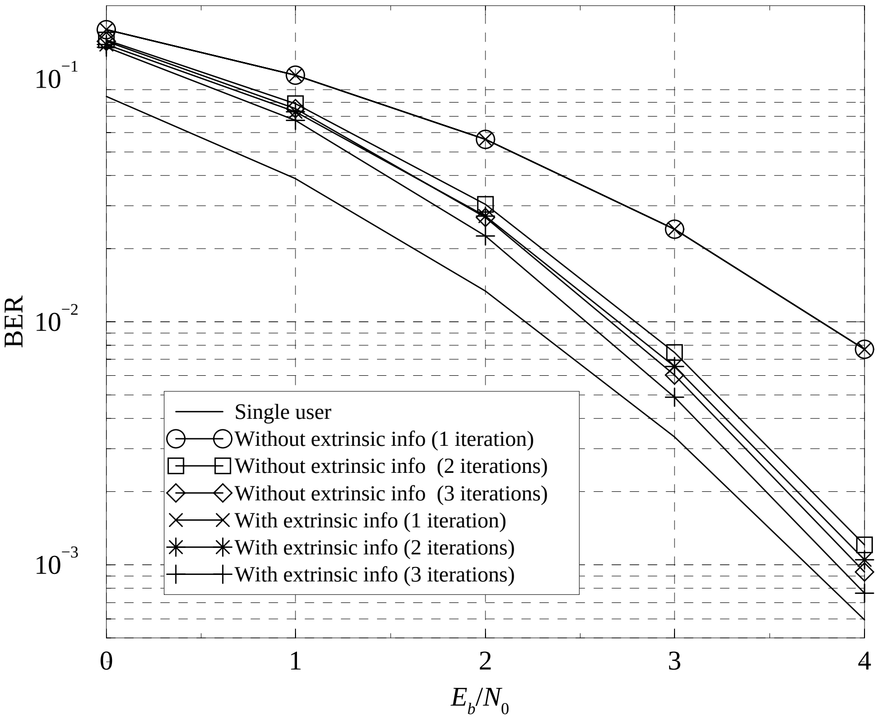

Figure 7.2: *BER against E_b/N_0 for a system with $K = 5$ users using the optimum Bayesian iterative multiuser detector*

Figure 7.3 shows the average BER against E_b/N_0 for 5 users in the same system. Graphs of the folded trellis sub-optimum iterative receiver are plotted up to 3 iterations. Further iterations do not produce significant improvement in performance. The simulation has been carried out for $n_m = 3$, and $n_s = 6$ or $n_s = 15$. Extrinsic information is considered. From the figure, it is found that if the number of symbol vectors considered is increased from 6 to 15, performance does not improve significantly. Comparing with the performance of the optimum iterative receiver shown in Figure 7.2, it is observed that considering 6 out of the 2^5 symbol vectors is sufficient to produce near-optimum performance.

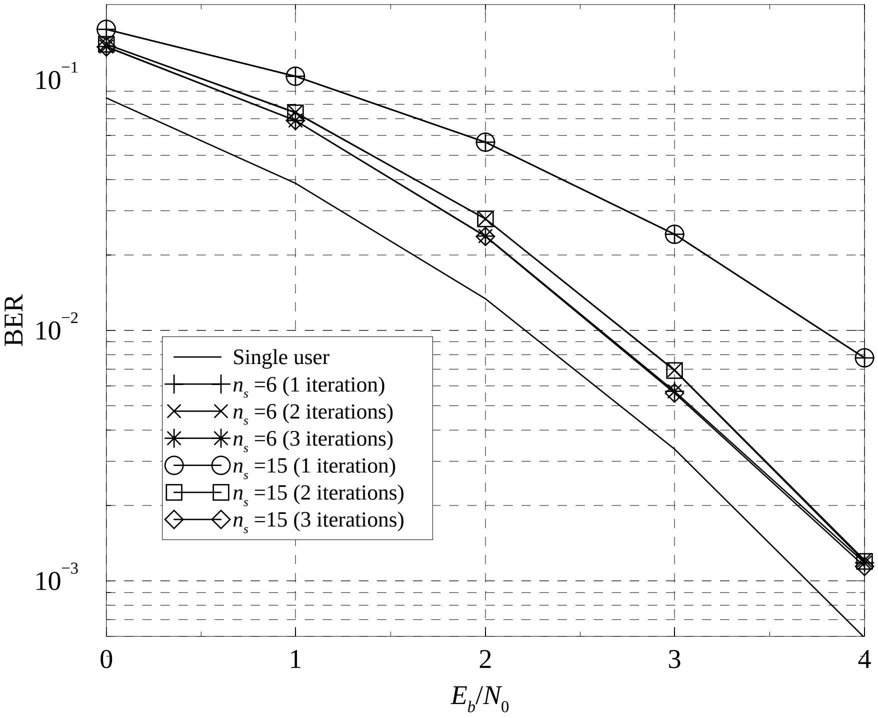

Figure 7.3: *BER against E_b/N_0 for a system with $K = 5$ users using the sub-optimum Bayesian multiuser detector ($n_m = 3$)*

Figure 7.4 shows the average BER against E_b/N_0 for 8 users in the same system. Hence, the system is highly loaded ($K/G > 1$). Graphs of the folded trellis sub-optimum partitioned detector/decoder and the optimum partitioned detector/decoder are both plotted up to 3 iterations. Further iterations do not produce significant improvement in performance. Extrinsic information is again considered for both receivers since it gives better performance. The folded trellis selects the $n_m = 3$ smallest metric sums and out of the associated symbol vectors, the $n_s = K + 1 = 9$ most frequently occurring ones are chosen. From the figure, it is observed that for each iteration, the performance of the optimum and sub-optimum detectors are very close to each other, except at higher E_b/N_0 where the higher power signals bring the performance of the optimum receiver even closer to the single-user bound. This shows that the reduction in complexity does not have an adverse effect on the approximation made in equation (7.3). Complexity has been reduced from $O(2^8)$ to $O(9)$. Comparing the graphs in Figure 7.2 and Figure 7.4, it is observed that increasing the number of users push the performance curves away from the single-user bound as expected. However, the degradation is not large despite the high loading.

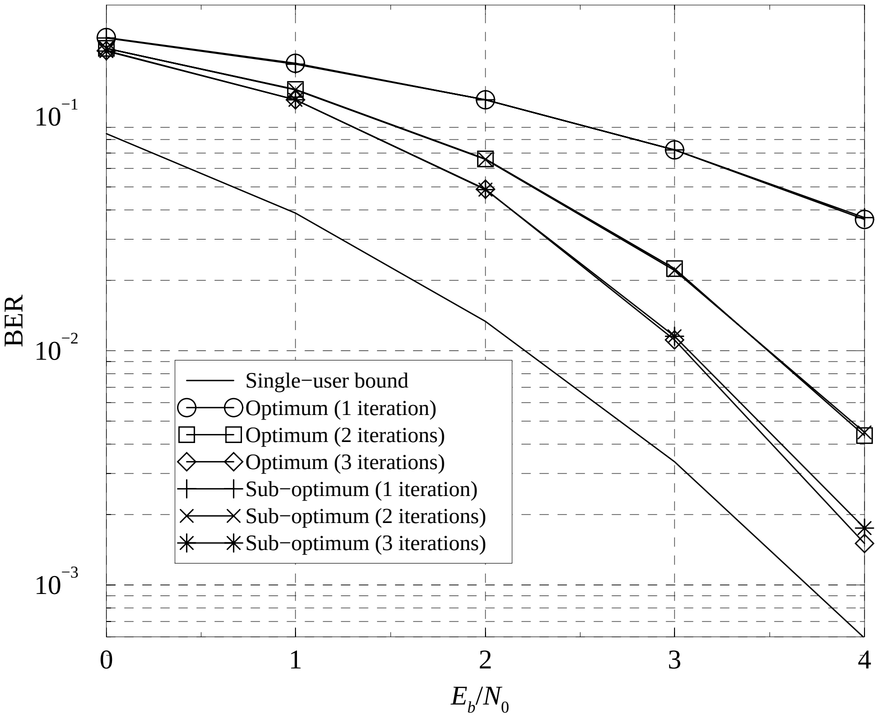

Figure 7.4: *BER against E_b/N_0 for $K = 8$ users, sub-optimum receiver uses $n_m = 3$ metric sums with $n_s = 9$ symbol vectors*

Simulation results of the BER against the number of active users are plotted in Figure 7.5. The E_b/N_0 is now fixed at 4 dB while the number of users increases from 4 to 8. For the sub-optimum iterative receiver, the value of n_m is fixed at 3, while the value of n_s which is set to be equal to $K + 1$, depends on the number of users, K. It is observed from the graphs that as MAI increases, the improvement due to the iterations widen. If $K = 4$, the third iteration does not improve performance further. Despite the increase in MAI, the degradation is small after 3 iterations, even for the sub-optimum iterative receiver.

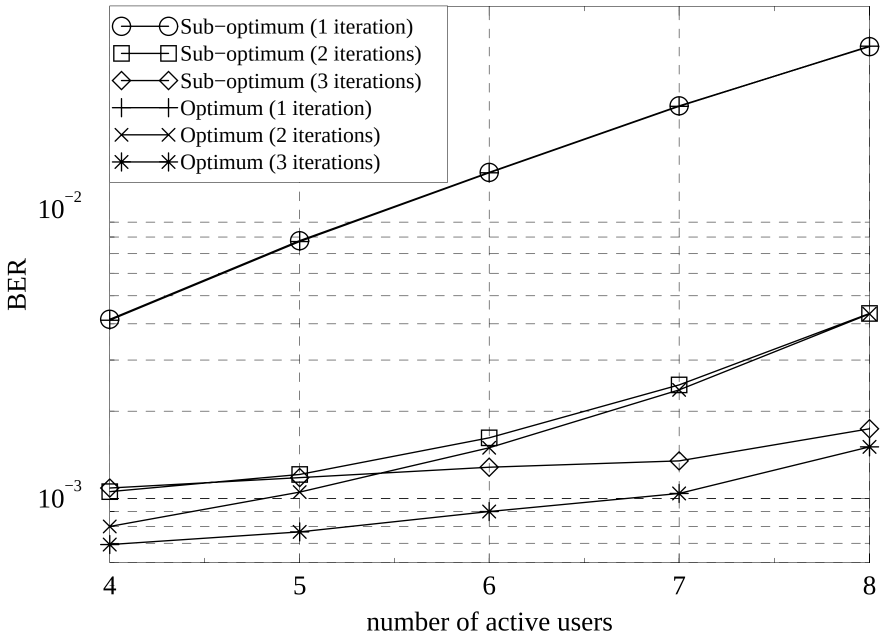

Figure 7.5: *BER against the number of active users for $E_b/N_0 = 4$ dB, $n_m = 3$ and $n_s = K + 1$*

7.5 Summary

In this chapter, an iterative multiuser receiver based on the iterative algorithm of Turbo codes is proposed. This receiver performs signal detection and decoding separately, but iterates between the two using soft-inputs and soft-outputs. The folded trellis sub-optimum MLSE is suggested to replace the optimum MLSE in the signal detection portion. It is shown that the performance is still close to the optimum receiver despite the large reduction in complexity from $O(2^K)$ to $O(K + 1)$. It has also been shown that if the extrinsic information extracted by the decoders is used in the detector, performance improves.

Chapter 8
Conclusions

This concluding chapter first gives a brief summary of the thesis. The following section outlines the contributions to knowledge arising from the work carried out in this thesis. Finally, possible future developments of the work are discussed in Section 8.3.

8.1 Summary

Chapter 1 gives a general overview of cellular communications, which includes the concept of cells in wireless communications, its importance and the need to go wireless. The various multiple access techniques used by the subscribers to access the network, such as FDMA, TDMA and CDMA, are discussed. The shortcomings of each technique are stated and the motivation for the work carried out in this thesis is addressed. The organisation of the thesis is also outlined.

Chapter 2 goes into the detailed discussion of direct-sequence (DS) CDMA communications systems which use code sequences directly for spreading. The principles of spread spectrum which is the fundamental to CDMA technologies are also covered. Next the models of the coded and uncoded communications systems are presented on which all discussions in the subsequent chapters are based. The code sequences or spreading sequences used in DS-CDMA systems are discussed with emphasis on orthogonal codes and random codes. Random codes are important because of their diverse implications to other more realistic systems. The conventional receiver using matched filters is discussed next, and the MAI arising from treating the interference from the other users as noise is shown. Finally, the FEC coding in the communications systems is presented. Convolutional coding is discussed as it is a popular FEC code in communications systems. Two decoding methods for convolutionally encoded signals, namely Viterbi algorithm and BCJR algorithm, are illustrated. The Turbo code, which is a concatenation of convolutional codes, is introduced briefly along with the BCJR decoder or MAP channel decoder needed for decoding.

Instead of using the conventional receiver, multiuser receivers can be used which improve the performance significantly. Hence, established multiuser receivers form the main topic of Chapter 3. Firstly, the cause of MAI and the various methods of mitigating this interference are discussed. Multiuser detection is introduced as a potential scheme of reducing MAI. The optimum MUDs for the uncoded systems, specifically the sequence-based and single-symbol based optimum MUDs, are formulated. It is verified by theoretical analysis that the MLSE detector, which is the optimum sequence-based MUD, can be implemented with the VA in a synchronous multipath environment, such that it becomes independent of the sequence length. However, the optimum MUD still has a prohibitive computational complexity (exponential in the number of users) that makes implementation difficult. Hence, sub-optimum MUDs, using linear or non-linear detection techniques are attractive. Various sub-optimum MUDs, such as DD, MMSE, IC, MSD and DFD, are reviewed and compared. The optimum multiuser receiver for the coded systems is far too complex and is not discussed further in the thesis, however the optimum iterative partitioned multiuser receiver is covered. The powerful iterative algorithm of Turbo codes has been employed in this multiuser detection/decoding, yielding a new class of receiver known as iterative multiuser receiver. The optimum iterative receiver iterates between the Bayesian detector and the single-user BCJR decoders.

Chapter 4 introduces the chip-based multiuser detectors by proving that the MLSE detector can also be operated at the chip level without any degradation in performance. Both implementations, that is at symbol level and at chip level, are explained in detail. It has been found that the CBMLSE has a higher complexity due to the multiplications needed at each chip. Other simplifications to the CBMLSE including using linear distance metrics and Hamming distances have also been investigated. The complexities of the MLSE and the LMCBMLSE are compared in the AWGN and multipath scenarios. It is found that complexities are still exponential in the number of users. Simulation results also show that a truncation length of 2 symbol intervals in the VA is sufficient for most multipath cases.

Based on the complete state-transition trellis of the CBMLSE, a folded state-transition trellis is proposed in Chapter 5. The original trellis is folded up using the discrete transmitted chip values as the states instead of the various possible symbol combinations. By doing so, the number of states involved is reduced. Each state can consist of 1 chip, or multiple chips can be combined into a state. When all the chips in a symbol interval are combined in a state, it becomes the MLSE. The structure of the folded trellis is also extended to the multipath channel.

The complexity of this scheme is then analysed in detail and complexity is found to be in the order of K and $\min(4^K, (K+1)^L)$ for the AWGN and multipath channels respectively. Finally, the simulation results are presented which show that performance is still near-optimum despite the great reduction in complexity.

In Chapter 6, the folded state-transition trellis is combined with the CBMLSE to yield a sub-optimum MLSE with folded trellis preselection. In this case, the single-chip folded trellis is used to preselect a subset of the possible symbol combinations that are likely to be transmitted based on the metric sums computed by the FTD first stage. The CBMLSE then calculates the joint likelihoods of the components in this subset. Since the subset is much smaller than the complete set, the complexity is very much reduced. This is analysed in detail. Furthermore, as the folded trellis has a high probability of selecting the actual transmitted symbol vector among symbol combinations in the subset, the performance is still favourable which is verified with simulation results. Performance is better than the single-chip FTD due to the CBMLSE second stage, however additional complexity due to the addition of the second stage is not large because the metrics needed have already been computed by the FTD. Comparing with the multi-chip FTD, this sub-optimum folded trellis MLSE is more attractive in terms of complexity.

Chapter 7 looks at the iterative multiuser receiver which iterates between the 2 coding dimensions, namely the spreading codes and the FEC codes. The system model for the coded DS-CDMA communications systems is first briefly reviewed, then the iterative multiuser receiver is discussed. The soft-in soft-out signal detection stage and the soft-in soft-out decoding stage are first described separately, and then the iteration process between the two stages is described. The optimum iterative multiuser receiver uses the optimum decentralised single-user Bayesian detector to generate the single-user extrinsic information for the single-user decoders. This chapter suggests modifying the sub-optimum detector in the previous chapter such that it can be utilised in the multiuser signal detection to yield a sub-optimum Bayesian detector, in order to reduce the dominating complexity of the optimum decentralised detector. Simulation results show that performance is very close to optimum and performance improves with diminishing returns at every iteration. The effect of the single-user decoders' extrinsic information on the signal detection stage is also investigated and found that extrinsic information improves performance. Complexity is favourable because of the low complexity of the sub-optimum folded trellis MLSE.

8.2 Contributions

So far most work on sub-optimum MUD usually starts from the linear detectors and then tries to improve performance by introducing non-linearity into the receivers in the form of feedback, regeneration and cancellation. For example, the IC, MSD and DFD all require a linear front-end such as DD, MMSE or simply the matched filter bank. However, introducing non-linearity also increases complexity significantly while the performance is restricted by the linear portion of the detector. The approach taken here is to look at the problem from another perspective. Beginning from the optimum detector, the aim here is to simplify the complexity while retaining the superior performance. There has not been much research in this area as the optimum detector with an exponential complexity in the number of users has always been regarded as useless and impracticable [105]. However, the optimum receiver may actually be feasible with some simplifications such as the FTD suggested in this work.

There has been no literature on optimum synchronous multiuser detector in multipath downlink channel. There are however articles addressing optimum MUD in asynchronous multipath [61] and single-path [59] fading, both of which are for the uplink environment. Hence, the work here provides a lower bound for all MUDs in the same area.

The first stage of the MUD is usually a matched filter bank used for despreading the spread received signal such that the remaining section of the receiver operates at symbol level. However, this is not a must, decoding can actually begins without despreading first. For example, the MMSE performs despreading and multiuser processing jointly with a linear FIR filter operating at chip level, and [116, 117] suggest using Kalman filter to estimate delays and amplitudes together with decoding. This implies that the MUD can also operate at chip level. The chip-based MLSE is first introduced in this thesis. Chip-by-chip maximum likelihood multiuser processing has never been investigated at the point of writing and opens up a new area of research. It allows the VA to operate at the chip level instead of the symbol level. By doing so, the matched filter bank is no longer needed.

Furthermore, no work has been done so far to check the truncation delay needed for the VA if employed in the multiuser DS-CDMA environments. This negligence is probably due to the fact that the optimum detector has often been viewed as useless. Although the implementation of the CBMLSE detector here is different from the MLSE detector, they are theoretically the same, hence the result can be generalised to that of the MLSE.

The novel technique of folding up the original state-transition trellis provides a mean of reducing the prohibitive complexity of the MLSE such that it is more suitable for practical implementation. It has been shown that the FTD has performance close to the MLSE. The FTD is not limited to the AWGN channel and can be extended to the multipath channel. The FTD can also be used as a preselection that determines the symbol vectors that are likely to be transmitted with high accuracy as shown in the sub-optimum folded trellis MLSE. Similarly, it is also used for the same purpose in the iterative folded trellis multiuser receiver to approximate the Bayesian detector.

It has been shown that using the extrinsic information in the optimum iterative multiuser receiver improves the performance. The extrinsic information is explicitly determined at the output of the single-user decoder by removing the *a priori* information of the symbol in question.

8.3 Future work

There are still many open issue regarding the work in this thesis. Many different areas of research can be extended from the proposed FTD scheme. A few possible issues to be addressed are discussed here.

In this thesis, only the stationary multipath channel has been considered for simplicity. It would be interesting to test the various receivers in a multipath fading channel such as the COST 207 channel model [118]. The performance degradation due to channel estimation is also worth investigating.

Currently, the work is only done for coherent detection method where the phase of the received signal is assumed known. A non-coherent detection method can also be implemented by considering the complex tap-weights instead of taking the magnitudes for the multipath channel. Consequently, this suggests QPSK modulation which is the modulation technique used in cdmaone and in IMT-2000. Here, the real and imaginary portions are processed separately and then combined together. BPSK modulation has been used throughout the thesis for simplicity, considering other modulation techniques is another option to look into.

Another interesting area where this work can be extended to is to investigate the idea of preselection. In the sub-optimum MLSE, the FTD is used as a pre-selection stage to determine the symbol vectors that are more likely to be transmitted and then the final decision is based on the optimum MLSE. The basic idea is to use a detector with low complexity for pre-selection and to use a detector with good performance for the final decision. This idea can be exploited with other combinations of detectors.

In the sub-optimum folded trellis MLSE, the preassigned values of the number of selected metric sums, n_m, and the number of selected symbol combinations, n_s, have not been chosen according to any fixed theoretical criteria. Hence, the optimal values of n_m and n_s have not been used. If a theoretical rule can be obtained to determine the optimal values, then the performance would be even better.

Another option is to apply soft-decision outputs with reliability information at the chip-level such that some forms of soft information are passed between each chip, instead of passing just the metric sums to the next chip [119–121]. The additional information indicates whether the decided chips are reliable and can be used to weight the path metrics in the VA.

Turbo coding has been applied to CDMA MUD in [122], it may replace the convolutional codes used in the iterative multiuser receiver. It is worthwhile investigating the performance improvement of the iterative scheme if iteration is carried out between SISO MUD detectors and Turbo codes instead.

Interleaving has been used in Turbo codes to reduce the correlations between the two codes. Although the spreading codes and convolutional codes used in the iterative multiuser receiver have small correlation values, the addition of interleaver between the two dimensions would probably improve performance further. On the other hand, it may be interesting to investigate the effect of using different convolutional codes for different users such that there would be less correlation between the users.

IMT-2000 supports multirate, that is the users can transmit at different data rates, thus the spreading codes of the users have different processing gains. There is no expected problem of extending the FTD to this system. Spreading codes with shorter code length simply have to be repeated to fill up a time slot, and the trellis of FTD would still span 1 time slot.

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, USA: Cambridge University Press, 2nd ed., 1992.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.

[3] T. Edwards, “Technology trends for personal communications terminals,” in *Proceedings of Colloqium on Personal Communications in the 21th Century part I and II, ref no. 1998/214 and 1998/242*, pp. 5/1–5/8, IEE, February 1998.

[4] A. Fukasawa, T. Sato, Y. Takizawa, R. E. Fisher, T. Kato, and M. Kawabe, “Wideband CDMA System,” in *Proceedings of the 1996 4th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’96*, vol. 1, (Mainz, Germany), pp. 244–248, IEEE, September 22-25 1996.

[5] E. Dahlman, P. Beming, J. Knutsson, F. Ovesjoe, M. Persson, and C. Roobol, “WCDMA–The Radio Interface for Future Mobile Multimedia Communications,” *IEEE Transactions on Vehicular Technology*, vol. 47, pp. 1105–1118, November 1998.

[6] E. Dahlman, B. Gudmundson, M. Nilsson, and J. Sköld, “UMTS/IMT-2000 Based on Wideband CDMA,” *IEEE Communications Magazine*, pp. 70–80, September 1998.

[7] F. Adachi, M. Sawahashi, and H. Suda, “Wideband DS-CDMA for Next-Generation Mobile Communications Systems,” *IEEE Communications Magazine*, pp. 56–69, September 1998.

[8] V. K. Garg and J. E. Wilkes, *Wireless and Personal Communications Systems*. Prentice Hall PTR, 1996.

[9] A. J. Paulraj and C. B. Papadias, “Space-Time Processing for Wireless Communications,” *IEEE Signal Processing Magazine*, pp. 49–83, November 1997.

[10] W. C. Y. Lee, “Overview of Cellular CDMA,” *IEEE Transactions on Vehicular Technology*, vol. VT-40, pp. 291–302, May 1991.

[11] J. E. Padgett, C. G. Günthur, and T. Hattori, “Overview of Wireless Personal Communications,” *IEEE Communications Magazine*, pp. 28–41, January 1995.

[12] L. Hano and R. steele, “The Pan-European Mobile Radio System,” *European Transactions on Communications*, vol. 5, pp. 245–276, March 1994.

[13] S. Hara and R. P. Prasad, “Overview of Multicarrier CDMA,” *IEEE Communications Magazine*, pp. 126–133, December 1997.

[14] A. J. Viterbi, “The Orthogonality-Random Waveform Dichotomy for Digital Mobile Personal Communications,” *IEEE Personal Communications*, pp. 18–24, First Quarter 1994.

[15] P. Jung, P. W. Baier, and A. Steil, “Advantages of CDMA and Spread Spectrum Techniques over FDMA and TDMA in Cellular Mobile Radio Applications,” *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 357–364, August 1993.

[16] P. W. Baier and P. Jung, “CDMA Myths and Realities Revisited,” *IEICE Transactions : Fundamentals*, vol. E79-A, pp. 1930–1937, December 1996.

[17] P. Jung, M. Nasshan, and Y. Ma, “Comparison of Optimum Detectors for Coherent Receiver Antenna Diversity in GSM Type Mobile Radio Systems,” in *Proceedings of the 1993 International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 93*, (Yokohama, Japan), pp. 54–58, September 1993.

[18] TIA/EIA IS-95A, *Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular system*. Telecommunication Industry Association, 1993.

[19] A. J. Viterbi, “Spread Spectrum Communications – Myths and Realities,” *IEEE Communications Magazine*, vol. 17, pp. 219–226, May 1979.

[20] R. A. Scholtz, “The Spread Spectrum Concept,” *IEEE Transactions on Communications*, vol. COM-25, pp. 121–128, August 1977.

[21] S. Haykin, *An Introduction to Analog and Digital Communications*. John Wiley and Sons Inc, 1989.

[22] A. Duel-Hallen, J. Holtzman, and Z. Zvonar, “Multiuser Detection for CDMA Systems,” *IEEE Personal Communications*, pp. 46–58, April 1995.

[23] S. Verdu, “Adaptive Multiuser Detection,” in *Proceedings of the 1994 3rd International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’94*, (Oulu, Finland), pp. 43–50, July 1994.

[24] V. M. DaSilva, E. S. Sousa, and V. Jovanovic, “Effect of Multipath Propagation on the Forward Link of a CDMA Cellular System,” *Wireless Personal Communications*, pp. 33–41, 1994.

[25] W. Mohr and M. Kottkamp, “Downlink Performance of IS-95 DS-CDMA under Multipath Propagation Conditions,” in *Proceedings of the 1996 4th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’96*, vol. 3, (Mainz, Germany), pp. 1063–1067, IEEE, September 1996.

[26] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[27] R. Kohno, “Pseudo-Noise Sequences and Interference Cancellation Techniques for Spread Spectrum Systems — Spread Spectrum Theory and Techniques in Japan —,” *IEICE Transactions : Communications*, vol. J74-B-I, pp. 1083–1092, May 1991.

[28] E. H. Dinan and B. Jabbari, “Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks,” *IEEE Communications Magazine*, pp. 48–54, September 1998.

[29] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, “Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance,” *IEEE Transactions on Communications*, vol. 46, pp. 1693–1699, December 1998.

[30] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel, “Iterative Multiuser Interference Reduction: Turbo CDMA,” *IEEE Transactions on Communications*, vol. 47, pp. 1008–1014, July 1999.

[31] G. L. Turin, “Introduction to Spread-Spectrum Antimultipath Techniques and Their Application to Urban Digital Radio,” *Proceedings of the IEEE*, vol. 68, pp. 328–353, March 1980.

[32] P. M. Grant, G. J. R. Povey, and R. D. Pringle, “Performance of a Spread Spectrum Rake Receiver Design,” in *Proceedings of the 1992 2nd International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’92*, pp. 71–74, IEEE, November 1992.

[33] S. Haykin, *Digital Communications*. John Wiley and Sons Inc, 1988.

[34] A. J. Viterbi, “Very Low Rate Convolutional Codes for Maximum Theoretical Performance of Spread-Spectrum Multiple-Access Channels,” *IEEE Journal on Selected Areas in Communication*, vol. 8, pp. 641–649, May 1990.

[35] R. J. McEliece and W. Lin, “The Trellis Complexity of Convolutional Codes,” *Jet Propulsion TDA Progress Report 42-123*, pp. 122–139, November 1995.

[36] G. D. Forney, Jr., “The Viterbi Algorithm,” *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[37] H.-L. Lou, “Implementing the Viterbi Algorithm,” *IEEE Signal Processing Magazine*, pp. 42–52, September 1995.

[38] J. Heller and I. Jacobs, “Viterbi Decoding for Satellite and Space Communication,” *IEEE Transactions on Communications*, vol. COM-19, no. 5, pp. 835–848, 1971.

[39] G. D. Forney, Jr., “Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference,” *IEEE Transactions on Information Theory*, vol. 18, pp. 363–378, May 1972.

[40] L. C. Barbosa, “Maximum Likelihood Sequence Estimators: A Geometric View,” *IEEE Transactions on Information Theory*, vol. 35, pp. 419–427, March 1989.

[41] G. Benelli, A. Garzelli, and F. Salvi, “Simplified Viterbi Processors for the GSM Pan-European Cellular Communication System,” *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 870–877, November 1994.

[42] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, “Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes,” *Jet Propulsion TDA Progress Report 42-124*, pp. 63–87, February 1996.

[43] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error-Correcting coding and decoding: Turbo-codes (1),” in *International Conference on Communication*, pp. 1064–1070, 1993.

[44] C. Berrou and A. Glavieux, “Near Optimum Error Correcting Coding And Decoding: Turbo-Codes,” *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, October 1996.

[45] B. Sklar, “A Primer on Turbo Code Concepts,” *IEEE Communications Magazine*, pp. 94–102, December 1997.

[46] D. Divsalar and F. Pollara, “On the Design of Turbo Codes,” *Jet Propulsion TDA Progress Report 42-123*, pp. 99–121, November 1995.

[47] D. Divsalar and R. J. McEliece, “Effective free distance of turbo codes,” *Electronics Letters*, vol. 32, pp. 445–446, February 1996.

[48] C. Berrou and M. Jezequel, “Frame-oriented convolutional turbo code,” *Electronics Letters*, vol. 32, pp. 1362–1364, July 1996.

[49] S. Benedetto and G. Montorsi, “Iterative decoding of serially concatenated convolutional codes,” *Electronics Letters*, vol. 32, pp. 1186–1188, June 1996.

[50] P. Jung, “Comparison of Turbo-Code Decoders Applied to Short Frame Transmission Systems,” *IEEE Journal on Selected Areas in Communication*, vol. 14, pp. 530–537, April 1996.

[51] J. Hagenauer and L. Papke, “Decoding Turbo-Codes with the Soft Output Viterbi Algorithm (SOVA),” in *Proceedings of the 1994 International Symposium on Information Theory*, p. 164, IEEE, June 1994.

[52] R. Wichman and A. Hottinen, “Multiuser Detection for Downlink CDMA Communications in Multipath Fading Channels,” in *Proceedings of the 1997 47th IEEE Vehicular Technology Conference*, (Phoenix, Arizona, USA), pp. 572–576, IEEE, May 4-7 1997.

[53] S. Verdu, “Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels,” *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

[54] P. Jung and P. D. Alexander, “A Unified Approach to Multiuser Detectors for CDMA and their Geometrical Interpretations,” *IEEE Journal on Selected Areas in Communication*, vol. 14, pp. 1595–1601, October 1996.

[55] W. V. Etten, “Maximum Likelihood Receiver for Multiple Channel Transmission Systems,” *IEEE Transactions on Communications*, pp. 276–283, February 1976.

[56] S. Verdu, “Computational Complexity of Optimum Multiuser Detection,” *Algorithmica*, vol. 4, pp. 303–312, 1989.

[57] E. A. Lee and D. G. Messerschmitt, *Digital Communication*. Kluwer Academic Publishers, 1988.

[58] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[59] Z. Zvonar and D. Brady, “Multiuser Detection in Single-Path Fading Channels,” *IEEE Transactions on Communications*, vol. 42, pp. 1729–1739, February/March/April 1994.

[60] S. Verdu, “Optimum Multiuser Asymptotic Efficiency,” *IEEE Transactions on Communications*, vol. 34, pp. 890–897, September 1986.

[61] Z. Zvonar and D. Brady, “Optimum Detection in Asynchronous Multiple-Access Multipath Rayleigh Fading Channels,” in *The Proceedings of the 1992 Conference on Information Science and Systems, Princeton, NJ*, March 1992.

[62] H. V. Poor and S. Verdu, “Single-User Detectors for Multiuser Channels,” *IEEE Transactions on Communications*, vol. 36, pp. 50–60, January 1988.

[63] D. G. M. Cruickshank, “Radial basis function receivers for DS-CDMA,” *IEE Electronics Letters*, vol. 32, pp. 188–190, February 1996.

[64] U. Mitra and H. V. Poor, “Neural Network Techniques for Adaptive Multiuser Demodulation,” *IEEE Journal on Selected Areas in Communication*, vol. 12, pp. 1460–1470, December 1994.

[65] R. Tanner and D. G. M. Cruickshank, “RBF Based Receivers for DS-CDMA with Reduced Complexity,” in *Proceedings of the 1998 5th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’98*, vol. 2, (Sun City, South Africa), pp. 647–651, September 2-4 1998.

[66] R. Tanner, D. G. M. Cruickshank, C. Z. W. H. Sweatman, and B. Mulgrew, “Receivers for nonlinearly separable scenarios in DS-CDMA,” *IEE Electronics Letters*, vol. 33, pp. 2103–2105, December 1997.

[67] M. K. Varanasi, “Group Detection for Synchronous Gaussian Code-Division Multiple-Access Channels,” *IEEE Transactions on Information Theory*, vol. 41, pp. 1083–1096, July 1995.

[68] R. E. Kamel and Y. Bar-Ness, “Reduced-Complexity Sequence Estimation Using State Partitioning,” *IEEE Transactions on Communications*, vol. 44, pp. 1057–1063, September 1996.

[69] Z. Xie, C. K. Rushforth, and R. T. Short, “Multiuser Signal Detection Using Sequential Decoding,” *IEEE Transactions on Communications*, vol. 38, pp. 578–583, May 1990.

[70] A. Klein, “Data Detection Algorithms Specially Designed for the Downlink of CDMA Mobile Radio Systems,” in *Proceedings of the 1997 47th IEEE Vehicular Technology Conference*, (Phoenix, Arizona, USA), pp. 203–207, May 4-7 1997.

[71] R. Lupas and S. Verdu, “Near-far Resistance of Multiuser Detectors in Asynchronous Channels,” *IEEE Transactions on Communications*, vol. 38, pp. 496–508, April 1990.

[72] Z. Zvonar and D. Brady, “Differentially Coherent Multiuser Detection in Asynchronous CDMA Flat Rayleigh Fading Channels,” *IEEE Transactions on Communications*, vol. 43, pp. 1252–1255, February/March/April 1995.

[73] U. Madhow and M. L. Honig, "MMSE Interference Suppression for Direct-sequence Spread-Spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, pp. 3178–3188, December 1994.

[74] A. Klein and P. W. Baier, "Simultaneous Cancellation of Cross Interference and ISI in CDMA Mobile Radio Communications," in *Proceedings of the 1992 International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 92*, (Boston, USA), pp. 118–122, 1992.

[75] R. Singh and L. B. Milstein, "Interference Suppression for DS/CDMA," *IEEE Transactions on Communications*, vol. 47, pp. 446–453, March 1999.

[76] S. Moshavi, E. G. Kanterakis, and D. L. Schilling, "Multistage Linear Receivers for DS-CDMA Systems," *International Journal of Wireless Information Networks*, vol. 3, no. 1, pp. 1–17, 1996.

[77] S. Moshavi, "A New Multiuser Detection Scheme for DS-CDMA Systems," in *Proceedings of MILCOM, San Diego, USA*, vol. 2, pp. 518–522, IEEE, November 1995.

[78] J. Shen and Z. Ding, "Edge Decision Assisted Decorrelators for Asynchronous CDMA Channels," *IEEE Transactions on Communications*, vol. 47, pp. 438–445, March 1999.

[79] U. Mitra and H. V. Poor, "Analysis of an Adaptive Decorrelating Detector for Synchronous CDMA Channels," *IEEE Transactions on Communications*, vol. 44, pp. 257–267, February 1996.

[80] Z. Zvonar and D. Brady, "Suboptimal Multiuser Detector for Frequency-Selective Rayleigh Fading Synchronous CDMA Channels," *IEEE Transactions on Communications*, vol. 43, pp. 154–157, February/March/April 1995.

[81] Z. Xie, R. T. Short, and C. K. Rushforth, "A Family of Suboptimum Detectors for Coherent Multiuser Communications," *IEEE Journal on Selected Areas in Communication*, vol. 8, pp. 683–690, May 1990.

[82] E. G. Ström and S. L. Miller, "Properties of the Single-Bit Single-User MMSE Receiver for DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 47, pp. 416–425, March 1999.

[83] H.-H. Chen and H.-K. Sim, "Quasi-Decorrelating Detector (QDD) And Its Spreading Codes Dependent Performance Analysis," *IEICE Transactions : Communications*, vol. E80-B, pp. 1337–1344, September 1997.

[84] D. Guo, L. K. Rasmussen, S. Sun, T. J. Lim, and C. Cheah, "MMSE-Based Linear Parallel Interference Cancellation in CDMA," in *Proceedings of the 1998 5th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA'98*, (Sun City, South Africa), pp. 917–921, September 2-4 1998.

[85] R. Kohno, P. B. Rapajic, and B. S. Vucetic, "An Overview of Adaptive Techniques for Interference Minimization in CDMA Systems," *Wireless Personal Communications*, pp. 3–21, 1994.

[86] U. Mitra and H. V. Poor, “Adaptive Receiver Algorithms for Near-Far Resistant CDMA,” *IEEE Transactions on Communications*, vol. 43, pp. 1713–1724, February/March/April 1995.

[87] D. G. M. Cruickshank, “Optimal and adaptive FIR filter receivers for DS-CDMA,” in *Proceedings of the 1994 International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 94*, vol. 32, pp. 1339–1343, September 1994.

[88] D. Raphaeli, “Nonlinear Multiuser Detector for Synchronous Coded CDMA on Severely Distorted Frequency Selective Channels,” in *Proceedings of the 1996 IEEE Global Communications Conference - GLOBECOM’96*, vol. 3, (London, UK), pp. 1578–1582, IEEE, November 1996.

[89] B. Aazhang and H. V. Poor, “An Analysis of Nonlinear Direct-Sequence Correlators,” *IEEE Transactions on Communications*, vol. 37, pp. 723–731, July 1989.

[90] L. B. Nelson and H. V. Poor, “Soft-Decision Interference Cancellation for AWGN Multi-User Channels,” in *Proceedings of the 1994 International Symposium on Information Theory*, p. 134, IEEE, 1994.

[91] P. M. Grant, S. Mowbray, and R. D. Pringle, “Multipath and Co-channel CDMA Interference Cancellation,” in *Proceedings of the 1992 2nd International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’92*, (Yokohama, Japan), pp. 83–86, November 1992.

[92] R. S. Mowbray, R. D. Pringle, and P. M. Grant, “Increased CDMA system capacity through adaptive cochannel interference regeneration and cancellation,” *IEE Proceedings - Communication*, vol. 139, pp. 515–524, October 1992.

[93] Y. C. Yoon, R. Kohno, and H. Imai, “A Spread-Spectrum Multiaccess System with Cochannel Interference Cancellation for Multipath Fading Channels,” *IEEE Journal on Selected Areas in Communication*, vol. 11, pp. 1067–1075, September 1993.

[94] D. G. M. Cruickshank, “Suppression of multiple access interference in a DS-CDMA system using wiener filtering and parallel cancellation,” *IEE Proceedings - Communication*, vol. 143, pp. 226–230, August 1996.

[95] R. M. Buehrer and S. P. Nicoloso, “Comments on “Partial Parallel Interference Cancellation for CDMA”,” *IEEE Transactions on Communications*, vol. 47, pp. 658–661, May 1999.

[96] Z.-L. Shi, W. Du, and P. F. Driessen, “A New Multistage Detector For Synchronous CDMA Communications,” *IEEE Transactions on Communications*, vol. 44, pp. 538–541, May 1996.

[97] Y.-H. Kim and S. Shamsunder, “Soft Decision Feedback Receiver for CDMA Communication in multipath,” in *Proceedings of the 1998 48th IEEE Vehicular Technology Conference*, pp. 1705–1709, 1998.

[98] Y. Bar-Ness, “Asynchronous Multiuser CDMA Detector Made Simpler: Novel Decorrelator, Combiner, Canceller, Combiner (DC^3) Structure,” *IEEE Transactions on Communications*, vol. 47, pp. 115–122, January 1999.

[99] M. K. Varanasi and B. Aazhang, “Multistage detection in asynchronous code-division multiple-access communications,” *IEEE Transactions on Communications*, vol. 38, pp. 509–519, April 1990.

[100] S. Chen, S. McLaughlin, B. Mulgrew, and P. M. Grant, “Bayesian decision feedback equaliser for overcoming co-channel interference,” *IEE Proceedings - Communication*, vol. 143, pp. 219–225, August 1996.

[101] L. Wei and C. Schlegel, “Synchronous DS-SSMA system with improved decorrelating decision-feedback multiuser detection,” *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 767–772, 1994.

[102] L. Wei, L. K. Rasmussen, and R. Wyrwas, “Near Optimum Tree-Search Detection Schemes for Bit-Synchronous Multiuser CDMA Systems over Gaussian and Two-Path Rayleigh-Fading Channels,” *IEEE Transactions on Communications*, vol. 45, pp. 691–700, June 1997.

[103] X. H. Chen and H. K. Sim, “Novel synchronous CDMA multiuser detection scheme: orthogonal decision-feedback detection and its performance study,” *IEE Proceedings: Communications*, vol. 144, pp. 275–280, August 1997.

[104] R. M. Buehrer, N. S. Correal, and B. D. Woerner, “A Comparison of Multiuser Receivers for Cellular CDMA,” in *Proceedings of the 1996 IEEE Global Communications Conference - GLOBECOM’96*, (London, UK), pp. 1571–1577, IEEE, November 1996.

[105] S. Verdu, “Demodulation in the Presence of Multiuser Interference: Progress and Misconceptions,” *Intelligent Methods in Signal Processing and Communications, Birkhauser, Boston, USA*, pp. 15–44, 1997.

[106] T. R. Giallorenzi and S. G. Wilson, “Multiuser ML Sequence Estimator for Convolutionally Coded Asynchronous DS-CDMA Systems,” *IEEE Transactions on Communications*, vol. 44, pp. 997–1008, August 1996.

[107] T. R. Giallorenzi and S. G. Wilson, “Suboptimum Multiuser Receivers for Convolutionally Coded Asynchronous DS-CDMA Systems,” *IEEE Transactions on Communications*, vol. 44, pp. 1183–1196, September 1996.

[108] M. Moher, “An Iterative Multiuser Decoder for Near-Capacity Communications,” *IEEE Transactions on Communications*, vol. 46, pp. 870–880, July 1998.

[109] X. Wang and V. Poor, “Iterative (Turbo) Soft Interference Cancellation and Decoding for Coded CDMA,” *IEEE Transactions on Communications*, vol. 47, pp. 1046–1061, July 1999.

[110] W. H. Tranter and K. L. Kosbar, “Simulation of Communication Systems,” *IEEE Communications Magazine*, pp. 26–35, July 1994.

[111] H. K. Sim and D. G. M. Cruickshank, “A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis,” *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, pp. 846–850, May 1999.

[112] H. K. Sim and D. G. M. Cruickshank, “A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis.” To appear in *IEEE Transactions on Communications*.

[113] H. K. Sim and D. G. M. Cruickshank, “A Sub-Optimum MLSE Detector with a Folded State-Transition Trellis Preselection Stage,” in *3G 2000 Mobile Communication Technologies (IEE Conference Publication 471)*, pp. 271–275, March 2000.

[114] H. K. Sim and D. G. M. Cruickshank, “A Sub-Optimum MLSE Detector with a Folded State-Transition Trellis Preselection Stage.” To appear in *IEE Proceedings: Communications*.

[115] H. K. Sim and D. G. M. Cruickshank, “An Iterative Multiuser Receiver for FEC Coded CDMA,” in *Proceedings of the 2000 6th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’00*, (New Jersey, USA), pp. 728–732, IEEE, September 6-8 2000.

[116] R. A. Iltis, “Joint Estimation of PN Code Delay and Multipath Using the Extended Kalman Filter,” *IEEE Transactions on Communications*, vol. 38, pp. 1677–1685, October 1990.

[117] R. A. Iltis, “A Digital Receiver for Demodulation of CDMA Waveforms with A-prior Unknown Delays and Amplitudes,” in *MILCOM’91*, pp. 113–116, 1991.

[118] D. I. Laurenson, D. G. M. Cruickshank, and G. J. R. Povey, “A Computationally Efficient Channel Simulator for the COST 207 Channel Models,” in *Proceedings of the IEE Colloquium on Computer Modelling of Communications Systems, Digest*, pp. 8/1–8/6, IEE, May 1994.

[119] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm with Soft-Decision Outputs and its Applications,” in *Proceedings of the 1989 IEEE Global Communications Conference - GLOBECOM’89*, (Dallas, Texas), pp. 47.11–47.17, IEEE, November 1989.

[120] M. Benthin and K. D. Kammeyer, “Viterbi Decoding of Convolutional codes with reliability Information for a Noncoherent RAKE-Receiver in a CDMA-Environment,” in *Proceedings of the 1994 IEEE Global Communications Conference - GLOBECOM’94*, (San Francisco, USA), pp. 1–5, IEEE, December 1994.

[121] C. Berrou, P. Adde, E. Angui, and S. Faudeil, “A Low Complexity Soft-Output Viterbi Decoder Architecture,” in *International Conference on Communication*, pp. 737–740, IEEE, 1993.

[122] P. Jung, M. Nasshan, and J. Blanz, “Application of Turbo-Codes to a CDMA Mobile radio System Using Joint Detection and Antenna Diversity,” *Proceedings of the 1994 IEEE 44th Vehicular Technology Conference*, pp. 770–774, 1994.

Appendix A
Application of the Viterbi Algorithm to signal detection

Communications through band-limited channels cause intersymbol interference (ISI) due to imperfect channels. The effects of one transmitted pulse are not allowed to die away completely before the transmission of the next in pulse-modulation systems give rise to ISI. This problem is especially prominent in high-rate digital transmission. The MLSE detector implemented with the Viterbi algorithm (VA) is the optimum demodulator and detector for digital transmission through a non-ideal band-limited channel with AWGN.

The optimum demodulator can be realised as a whitening matched filter, followed by a sampler operating at the symbol rate and a subsequent processing algorithm implemented with the VA. The samples at the output of the matched filter provide a set of sufficient statistic for the estimation of the input sequence. It is shown in [26, 39, 57, 58] that the use of a whitening matched filter develops an equivalent discrete-time model for the continuous-time system. The interest here is the application of the VA to the equalisation and detection problem.

The equivalent discrete-time ISI channel can be modelled as a discrete-time transversal filter and a white Gaussian noise sequence, n_i, with zero mean and variance $N_0/2$. Let $\{g_0, \ldots, g_L\}$ be the $L + 1$ tap coefficients of the filter, and let v_i and y_i be the transmitted sequence and the received sequence at the input and output of the channel respectively, where i is the time index. Thus, the relationship between these terms is

$$y_i = \sum_{j=0}^{L} g_j v_{i-j} + n_i. \tag{A.1}$$

Assume that the information symbols are binary, the channel can be described by a trellis with 2^L states. The VA can then be applied to the trellis to find the most likely path through the trellis. Let a_i be the i-th bit of the unmodulated information sequence. For BPSK modulation, the relationship between a_i and v_i is

$$v_i = \sqrt{E_s}(2a_i - 1), \tag{A.2}$$

where E_s is the energy of the symbol. The state at time i is made up of L most recent input bits, $(a_{i-L}, a_{i-L+1}, \ldots, a_{i-1})$, where $a_i = 0$ for $i < 0$. Hence, the number of bits per state and the number of states in the trellis are directly dependent on the length of the ISI channel. The metrics used in this trellis are akin to the metrics used in the decoding of convolutional codes described in section 2.6.2. Upon reception of y_i, the transition metric from state $(a_{i-L}, a_{i-L+1}, \ldots, a_{i-1})$ to state $(a_{i-L+1}, a_{i-L+2}, \ldots, a_i)$ is calculated as

$$M^T(a_{i-L}, a_{i-L+1}, \ldots, a_i) \quad = \quad \left(y_i - \sum_{j=0}^{L} g_j v_{i-j} \right)^2 . \tag{A.3}$$

Note that there are 2^{L+1} possible transition metrics at each stage, corresponding to the 2^{L+1} possible sequences $(a_{i-L}, a_{i-L+1}, \ldots, a_i)$. The path metric of state $(a_{i-L+1}, a_{i-L+2}, \ldots, a_i)$ is expressed as

$$\begin{aligned} & M^P(a_{i-L+1}, a_{i-L+2}, \ldots, a_i) \\ & = \min_{a_{i-L} \in A} \left[M^P(a_{i-L}, a_{i-L+1}, \ldots, a_{i-1}) + M^T(a_{i-L}, a_{i-L+1}, \ldots, a_i) \right], \end{aligned} \tag{A.4}$$

where A is the set containing all the possible values of a_{i-L} and in this case the size of the set is only two. Hence, the 2^{L+1} transition metrics are subdivided into 2^L groups, with each group containing 2 transition metrics that terminate in the same state $(a_{i-L+1}, a_{i-L+2}, \ldots, a_i)$ and differ only in the bit a_{i-L}. The 2 transition metrics in each group are then summed with their respective path metric of the previous stage, and out of the 2 resulting sums, the one with the smaller value is selected as the new path metric for this group (see equation A.4).

Once the path metric has been determined, the value of a_{i-L} in set A, say a_{i-L}^{min}, that gives rise to the minimum metric is also noted. The transition that corresponds to a_{i-L}^{min} is then appended to the survivor path of state $(a_{i-L}^{min}, a_{i-L+1}, \ldots, a_{i-1})$ to form the survivor path of $(a_{i-L+1}, a_{i-L+2}, \ldots, a_i)$. The initial path metrics, $M^P(a_{-L}, a_{-L+1}, \ldots, a_{-1})$, are all set to zero. Therefore, by recursively calculating the path metrics and forming the survivor path until the end of the trellis, the global survivor path can be traced out just as it is done in the decoding of convolutional codes. The delay in detecting each bit is variable, so in practice the variable delay is avoided by truncating the surviving sequences. The length of the truncated surviving sequence is usually fixed at a value greater than $5L$ to prevent a degradation in performance.

The following illustrates with a simplified example. Consider a BPSK modulated band-limited system transmitting 4 symbols. The ISI channel can be modelled with an equivalent discrete-time transversal filter with three taps. Hence, the trellis has $2^L = 4$ states as shown in Figure A.1. The value in each transition represents one of the two possible values of a_i at time i. After a_3 is transmitted, when transmission has ended, there is still residue of the previous symbols due to ISI effect in the channel. Hence, the trellis extends until the residual effect of the last bit is over.

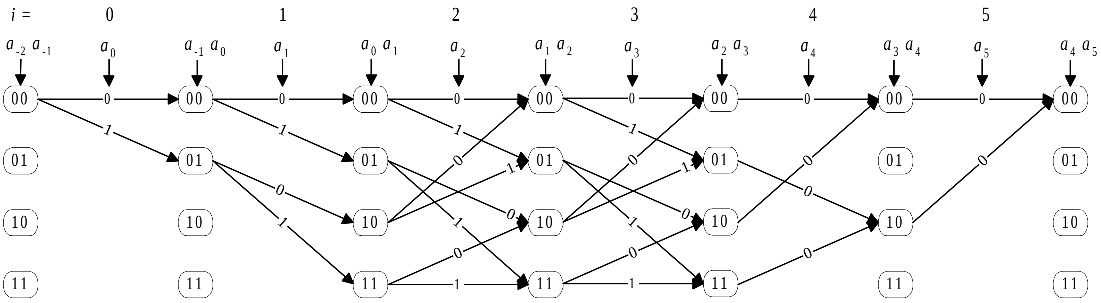

Figure A.1: *Viterbi trellis for the signal detection of a BPSK modulated band-limited system with a ISI discrete-time channel model of 3 taps*

Appendix B
Publications

Lists of publications:

- H. K. Sim and D. G. M. Cruickshank, ”A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using Linear Distance Metrics,” accepted for *Postgraduate Mobile Communications Research Symposium*, 1998.

- H. K. Sim and D. G. M. Cruickshank, “A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis,” in *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, pp. 846–850, May 1999.

- H. K. Sim and D. G. M. Cruickshank, “A Sub-Optimum MLSE Detector with a Folded State-Transition Trellis Preselection Stage,” in *3G 2000 Mobile Communication Technologies (IEE Conference Publication 471)*, pp. 271–275, March 2000.

- H. K. Sim and D. G. M. Cruickshank, “An Iterative Multiuser Receiver for FEC Coded CDMA,” in *Proceedings of the 2000 6th International Symposium on Spread Spectrum Techniques and Applications, ISSSTA’00*, (New Jersey, USA), pp. 728–732, IEEE, September 6-8 2000.

- H. K. Sim and D. G. M. Cruickshank, “A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis,” accepted for publication in *IEEE Transactions on Communications*.

- H. K. Sim and D. G. M. Cruickshank, “A Sub-Optimum MLSE Detector With A Folded State-Transition Trellis Preselection Stage,” accepted for publication in *IEE Proceedings - Communication*.

- H. K. Sim and D. G. M. Cruickshank, “Near-Optimum Iterative Multiuser Receiver for FEC Encoded DS-CDMA,” submitted to *Wireless Personal Communications*.

A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using Linear Distance Metrics

Hak Keong Sim and David G M Cruickshank

Signals and Systems Group,
Department of Electronics and Electrical Engineering,
The University of Edinburgh, The King's Buildings,
Mayfield Rd, Edinburgh EH9 3JL.
Tel (+44) 131 650 5655. Fax (+44) 131 650 6554.
Email hks@ee.ed.ac.uk, dgmc@ee.ed.ac.uk.

Abstract

The maximum-likelihood sequence estimation (MLSE) multiuser detector has very good performance but at the expense of high computational complexity. This detector searches through all the possible input bit sequences and selects the one closest to the received sequence based on the Euclidean distance. It can be implemented using a matched filter bank front-end and based on the sufficient statistics of these matched filters' output, a maximum likelihood ratio comparison is then carried out at the bit level.

The idea of the proposed detector is to operate the MLSE at the chip-level instead of the bit-level. This negates the need for a matched filter front-end, consequently, despreading, symbol-decoding and channel-equalisation are performed jointly. Immediately, it allows us to consider the absolute linear distance instead of the Euclidean (squared) distance. As squaring is not needed, no multiplication is involved at runtime. For the downlink, the number of multiplications needed to be performed by the bit-based MLSE is exponential in the number of users.

In a multipath channel, where the bits from different intervals are dependent on each other, the Viterbi algorithm can be applied at chip-level, to determined the linear distance. The state-transition trellis formed in this case has a number of expansions and mergings in each bit interval, and repeats every bit interval. The transitions between chips do not cross each other when there is no multipath effect from adjacent bits and no decision is made. The trellis expands when multipath effect arises and collapses when the effect is gone. At the state where transitions merge, a decision can be made. The symbols are decoded when all survival paths originate from a state. An advantage of operating at chip level is that decisions can be made once the trellis collapses and do not have to be delayed until the end of the bit interval to produce the same result.

The proposed detector was simulated in a downlink environment with a Gaussian channel, and both stationary and time-varying (COST 207) multipath fading channels. Other detectors simulated include the conventional detector, the RAKE receiver and the minimum mean squared error detector. Generally, the proposed scheme performs better than all except the MLSE. It was found that the performance for random codes and Walsh codes is close to the MLSE detector when the survival path lengths are unlimited. For practice considerations, a fixed delay needs to be employed instead. It was observed from simulation results that for a delay of just 2 bit-interval, the performance is already indistinguishable from that of the unlimited delay case.

1 Introduction

In a direct-sequence code-division multiple-access (DS-CDMA) system, users are distinguished by using spreading codes. For the downlink, the base station transmits synchronously to all the mobile users. In an ideal scenario, each mobile receives the joint signal that is passed through an additive white Gaussian noise (AWGN) channel. With orthogonal codes, each individual user's component in the joint signal is independent of the other users, and does not interfere with one another. However, in a practical scenario, there is multipath fading effect in the channel, resulting in intersymbol interference. This destroys any orthogonality of the spreading codes and causes multiple access interference (MAI) from the other users. To overcome the MAI, multiuser detectors (MUDs) have been suggested where the users are decoded jointly for their mutual benefits. The MAI is not an inherent problem of DS-CDMA, but of the conventional detector — single-user matched filter or the RAKE receiver, which decodes and equalises the users independently, treating the MAI as Gaussian noise [1].

The maximum likelihood sequence estimator (MLSE) [2] gives near-optimum performance when the transmitted bit sequences of the users are equiprobable and independent. Note that the optimum detector has to be based on the maximum-a-posteriori (MAP) or Bayesian techniques such as the radial basis function (RBF) detector [3]. The RBF finds the optimum data bit for each individual user while the MLSE finds the optimum data bit vector that results in minimum joint probability. However, this optimum data bit vector does not give the optimum performance for each individual user only the best performance on average. Fortunately, the difference in performance is not too significant. Due to the high complexity of the RBF, we concentrate mainly on the MLSE. This detector searches through all the possible input bit sequences and selects the one that minimises the squared Euclidean distance between the observation and the bit sequences. For the ideal channel, where the bit intervals do not interfere with each other, the one-shot approach is sufficient to produce the optimum performance. Under these conditions, it can be implemented using a matched filter bank front-end and based on the sufficient statistics of these matched filters' output, a maximum likelihood ratio comparison is then carried out at the bit-level. However, in a realistic environment, symbols are interfered with by symbols in adjacent intervals, and these symbols are in turn interfered by other symbols. Hence, additional information about the received signal in the bit interval in question can be obtained by considering other bit intervals as well, and the optimum approach is thus to consider the entire observation interval. The one-shot approach can be implemented by a conventional matched filter bank front-end followed by a maximum likelihood decision algorithm [4]. It can be shown that the MLSE detector for a multipath channel is to use a RAKE receiver front-end followed by a forward dynamic programming decision algorithm (the Viterbi algorithm [5]). Although the complexity is no longer dependent on the transmission length due to the Viterbi algorithm, it is still exponential in the number of users.

In this paper, we investigate the MLSE that is implemented at the chip level instead of the bit-level. A chip-based MUD negates the need for a matched filter front-end and decoding begins even before despreading is completed. We can immediately simplify this detector by considering the absolute linear distance instead of the Euclidean (squared) distance. The absolute value can be obtained simply by dropping the signed bit in the DSP, no multiplication is involved during run-time. The absolute values of the differences between the received value and all the possible noiseless received values are taken at each chip. In a multipath channel, where the bits from different intervals are dependent on each other, the Viterbi algorithm can be applied at the chip-level to determine the linear distance. Consequently, despreading, channel-equalisation and symbol-decoding are performed jointly using a state-transition trellis.

Section 2 of this paper describes the multipath fading channel model of the downlink of a DS-CDMA. This is followed by a discussion of the bit-based and chip-based MLSE where it is shown that the output of the RAKE receiver provides a sufficient statistics for the bit-based MLSE in a multipath environment. The application of the Viterbi algorithm to the chip-based MLSE is also shown in this section. Section 4 introduces the proposed detector in an AWGN environment and then extends it to the multipath fading scenario. State-transition trellis diagrams are used to explain the decoding procedure at the chip-level. The complexity of the proposed detector is then compared with the bit-based MLSE in the following section. The number of addition and multiplication operations needed both at run-time and pre-computation are given in detail for both the AWGN channel and the multipath channel. Finally, in Section 6, simulation results under AWGN, stationary multipath and time-varying multipath channels are presented. The convergence rate of the Viterbi algorithm is also investigated.

2 The DS-CDMA Downlink Channel Model

The equivalent discrete-time mathematical model for the downlink of a DS-CDMA system will be discussed in this section. A synchronous multipath channel with K users sharing the same bandwidth is shown in Figure 1. Each user in the system has a pre-assigned spreading waveform,

$$s_k(t) = \sum_{i=0}^{G-1} c_{i,k} p(t - iT_c), \qquad (k = 1, \ldots, K) \tag{1}$$

where G is the processing gain of the spreading waveform, $c_{i,k} = \{-1, +1\}$ is the i-th chip value of the k-th user, T_c is the chip duration and $p(t)$ is the rectangular chip waveform which is zero outside $[0, T_c)$. Each waveform, $s_k(t)$, is thus restricted to a symbol duration, $T = GT_c$, i.e., zero outside $[0, T)$. The spreading waveform can also be expressed as a z-transform spreading code sequence of length G,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}. \qquad (k = 1, \ldots, K) \tag{2}$$

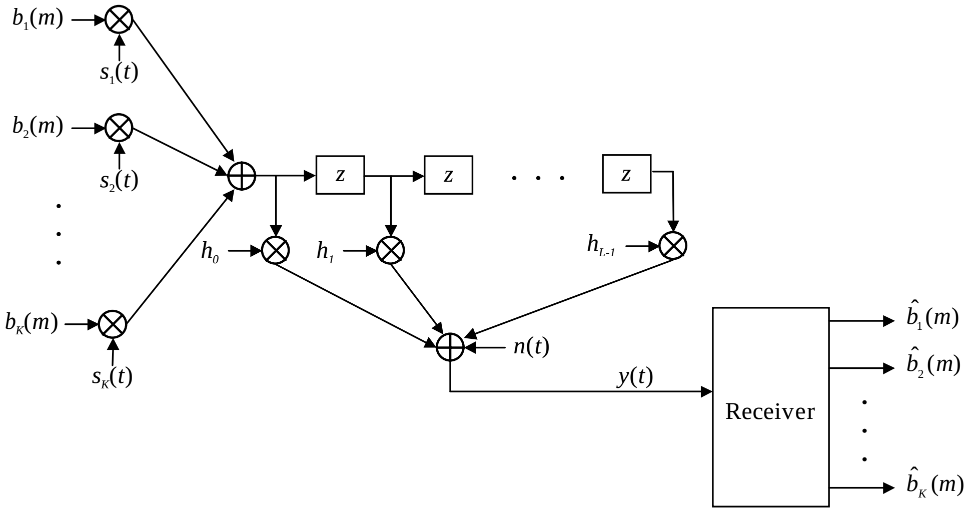

Figure 1: Synchronous multipath system model

Consider a BPSK modulated system that uses only short spreading codes (one bit duration is covered by exactly one period of the codes). The input bit sequences for different users are identically independently distributed (i.i.d) and equiprobable. Consider a multipath fading channel with L paths and a delay spread greater than one chip period. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink. The z-transform of the normalised channel impulse response can be written as

$$H(z) = \sum_{l=0}^{L-1} h_l z^{-l}, \tag{3}$$

where h_l is the path coefficient of the l-th path.

Assume that each user transmits M bits with constant power. The noiseless received signal of the entire data transmission, can be expressed as

$$\begin{aligned} X(z) &= H(z) \sum_{m=0}^{M-1} z^{-Gm} \sum_{k=1}^{K} S_k(z) \sqrt{P_k} b_k(m) \\ &= \sum_{k=1}^{K} R_k(z) \sqrt{P_k} B_k(z) \\ &= \mathbf{R}^T(z) \mathbf{P} \mathbf{B}(z), \end{aligned} \tag{4}$$

where:

$$B_k(z) = \sum_{m=0}^{M-1} b_k(m) z^{-Gm} \qquad (k = 1, \ldots, K) \tag{5}$$

represents the entire transmission of the input bits for the k-th user, with $b_k(m) = \{-1, +1\}$ representing the m-th input bit symbol,

$R_k(z) = H(z)S_k(z)\,(k = 1, \ldots, K)$, which is a sequence of length $G + L - 1$, denotes the convolution of the channel response and the k-th user spreading sequence,

$P_k\ (k = 1, \ldots, K)$ is the received power of the k-th user,

$\mathbf{R}(z) = [R_1(z), \ldots, R_K(z)]^T$ is the overall channel response due to spreading and multipath effect,

$\mathbf{P} = diag\{\sqrt{P_1}, \ldots, \sqrt{P_K}\}$ is the $K \times K$ diagonal matrix representing the channel attenuations of the users,

$\mathbf{B}(z) = [B_1(z), \ldots, B_K(z)]^T$ is the entire input bit sequence vector.

The z-transform of the received signal, $y(t)$, is thus $Y(z) = X(z) + N(z)$, where $N(z)$ is the z-transform of the AWGN signal, $n(t)$, which has zero mean and two-sided power spectral density equal to $N_o/2$. Two assumptions were made here: the receiver knows exactly the received power of the users and the channel attenuation is generalised to be real for analysing coherent methods.

3 The Bit-based and Chip-based MLSE Detectors

The MLSE detector is optimum in the sense that it finds the input bit sequence which maximises the conditional probability, or likelihood of the observation sequence. This implies performing a search over all the possible transmitted sequences by comparing the likelihood ratios. It can be shown that this is equivalent to comparing the Euclidean distances between the received sequence and all the possible noiseless received sequences when the transmitted bit of the users are equiprobable and independent [4]. The sequence, $X(z)$, that is closest to the observation in terms of the Euclidean distance (minimum Euclidean distance) is then selected as the decoded sequence. Mathematically, it is

$$\begin{aligned}\hat{\mathbf{B}}(z) &= \arg\min_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \sum_{i=0}^{G(M-1)+L-2} [y(i)-x(i)]^2 && (6)\\ &= \arg\max_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \sum_{i=0}^{G(M-1)+L-2} \left[2y(i)x(i)-x^2(i)\right]. && (7)\end{aligned}$$

where $x(i)$ and $y(i)$ are the coefficients of z^{-i} in $X(z)$ and $Y(z)$ respectively. The term, $y^2(i)$, is common to all $\mathbf{B}(z)$, so it can be dropped. However, this implies an exhaustive search over the 2^{MK} possible sequences. Fortunately, $\hat{\mathbf{B}}(z)$ has the right structure to employ the Viterbi algorithm that results in a significantly better complexity. Implementing the MLSE using Viterbi algorithm in the asynchronous uplink, for both the AWGN channel and the multipath channel, have been shown in [2] and [6] respectively. For the AWGN channel, the front-end is the matched filter bank, and the back-end is the Viterbi algorithm that performs the likelihood ratios comparison. The entire outputs of the matched filter provides the sufficient statistics to perform Viterbi algorithm at the bit-level. As for the multipath uplink, the front-end is replaced by the RAKE receiver which despreads the signal as well as equalises the channel. In the downlink, where the system is synchronous, Viterbi algorithm can also be employed when there is multipath in the channel. Appendix A shows how the bit-based MLSE can be implemented with a RAKE receiver front-end without any degradation in performance. It should be emphasised here that the output of the RAKE receiver here should not be viewed as a set of pre-processed signals but a set sufficient statistics.

The MLSE can also be implemented at chip-level, that is without first matched filtering the signal with the spreading codes of the users. Instead of implementing eqn (7), eqn (6) can be used. At each chip, the squared of the differences between the received chip value and all the possible noiseless chip values are computed. We will call each of these values the Euclidean distance metric. Although the noiseless chip values, $x(i)$, can be calculated using eqn (4), it not only has to be based on the entire transmitted bit sequences but is also exponential in the message length. We proceed to show that a discrete-time system can be found such that it can be solved efficiently by the forward dynamic programming algorithm (the Viterbi algorithm). Starting from eqn (4),

$$\begin{aligned}X(z) &= \sum_{m=0}^{M-1}\left[\sum_{i=0}^{G+L-2}\mathbf{r}^T(i)z^{-i}\right]\mathbf{Pb}(m)z^{-Gm}\\ &= \sum_{m=0}^{M-1}\left[\sum_{i=0}^{G-1}\mathbf{r}^T(i)z^{-i}+\sum_{i=G}^{G+L-2}\mathbf{r}^T(i)z^{-i}\right]\mathbf{Pb}(m)z^{-Gm}\\ &= \sum_{m=0}^{M-1}\left[\sum_{i=0}^{G-1}\mathbf{r}^T(i)z^{-i}+\sum_{j=0}^{L-2}\mathbf{r}^T(G+j)z^{-G-j}\right]\mathbf{Pb}(m)z^{-Gm}\\ &= \sum_{m=0}^{M}\left[\sum_{i=0}^{G-1}\mathbf{r}^T(i)z^{-i}\mathbf{Pb}(m)+\sum_{j=0}^{L-2}\mathbf{r}^T(G+j)z^{-j}\mathbf{Pb}(m-1)\right]z^{-Gm}\\ &= \sum_{m=0}^{M}z^{-Gm}\sum_{i=0}^{G-1}x_i(\mathbf{b}(m-1),\mathbf{b}(m))z^{-i}, && (8)\end{aligned}$$

where $\mathbf{r}(i)=[r_1(i),\ldots,r_K(i)]^T$ $(i=0,\ldots,G+L-2)$, and $r_k(i)$ is the coefficient of z^{-i} in $R_k(z)\,(k=1,\ldots,K)$. We have also made use of the fact that $\mathbf{b}(m)=\mathbf{0}$ for $m=-1$ and $m=M$. The inner summation consists of all the noiseless chip values, $x_i(\mathbf{b}(m-1),\mathbf{b}(m))\,(i=0,\ldots,G-1)$, in the m-th message bit interval. The i-th chip value of $x_i(.)$ is independent of the previous bits if $i>L-2$. Only the first $L-2$ chips are interfered with by the previous bits. Eqn (8) shows that the noiseless chip values can be calculated just based on the current bits and (if necessary) the previous bits. These chip values also repeat every bit interval, so they can be calculated easily regardless of the message length. Decoding can also begin immediately once the signal is received and does not have to wait until transmission ends. Hence, the Viterbi algorithm can be readily applied to calculate the Euclidean distance metrics at each chip. A point to note here is that this implementation yields the same result as that implemented at bit-level since we are not making any assumptions or approximations. However, the chip-based MLSE has a higher complexity due to the multiplication at the each chip.

4 The Chip-based detector using Linear Distance Metrics

The chip-based MLSE detector yields the optimum performance, however the cost of such good performance is the multiplication (squaring) that has to be performed at each chip. Multiplications are very computational intensive for a DSP. If the absolute value is taken instead of the squared distance, then no multiplication is needed at all. However, this also means an inevitable trade-off in the performance of course. Our work here is to investigate whether the trade-off in complexity and performance is appropriate for practical implementation. Mathematically, the decision rule is now

$$\hat{\mathbf{B}}(z) = \arg \min_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \sum_{i=0}^{G(M-1)+L-2} |y_i - x_i|, \tag{9}$$

Hence, each linear distance metric value, $|y_i - x_i|$, is simply taking the absolute value of the difference between the received value and one of the possible noiseless received values at each chip. The absolute value can be taken simply by discarding the sign bit. Note that by implementing the receiver at the chip level, the matched filter bank is no long required, despreading, symbol-decoding as well as channel-equalisation are processed jointly.

For the synchronous AWGN channel, each bit interval is independent of the other bit intervals without intersymbol interference. This can be seen clearly from eqn (8) where the summation term that is dependent on the previous bit vector equates to zero since $L = 1$ here. Hence, a decision can be made once the complete sequence in a bit duration is received. In a bit interval, there are 2^K possible transmitted bit vectors. The corresponding 2^K possible metrics can be easily calculated at each chip, provided the spreading codes, channel response and received power are known. The decoding procedure is best illustrated with the state-transition trellis diagram in Figure 2, assuming a 2-user system. Each state, ($\mathbf{b}_j, i$), is determined by a possible bit vector, $\mathbf{b}_j$ $(j = 1, \ldots, 2^K)$ and the chip position, i $(i = 0, \ldots, G-1)$. The initial state is set to be an arbitrary state, (00,-1), since it does not have any effect on the remaining states. In the diagram, each transition is labelled with the value of $x_i(\mathbf{b}_j)$, which is independent of the previous bits. In this case, there is only one possible transition from one state to the next, since the transmitted bits remain unchanged in a bit duration. Hence, state ($\mathbf{b}_j, i$) can only transit to state ($\mathbf{b}_j, i+1$). The sum of all the metrics in each path (metric sum) determines the best path (path with the smallest metric sum) which corresponds to the best bit vector. The trellis diagram here is very simple and is not even necessary for decoding, but it gives an insight into decoding at chip level and is especially for the Viterbi algorithm, which is not necessary in an AWGN channel but essential in a multipath channel.

If the channel exhibits multipath properties, then the trellis has to be extended to the entire transmission. Now the trellis is not so straight forward, it has to be expanded between bits whenever multipath effect arises, and then collapsed when the interference from the previous bits dies away. For illustration, consider the same system above but now with a multipath of length $L = 4$. The state-transition trellis is given in Figure 3

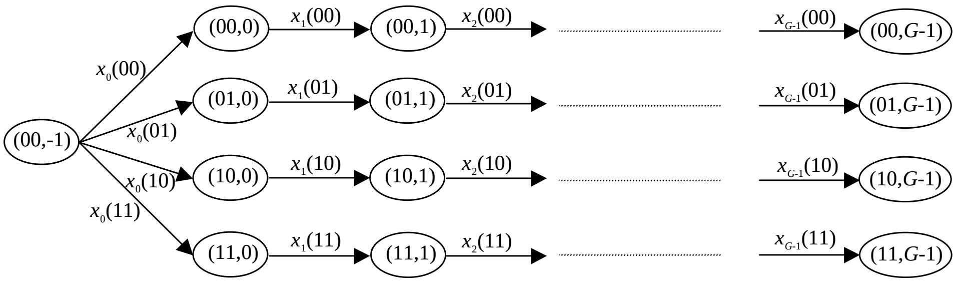

Figure 2: State-transition trellis for AWGN channel

which shows the transition between 2 adjacent bit-intervals, the $(m-1)$-th and m-th bit-intervals. The trellis is expanded when transiting from the last chip of the $(m-1)$-th interval to the first chip of the m-th interval. The state of the expanded trellis is described as $(\mathbf{b}_j(m-1), \mathbf{b}_k(m), i)$, which has to take the previous bit vector into consideration too. Each state in the $(m-1)$-th interval is expanded into 2^K states, resulting in a total of 4^K states and transitions each. For the second chip, it is still affected by the previous bits, so only horizontal transitions are possible. This ensures that at each state, there is only 1 incoming transition and thus preventing any decision making. Even though the third chip is still interfered by the previous bits, the trellis can be collapsed since the next chip is independent of the previous bits and so some form of decision can be made here. There are still 4^K transitions, only that 2^K of them converge into a state. At each terminating state of this transition, a decision can be made to select the survival path, which is the one with the smallest metric sum. Therefore, there will be a total of 2^K survival paths. The fourth chip $(i = L-1)$ is only dependent on the current bits, hence only 2^K noiseless received chip values or transitions are possible.

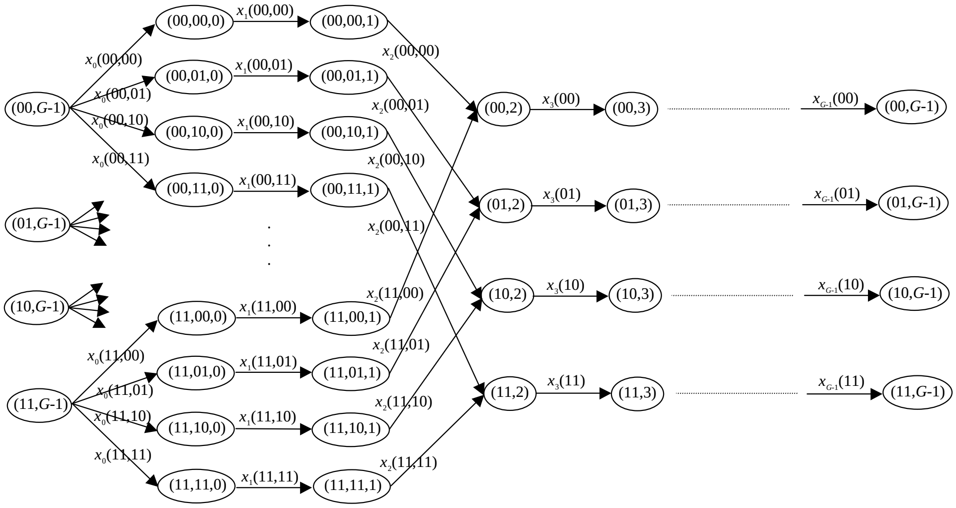

Figure 3: State-transition trellis for multipath channel

The values, $x_i\ (i = 1, \ldots, G-1)$, repeat after each bit interval (shown in eqn (8)), hence the trellis of 1 bit duration is sufficient to compute the metric values for the entire transmission. The metric associated with a transition from a state will be added to the metric sums of all incoming paths into that state. If any other path to each state replaces the survival path in any overall path, then the path metric would be larger. This is the basis of the Viterbi algorithm. Once the survival paths have been selected, then they can be traced back. If all the survival paths originate from a particular state, then the bit vector associated with this state is the decoded bit vector.

5 Complexity Analysis

The complexity of this proposed scheme is now compared with the bit-based MLSE under the AWGN channel downlink condition where all the users have equal power and $L = 1$. The proposed scheme does not involve any multiplication operation during run-time as the values of $x_i\ (i = 0, \ldots, G-1)$ are pre-calculated, only 2^K addition operations corresponding to the 2^K possible states are carried out at each chip. In one bit interval, total number of arithmetic operations is $G(G-1)2^K$ additions for the 2^K metric sums. The complexity of the bit-based MLSE at run-time is to be determined next. Assume that the matched filter bank does not require any multiplication since the matched filters only involve +1 and -1, and multiplication with -1 can be done by a simple inversion. However, $G-1$ additions are required for each matched filter. For maximum likelihood ratio comparisons, we make use of eqn (17). Since $2\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle\mathbf{P}$ is common to all the possible states, only K multiplications and $K(G-1)+1$ additions are required for this partial term. Hence, a total of K multiplications and $(K-1)2^K + K(G-1)+1$ additions for the calculation of all the 2^K possible values of $2\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle\mathbf{P}\mathbf{b}(m)$. During run-time, the computation of the 2^K possible metrics, $\lambda_m(\mathbf{b}(m))$, needs K multiplications and $K(2^K + G - 1) + 1$ additions. Therefore, the proposed scheme does not require any multiplications, while the bit-based MLSE is linear in the number of users. However, for addition, the dominant factors in the bit-based MLSE and the chip-based proposed detector are respectively $K2^K$ and $G^2 2^K$. As G is usually greater than K, the number of additions in the proposed scheme is expected to be higher. It should be emphasised here that multiplication operation is the crucial factor in terms of complexity. For any modulation scheme higher than the QPSK, the number of multiplications in the bit-based MLSE would be exponential in the number of users while the proposed scheme still remains at 0.

	Multiplications	Additions
Run-time		
Chip-based proposed detector		
Each transition, $\lvert y_i - x_i(\mathbf{b}(m))\rvert$	0	1
Total for all possible states	0	$G(G-1)2^K$
Bit-based MLSE		
$\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle$ for all users	0	$K(G-1)$
$2\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle\mathbf{P}\mathbf{b}(m)$	K	$K(G-1)$
for all states		$+1+(K-1)2^K$
$\lambda_m(\mathbf{b}(m))$ for all states	K	$1+K(2^K+G-1)$
Pre-calculation		
Chip-based proposed detector		
$\sum_{i=0}^{G-1}\mathbf{r}^T(i)\mathbf{P}\mathbf{b}(m)$ for all states	KG	$(K-1)2^K G$
Bit-based MLSE		
$\mathbf{b}^T(m)\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}(m)$	K^2+1	$(G-1)K^2$
for all states		$+2^K(K^2-1)$

Table 1: Complexity comparison between the chip-based linear distance metric detector and the bit-based MLSE detector in a AWGN channel.

For pre-calculation, proposed scheme needs K multiplications (for equal power case, $\mathbf{P}$ is just a scalar) and $K-1$ additions per chip per state using eqn (8) and taking note that the second term is zero. The value of $\mathbf{r}^T(i)\mathbf{P}$ (K multiplications) is the same for all states, hence giving a total of GK multiplications and $G2^K(K-1)$ additions for all the 2^K possible states in every bit. From eqn (17), the bit-based MLSE requires $1+K^2$ multiplications (assuming no multiplication for the calculation of correlation matrix and whenever multiply with +1 and -1) and $K^2(G-1)$ additions for the calculation of $\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)\rangle\mathbf{P}$, which remains unchanged for all the states. The last term in eqn (17) is dropped due to the AWGN channel. The total number of arithmetic operations for the bit-based MLSE is thus $1+K^2$ multiplications and $2^K(K^2-1)+K^2(G-1)$ additions. Clearly, the pre-calculation of the MLSE is of the same order as the proposed scheme. Table 1 gives a summary of the complexity comparison.

We proceed to compare the complexities of these 2 detectors under the multipath conditions. For the MLSE, the front-end is the RAKE receiver with L fingers if the number of multipath is L. Hence, the RAKE receiver for each user requires L multiplications and $GL-1$ additions. From eqn (17), only the first term has to be computed at run-time since it involves the observation, the other two can be precomputed jointly to yield the 4^K possibilities. The total number of arithmetic operations for the 2^K possible first term is $K(L+1)$ multiplications and $K(GL-1)+2^K(K-1)$ additions. Hence, the overall complexity in computing the bit metric given by eqn (17) is $K(L+1)$ multiplications and $K(GL-1)+2^K(K-1)+4^K$ additions. For the proposed scheme, the calculation of the linear distance metric once again does not require any multiplication. The number of additions needed for all the metrics calculation in the expanded and the rest of the trellis are $(L-1)4^K$ and $(G-L+1)2^K$ respectively. Hence, for the metric sums in a bit interval, total number of additions is $2(L-1)4^K+(G-L+1)2^{K+1}$. Number of multiplications in the bit-based MLSE is now dependent not only on the number of users but also the number of multipath. On the other hand, the proposed scheme still does not require any multiplication at all. Although both detectors have the same order of complexity ($O(4^K)$) in terms of additions, the proposed scheme needs more additions due to the scaling factor $2L$ in the dominant term.

Next, we compare the complexity of the pre-computation. Eqn (17) is again used for the bit-based MLSE. The correlation matrix of $\mathbf{R}(z)$ including the power, $\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)\rangle\mathbf{P}$, just have to be computed once as it is common to all the states. Similarly, $2\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}$ in the last term needs to be computed only once. Hence, the complexity of the common matrices in the second and third terms are respectively $(G+L-1)K^2+1+K^2$ multiplications and $(G+L-2)K^2$ additions, and $(L-1)K^2+1+K^2$ multiplications and $(L-2)K^2+1$ additions (using an addition to compute the scaling by a factor of 2 in the last term). Total complexity of the second term is thus $(G+L)K^2+1$ multiplications and $(G+L-2)K^2+2^K(K^2-1)$ additions for the 2^K possible states. The last term is harder to compute as it involves 2 bit-interval, resulting in 4^K possible states. Making use of the fact that each of the 2^K possible $2\mathbf{b}^T(m-1)\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}$ can be reused by all the 2^K possible $\mathbf{b}(m)$, the last term requires LK^2+1 multiplications and $(L-2)K^2+2^KK(K-1)+4^K(K-1)+1$ additions. Therefore, the total precomputation is $(G+2L)K^2+2$ multiplications and $(G+2L-4)K^2+2^K(2K^2-K-1)+4^K+1$ additions. For the proposed scheme, we make use of eqn (8) to find the pre-calculation complexity. Now the second term has to be considered as $L>1$ and it needs $(L-1)K$ multiplications and $(L-1)2^K(K-1)$ additions every bit. The complexity of the first term is the same as that computed in the AWGN channel scenario. Thus, overall precomputation is $(G+L-1)K$ multiplications and $(G+L-1)2^K(K-1)+4^K$ additions every bit interval. In the multipath case, the number of multiplications in the MLSE is an order higher than the proposed detector. Although both detector have the same order of complexity, ($O(4^K)$), in terms of additions, the MLSE still has a marginally higher complexity due to the additional K^2 term. Refer to Table 2 for a summary of the complexity comparison in the multipath scenario.

To illustrate the complexity gain, we consider two examples. Both with 3 paths ($L=$

	Multiplications	Additions
Run-time		
Chip-based proposed detector		
Each transition, $\lvert y_i - x_i(\mathbf{b}(m-1), \mathbf{b}(m))\rvert$	0	1
Total for all possible states	0	$2(L-1)4^K$
		$+(G-L+1)2^{K+1}$
Bit-based MLSE		
$\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle$ for all users	KL	$K(GL-1)$
$2\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle\mathbf{P}\mathbf{b}(m)$	$K(L+1)$	$K(GL-1)$
for all states		$+1+(K-1)2^K$
$\lambda_m(\mathbf{b}(m-1), \mathbf{b}(m))$	$K(L+1)$	$1+K(GL-1)$
for all states		$+(K-1)2^K+4^K$
Pre-calculation		
Chip-based proposed detector		
$\sum_{i=0}^{G-1}\mathbf{r}^T(i)\mathbf{P}\mathbf{b}(m)$ for all states	KG	$(K-1)2^KG$
$\sum_{j=0}^{L-2}\mathbf{r}^T(G+j)\mathbf{P}\mathbf{b}(m)$ for all states	$K(L-1)$	$(K-1)2^K(L-1)$
Total for all states	$K(G+L-1)$	$(K-1)2^K(G+L-1)$
		$+4^K$
Bit-based MLSE		
$\mathbf{b}^T(m)\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}(m)$	$(G+L)K^2+1$	$(G+L-2)K^2$
for all states		$+2^K(K^2-1)$
$2\mathbf{b}^T(m-1)\mathbf{P}\langle\mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}\mathbf{b}(m)$	LK^2+1	$1+(L-2)K^2$
for all states		$+2^K(K-1)K$
		$+4^K(K-1)$
Total for all states	$(G+2L)K^2+2$	$(G+2L-4)K^2$
		$+2^K(2K+1)(K-1)$
		$+4^K+1$

Table 2: Complexity comparison between the chip-based linear distance metric detector and the bit-based MLSE detector in a multipath channel.

3). One has 8 users ($K = 8$) and a processing gain of $G = 16$, the other example has 4 users and $G = 8$. The complexity is tabulated in Table 3.

6 Simulation Results

In order to find the performance of the proposed scheme, Monte-Carlo simulations were used to obtain the bit error rate (BER). Initially, only the synchronous case in AWGN channel was considered. The proposed scheme was simulated for a system with equal-power users and random spreading codes of length, $G = 7$. Figure 4 shows the BER averaged over all users in the system plotted against the number of active user for $E_b/N_0 = 9$ dB. The simulation was carried out for 1 million trials to ensure reliable data for the low BER. Other MUDs such as the conventional detector, the MMSE, and both the chip-based and the bit-based MLSE, were also simulated so as to compare the performance. The graphs labelled “Euclidean distance” and “MLSE” are the chip-based and bit-based MLSE detectors respectively. Theoretically, this detector and the bit-based MLSE give the same result. This is verified as observed from the plot. The purpose of this comparison is to ensure that the simulation for the proposed algorithm was simulated correctly. The conventional detector performs poorly especially when the number of users (MAI) increases. The MMSE detector shows substantial improvement in performance over the conventional detector but still far from optimum. The proposed scheme (“Linear distance”) only degrades slightly from optimum and is even indistinguishable from

AWGN	Multiplications	Additions	Multiplications	Additions
Run-time	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
Proposed	0	61440	0	896
MLSE	8	2169	4	93
Pre-calculation	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
Proposed	128	28672	32	384
MLSE	65	17088	17	352

Multipath	Multiplications	Additions	Multiplications	Additions
Run-time	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
Proposed	0	269312	0	1216
MLSE	32	67705	16	397
Pre-calculation	$K = 8, G = 16$	$K = 8, G = 16$	$K = 4, G = 8$	$K = 4, G = 8$
Proposed	144	97792	40	736
MLSE	1410	97153	226	849

Table 3: Complexity comparison between the chip-based linear distance metric detector and the bit-based MLSE detector for AWGN channel and multipath ($L = 3$) channel.

the optimum for large MAI. This is probably because the Gaussian noise becomes insignificant as compared to the large MAI. The metrics that do not correspond to the actual transmitted bits are very large as compared to the near zero value of the matrices that are associated with the actual transmitted bits. Since these metrics are already very large, squaring them does not improve performance significantly.

A stationary multipath channel is considered next. The channel has impulse response

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}, \tag{10}$$

which is the channel used in [3]. As would be the case in a downlink, all signals are assumed to pass through the same channel. Both the “Linear” and “Euclidean” detectors were simulated for random codes and Walsh codes, both of length $G = 8$ and $E_b/N_0 = 7$ dB. Refer to Figure 5. Assume all users are of equal power again. To ensure that the codes selected are unbiased, the codes selected for the active users are randomised. The simulations with 100000 trials were repeated 200 times, and each time, a new set of codes was randomly selected for the users. Walsh codes are orthogonal codes that are ideal for a perfect channel, but degrade disastrously in an imperfect channel. This is observed in the simulation plot. Even though the two types of codes are of the same processing gain, the random codes show much better performance. The RAKE receiver was also simulated for the two kinds of codes and the BER results are also plotted in Figure 5. It is observed that here the Walsh codes perform slightly better when the number of active users is more than 3. Under no MAI condition, the RAKE receiver performs close to the proposed scheme and the optimum receiver, but degrades rapidly when MAI increases, regardless of the codes used.

The graph of the BER against E_b/N_0 for the same multipath channel was also plotted as shown in Figure 6. A 4-user system using spreading code of processing gain 8 is considered here. The performance of the chip-based “Linear distance” and “Euclidean distance” is much better than the RAKE receiver. The random codes once again perform better than the Walsh codes except for the RAKE receiver. For a BER of -1.6, it is found that the proposed scheme requires a E_b/N_0 of just about 0.5 dB more to attain the same performance as the optimum case for both the codes. Note that this performance is unachievable by the RAKE receiver.

The length of the survival paths in the Viterbi algorithm considered so far is unlimited. If the convergence is slow, then a large amount of storage is needed to store the long survival paths and the delay may be intolerable. For practical implementation, the length is preset to a sufficiently small finite value, which means that if the algorithm has

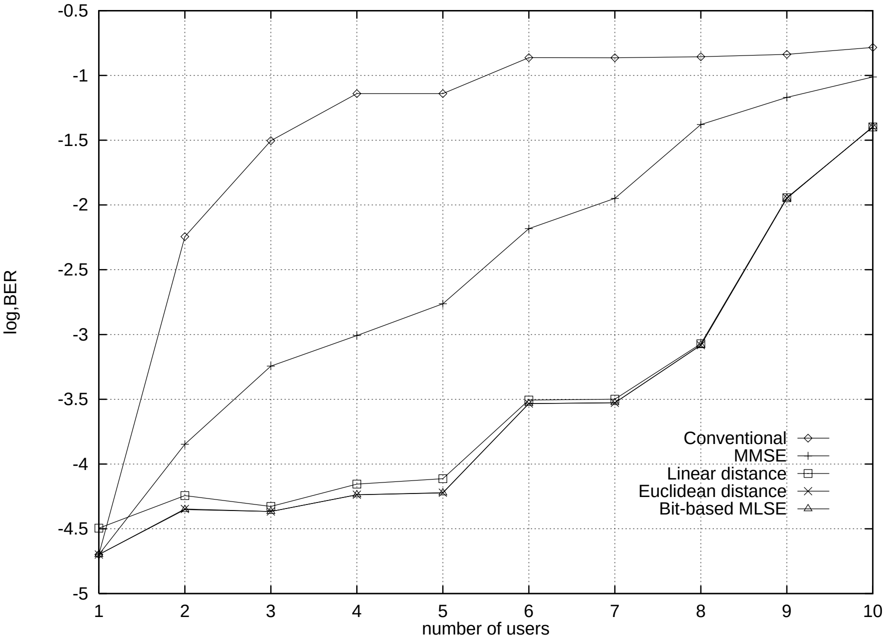

Figure 4: BER against number of active users for $E_b/N_0 = 9$ dB in AWGN channel.

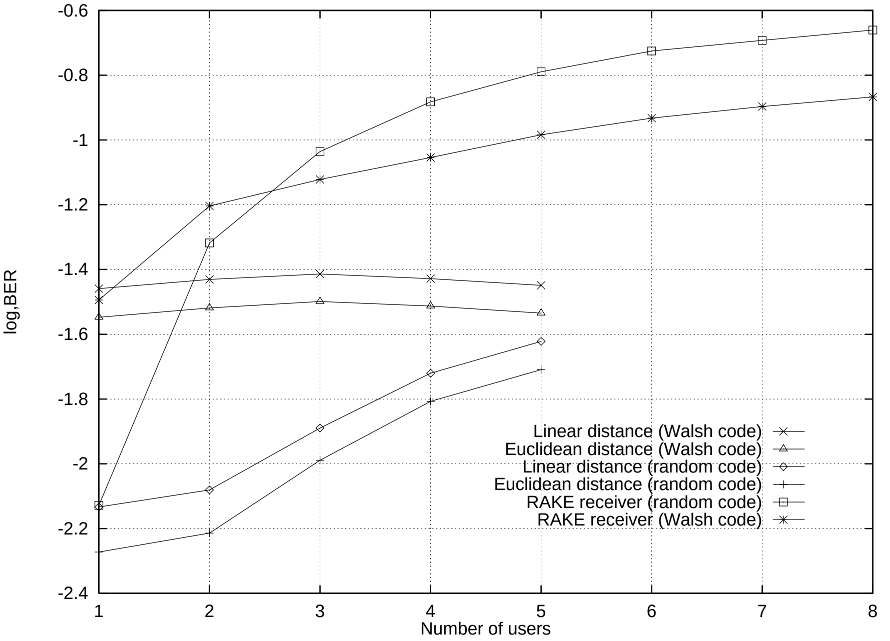

Figure 5: BER against number of active users for $E_b/N_0 = 7$ dB in multipath channel.

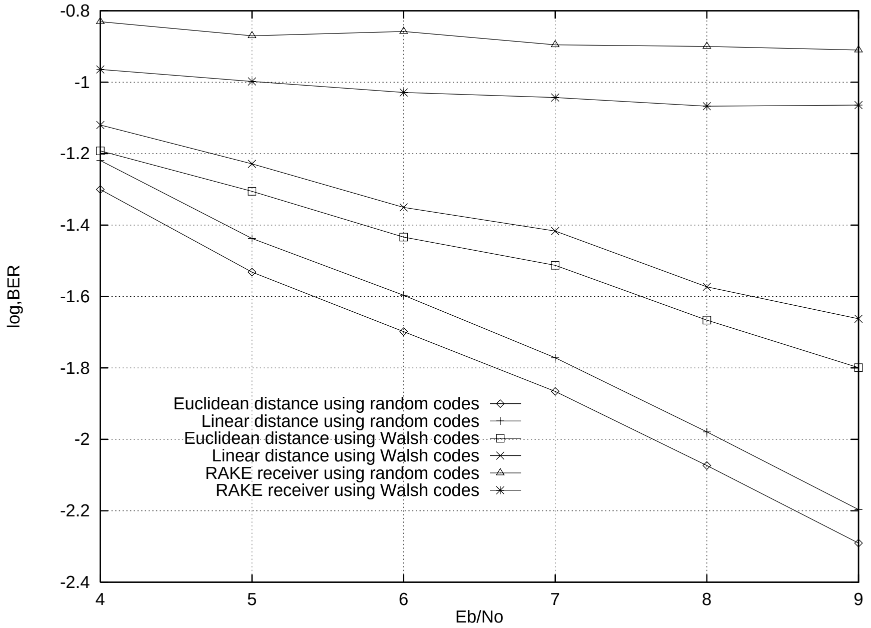

Figure 6: BER against E_b/N_0 for 4 users using codes of processing gain $G = 8$ in multipath channel.

not converged after this preset delay, the bit vector corresponding to the best survival path is selected. Hence, the rate of convergence of the Viterbi algorithm in the proposed scheme is a crucial factor. Simulations were again used to determine the performance when the path length varies from 1 bit to 5 bits. A path length of 1 bit means that decision for the m-th bit interval is made immediate at the next $(m+1)$ interval. Since the transitions are all horizontal after the trellis converges in the $(m+1)$-th interval, there is no point delaying decision-making until the end of the interval. A decision made immediately once the trellis converges (in this example where $L = 3$, it is the second chip) would give the same result. Generally, a delay of j bits implies making a decision at chip $i = L-2$ of the $(m+j)$-th bit-interval. Figure 7 shows the convergence rate of the Viterbi algorithm for a 4-user system. It is found that convergence is very rapid and for a delay of just 2 bits, the performance is already very close to that of the unlimited path length for both the proposed “Linear distance” and the optimum “Euclidean distance” chip-based detectors. The same trend is observed in both codes although the Walsh codes with the same processing gain perform much worse. The maximum survival path lengths under unlimited delay condition are also indicated in the figure and they are very much larger than the fixed delay. Hence, the well-known advantage that little degradation of performance occurs when the Viterbi algorithm uses a sufficient fixed finite decision delay in real-time implementation has been verified. The RAKE receiver was also simulated for the two kinds of codes in the multipath channel as shown in the plot. Notice the wide gap between the RAKE performance and the proposed scheme.

A mobile communication channel is rarely stationary since the paths to the mobiles vary rapidly especially when the mobiles are in motion. Hence, in order to model a realistic mobile communication channel, the COST 207 models, used for simulating GSM systems, is employed to test the proposed scheme. The chip rate channel simulator used for this model has been taken from [7]. The channel model used here is TU6. The fading tap-weights were generated at a rate of 800Hz and a maximum Doppler frequency shift

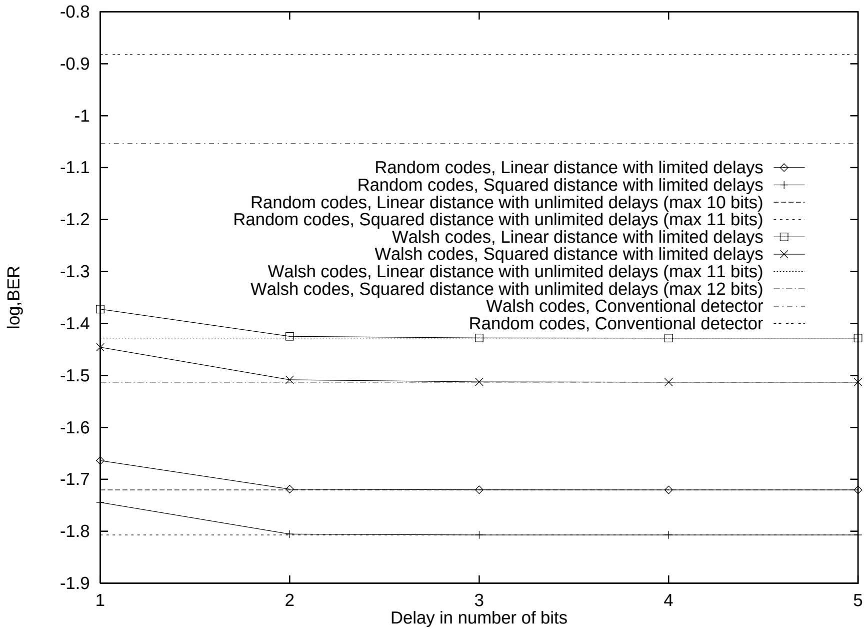

Figure 7: BER against number of delays for $E_b/N_0 = 7$ dB and 4 users using codes of length 8 in stationary multipath channels.

of 30Hz. A chip rate of 1.2288 Mchips / sec was selected since it corresponds to that used in the IS95 standard. Hence, the channel changes every 1536 chips and it is assumed here that the receiver can track this variation accurately. These parameters are also used in [7]. Since the proposed scheme assumed coherent detection and real channel attenuation, the magnitudes of the complex fading weights, which were generated by the simulator, were used instead. Simulations on the non-stationary channel were carried out on both random codes and Walsh codes for $G = 8$ and $G = 16$. The convergence rate of the Viterbi algorithm also follows similar trend as in the stationary channel case. This plotted in Figure 8. It is also found that doubling the processing gain does not improve the Walsh codes' performance significantly, while the random codes show great improvement. Moreover, the maximum survival path lengths of the Walsh codes are much longer than that of the random codes. Therefore, Walsh codes are not recommended for the downlink of a DS-CDMA system.

7 Conclusions

In this paper, a MUD for the downlink of the DS-CDMA systems is proposed. This detector operates at chip-level and uses the linear distance metrics for its decision rule instead of the Euclidean distance metrics. The paper shows that the optimum MLSE detector can be implemented at both the bit and chip level. Simulation results of the proposed detector were compared with the chip-based MLSE, while the complexity comparison was carried out on the bit-based MLSE since it has a simpler implementation. It was found that the proposed detector does not require any multiplication operation at runtime both for the AWGN channel as well as the multipath channel, while the bit-based MLSE needs an exponentially increasing number of multiplications. The overall (including precomputation) number of multiplication operations of the proposed detector is linear with the number of active users in the system. Although the number of addition

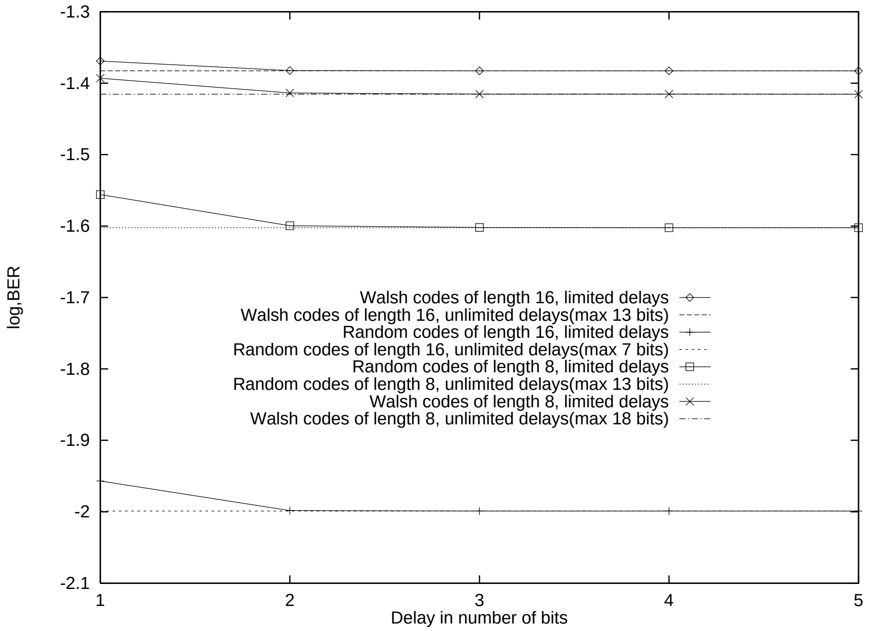

Figure 8: BER against delays for $E_b/N_0 = 7$ dB and 4 users using the proposed chip-based linear distance metric detector in COST 207 time-varying channels.

operations is still exponential in the number of active users, the implementation of the detector is more realistic now, since multiplication being the critical factor in the DSP has been greatly reduced. Furthermore, in the Universal Mobile Telecommunication System (UMTS) Delta proposal where the average number of users in a time-frame is only 8 and the processing gain of the codes is only 16, the practical implementation of the proposed detector is even more encouraging. It has been observed in the simulation results that the proposed scheme degrades only slightly from the optimum performance, under the AWGN channel, stationary and time-varying multipath fading channels scenarios. The favourable convergence rate of the Viterbi algorithm has also been verified by simulations, which shows that for just a delay of 2 bit-intervals, the performance has already converged to the unconstrained Viterbi algorithm.

A Implementing the bit-based MLSE with Viterbi Algorithm

The following proof shows the implementation of the MLSE using the Viterbi algorithm in the downlink of a DS-CDMA system. Eqn (7) can be rewritten as

$$\hat{\mathbf{B}}(z) = \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \left[2\langle Y(z), X(z)\rangle - \langle X(z), X(z)\rangle\right], \tag{11}$$

where

$$\langle Y(z), X(z)\rangle = \sum_i y(i)x(i). \tag{12}$$

First, consider

$$\begin{aligned}\langle Y(z), X(z)\rangle &= \langle Y(z), \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z)\rangle \\ &= \sum_{m=0}^{M-1} \langle Y(z), \mathbf{R}^T(z)\mathbf{P}\mathbf{b}(m)z^{-Gm}\rangle \\ &= \sum_{m=0}^{M-1} \langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle \mathbf{P}\mathbf{b}(m), \qquad (13)\end{aligned}$$

where $\mathbf{b}(m) = [b_1(m), \ldots, b_K(m)]^T$ is the m-th input bit vector and $\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle$ is actually the output of the RAKE matched filter bank in the m-th message interval. Even though each value of $\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle$ is not a sufficient statistics for the detection of $\mathbf{b}(m)$, the entire sequence of outputs of the RAKE receiver is a sufficient statistics for the selection of the most likely sequence, $\mathbf{B}(z)$.

Next, the second term is expanded as,

$$\begin{aligned}\langle X(z), X(z)\rangle &= \langle \mathbf{B}^T(z)\mathbf{P}\mathbf{R}(z), \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z)\rangle \\ &= \left\langle \sum_{j=0}^{M-1} \mathbf{b}^T(j)z^{-Gj}\mathbf{P}\mathbf{R}(z), \sum_{m=0}^{M-1} \mathbf{R}^T(z)\mathbf{P}\mathbf{b}(m)z^{-Gm} \right\rangle \\ &= \sum_{m=0}^{M-1} \left[\mathbf{b}^T(m)\mathbf{P}\langle \mathbf{R}(z), \mathbf{R}^T(z)\rangle \mathbf{P}\mathbf{b}(m)\right. \\ &\quad \left. + 2\mathbf{b}^T(m-1)\mathbf{P}\langle \mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle \mathbf{P}\mathbf{b}(m)\right], \qquad (14)\end{aligned}$$

making use of the fact that $\mathbf{b}(m) = \mathbf{0}$ for $m < 0$ and

$$\langle \mathbf{R}(z)z^{-Gj}, \mathbf{R}^T(z)z^{-Gm}\rangle = \begin{cases} \langle \mathbf{R}(z), \mathbf{R}^T(z)\rangle & \text{if } j = m \\ 0 & \text{if } |j-m| > 1 \end{cases}. \qquad (15)$$

The second condition is due to the fact that the all components of $\mathbf{R}(z)$ are of length $G + L - 1 < 2G$, or simply, the bits that are more than 1 bit interval away from each other do not interfere with each other.

Hence, the decision rule can be expressed as a sum of metrics,

$$\hat{\mathbf{B}}(z) = \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \sum_{m=0}^{M-1} \lambda_m(\mathbf{b}(m-1), \mathbf{b}(m)), \qquad (16)$$

with the transition metric given as

$$\begin{aligned}\lambda_m(\mathbf{b}(m-1), \mathbf{b}(m)) &= 2\langle Y(z), \mathbf{R}^T(z)z^{-Gm}\rangle \mathbf{P}\mathbf{b}(m) \\ &\quad - \mathbf{b}^T(m)\mathbf{P}\langle \mathbf{R}(z), \mathbf{R}^T(z)\rangle \mathbf{P}\mathbf{b}(m) \\ &\quad - 2\mathbf{b}^T(m-1)\mathbf{P}\langle \mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle \mathbf{P}\mathbf{b}(m). \qquad (17)\end{aligned}$$

Each metric can be computed using the current RAKE receiver output, a subset of control sequences, $\mathbf{b}(m)$, and the set of transmitted sequences, $\mathbf{b}(m-1)$. Therefore, this is a discrete time deterministic control problem with additive cost and finite input and state spaces, so the Viterbi algorithm can be readily applied.

If there is no multipath in the channel, then $R_k(z) = S_k(z)\,\forall(k = 1, \ldots, K)$ and the RAKE receiver reduces to the conventional matched filter bank. Since $S_k(z)$ is of length G, there is no interference from the other bit intervals. The last term in eqn (17) becomes 0 since $\langle \mathbf{R}(z), \mathbf{R}^T(z)z^{-G}\rangle = 0$. Hence, the transition metric, $\lambda_m(.)$, does not involve any other bit-interval and a decision can be made for the bit interval in question based on these metrics.

References

[1] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[2] S. Verdu, "Minimum Probability of Error for Asynchtonous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

[3] D. G. M. Cruickshank, "Radial basis function receivers for DS-CDMA," *IEE Electronics Letters*, vol. 32, pp. 188–190, February 1996.

[4] W. V. Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276–283, February 1976.

[5] G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[6] Z. Zvonar and D. Brady, "Optimum Detection in Asynchronous Multiple-Access Multipath Rayleigh Fading Channels," in *The Proceedings of the 1992 Conference on Information Science and Systems, Princeton, NJ*, March 1992.

[7] D. I. Laurenson, D. G. M. Cruickshank, and G. J. R. Povey, "A Computationally Efficient Channel Simulator for the COST 207 Channel Models," in *Proceedings of the IEE Colloquium on Computer Modelling of Communications Systems, Digest*, pp. 8 / 1–8 / 6, IEE, May 1994.

A CHIP BASED MULTIUSER DETECTOR FOR THE DOWNLINK OF A DS-CDMA SYSTEM USING A FOLDED STATE-TRANSITION TRELLIS

Hak K. Sim, David G. M. Cruickshank

Department of Electronics and Electrical Engineering
The University of Edinburgh
The King's Buildings, Mayfield Rd, Edinburgh EH9 3JL
PH: (+44) 131 650 5655, FAX: (+44) 131 650 6554
e-mail: hks@ee.ed.ac.uk, dgmc@ee.ed.ac.uk

Abstract – The maximum-likelihood sequence estimation (MLSE) multiuser detector has very good performance but at the expense of the exponentially increasing number of states. We propose a scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved. By using all the possible discrete chip values as the states in the transition trellis diagram and operating the Viterbi algorithm at the chip level, the number of states is reduced from 2^K to $K+1$ where K is the number of users.

I. INTRODUCTION

In a direct-sequence code-division multiple-access (DS-CDMA) system, users are distinguished by using spreading codes. However, the multipath fading effect in the channel causes intersymbol interference. This destroys any orthogonality of the spreading codes and causes multiple access interference (MAI) from the other users. The MAI is not an inherent problem of DS-CDMA, but of the conventional single-user matched filter receiver [1]. To overcome the MAI, multiuser detectors (MUDs) have been suggested where the users are decoded jointly for their mutual benefits.

The maximum likelihood sequence estimator (MLSE) [2], which finds the optimum data bit vector that results in minimum joint probability, is exponential in the number of users. The various bit combinations in the MLSE are taken to be the different states in the decoding trellis. We proposed a scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved. Due to the fact that the combined chip values of the spreading codes only take discrete values, the number of states can be reduced by using these discrete values as the states in the transition trellis diagram and operating the Viterbi algorithm [3] at the chip level.

We have also investigated the possibility of using the absolute linear distance metric instead of the Euclidean distance metric. The former does not involve any multiplication, while the latter performs squaring for each metric calculation.

Section ii of this paper describes the DS-CDMA channel model of the downlink. This is followed by a discussion of the bit- and chip-based MLSE. Section iv introduces the proposed detector in an AWGN environment. State-transition trellis diagrams are used to explain the decoding procedure at the chip-level. The trellis can also operate with multi-chip states in order to improve performance. Simulation results are presented in Section v. Finally, in Section vi some conclusions are drawn.

II. THE DS-CDMA DOWNLINK CHANNEL MODEL

The equivalent discrete-time mathematical model for the downlink of a DS-CDMA system will be discussed in this section. A bit- and chip-synchronous AWGN channel with K users sharing the same bandwidth is shown in Figure 1. Each user in the system has a pre-assigned spreading waveform,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}. \qquad (k = 1, \ldots, K) \qquad (1)$$

where G is the processing gain of the spreading waveform, $c_{i,k} = \{-1, +1\}$ is the i-th chip value of the k-th user. It is assumed that the samples are taken at chip rate. Consider a BPSK modulated system. The input bit sequences for different users are identically independently distributed (i.i.d) and equiprobable. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink.

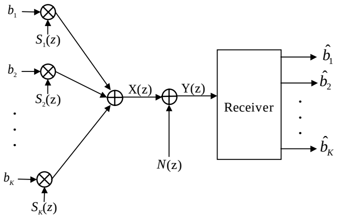

Figure 1: Synchronous AWGN system model

Assume that each user transmits with constant power. In the synchronous AWGN downlink channel, each bit interval is independent of the other bit intervals without intersymbol interference. Hence, we can simply consider a bit interval. The noiseless received signal in a bit interval, can be expressed as

$$X(z) \quad = \quad \sum_{k=1}^{K} S_k(z)\sqrt{P_k}b_k = \sum_{i=0}^{G-1} x_i(\mathbf{b})z^{-i}, \quad (2)$$

where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the input bit vector, $b_k = \{-1, +1\}$ and P_k represents the input bit symbol and the received power of the k-th user respectively, and $x_i(\mathbf{b})$ is the i-th noiseless received chip signal in a bit interval as a result of transmitting the data vector $\mathbf{b}$.

The received signal, is $Y(z) = X(z) + N(z)$, where $N(z)$ is the AWGN signal, which has zero mean and two-sided power spectral density equal to $N_o/2$. Three assumptions are made here: the receiver knows exactly the received power and the spreading codes of the users, and the channel attenuation is generalised to be real for analysing coherent methods.

III. THE BIT- AND CHIP-BASED MLSE DETECTORS

The MLSE detector is optimum in the sense that it finds the input bit sequence which maximises the conditional probability, or likelihood of the observation sequence. This implies comparing the Euclidean distances between the received sequence, $Y(z)$, and all the possible noiseless received sequences, $X(z)$, when the transmitted bits of the users are equiprobable and independent [4]. Mathematically, the input bit vector for one interval is decoded as

$$\hat{\mathbf{b}} \quad = \quad \arg \min_{\mathbf{b} \in \{-1,+1\}^K} \sum_{i=0}^{G-1} [y_i - x_i(\mathbf{b})]^2 \qquad (3)$$

$$\quad = \quad \arg \max_{\mathbf{b} \in \{-1,+1\}^K} \sum_{i=0}^{G-1} \left[2y_i x_i(\mathbf{b}) - x_i^2(\mathbf{b})\right] .(4)$$

where y_i is the coefficient of z^{-i} in $Y(z)$. However, this implies an exhaustive search over the 2^K possible sequences. Implementing the MLSE using Viterbi algorithm in the asynchronous uplink, for both the AWGN channel and the multipath channel, has been shown in [2] and [5] respectively. It can be shown easily that the outputs of the conventional matched filters provide sufficient statistics to perform eqn (4) at the bit-level.

The MLSE can also be implemented at chip-level, that is without first matched filtering the signal with the spreading codes of the users. Eqn (3) is used instead. A point to note here is that this implementation yields the same result as that implemented at bit-level since we are not making any assumption or approximation. However, the chip-based MLSE has a higher complexity due to the multiplication at each chip. At each chip, the square of the differences between the received chip value and all the possible noiseless chip values are computed. We will call each of these values the Euclidean metric. Note that a simple simplification can be made here by using the absolute distance instead, ie using $|y_i - x_i(\mathbf{b})|$. Multiplications are very computational intensive for a DSP. Since the absolute value can be taken simply by discarding the sign bit, no multiplication is needed at all. However, this also means an inevitable loss of performance which in practice turns out to be small.

In a bit interval, there are 2^K possible transmitted bit vectors. The corresponding 2^K possible metrics can easily be calculated at each chip using eqn (2), provided the spreading codes and received power are known. The decoding procedure is best illustrated with the state-transition trellis diagram in Figure 2, assuming a 2-user system. Each state, $(\mathbf{b}_j, i)$, is defined by a possible bit vector, $\mathbf{b}_j$ $(j = 1, \ldots, 2^K)$ and the chip position, i $(i = 0, \ldots, G-1)$. The initial state is arbitrary, since it does not have any effect on the remaining states. In the diagram, each transition is labelled with the value of $x_i(\mathbf{b}_j)$. As the transmitted bits remain unchanged in a bit duration, state

$(\mathbf{b}_j, i)$ can only transit to state $(\mathbf{b}_j, i+1)$. The sum of all the metrics in each path (metric sum) determines the best path (path with the smallest metric sum) which corresponds to the best bit vector. The trellis diagram here is not necessary for decoding, but it gives an insight into decoding at chip level and provides the basis for the folded trellis to be discussed next.

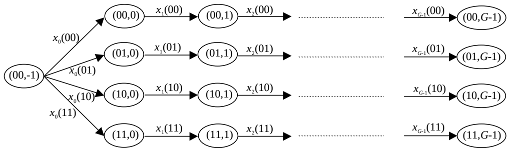

Figure 2: State-transition trellis for AWGN channel

IV. THE FOLDED STATE-TRANSITION TRELLIS

In this section, we devise a scheme to fold up the state-transition trellis shown in Figure 2. Without loss of generality, we assume equal power for all the users ($P = P_k \, (k = 1, \ldots, K)$). The noiseless received signal in eqn (2) becomes

$$X(z) = \sqrt{P} \sum_{i=0}^{G-1} v_i(\mathbf{b}) z^{-i} \qquad (5)$$

where $v_i(\mathbf{b})$ only takes discrete values from $-K$ to K in steps of 2 since $c_{i,k}$ and b_k only take the two possible values -1 or +1 for $i = 0, \ldots, G-1$ and $k = 1, \ldots, K$. Total number of discrete values is hence $K+1$. Let $U(d) \, (d = 0, \ldots, K)$ be the set consisting of all the $K+1$ possible values of $v_i(\mathbf{b})$. If all these values, $U(d)$, are used as the states in the trellis, the trellis is folded up. For example, consider a 2-user case with spreading codes {-1,-1,1,-1,-1,-1,1} and {-1,1,1,-1,-1,1,1}. Hence the resulting complete AWGN trellis is shown in Figure 3. Using the $K + 1 = 3$ discrete values of -2, 0 and 2 as the states in the transition trellis diagram, the trellis can be folded up as shown in Figure 4.

From eqn (5), the transition value pertaining to the transition path entering state $U(d)$ is $\sqrt{P}U(d) = x_i(\mathbf{b}) = x_i(U(d))$ since $U(d)$ and $\mathbf{b}$ are related. As more than 1 value of $\mathbf{b}$ can result in the same $U(d)$, more than 1 bit combination can be associated with each path. For example, the transition from node (0,2) to node (0,3) in Figure 4 corresponds to nodes (01,2) and (10,2) in Figure 3, hence the associated bit combinations are 01 and 11. The number of states is reduced from 2^K (exponential) to $K+1$ (linear). Hence, the number of transitions as well as complexity is reduced from $N_c = 2^K$ to $N_f \leq \min(2^K, (K+1)^2)$. The number of transitions is obviously limited by 2^K (complete trellis case), and under the worst scenario, there might be $K+1$ transitions emerging from each of the $K+1$ states, resulting in a total of $(K+1)^2$ transitions. However, in most cases, paths coincide and the actual number of transitions is usually less than the upper bound, for example there are only 3 transitions from chip 2 to chip 3 in Figure 4.

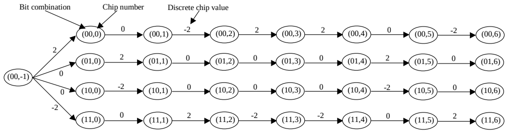

Figure 3: Complete state-transition trellis

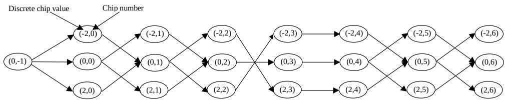

Figure 4: Single-chip folded state-transition trellis

Despite the reduction in the number of states, a one-to-one correspondence still exists between the two trellis and no information is lost. The Euclidean metric pertaining to each transition in the trellis can be calculated in the same way as in the complete trellis case. As a result of the crossings in the folded trellis, a decision is made based on the metric sums, whenever two or more transitions converge at a state (Viterbi algorithm). The path associated with the smallest metric sum is selected and the other paths discarded. The metric values pertaining to the paths emerging from this node add on to this selected metric sum. At the end of the bit interval, the survival path which corresponds to the smallest metric sum can be traced out. As each transition is associated with 1 or more bit combinations, the total number of occurrences of each bit combination is determined for the entire survival path. Hence, as we proceed along the trellis, we add metrics as well as the number of occurrences of the various bit combinations. The bit combination with the highest frequency of occurrence at the end of the interval is selected to be the decoded bits. If 2 or more bit combinations have the same highest frequency of occurrence, then the complete trellis search is performed on this subset.

Mutiple chips can be grouped together and taken as the states too. Consequently, the number of transitions per bit interval is reduced and the number of states is increased. If n_c is the number of chips per state, the states used are all the possible combinations n_c of the $K+1$ discrete values $U(d)\,(d = 0,\ldots,K)$. Let $U_{n_c}(d)\,(d = 0,\ldots,(K+1)^{n_c}-1)$ be the new set of states for the first $g = [G/n_c]$ groups where $[x]$ is the largest integer not greater than x. Each state in these groups has n_c chips, ie group $n_g\,(n_g = 0,\ldots,g-1)$ uses chip $(n_g n_c)$,...,chip $((n_g+1)n_c-1)$. The last group $(n_g = g)$ uses the remaining chips, ie chip (gn_c),...,chip $(G-1)$ and has $(K+1)^{G-gn_c}$ different states. Except for the transitions in the last stage, all other transitions in the previous stages of the trellis comprise of n_c noiseless received chip values, $x_i(\bullet),\ldots,x_{i+n_c-1}(\bullet)\,(i = n_g n_c; n_g = 0,\ldots,g-1)$. The transition values in the last stage are $x_{gn_c}(\bullet),\ldots,x_{G-1}(\bullet)$. The Euclidean metric for each transition path is calculated as

$$M(n_g, U_{n_c}(d)) = \begin{cases} \sum_{j=n_g n_c}^{(n_g+1)n_c-1}(y_j - x_j(U_{n_c}(d)))^2 \\ \qquad\qquad n_g = 0,\ldots,g-1 \\ \sum_{j=n_g n_c}^{G-1}(y_j - x_j(U_{n_c}(d)))^2 \\ \qquad\qquad n_g = g \end{cases} . \quad (6)$$

Using multiple chips per state reduces the number of discarded paths and hence improves the performance since information is lost whenever a decision is made. Obviously, if all the chips are grouped together as 1 state, then the performance would approach that of the MLSE since there would be no decision-making until the end of the bit-interval. The number of transitions is $N_f \le \min(2^K, (K+1)^{2n_c})$. The performance/complexity tradeoff of the proposed scheme can easily be adjusted by varying the number of chips per state.

V. SIMULATION RESULTS

Monte-Carlo simulations were used to obtain the bit error rate (BER). The proposed scheme was simulated for a system with equal-power users using random spreading codes with processing gain, $G = 7$. Figure 5 shows the average BER for $E_b/N_0 = 9$ dB. Other receivers such as the conventional matched filter detector, the MMSE [6], and both the chip- and the bit-based MLSE, were also simulated. The plot verifies that the chip- and bit-based MLSE detectors give the same result. The proposed folded-trellis scheme performs better than the MMSE for both linear and Euclidean metrics. The linear metric scheme only degrades slightly from the Euclidean metric scheme and is even indistinguishable from the optimum for large MAI. Notice that for the single-user case, the proposed scheme has the same performance as the MLSE because the 2 possible paths never crossed.

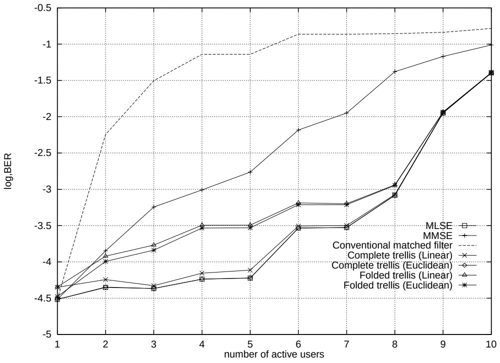

Figure 5: BER against number of active users for $E_b/N_0 = 9$ dB and $G = 7$

If G is now increased to 16 and E_b/N_0 reduced to 7 dB, the resulting performances of the proposed detector using different number of chips per state are plotted in Figure 6. The performance of the single-chip trellis is worse than the MMSE. This is because G has increased by more than twice its original value, and this greatly benefited the MMSE detector. On the other hand, the proposed scheme suffers because a hard decision is made whenever 2 or more paths converge at a state regardless of the length of the trellis which is directly proportional to the processing gain. If a wrong decision is made in the early stages of the trellis, the detrimental effect will propagate down the trellis. A wrong decision is made when the noise value in a chip is so large that it causes the noisy received signal vector to fall in the decision region of an incorrect noiseless received vector. Thus a wrong decision is mainly due to the noise and not the MAI. The erroneous effect is most detrimental for small number of users because the small MAI (greatly reduced due to the high G) is very beneficial to both the conventional receiver and the MMSE but not the proposed scheme. However, for large MAI, the latter degrades less.

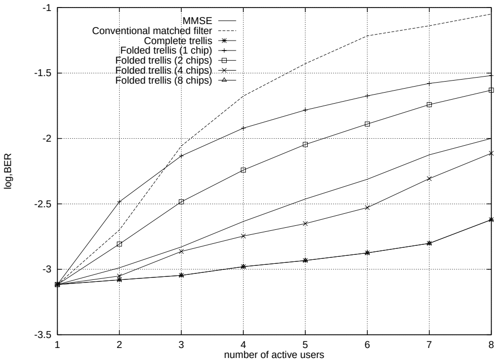

Figure 6: BER against number of active users for $E_b/N_0 = 7$ dB and $G = 16$ using multi-chip folded trellis

It is also observed that the performance improves by increasing the number of chips per state. If the number of chips is 4, the proposed scheme has outperformed the MMSE. If 8 chips per state is considered then it has reached the performance of the MLSE.

Next, we investigate the performance of the receiver using linear and Euclidean metrics. Figure 7 compares the performance of the linear and Euclidean metrics. It is found that as MAI increases, the degradation due to the linear metrics minimises. This is probably because the errors due to MAI are much more significant than that caused by the approximation.

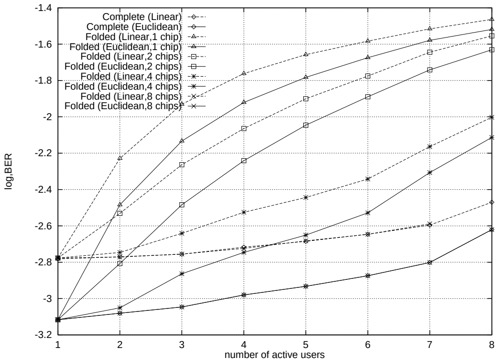

Figure 7: Comparison between linear and Euclidean metrics for $E_b/N_0 = 7$ dB and $G = 16$

VI. CONCLUSIONS

In this paper, a MUD for the downlink of the DS-CDMA systems is proposed. It has been shown that the MLSE can also operate at the chip level using a trellis diagram. We propose to fold up the original trellis and use all the possible transmitted discrete chip values as the states instead of all the possible bit combinations. The number of states is reduced from 2^K to $K + 1$. The receiver can also use the linear metrics for its decision rule instead of the Euclidean metrics without significant degradation in the performance. It has been observed in the simulations that the proposed detector does not gain very much from the processing gain due to the hard decision makings along the trellis. Fortunately, by using more chips per state, performance can be improved and eventually approaches that of the MLSE.

Future work to be carried out involves extending the folded trellis to a multipath fading downlink channel and to apply soft decision-makings used in Turbo codes into this scheme.

REFERENCES

(1) J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

(2) S. Verdu, "Minimum Probability of Error for Asynchtonous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

(3) G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

(4) W. V. Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276–283, February 1976.

(5) Z. Zvonar and D. Brady, "Optimum Detection in Asynchronous Multiple-Access Multipath Rayleigh Fading Channels," in *The Proceedings of the 1992 Conference on Information Science and Systems, Princeton, NJ*, March 1992.

(6) U. Madhow and M. L. Honig, "MMSE Interference Suppression for Direct-sequence Spread-Spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, pp. 3178–3188, December 1994.

A SUB-OPTIMUM MLSE DETECTOR WITH A FOLDED STATE-TRANSITION TRELLIS PRESELECTION STAGE

H K Sim and D G M Cruickshank

The University of Edinburgh, UK

Abstract – The maximum-likelihood sequence estimation (MLSE) multiuser detector has very good performance but at the expense of the exponentially increasing number of states with the number of users. A scheme that folds up the original trellis to reduce the number of states involved has been proposed in [1]. In this paper, the folded trellis detector is combined with the MLSE. The main idea here is to use the folded trellis which has a linear complexity to preselect a subset of the bit-combinations and then apply the MLSE to this much smaller set.

1 INTRODUCTION

In a direct-sequence code-division multiple-access (DS-CDMA) system, users are distinguished by using unique spreading codes. However, the multipath fading effect in the channel causes intersymbol interference. This destroys any orthogonality of the spreading codes and causes multiple access interference (MAI) from the other users. The MAI is not an inherent problem of DS-CDMA, but of the conventional single-user matched filter receiver [2]. To overcome the MAI, multiuser detectors (MUDs) have been suggested where the users are decoded jointly for their mutual benefits.

The maximum likelihood sequence estimator (MLSE) [3], which finds the optimum data bit vector that results in minimum joint probability has exponential complexity growth with the number of users. The various bit-combinations in the MLSE are taken to be the different states in the decoding trellis. A scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved has been proposed in [1]. There is of course a tradeoff between the performance and the complexity. The complexity of this scheme depends on the number of chips used per state. Small number of chips per state is favourable due to its low complexity but at the expense of performance. In this paper, we suggest a better utilisation of the resources, since similar performance to MLSE receiver can be achieved with a lower complexity. The basic idea here is to use the original folded trellis with 1 chip per state to preselect those bit-combinations that are more likely to be transmitted, after which the MLSE criteria is performed on this much smaller subset of bit-combinations. If the transmitted bit-combination is always among the selected bit-combinations, the performance would be that of the MLSE. All the metrics have already been calculated by the folded trellis, the MLSE only has to calculate the sum of the metrics. Furthermore, as the folded trellis has discarded most of the unlikely bit-combinations, the MLSE is only applied to a few bit-combinations, thereby limiting the complexity to a reasonable level.

The next section describes the downlink channel model. Section 3 recaps the MLSE and the folded trellis detector. The following section looks at the new detector which comprises a folded trellis detector and a MLSE detector. Simulation results are presented in Section 6. Finally, in Section 7 some conclusions are drawn.

2 THE DS-CDMA DOWNLINK CHANNEL MODEL

The equivalent discrete-time mathematical model for the downlink of a DS-CDMA system will be discussed in this section. We consider a bit- and chip-synchronous AWGN channel with K users sharing the same bandwidth. Each user in the system has a pre-assigned spreading waveform,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}. \qquad (k = 1, \ldots, K) \qquad (1)$$

where G is the processing gain of the spreading waveform, $c_{i,k} \in \{-1, +1\}$ is the i-th chip value of the k-th user. It is assumed that the received samples are taken at the chip rate. We consider a BPSK modulated system. The input bit sequences for different users are identically independently distributed (i.i.d) and equiprobable. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink.

We assume that each user transmits with constant power. In the synchronous AWGN downlink channel, each bit interval is independent of the other bit intervals without intersymbol interference. Hence, we can simply consider a bit interval. The noiseless received signal in a bit interval can be expressed as

$$X(z) \quad = \quad \sum_{k=1}^{K} S_k(z) \sqrt{P_k} b_k = \sum_{i=0}^{G-1} x_i(\mathbf{b}) z^{-i}, \quad (2)$$

and $(3G-1)(min(2^K,(K+1)^2))$ respectively. Comparing the $K+1$ metric sums at the end of the trellis and selecting the n_m smallest sums requires $(K+1)!/(K-n_m+1)!$ FLOP. Tracing back the survival paths and selecting the n_s most frequently occurring bit-combinations only requires fixed-point operations.

The chip-based MLSE second stage has a relatively low complexity of $(G-1)n_s$ FLOP. Note that the optimum chip-based MLSE requires to sum up the metrics that correspond to all the 2^K bit-combinations. Thus the reduction here is by a factor of $2^K/n_s$ and that the second stage complexity is negligible as compared with the first stage.

Now, we consider the complexity of the optimum bit-based MLSE using eqn (4). The total run-time complexity is $K(2^K+G)$ FLOP, which includes the K matched filters $K(G-1)$ FLOP. The remaining operation is the calculation of the metric sums. If the bit-based MLSE is used as the second stage in the proposed detector, then its complexity would be $K(n_s+G)$ as compared to $(G-1)n_s$ FLOP, even without considering the additional pre-calculations needed.

The complexity is summarised in Table 1. As n_m and n_s are small, we observed that the dominating factors in the proposed detector and the bit-based MLSE are $min(2^K,(K+1)^2)$ and 2^K respectively. Hence, there is significant reduction in complexity.

	Complexity (FLOP per information bit)
Folded trellis first stage:	
metric sums	$\geq (3G-1)(K+1)$
	$\leq (3G-1)(min(2^K,(K+1)^2))$
metric sums selection	$K!/(K-n_m)!$
MLSE second stage:	
Chip-based MLSE	$(G-1)n_s$
Optimum MLSE:	
Bit-based MLSE	$K(2^K+G)$

Table 1: Complexity comparison between the proposed sub-optimum MLSE and the bit-based optimum MLSE detector in a AWGN channel.

6 SIMULATION RESULTS

Monte-Carlo simulations were used to obtain the bit error rate (BER). The proposed scheme was simulated for a system with equal-power users using random spreading codes of $E_b/N_0 = 7$ dB. These codes, which are randomly generated for every trial in the simulation, are considered because they allow for interferer diversity. Figure 3 shows the average BER against the number of active users for $n_m = 3$. Graphs of the sub-optimum MLSE are plotted for $n_s = 1,\ldots,4$. Other MUDs such as the MMSE [5], the optimum MLSE and the 1-chip original folded trellis detector were also simulated. The proposed sub-optimum MLSE scheme performs better than the MMSE and very close to the optimum MLSE for $n_s > 1$. As n_s increases from 1 to 3, the performance improves. Thereafter, no further improvement is observed. Note the great improvement over the 1-chip original folded trellis (which is in fact the proposed sub-optimum MLSE with $n_m = 1$ and $n_s = 1$), when the MLSE second stage is added. When $n_s = 1$, the undesirable performance shows that the most frequently occurring state is not necessary the correct one, especially when metric sums, other than the smallest one are considered. Furthermore, by selecting just 1 bit-combination, we do not exploit the advantage of the second stage MLSE. It is simply the original folded trellis detector taking into consideration other unreliable metric sums. Therefore, we can conclude that increasing n_m without increasing n_s has an adverse effect on the performance and n_s should be kept close to n_m.

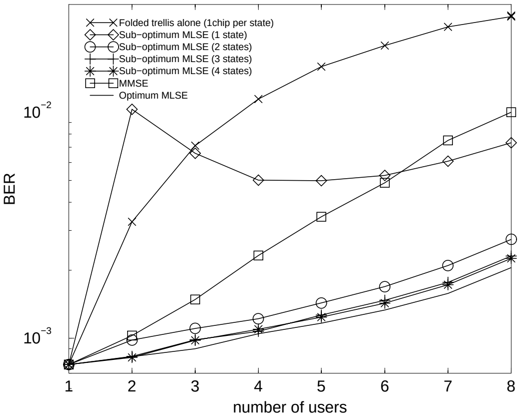

Figure 3: BER against number of active users for $E_b/N_0 = 7$ dB, $G = 16$ and $n_m = 3$.

The conditions used in Figure 4 are the same as in Figure 3 except that here n_s is fixed at 2 and n_m varies from 1 to 4. It is observed that the performance is improved as n_m is increased up to 3. For $n_m = 4$, there is no significant improvement. For this case, setting $n_m = 3$ and $n_s = 2$ is the optimal trade-off with respect to performance and complexity. Even for 8 users, when there are 256 bit-combinations, the sub-optimum MLSE only has to consider 2 bit-combinations to give near optimum performance.

In Figure 5, we investigate the effect of overloading the system. Here the processing gain, G, is set to be 7 and the number of users is increased up to 10 which is greater than G. From the figure, it is clear that with just $n_m = 2$ and $n_S = 2$, the performance of the proposed sub-optimum MLSE has already reached that of the MLSE even under excessive loading. For where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the input bit vector, $b_k \in \{-1, +1\}$ is the input bit symbol of the k-th user, P_k represents the received power of the k-th user and $x_i(\mathbf{b})$ is the i-th noiseless received chip signal in a bit interval as a result of transmitting the data vector $\mathbf{b}$.

The received signal, is $Y(z) = X(z) + N(z)$, where $N(z)$ is the AWGN signal, which has zero mean and two-sided power spectral density equal to $N_o/2$. We have made the assumptions that the receiver knows exactly the received power and the spreading codes of the users, and the channel attenuation is generalised to be real for analysing coherent methods.

3 THE MLSE AND THE FOLDED TRELLIS DETECTORS

The MLSE detector

The MLSE detector is optimum in the sense that it finds the input bit sequence which maximises the conditional probability, or likelihood of the observation sequence. Mathematically, the input bit vector for one interval is decoded as

$$\hat{\mathbf{b}} = \arg \min_{\mathbf{b} \in \{-1,+1\}^K} \sum_{i=0}^{G-1} [y_i - x_i(\mathbf{b})]^2 \qquad (3)$$

$$= \arg \max_{\mathbf{b} \in \{-1,+1\}^K} \left\{ 2 \sum_{k=1}^{K} \sqrt{P_k} b_k \sum_{i=0}^{G-1} y_i c_{i,k} - \sum_{i=0}^{G-1} x_i^2(\mathbf{b}) \right\}. \qquad (4)$$

where y_i is the coefficient of z^{-i} in $Y(z)$ and $\sum_{i=0}^{G-1} y_i c_{i,k}$ is the output of the k-th user's matched filter [3]. However, this implies an exhaustive search over the 2^K possible sequences.

The MLSE operates at the chip level if eqn (3) is used and at the bit level if eqn (4) is used instead. We call the respective detectors the chip-based MLSE and the bit-based MLSE. Both implementations yield exactly the same performance thereotically.

The folded trellis detector

The folded trellis detector operates at the chip level using all the possible unit-power transmitted chip values as the states in the state-transition trellis, instead of using the bit-combinations. This reduces the number of states from 2^K (exponential) to $K+1$ (linear) and folds up the state-transition trellis, resulting in a reduction of complexity. For example, a 2-user case with spreading codes {-1,-1,1,-1,-1,-1,1} and {-1,1,1,-1,-1,1,1}, results in a single-chip folded trellis as shown in Figure 1 [1].

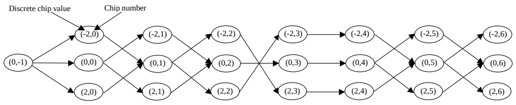

Figure 1: Single-chip folded state-transition trellis

Let

$$U(d_i) = 2d - K \qquad d = 0, \ldots, K \qquad (5)$$

represent the $K+1$ discrete values at the i-th chip and $\mathbf{U} = [U(d_0), \ldots, U(d_{G-1})]^T$ be the vector that represents a path in the trellis. If $\mathbb{U}_1$ is the set that consists of all the 2^K possible paths that correspond to the 2^K bit-combinations, then the minimum metric sum is given as

$$M_{\text{min}} = \min_{\mathbf{U} \in \mathbb{U}_1} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} U(d_i) \right]^2. \qquad (6)$$

The metric sum measures the distance between the received signal vector, $Y(z)$, and one of the 2^K possible noise free signal vectors, $X(z)$. The signal vector $X(z)$ closest (minimum metric sum) to the observation $Y(z)$ has the highest likelihood of being the transmitted signal.

Up to this point, the folded trellis operates exactly as the MLSE except at the chip level. However, as a result of the crossings in the folded trellis, there are more possible paths, not limited to 2^K, as observed in Figure 1. Let $\mathbb{U}_2$ be the set that consists of all the other paths, and $\mathbb{U} = \mathbb{U}_1 \cup \mathbb{U}_2$ be the joint set. If $\mathbb{U}_2$ is included into the metric sums calculation, then we have

$$M_{\text{min}} = \min_{\mathbf{U} \in \mathbb{U}} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} U_i \right]^2. \qquad (7)$$

Initially, it seems that by doing this, the computation increases. Fortunately, the Viterbi algorithm [4] can be used to carry out eqn (7). Consequently, limiting the number of transitions at each chip to a maximum of $min(2^K, (K+1)^2)$ [1]. In the worst scenario, $K+1$ transitions converge at each of the $K+1$ nodes. Thus, the number of transitions as well as complexity is reduced from $N_c = 2^K$ to $N_f \leq min(2^K, (K+1)^2)$.

After the Viterbi algorithm is applied to the trellis, the survival path with the minimum metric sum is selected. As each chip state is associated with 1 or more bit-combinations, the total number of occurrences of each bit-combination is determined for the entire path. The bit-combination with the highest frequency of occurrence at the end of the interval is selected to be the decoded bits. If 2 or more bit-combinations have the same highest frequency of occurrence, then the complete trellis search (MLSE) is performed on this subset.

4 THE SUB-OPTIMUM MLSE

In this section, the sub-optimum MLSE detector is introduced. It consists of a folded trellis first stage, followed by the MLSE with a reduced number of states. The idea here is to use the reliability information inherent in the folded trellis to preselect a number of bit-combinations that are more likely to be transmitted, based on the received signals. If the MLSE is then applied to this reduced subset, the complexity is greatly reduced.

Figure 2 shows the block diagram of the new receiver. The first stage consists of the folded trellis detector and the second stage the chip-based MLSE detector. The preselected n_s bit combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$, are passed to the MLSE which makes used of this information to decode the received signal.

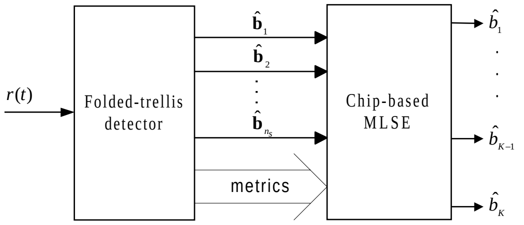

Figure 2: Block diagram of the sub-optimum MLSE detector

Folded trellis first stage

Instead of selecting the smallest metric sum at the end of the trellis and tracing back along the trellis, the new scheme selects the n_m smallest metric sums and then traces back from all the nodes associated with these metric sums. For each trace-back, the frequencies of occurrences of the various bit-combinations are again determined. The frequencies of all these bit-combinations are then compared and ranked in a decreasing order. The first n_s bit-combinations are chosen and passed to the next stage, which is the MLSE. The values of n_m and n_s are preassigned. If trace-back performs on all of the $K+1$ metric sums, complexity would increase, hence n_m is used to limit the number of metric sums considered. The value of n_s affects the complexity of the second stage MLSE and must be kept small. However, it should be greater than 1, otherwise the MLSE becomes redundant.

There are $K+1$ metric sums, $M_1, \ldots, M_{K+1}$, at the end of the trellis corresponding to the $K+1$ chip states. Without loss of generality, we assume that M_1 is the smallest metric sum and the other metric sums are ranked in the order of increasing distance. We then trace back along the trellis starting from the states associated with $M_1, \ldots, M_{n_m}$. Assume that there are N_i bit-combinations associated with the trace-back path of metric sum $M_i \quad (i = 1, \ldots, n_m)$. Out of the $N_1 + \ldots + N_{n_m}$ sets of bit-combinations, the n_s most frequently occurring ones are selected and this information is passed to the MLSE stage.

Chip-based MLSE second stage

In the original folded trellis detector proposed in [1], the MLSE criteria is used to make a decision only when there are 2 or more bit-combinations with the same highest frequency of occurrences. We extend the idea here to include the n_s most frequently occurring bit-combinations from the n_m smallest metric sums.

The inputs of the MLSE are the metrics and the n_s preselected bit-combinations from the previous stage. Let $\mathbb{B}$ be the set that consists of these n_s bit-combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$. Hence, the decoding rule becomes

$$\hat{\mathbf{b}} \quad = \quad \arg\min_{\mathbf{b}\in\mathbb{B}} \sum_{i=0}^{G-1} [y_i - x_i(\mathbf{b})]^2 \,, \qquad (8)$$

instead of eqn (3). The MLSE selects from among the much smaller set of bit-combinations, the one with the smallest Euclidean distance. By limiting the values of n_m and n_s to a reasonable values the complexity is maintained at a practical level.

Furthermore, we recommend that the second stage is implemented with the chip-based MLSE since all the metrics, $[y_i - x_i(\mathbf{b})]^2 \quad (i = 0, \ldots, G-1, \mathbf{b} \in \mathbb{B}$, have already been computed in the previous stage. The chip-based MLSE only has to carry out the summation over the G chips for the n_s bit-combinations.

5 COMPLEXITY

In this section, we will compare the complexity of the proposed sub-optimum detector and the optimum bit-based MLSE detector. We determine the number of floating-point operations (FLOP) required per information bit transmitted.

First, we look at the complexity of the proposed scheme. As the nature of the transitions in the folded trellis depends on the spreading codes of the users, the exact number of transitions per chip cannot be determined for the general case. However, we know that the maximum and minimum numbers of transitions possible are $min(2^K, (K+1)^2)$ and $K+1$ respectively. For the minimum case, there is only 1 transition that converges to each node. For every transition, the associated metric, which is the square of the difference between the observation and the hypothesis, is calculated. Hence, the lower and upper complexity bounds of the metric sums calculation are of the order of $(3G-1)(K+1)$

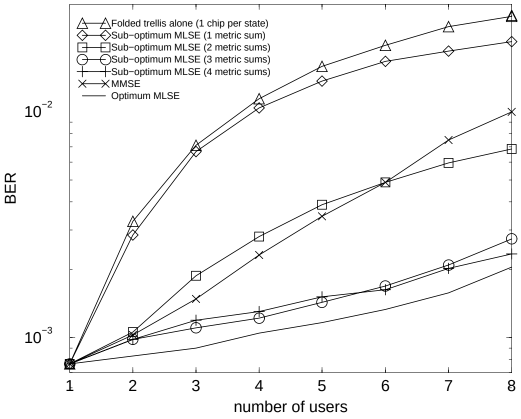

Figure 4: BER against number of active users for $E_b/N_0 = 7$ dB, $G = 16$ and $n_s = 2$.

a BER of 0.05, the proposed detector can have 2.5 users more than the MMSE. Using the complexity table, it is found that for 10 users, the complexity of the MLSE is at least 4 times and at most 32 times greater than the proposed MUD. The graph for $n_m = 1$ and $n_S = 1$, which is equivalent to the original folded trellis without the MLSE second stage, is also plotted. Here we observe the great performance gain over the MMSE especially for high loading, without even exploiting the MLSE second stage.

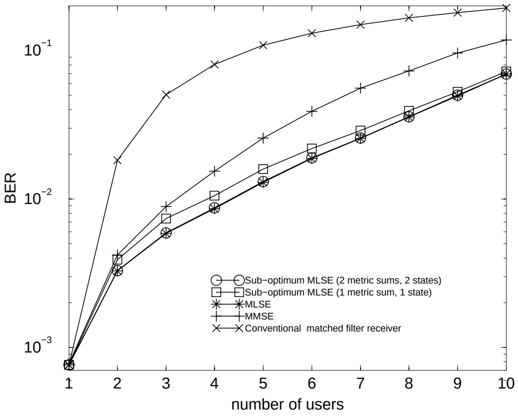

Figure 5: BER against the number of active users for $E_b/N_0 = 7$ dB and $G = 7$.

7 CONCLUSIONS

In this paper, a MUD for the downlink of the DS-CDMA systems is proposed. It has been shown that by using the folded trellis as a pre-selection stage for the MLSE, the performance of the folded trellis can be near the optimum MLSE. The complexity of the MLSE is greatly reduced due to the small subset of bit-combinations selected by the folded trellis. Moreover, the metrics needed by the MLSE has already been calculated by the folded trellis first stage. In short, we have exploited the low complexity of the folded trellis and the high performance of the MLSE, resulting in a better trade-off between complexity and performance.

REFERENCES

[1] H. K. Sim and D. G. M. Cruickshank, "A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis," *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, pp. 846–850, May 1999.

[2] J. G. Proakis, *Digital Communications.* McGraw-Hill, 1995.

[3] S. Verdu, "Minimum Probability of Error for Asynchtonous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

[4] G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[5] U. Madhow and M. L. Honig, "MMSE Interference Suppression for Direct-sequence Spread-Spectrum CDMA," *IEEE Transactions on Communications*, vol. 42, pp. 3178–3188, December 1994.

IEEE 6th Int. Symp. on Spread-Spectrum Tech. & Appli., NJIT, New Jersey, USA, Sept. 6-8, 2000

An Iterative Multiuser Receiver for FEC Coded CDMA

H. K. Sim and D. G. M. Cruickshank
Signal and Systems Group
Department of Electronics and Electrical Engineering
The University of Edinburgh
The King's Buildings, Mayfield Rd
Edinburgh EH9 3JL, UK
PH: (+44) 131 650 5655, FAX: (+44) 131 650 6554
email: hks@ee.ed.ac.uk, dgmc@ee.ed.ac.uk

Abstract — **The powerful iterative algorithm of Turbo code is employed in the proposed iterative multiuser receiver for the downlink of a forward error correction (FEC) coded direct-sequence code-division multiple access (DS-CDMA) environment. The receiver iterates between the 2 coding dimensions, namely the spreading codes and the FEC codes. The optimum iterative multiuser receiver uses the optimum decentralised single-user detector to generate the single-user extrinsic information for the single-user decoders. This paper suggests a sub-optimum detector using a folded trellis preselection stage for the multiuser signal detection, in order to reduce the dominating complexity of the optimum decentralised detector. Simulation results show that performance is close to optimum. The effect of the single-user decoders' extrinsic information on the signal detection stage is also investigated and found that extrinsic information improves performance.**

I. Introduction

Direct-sequence code-division multiple-access (DS-CDMA) is a multiple access technology that has attracted much attention over the past decade. As users' signals overlap in time and frequency in this system, the conventional single-user detection strategy causes multiple access interference (MAI) among the users due to the multipath fading effect and the asynchonisation in the channel [1]. The channel destroys the orthogonal property of the spreading codes which is essential to distinguish the users.

Multiuser detection is a scheme that can counteract the MAI problem, but the high computational complexity of the optimum multiuser detector (which is exponential in the number of active users) has made implementation very difficult. Consequently, a lot of interest on low-complexity sub-optimum multiuser detectors has been raised. Initial work in this area focused on uncoded CDMA systems while more recent work addresses FEC coded CDMA systems.

The optimal multiuser sequence estimator for an asynchronous convolutionally coded DS-CDMA system [2] performs multiuser detection and the convolutional decoding jointly, resulting in a prohibitive computational complexity that is exponential in the product of the number of users and the constraint length of the convolutional code. In [3], a suboptimal scheme that performs symbol detection and decoding separately is proposed. Consequently, the complexity of this partitioned scheme is reduced.

In 1993, a new coding scheme, called Turbo codes [4], was proposed. This has received a lot of attention not only because it achieves near capacity performance close to the Shannon limit, but also because of its powerful iterative algorithm. Recently, the iterative technique has been applied to the multiuser detection of a FEC coded CDMA system [5–7]. This scheme is similar to the partitioned scheme whereby the multiuser detector and the channel decoders are separated, and different in that the multiuser detector and the single-user decoders iterate between one another, each producing soft outputs. The main difference between these works is in the multiuser detection portion since its complexity is the dominant factor in the overall complexity. [5] uses the M-algorithm to find a subset of the possible symbol vectors. The same algorithm is also used in [7], but a pre-whitening filter is added. This technique is similar to the improved decorrelating decision-feedback detector [8] which has a decorrelating or whitening filter before the tree search. However, [7] implements the whitening filter with the Gram-Schmidt orthogonal filter which is suggested in [9]. The work in [6] is somewhat different as the soft outputs of the single-user decoders are used to modify the received signal by removing the estimated MAI. After a MMSE filtering, it is assumed that all MAI is removed and the channel becomes purely Gaussian. The extrinsic information needed by the single-user decoders can then be calculated. The work in this paper is similar to that of [5, 7], except that instead of the M-algorithm, the folded-trellis sub-optimum MLSE [10, 11] is employed.

The next section describes the FEC coded DS-CDMA downlink system model. Section III looks at the iterative multiuser receiver which is a partitioned receiver that iterates between multiuser signal detection and channel decoding. A sub-optimum scheme is suggested to reduce the high computational complexity of the signal detection portion. The following section analyses the complexity of the proposed sub-optimum structure with respect to the optimum architecture. Simulation results are presented in Section V. Finally, in Section VI some conclusions are drawn.

II. The FEC coded DS-CDMA Downlink System Model

This section discusses the model of the downlink of a discrete-time FEC coded DS-CDMA system. We consider a FEC coded bit- and chip-synchronous AWGN channel with K active users. The results here can also be extended to the multipath channel. Figure 1 shows the system model which has K encoders for the K users and an iterative receiver that detects and decodes the noisy received signals. Each user in the system has a pre-assigned spreading waveform,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}, \qquad (k = 1, \dots, K) \tag{1}$$

where G is the processing gain of the spreading waveform, $c_{i,k} = \{-1, +1\}$ is the i-th chip value of the k-th user. It is assumed that the received samples are taken at the chip rate. The binary information data, $d_k(j) \quad (k = 1, \dots, K; j = 1, \dots, J)$, are convolutionally encoded with constraint length v and code rate R. Without loss of generality, we assume $R = 1/N$, for every input data bit, each encoder produces N symbols. The J data bits include $v - 1$ trailing zeros, thus the actual number of transmitted data bits is

$J - v + 1$. The code-bits are BPSK modulated, yielding data symbols $b_k(t) = \{-1, +1\}$ $(k = 1, \ldots, K; t = 1, \ldots, L)$, where $L = JN$ is the number of symbols in one frame. The information data sequences for different users are identically independently distributed (i.i.d) and equiprobable. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink.

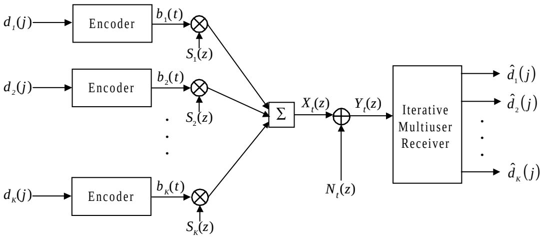

Fig. 1: A FEC coded DS-CDMA system model

We assume that each user transmits with equal and constant power. Each symbol is modulated by a spreading waveform and the noiseless received signal in the t-th symbol interval can be expressed as

$$X_t(z) \quad = \quad \sum_{k=1}^{K} S_k(z)\sqrt{P_k}\, b_k(t) = \sum_{i=0}^{G-1} x_i(\mathbf{b}(t)) z^{-i}, \qquad (2)$$

where $\mathbf{b}(t) = [b_1(t), \ldots, b_K(t)]^T$ is the t-th input symbol vector, P_k represents the received power of the k-th user, and $x_i(\mathbf{b}(t))$ is the i-th noiseless received chip signal in the t-th symbol interval as a result of transmitting the symbol vector $\mathbf{b}(t)$.

The received signal, is $Y_t(z) = X_t(z) + N_t(z)$, where $N_t(z)$ is the AWGN signal, which has zero mean and two-sided power spectral density equals to $N_o/2$. We have made the assumption that the receiver knows exactly the received power and the spreading codes of the users.

III. The Iterative Multiuser Receiver

The iterative multiuser receiver is shown in Figure 2. It comprises of a CDMA Bayesian detector and single-user maximum-a posteriori (MAP) decoders. Due to the fact that only the single-user FEC decoders are used for the decoding instead of joint decoding, the Bayesian detector needs to generate single-user information. This single-user information, $p(Y_t(z)|b_k(t))$, can be viewed as the extrinsic information which is the knowledge about the code-bit based on the structure of the multiuser signal. The single-user information, $\Pr(b_k(t)|\mathbf{Y}_1^L(z)$, provided by the single-user decoder is the extrinsic information about the code-bit based on the constraint of the FEC. The basic idea of the iterative scheme is to produce soft information both across the users and across the trellis in a frame by using a soft-output multiuser detector and soft-input soft-output single-user decoders respectively. The soft information across the two separate dimensions improves with every iteration.

1. Bayesian Detector

In this section, we first discuss the optimum Bayesian detector which produces the soft output for the single-user decoders. Next, a sub-optimal detector is suggested to reduce the prohibitive computational complexity.

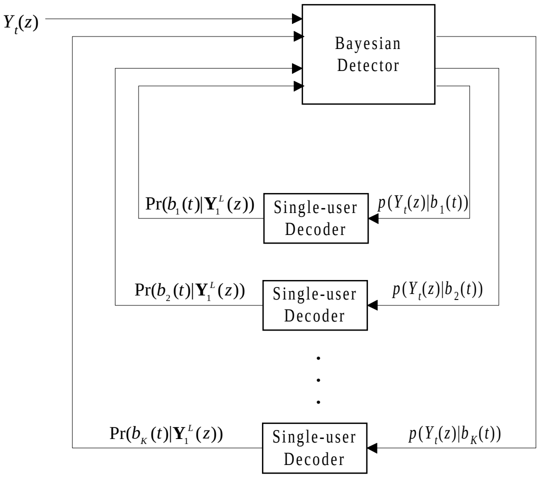

Fig. 2: Iterative multiuser receiver

1.1 Optimum Bayesian Detector

Based on the Bayes theorem, the observation probability density at time t conditioned on the hypothesis that $b_k(t) = b$ is

$$p(Y_t(z)|b_k(t) = b) \quad = \quad \sum_{B_k^b} p(Y_t(z)|\mathbf{b}(t)) \prod_{\substack{i=1 \\ (i \neq k)}}^{K} \Pr(b_i(t)),$$
$$(k = 1, \ldots, K; t = 1, \ldots, L) \qquad (3)$$

where B_k^b is defined as a set that consists of all the symbol vectors, $\mathbf{b}(t)$, with $b_k(t) = b$ [5]. This is the extrinsic information or the extra knowledge based on the structure of the spreading sequences. The prior information, $\Pr(b_i(t))$, on the other hand, is derived from the trellis of the convolutional codes which will be discussed in Section III.

Note that if the symbols are equally likely, then the single-user Bayesian decision rule,

$$\hat{b}_k(t) \quad = \quad \mathrm{sgn}(\Pr(b_k(t) = +1|Y_t(z)) - \Pr(b_k(t) = -1|Y_t(z)))$$
$$= \quad \mathrm{sgn}\left(\sum_{B_k^{+1}} p(Y_t(z)|\mathbf{b}(t)) - \sum_{B_k^{-1}} p(Y_t(z)|\mathbf{b}(t))\right),$$
$$(k = 1, \ldots, K; t = 1, \ldots, L) \qquad (4)$$

is in fact the optimum one-shot detector for decentralised detection of a single-user in an uncoded CDMA system [12] where the signum function returns the sign of its operand. If hard-decisions are made based on the initial (first iteration) output of the Bayesian detector, the result would be the same as the optimum single-user detector for an uncoded system. This is because the initial values of the *a prior* probabilities, $\Pr(b_i(t))$, are 0.5 for all users' symbols. The subsequent values of the *a prior* probabilities are determined by the single-user FEC decoders.

The conditional probability in the summation of eqn (3) is the likelihood of the received sequence $Y_t(z)$ based on the hypothesis that $\mathbf{b}(t)$ is transmitted. As the hypothesis is on the entire t-th symbol vector instead of just $b_k(t)$, we call this the joint likelihood at time instant t and can be computed as

$$p(Y_t(z)|\mathbf{b}(t)) = \frac{1}{(\pi N_0)^{G/2}} \exp\left(-\frac{\sum_{i=0}^{G-1}(y_i(t) - x_i(\mathbf{b}(t)))^2}{N_0}\right). \quad (t = 1, \ldots, L) \qquad (5)$$

where $y_i(t)$ is the coefficient of z^{-i} in $Y_t(z)$. Note that this joint likelihood is considered in the optimum MLSE [13]. The numerator in the exponential term is the Euclidean distance which can be determined by the chip-based optimum MLSE, discussed in [10]. Since $1/(\pi N_0)^{G/2}$ in eqn (5) is common to all $\mathbf{b}(t)$, it can be ignored.

However, the computational complexity of the optimum Bayesian detector is even higher than the optimum MLSE which is of the order of 2^K. The additional computations stem from the summation of 2^{K-1} terms in eqn (3) which has to be carried out twice for the 2 possible bits of $b_k(t)$ and then repeated for the other $K-1$ users. Thus, even though the partitioned detector/decoder reduces complexity substantially, the complexity in the multiuser detector is still the bottleneck.

1.2 Sub-Optimum Bayesian Detector

To reduce complexity, we propose to use the sub-optimum MLSE suggested in [11] to find a subset of the 2^K possible joint likelihoods. The sub-optimum MLSE uses a chip-based folded trellis [10] preselection first stage to choose n_m of the $K+1$ metric sums, and then out of the associated symbol vectors, $\mathbf{b}(t)$, the $n_s \ll 2^K$ ones that are more likely to be transmitted are selected. This piece of information is passed to the chip-based MLSE second stage which computes the Euclidean distances of these n_s preselected symbol vectors. Utilising these calculated Euclidean distances, the joint likelihoods of these symbol vectors are then determined using eqn (5). Define U_k^b as the set that consists of all the symbol vectors, $\mathbf{b}(t)$, in this subset that has $b_k(t) = b$. Hence, the single-user likelihood is approximated as

$$p(Y_t(z)|b_k(t) = b) \approx \sum_{U_k^b} \exp\left(-\frac{M(Y_t(z), \mathbf{b}(t))}{N_0}\right) \prod_{\substack{i=1 \\ (i \neq k)}}^{K} P(b_i(t)), \quad (k = 1, \ldots, K; t = 1, \ldots, L) \qquad (6)$$

where $M(Y_t(z), \mathbf{b}(t))$ is the metric sum generated by the chip-based MLSE second stage in the sub-optimum MLSE. This approximation is accurate if the actual transmitted symbol vector falls within this subset since it has the highest likelihood value if $N_0/2 \to 0$. The folded trellis has high probability of selecting the actual transmitted symbol vectors and other symbol vectors with high likelihood values. Moreover, the likelihood function is exponential, thus the less likely symbol vectors have very low likelihood values and can be ignored. This method is similar to that proposed in [5] but the latter uses the M-algorithm to find the subset.

2. Single-User MAP Decoder

As each user has exactly the same MAP decoder, the following discussion is based solely on the k-th user. The input of the k-th single-user decoder is a sequence of the k-th single-user likelihoods, $p(Y_t(z)|b_k(t) = b)$ $(b = -1, +1; t = 1, \ldots, L)$, in a frame. The output of the decoder is a sequence of *a posteriori* probabilities, $\Pr(b_k(t)|\mathbf{Y}_1^L(z))$, where $\mathbf{Y}_{t'}^t(z) = [Y_{t'}(z), \ldots, Y_t(z)]^T$ represents a vector form of the received sequence from time t' to time t in a frame. The *a posteriori* probability is conditioned on the hypothesis of the joint received signal and not just the received signal of the individual user since we are unable to distinctly separate them.

The single-user decoder calculates the *a posteriori* probabilities using the algorithm proposed in [14]. This soft-output algorithm has been used in the iterative decoding of Turbo codes [15]. Here we derive a similar algorithm to extract the soft-output extrinsic information needed for iteration. Define the k-th user's joint probability at time j from state $S_{j-1}^k = m^-$ to state $S_j^k = m$ as

$$\begin{aligned} \sigma_j^k(m^-, m) &= p(S_{j-1}^k = m^-; S_j^k = m; \mathbf{Y}_1^L(z)) \\ &= (h_\sigma)_j^k \alpha_{j-1}^k(m^-)\gamma_j^k(m^-, m)\beta_j^k(m), \\ &\qquad (j = 1, \ldots, J; k = 1, \ldots, K) \end{aligned} \qquad (7)$$

where $(h_\sigma)_j^k$ is a constant. The forward state probability can be obtained by forward recursion

$$\begin{aligned} \alpha_j^k(m) &= \Pr(S_j^k = m|\mathbf{Y}_1^{jN}(z)) \\ &= (h_\alpha)_j^k \sum_{m^-} \alpha_{j-1}^k(m^-)\gamma_j^k(m^-, m), \\ &\qquad (j = 1, \ldots, J; k = 1, \ldots, K) \end{aligned} \qquad (8)$$

where $(h_\alpha)_j^k$ is such that

$$\sum_m \alpha_j^k(m) = 1. \qquad (j = 1, \ldots, J; k = 1, \ldots, K) \qquad (9)$$

Similarly the reverse state probability at time j for a transition from state m to state m^+ can be obtained by backward recursion

$$\begin{aligned} \beta_j^k(m) &= \Pr(S_j^k = m|\mathbf{Y}_{jN+1}^L(z)) \\ &= (h_\beta)_j^k \sum_{m^+} \beta_{j+1}^k(m^+)\gamma_{j+1}^k(m, m^+), \\ &\qquad (j = 1, \ldots, J-1; k = 1, \ldots, K) \end{aligned} \qquad (10)$$

where $(h_\beta)_j^k$ is such that

$$\sum_m \beta_j^k(m) = 1. \qquad (j = 1, \ldots, J-1; k = 1, \ldots, K) \qquad (11)$$

The boundary conditions are $\alpha_0^k(0) = 1$, $\alpha_0^k(m) = 0$ for $m \neq 0$, $\beta_J^k(0) = 1$ and $\beta_J^k(m) = 0$ for $m \neq 0$. For a valid transition from state m^- to state m, the transition probability is given as

$$\gamma_j^k(m^-, m) = \prod_{t=(j-1)N+1}^{jN} p(Y_t(z)|b_k(t) = b), \quad (j = 1, \ldots, J; k = 1, \ldots, K) \qquad (12)$$

where $[b_k((j-1)N+1), \ldots, b_k(jN)]$ is the codeword associated with the transition from state m^- to state m. The *a priori* information, $p(Y_t(z)|b_k(t) = b)$, needed in eqn (12), is the single-user likelihood information determined by the Bayesian detector.

Let $W_{j,t,k}^b$ be the set of transitions, $S_{j-1}^k = m^- \to S_j^k = m$, such that $b_k(t) = b$ where the value of t lies between $(j-1)N+1$ and jN. Then the joint probability of the state transition is computed as

$$p(b_k(t) = b; \mathbf{Y}_1^L(z)) = \sum_{W_{j,t,k}^b} \sigma_j^k(m^-, m). \quad (j = 1, \ldots, J; k = 1, \ldots, K; t = 1, \ldots, L) \qquad (13)$$

In order to obtain the extra information of the symbol, $b_k(t)$, gleaned from the prior information about the other symbols based on the structure of the spreading codes and independent of the symbol in question, we have to remove this prior information. Hence, following similar method done in [6], the extrinsic information obtained across the convolutional code for symbol $b_k(t)$ is

$$p_{ext}(b_k(t)=b;\mathbf{Y}_1^L(z)) = \frac{\sum_{W_{j,t,k}^b} \sigma_j^k(m^-,m)}{p(Y_t(z)|b_k(t)=b)}. \\ (j=1,\ldots,J;k=1,\ldots,K;t=1,\ldots,L) \qquad (14)$$

The extrinsic *a posteriori* probabilities are therefore

$$\Pr_{ext}(b_k(t)=b|\mathbf{Y}_1^L(z)) = \\ \frac{p_{ext}(b_k(t)=b;\mathbf{Y}_1^L(z))}{p_{ext}(b_k(t)=+1;\mathbf{Y}_1^L(z))+p_{ext}(b_k(t)=-1;\mathbf{Y}_1^L(z))}. \\ (k=1,\ldots,K;t=1,\ldots,L) \qquad (15)$$

Note that the scaling factor, $(h_\sigma)_j^k$, in eqn (7) does not have to be computed since it is cancelled away in the ratio of eqn (15).

By making the following assignment,

$$\Pr(b_k(t)=b) \quad = \quad \Pr(b_k(t)=b|\mathbf{Y}_1^L(z)), \\ (k=1,\ldots,K;t=1,\ldots,L) \qquad (16)$$

the *a posteriori* probabilities generated by the K single-user decoders can be used as the *a priori* information in the Bayesian detector. This is valid since the correlation between the spreading codes and the convolutional codes is small [5].

To decode the signal after the appropriate number of iterations, let $V_{j,k}^d$ be the set of states, S_j^k, such that $d_k(j)=d\in\{0,1\}$, then the decision rule is thus given as

$$\sum_{V_{j,k}^0} \alpha_j^k(m)\beta_j^k(m) \underset{d_k(j)=1}{\overset{d_k(j)=0}{\gtrless}} \sum_{V_{j,k}^1} \alpha_j^k(m)\beta_j^k(m), \\ (j=1,\ldots,J-v+1;k=1,\ldots,K) \qquad (17)$$

since

$$\alpha_j^k(m)\beta_j^k(m) = h_j^k p(S_j^k=m|\mathbf{Y}_1^L(z)), \\ (j=1,\ldots,J-v+1;k=1,\ldots,K) \qquad (18)$$

where h_j^k is a constant that will be cancelled away in eqn (17).

IV. Complexity

Since single-user decoders are used, the complexity of the decoder is $O(2^v)$ which is low as v is usually small. The optimum Bayesian detector has a complexity of $O(2^K)$ which dominates for a system with a large number of users. Hence, we suggest to replace the optimum MLSE in the optimum Bayesian detector with the sub-optimum MLSE. The complexity reduction of the folded trellis sub-optimum MLSE is discussed in [11]. The complexity of the folded trellis, which has $K+1$ states, is in the order of $O(K+1)$ since the transition paths converging into the same state has the same transition metric and recomputation is unnecessary. The complexity of the Euclidean distance ($M(Y_t(z),\mathbf{b}(t))$) computation depends on the number of possible symbol vectors selected (n_s), and is hence of the order $O(n_s)$. The computation of the joint likelihoods by the sub-optimum MLSE does not have to be repeated every iteration. For every iteration, eqn (5) is recomputed, only the *a prior* information, $\Pr(b_k(t)=b)$, changes, the joint likelihoods, $p(Y_t(z)|b_k(t)=b)$, remain the same throughout. The computational complexity is thus $O(n_s)$ per iteration. By keeping n_s to a small value such as $K+1$, the complexity is kept low, in the order of $O(K+1)$.

V. Simulation Results

Monte-Carlo simulations were used to obtain the bit error rate (BER). We consider a system with equal-power users using random spreading codes of processing gain, $G=7$. Random codes are considered in a synchronous environment because the result can be generalised to a asynchronous system and/or a system with multipath effect using orthogonal codes. The same convolution codes with rate, $R=1/2$, constraint length, $v=3$, and generators (7,5) in octal notation, are employed for all users. The size of each frame is 128 data information bits, giving $J=130$. A total of 10,000 frames are transmitted. The $L=260$ sets of encoded symbols of the K users in each frame are transmitted with different sets of randomly generated spreading codes.

In [5], the single-user decoders do not extract the extrinsic information. Instead of using eqn (15), the following equation is equivalent to that used:

$$\Pr(b_k(t)=b|\mathbf{Y}_1^L(z)) = \\ \frac{p(b_k(t)=b;\mathbf{Y}_1^L(z))}{p(b_k(t)=+1;\mathbf{Y}_1^L(z))+p(b_k(t)=-1;\mathbf{Y}_1^L(z))}. \\ (k=1,\ldots,K;t=1,\ldots,L) \qquad (19)$$

By simulation, we found that if extrinsic information is used, the performance is better. The results in terms of BER for 3 iterations are plotted in Figure 3. In the first iteration, the receivers show the same performance since the extrinsic information only affects the subsequent iterations which that the use of extrinsic information improves performance. Note the huge improvement in performance at the second iteration for both cases. As the soft outputs become more and more correlated after every iteration, the improvement drops rapidly. Further iterations show similar trend. The FEC coded single-user bound has been included to show how close to single-user performance the iterative multiuser detector/decoder can achieve.

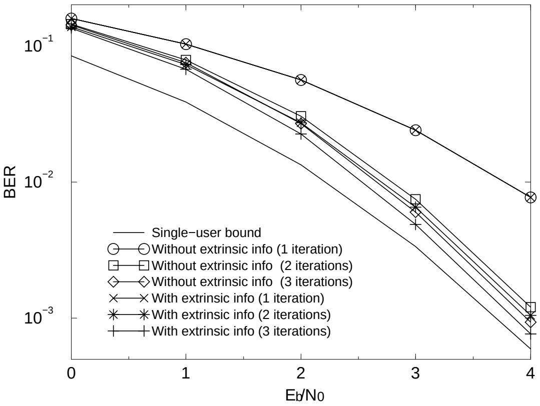

Fig. 3: BER against E_b/N_0 for a system with $K=5$ users using the optimum Bayesian multiuser detector.

Figure 4 shows the average BER against E_b/N_0 for 8 users in the same system. Hence, the system is highly loaded ($K/G>1$). Graphs of the folded trellis sub-optimum partitioned detector/decoder and the optimum partitioned detector/decoder are both plotted up to 3 iterations. Further iterations do not produce significant improvement in performance. Extrinsic information is considered for both receivers since it gives better performance. The folded trellis selects the $n_m = 3$ smallest metric sums and out of the associated symbol vectors, the $n_s = K + 1 = 9$ most frequently occurring ones are chosen. From the figure, it is observed that for each iteration, the performance of the optimum and sub-optimum detectors are very close to each other, except at higher E_b/N_0 where the higher power signals bring the performance of the optimum receiver even closer to the single-user bound. This shows that the reduction in complexity does not have an adverse effect on the approximation made in eqn (6). Complexity has been reduced from $O(2^8)$ to $O(9)$. Comparing the graphs in Figure 3 and Figure 4, we observed that increasing the number of users push the performance curves away from the single-user bound as expected. However, the degradation is not large despite the high loading.

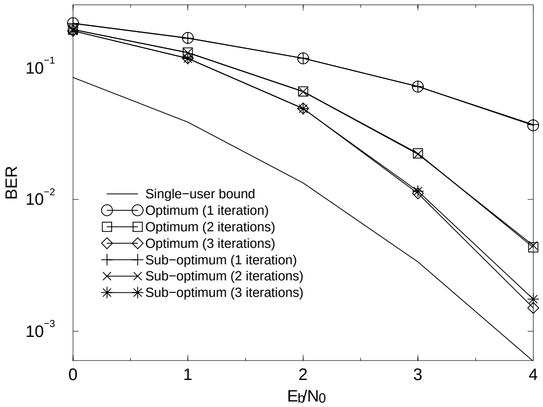

Fig. 4: BER against E_b/N_0 for $K = 8$ users, sub-optimum receiver uses $n_m = 3$ metric sums with $n_s = 9$ symbol vectors.

VI. Conclusions

In this paper, an iterative multiuser receiver based on the iterative algorithm of Turbo codes is proposed. This receiver performs signal detection and decoding separately, but iterates between the two using soft-inputs and soft-outputs. The folded trellis sub-optimum MLSE is suggested to replace the optimum MLSE in the signal detection portion. It is shown that the performance is still close to the optimum receiver despite the large reduction in complexity from $O(2^K)$ to $O(K + 1)$. It has also been shown that if the extrinsic information extracted by the decoders is used in the detector, performance improves.

References

[1] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[2] T. R. Giallorenzi and S. G. Wilson, "Multiuser ML Sequence Estimator for Convolutionally Coded Asynchronous DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 44, pp. 997–1008, August 1996.

[3] T. R. Giallorenzi and S. G. Wilson, "Suboptimum Multiuser Receivers for Convolutionally Coded Asynchronous DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 44, pp. 1183–1196, September 1996.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting coding and decoding: Turbo-codes (1)," in *International Conference on Communication*, pp. 1064–1070, 1993.

[5] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, "Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance," *IEEE Transactions on Communications*, vol. 46, pp. 1693–1699, December 1998.

[6] X. Wang and V. Poor, "Iterative (Turbo) Soft Interference Cancellation and Decoding for Coded CDMA," *IEEE Transactions on Communications*, vol. 47, pp. 1046–1061, July 1999.

[7] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel, "Iterative Multiuser Interference Reduction: Turbo CDMA," *IEEE Transactions on Communications*, vol. 47, pp. 1008–1014, July 1999.

[8] L. Wei and C. Schlegel, "Synchronous DS-SSMA system with improved decorrelating decision-feedback multiuser detection," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 767–772, 1994.

[9] X. H. Chen and H. K. Sim, "Novel synchronous CDMA multiuser detection scheme: orthogonal decision-feedback detection and its performance study," *IEE Proceedings: Communications*, vol. 144, pp. 275–280, August 1997.

[10] H. K. Sim and D. G. M. Cruickshank, "A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis," *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, pp. 846–850, May 1999.

[11] H. K. Sim and D. G. M. Cruickshank, "A Sub-Optimum MLSE Detector with a Folded State-Transition Trellis Preselection Stage," in *3G 2000 Mobile Communication Technologies (IEE Conference Publication 471)*, pp. 271–275, March 2000.

[12] H. V. Poor and S. Verdu, "Single-User Detectors for Multiuser Channels," *IEEE Transactions on Communications*, vol. 36, pp. 50–60, January 1988.

[13] S. Verdu, "Minimum Probability of Error for Asynchtonous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

[14] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.

[15] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes," *Jet Propulsion TDA Progress Report 42-124*, pp. 63–87, February 1996.

A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis

Hak K. Sim and David G. M. Cruickshank

Abstract

The maximum-likelihood sequence estimation (MLSE) multiuser detector has very good performance but at the expense of high computational complexity. The Viterbi algorithm [1] is employed to implement the MLSE in the asynchronous uplink [2]. In this scheme, the various bit-combinations are taken to be the different states in the decoding trellis. However, the number of states increases exponentially with the number of active users. We propose a scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved. Each transmitted chip value can only take a certain number of possible discrete values. Using these discrete values as the states in the transition trellis diagram and operating the Viterbi algorithm at the chip level, the number of states is reduced from 4^K to $(K+1)^L$ where K is the number of users and L is the number of paths in the multipath channel.

Keywords

Multiuser detection, Code division multiple access, Viterbi detection.

I. Introduction

In a direct-sequence code-division multiple-access (DS-CDMA) system, users are distinguished by using spreading codes. For the downlink, the base station transmits synchronously to all the mobile users. In an ideal scenario, each mobile receives the joint signal that is passed through an additive white Gaussian noise (AWGN) channel. With orthogonal codes, each individual user's component in the joint signal is

The authors are from Signals and Systems Group, Department of Electronics and Electrical Engineering, The University of Edinburgh, The King's Buildings, Mayfield Rd, Edinburgh EH9 3JL; PH: (+44) 131 650 5655, FAX: (+44) 131 650 6554; email: hks@ee.ed.ac.uk, dgmc@ee.ed.ac.uk

independent of the other users, and they do not interfere with one another. However, in a practical scenario, there is a multipath fading effect in the channel, resulting in intersymbol interference [3]. This destroys any orthogonality of the spreading codes and causes multiple access interference (MAI) from the other users [4]. The conventional single-user matched filter or RAKE receiver, which decodes and equalises the users independently, treating the MAI as Gaussian noise [5], could not mitigate the MAI. To overcome the MAI, multiuser detectors (MUDs) [6, 7] have been suggested where the users are decoded jointly for their mutual benefits.

The maximum likelihood sequence estimator (MLSE) [2] finds the optimum data bit vector that results in minimum joint probability. In a realistic multipath environment, symbols are interfered with by symbols in adjacent intervals. Hence, additional information about the received signal in the bit interval in question can be obtained by considering other bit intervals as well, and the optimum approach is thus to consider the entire observation interval [8]. It can be shown that the MLSE detector for a multipath synchronous channel is to use a RAKE receiver front-end followed by a forward dynamic programming decision algorithm (the Viterbi algorithm [1, 9]). Although the complexity is no longer dependent on the transmission length due to the Viterbi algorithm, it is still exponential in the number of users.

The various bit-combinations in the MLSE are taken to be the different states in the decoding trellis. We proposed a scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved. Due to the fact that the combined chip values of the spreading codes can only take discrete values, the number of states can be reduced by using these discrete values as the states in the transition trellis diagram and operating the Viterbi algorithm at the chip level. This idea was introduced in [10] for the AWGN channel. This paper extends the basic idea to the multipath environment. Thus, while the number of states can be reduced, this reduction comes at the expense of running the Viterbi algorithm at a much faster rate (chip rate).

Section II of this paper describes the multipath fading channel model of the downlink of a DS-CDMA. This is followed by a discussion of the bit- and chip-based MLSE where it is shown that the outputs of the bank of

RAKE receivers provide a set of sufficient statistics for the bit-based MLSE in a multipath environment. The application of the Viterbi algorithm to the chip-based MLSE is also shown in this section. Section IV gives a brief introduction to the proposed detector in an AWGN environment and then extends it to the multipath scenario. State-transition trellis diagrams are used to explain the decoding procedure at the chip-level. The next section gives a detailed analysis of the complexity of the proposed detector in comparison to the optimum MLSE detector. Simulation results with stationary multipath channels are presented in Section VI. Finally, in Section VII some conclusions are drawn.

II. The DS-CDMA Downlink Channel Model

The equivalent discrete-time mathematical model for the downlink of a DS-CDMA system will be discussed in this section. A bit- and chip-synchronous multipath channel with K users sharing the same bandwidth is shown in Figure 1. Each user in the system has a pre-assigned spreading waveform,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}. \qquad (k = 1, \ldots, K) \tag{1}$$

where G is the processing gain of the spreading waveform, $c_{i,k} = \{-1, +1\}$ is the i-th chip value of the k-th user. It is assumed that the samples are taken at chip rate. Consider a BPSK modulated system that uses only short spreading codes. The assumptions of short codes and BPSK are not limitations of the proposed scheme but to simplify description and analysis. The proposed scheme can even be extended to systems with different users using different processing gains without much difficulty. The input bit sequences for different users are identically independently distributed (i.i.d) and equiprobable.

Consider a multipath channel with L paths and a delay spread greater than one chip period. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink. The normalised channel impulse response is

$$H(z) = \sum_{l=0}^{L-1} h_l z^{-l}, \tag{2}$$

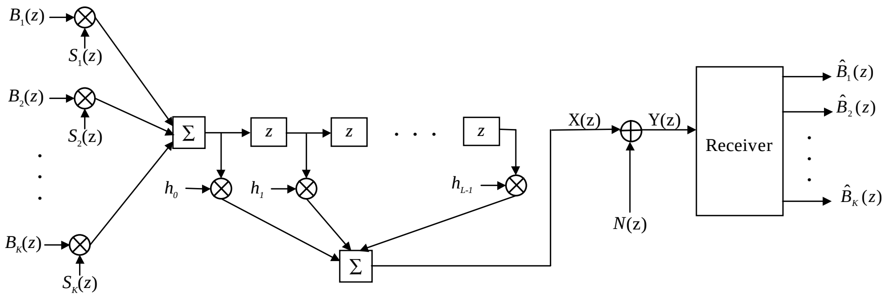

Fig. 1. Synchronous multipath system model

where h_l is the path coefficient of the l-th path. We assume that each user transmits M bits with constant power. The noiseless received signal of the entire data transmission is

$$\begin{aligned} X(z) &= H(z)\sum_{m=0}^{M-1} z^{-Gm}\sum_{k=1}^{K} S_k(z)\sqrt{P_k}b_k(m) \\ &= \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z). \end{aligned} \tag{3}$$

The vector, $\mathbf{R}(z) = [R_1(z), \ldots, R_K(z)]^T$, is the overall channel response due to spreading and multipath effect, where $R_k(z) = H(z)S_k(z)$ is a sequence of length $G + L - 1$. The $K \times K$ diagonal matrix, $\mathbf{P} = diag\{\sqrt{P_1}, \ldots, \sqrt{P_K}\}$, represents the channel attenuations of the users, where P_k is the received power of the k-th user. The vector,

$$\mathbf{B}(z) = \sum_{m=0}^{M-1} \mathbf{b}(m)z^{-Gm} \tag{4}$$

is the entire input bit sequence vector, where $\mathbf{b}(m) = [b_1(m), \ldots, b_K(m)]^T$ is the input bit vector at the m-th interval, with $b_k(m) = \{-1, +1\}$ representing the m-th input bit symbol of the k-th user.

The received signal is thus $Y(z) = X(z) + N(z)$, where $N(z)$ is the AWGN signal, which has zero mean and two-sided power spectral density equal to $N_o/2$. Two assumptions are made here: the receiver knows exactly the received power and the spreading codes of the users, and the channel attenuation is generalised to be real for analysing coherent methods.

III. The Bit-based and Chip-based MLSE Detectors

The MLSE detector is optimum in the sense that it performs a search over all the possible transmitted sequences, $X(z)$, by comparing the likelihood ratios when the transmitted bits of the users are equiprobable and independent [11]. The sequence that is closest to the observation, $Y(z)$, is then selected as the decoded sequence. Mathematically, it is

$$\begin{aligned}\hat{\mathbf{B}}(z) &= \arg \min_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \left\{ \sum_{i=0}^{G(M-1)+L-2} \left[y_i - x_i(\mathbf{B}(z))\right]^2 \right\} \qquad (5)\\ &= \arg \max_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \left\{ \sum_{i=0}^{G(M-1)+L-2} \left[2y_i x_i - x_i^2(\mathbf{B}(z))\right] \right\}, \qquad (6)\end{aligned}$$

where $x_i(.)$ and y_i are the coefficients of z^{-i} in $X(z)$ and $Y(z)$ respectively. Brute-force exhaustive search would have complexity 2^{MK}; however $\hat{\mathbf{B}}(z)$ has the right structure to employ the Viterbi algorithm which results in significantly better complexity. Implementing the MLSE using Viterbi algorithm in the asynchronous uplink, for both the AWGN channel and the multipath channel, has been shown in [2] and [12] respectively. The outputs of the bank of matched filters or the RAKE receivers provide the sufficient statistics to perform the Viterbi algorithm at the bit-level. In the downlink, where the system is synchronous, Viterbi algorithm can also be employed if there is multipath in the channel. Appendix A shows how the bit-based MLSE can be implemented with a RAKE receiver front-end without any degradation in performance. It should be emphasised here that the outputs of the RAKE receivers should not be viewed as a set of pre-processed signals but as a set of sufficient statistics.

The MLSE can also be implemented at chip-level, that is without first matched filtering the signal with the spreading codes of the users. Instead of implementing eqn (6), eqn (5) can be used. At each chip, the squares of the differences between the received chip value and all the possible noiseless chip values are computed. We will call each of these values the Euclidean metric.

Although the noiseless chip values, $x_i(\mathbf{B}(z))$, can be calculated using eqn (3), it not only has to be based on the entire transmitted bit sequences but is also exponential in the message length. We proceed to show that a discrete-time system can be found such that it can be solved efficiently by the forward dynamic programming algorithm (the Viterbi algorithm). Starting from eqn (3),

$$\begin{aligned} X(z) &= \sum_{m=0}^{M} \left[\sum_{i=0}^{G-1} \mathbf{r}_i^T z^{-i} \mathbf{P}\mathbf{b}(m) + \sum_{j=0}^{L-2} \mathbf{r}_{G+j}^T z^{-j} \mathbf{P}\mathbf{b}(m-1) \right] z^{-Gm} \\ &= \sum_{m=0}^{M} z^{-Gm} \sum_{i=0}^{G-1} x_i(\mathbf{b}(m-1), \mathbf{b}(m)) z^{-i}, \end{aligned} \tag{7}$$

where $\mathbf{r}_i = [r_{i,1}, \ldots, r_{i,K}]^T$, and $r_{i,k}$ is the coefficient of z^{-i} in $R_k(z)$. We have also made use of the fact that $\mathbf{b}(m) = \mathbf{0}$ for $m = -1$ and $m = M$.

The inner summation consists of all the noiseless chip values, $x_i(\mathbf{b}(m-1), \mathbf{b}(m))$, in the m-th message bit interval. The i-th chip value of $x_i(.)$ is independent of the previous bits if $i > L-2$. Only the first $L-1$ chips are interfered with by the previous bits. Eqn (7) shows that the noiseless chip values can be calculated just based on the current bits and (if necessary) the previous bits. These chip values repeat every bit interval, so they can be calculated regardless of the message length. Decoding can begin immediately once the signal is received and does not have to be delayed until transmission ends. Hence, the Viterbi algorithm can be readily applied to calculate the Euclidean metrics at each chip. Note that this implementation yields the same result as that implemented at bit-level since we are not making any assumptions or approximations, though it has to run at a higher speed since processing is carried out at chip level.

The state-transition trellis of a 2-user system for the AWGN channel is shown in Figure 2 and was described in [10]. It is a 2^K-state trellis with only “horizontal” transitions allowed, which is equivalent to making memoryless decisions in a 2^K-symbol alphabet. If the channel exhibits multipath properties, the trellis expands between bits whenever intersymbol interference arises, and collapses when the interference from the previous bits dies away. For illustration, consider a 2-user system with a multipath of length $L = 4$. The state-transition trellis is given in Figure 3 which shows the transition between 2 adjacent bit-intervals, the $(m-1)$-th and m-th bit-intervals. The trellis is expanded when transiting from the last chip of the $(m-1)$-th interval to the first chip of the m-th interval. The state of the expanded trellis at the i-th chip is described as

($\mathbf{b}_j(m-1), \mathbf{b}_p(m), i$), where $\mathbf{b}_j(m-1)\,(j = 1, \ldots, 2^K)$ and $\mathbf{b}_p(m)\,(p = 1, \ldots, 2^K)$ are possible bit vectors in the $(m-1)$-th and m-th intervals respectively. Each state in the $(m-1)$-th interval is expanded into 2^K states, resulting in a total of 4^K states and transition paths each. Each transition path is labelled with the value of $x_i(\mathbf{b}_j(m-1), \mathbf{b}_p(m))$. Even though the trellis collapses at the third chip, there are still 4^K transition paths, only that 2^K of them converge into a state. At each terminating state of this transition path, a decision can be made to select the survival path, which is the one with the smallest metric sum. Therefore, there will be a total of 2^K survival paths. The fourth chip ($i = L - 1$) is only dependent on the current bits, hence only 2^K noiseless received chip values or transition paths are possible. The states, ($\mathbf{b}_j(m), i$), and transitions, $x_i(\mathbf{b}_j(m))$, in the collapsed trellis, do not involve the previous bit vector.

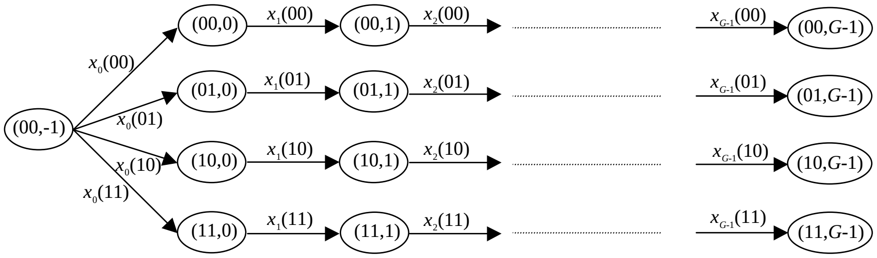

Fig. 2. State-transition trellis for AWGN channel

If any other path to each state replaces the survival path in any overall path, then the path metric would be larger. This is the basis of the Viterbi algorithm. Once the survival paths have been selected, they can be traced back. If all the survival paths originate from a particular state, then the bit vector associated with this state is the decoded bit vector.

IV. The folded state-transition trellis

In this section, we devise a scheme to fold up the state-transition trellis such that the number of states involved is reduced. We assume equal power for all the users ($P = P_k\,(k = 1, \ldots, K)$). The transmitted

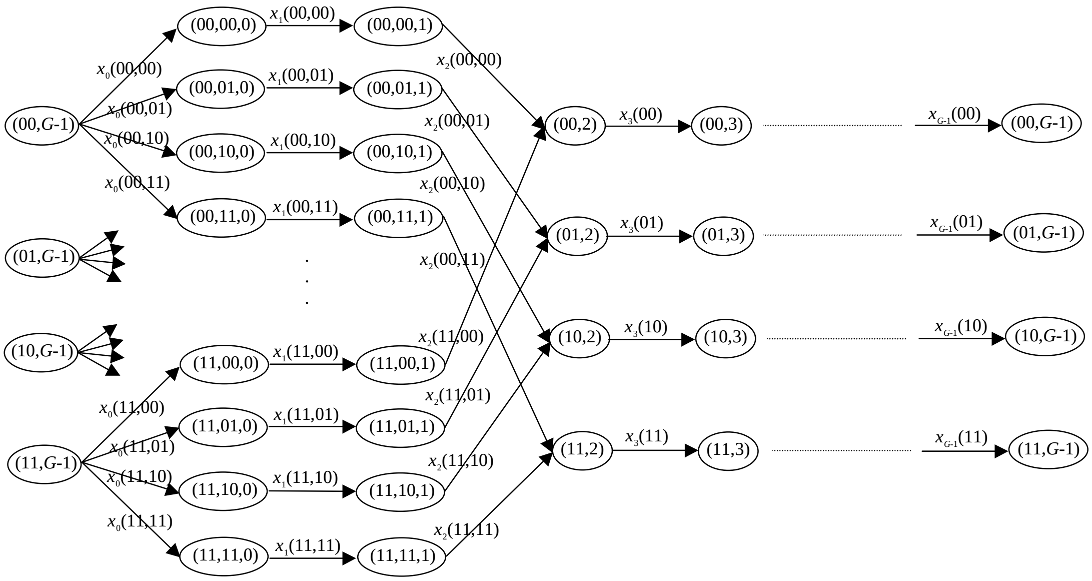

Fig. 3. State-transition trellis for multipath channel

signal is given by

$$V(z) \quad = \quad \sqrt{P}\sum_{m=0}^{M-1} z^{-Gm}\sum_{i=0}^{G-1} v_i(\mathbf{b}(m))z^{-i}, \tag{8}$$

where

$$v_i(\mathbf{b}(m)) \quad = \quad \sum_{k=1}^{K} c_{i,k}b_k(m) \qquad (i = 0, \dots, G-1) \tag{9}$$

takes discrete values from $-K$ to K in steps of 2 since $c_{i,k}$ and $b_k(m)$ only take the two possible values -1 or +1. Total number of discrete values is $K+1$. Let $v_{i,d}\,(d = 0, \dots, K)$ be all the $K+1$ possible values of $v_i(\mathbf{b}(m))$ and if all these values are used as the states in the trellis, the trellis is folded up. The folded trellis is generated based on the transmitted signal given by eqn (8) and not the noiseless received signals even though they are scaled version of each other in an AWGN channel. To illustrate the folding of the trellis, consider a 2-user case with spreading codes $\{$-1,-1,1,-1,-1,-1,1$\}$ and $\{$-1,1,1,-1,-1,1,1$\}$. Hence the resulting complete

AWGN trellis is shown in Figure 4. Using the $K+1=3$ discrete values of -2, 0 and 2 as the states in the transition trellis diagram, the trellis can be folded up as shown in Figure 5.

The transition value pertaining to the transition path entering state $v_{i,d}$ is $x_i(\mathbf{b}) = \sqrt{P}v_{i,d} = x_i(v_{i,d})$ since $v_{i,d}$ and $\mathbf{b}$ are related. As more than one value of $\mathbf{b}$ can result in the same $v_{i,d}$, more than one bit-combination can be associated with each state. The bit-combinations associated with each state are given in the transition paths of the 3-state trellis (see Figure 5). For example, node (0,2) in Figure 5 corresponds to nodes (01,2) and (10,2) in Figure 4, hence the associated bit-combinations are 01 and 10.

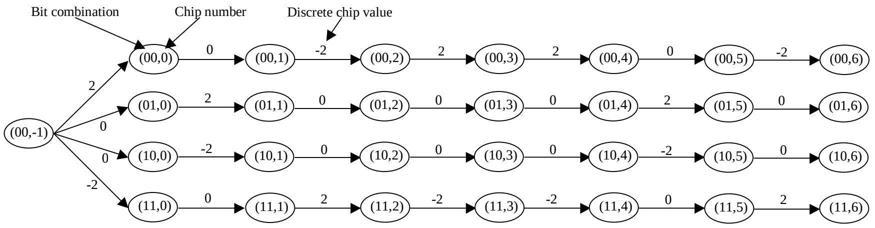

Fig. 4. Complete state-transition trellis

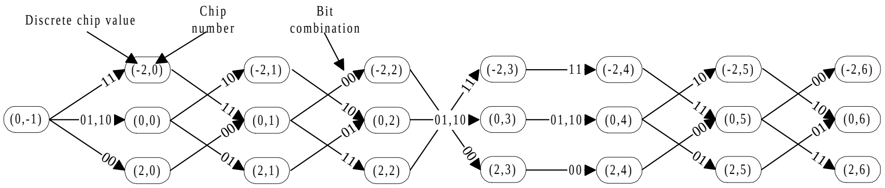

Fig. 5. Single-chip folded state-transition trellis

Despite the reduction in the number of states, a one-to-one correspondence still exists between the two trellis and no information is lost as all the possible chip sequences can still be traced out. Although one-to-one mapping between the combinations of transmitted bits and combinations of transmitted chips exists along the trellis, it is lost when the Viterbi algorithm is applied. The reason is that we do not have enough paths between stages to represent all possible combinations of the transmitted bits. As a result of the crossings in the folded trellis, a decision is made based on the metric sums, whenever two or more transition paths converge at a state (Viterbi algorithm). The path associated with the smallest metric sum is selected and the other paths discarded. The metric values pertaining to the paths emerging from this node add on to this selected metric sum. At the end of the bit interval, the survival path which corresponds to the smallest metric sum can be traced out. For example, if the survival path is {(0,-1),(-2,0),(0,1),(-2,2),(2,3),(2,4),(0,5),(2,6)}, then the bit-combinations associated with the survival path and the respective states are given in Table I. As each state is associated with 1 or more bit-combinations, the total number of occurrences of each bit-combination is determined for the entire survival path. The bit-combination with the highest frequency of occurrences at the end of the interval is selected to be the decoded bits. In the example, bit-combination 00 occurs 5 times while bit-combination 11 has 4 occurrences. Hence, 00 is the decoded bit-combination. If 2 or more bit-combinations have the same highest frequency of occurrences, then the complete trellis search is performed on this subset.

state	(-2,0)	(0,1)	(-2,2)	(2,3)	(2,4)	(0,5)	(2,6)
bit-combination	11	00,11	00	00	00	00,11	11

TABLE I

BIT-COMBINATIONS ASSOCIATED WITH THE STATES OF THE SURVIVAL PATH

The folded trellis is sub-optimum because a decision has to be made whenever two or more paths converge towards a state (Viterbi algorithm) within the folded trellis. For example, at node (0,1) in Figure 5, the metric sums corresponding to path {(0,-1),(-2,0),(0,1)} and path {(0,-1),(2,0),(0,1)} are compared, and the path with the smaller metric sum is selected while the other path is discarded. Path {(0,-1),(-2,0),(0,1)} is selected in the survival path shown in Table I. A wrong decision at this point due to large noise power causes the wrong transition path to be selected and thus the correct bit-combination associated with the correct transition path is lost at this stage. It is obvious that no such decision has to be made for the complete trellis case and thus optimum performance is obtained. A wrong decision at a node does not necessary imply the wrong bit-combination is decoded eventually. Consider the survival path given in Table I, the actual transmitted bit-combination is 00, hence a wrong decision has been made at node (0,1) since the survival path contains node (-2,0) instead of (2,0). However, the right decision is still made eventually.

Multiple chips can be grouped together and taken as the states too. Consider the same 2-user example, Figure 6 illustrates how this can be achieved. The states used are all the possible combinations of n_c of the $K+1$ discrete values $v_{i,d}\,(d=0,\ldots,K)$ if there are n_c chips per state. As $v_{i,d}$ denotes the d-th discrete value of the i-th chip, the noiseless received chip signals or the transition values can be expressed in terms of the discrete values as $x_i(v_{i,d})$ [10]. Using multiple chips per state reduces the number of discarded paths and hence improves the performance since information is lost whenever a decision is made. Obviously, if all the chips are grouped together as one state, then the performance would approach that of the MLSE since there would be no decision-making until the end of the bit-interval. Therefore, by varying the number of chips used per state, the performance lies between that of the MLSE and the single-chip folded trellis.

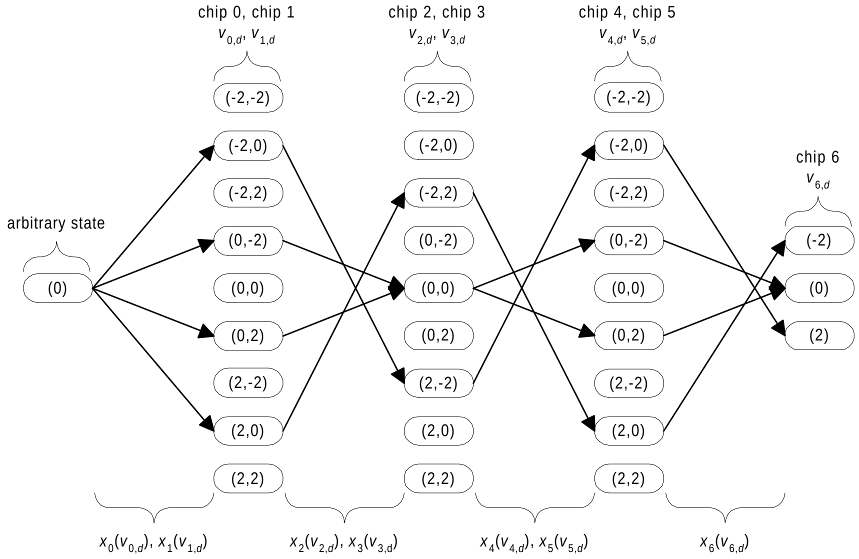

Fig. 6. Folded state-transition trellis for AWGN channel with 2 chips per state

The folded trellis is also applicable to the multipath environment. From eqn (8), the noiseless received signal, $X(z) = H(z)V(z)$, is given as

$$\begin{aligned} X(z) &= \sqrt{P}\sum_{l=0}^{L-1} h_l z^{-l} \sum_{m=0}^{M-1} z^{-Gm} \sum_{i=0}^{G-1} v_i(\mathbf{b}(m))z^{-i} \\ &= \sqrt{P}\sum_{m=0}^{M} z^{-Gm} \sum_{i=0}^{G-1}\sum_{l=0}^{L-1} h_l v_{rem(i-l+G,G)}(\mathbf{b}(m+\lfloor (i-l)/G \rfloor))z^{-i} \\ &= \sum_{m=0}^{M} z^{-Gm} \sum_{i=0}^{G-1} x_i(v_{rem(i-L+1+G,G),d}, \dots, v_{rem(i+G,G),d})z^{-i}, \end{aligned} \tag{10}$$

where $\lfloor x \rfloor$ is the largest integer that is not greater than x and rem(x, y) is the remainder of x divided by y. Again, we have assumed that $\mathbf{b}(m) = \mathbf{0}$ for $m = -1$ and $m = M$. Eqn (10) and eqn (7) are similar except that one expresses $x_i(\boldsymbol{.})$ in terms of the transmitted discrete chip values while the other expresses it in terms of the bit-combinations.

Due to the multipath effect, each received chip signal is affected by L consecutive chips if there are L paths as shown in eqn (10). Hence, the number of states in the trellis is affected by the number of paths in the channel. For example, if $L = 3$, then the number of chips used must be greater than 1. This is because each noiseless received chip signal is dependent on 3 chips but each transition path only involves 2 successive chips if there is only 1 chip per state. Hence, the number of chips per state must be greater or equal to $L - 1$. We have chosen L chips per state in all examples so as not to increase complexity too much. The number of paths is about 5 in a 1 Mchips/s IS-95 downlink environment with a delay spread of 5μs.

Considering the same example, Figure 7 shows the folded trellis in a multipath case, assuming that the received signal comes from 2 different paths, i.e. $L = 2$. In general, the G chips are first divided into $g = \lfloor G/L \rfloor$ groups with L non-repeating chips each. Let $V_{n,s}\,(n = 0, \dots, G-(g-1)(L-1)-1; s = 0, \dots, (K+1)^L - 1)$ be the s-th state of group n. The states in group $n\,(n = 0, \dots, g-1)$ are hence $V_{n,s} = (v_{nL,d}, \dots, v_{(n+1)L-1,d})$. The remaining chips are grouped such that each time there is only 1 new chip, the other $L - 1$ chips come from the previous stage, i.e., the states in group $n\,(n = g, \dots, G-(g-1)(L-1)-1)$ are $V_{n,s} = (v_{\text{rem}((g-1)(L-1)+n,G),d}, \dots, v_{\text{rem}(g(L-1)+n,G),d})$. The trellis extends to the next bit interval and the groups repeat. The noise-free received chip signals or transition values in eqn (10), $x_i(v_{rem(i-L+1+G,G),d},\ldots,v_{rem(i+G,G),d}) = x_i(V_{n,s},V_{n+1,s})\,(i=0,\ldots,G-1)$, are related to the discrete transmitted chip values, and in turn dependent on the states where each transition path originates and/or terminates. Note that if all the related chip values can be found in one state, i.e., $V_{n,s} = (v_{rem(i-L+1+G,G),d},\ldots,v_{rem(i+G,G),d})$, then the transition value is only dependent on one state, $x_i(V_{n,s})$. For the transition paths between the first g groups, there are L noise-free received chip signals associated with each path. These L associated transition values for each transition path emerging from state $V_{n,s}\,(n=0,\ldots,g-2)$ are $x_i(V_{n,s},V_{n+1,s}),\ldots,x_{i+L-1}(V_{n,s},V_{n+1,s})$ where $i=(n+1)L-1$. As for the transition paths between the remaining groups, only one noise-free received chip signal is associated with each path. Hence, for transition paths emerging from state $V_{n,s}\,(n=g-1,\ldots,G-(g-1)(L-1)-1)$, the noise-free received chip value for each path is $x_i(V_{n,s})$ where $i=\text{rem}(g(L-1)+n,G)$. The Euclidean metric for each transition path is calculated as

$$M_m(n,V_{n,s},V_{n+1,s}) = \begin{cases} (y_j - x_j(V_{n,s}))^2 & n=g-1,\ldots,G-(g-1)(L-1)-1; \\ & j=\text{rem}(g(L-1)+n,G) \\ \sum_{j=(n+1)L-1}^{(n+2)L-2}(y_j - x_j(V_{n,s},V_{n+1,s}))^2 & n=0,\ldots,g-2 \end{cases} . \tag{11}$$

Again, each transition path is associated with 1 or more bit-combinations. The occurrences of the bit-combinations determine the final decision on the decoded bit vector as shown in the decoding procedure [10]. The repetition of the chips during the multipath effected transition stages repeats the occurrences of the bit-combinations, and since the metric sums of the paths are more reliable at the end of the interval due to the accumulation of the metrics, decisions made here should be more reliable than those made early in the bit interval and thus put more weight on the correct bit-combination, in a sense biasing the occurrences of the bit-combinations. The right decisions for the last $L-1$ chips of the current interval imply the right decisions for the first $L-1$ chips in the next interval. Another reason for repeating the chips is to reduce the complexity which is discussed in Section V.

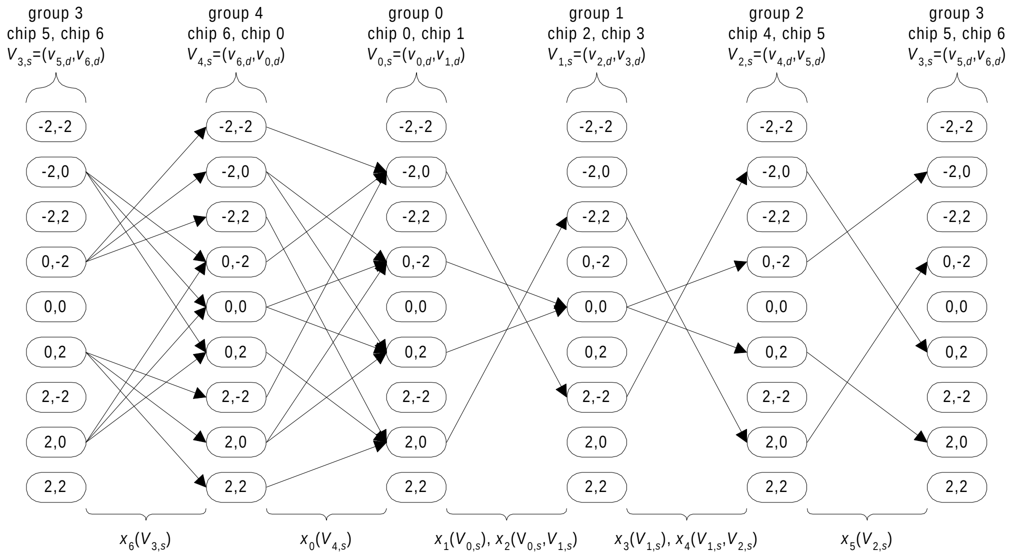
group 3
chip 5, chip 6
$V_{3,s}=(v_{5,d},v_{6,d})$
group 4
chip 6, chip 0
$V_{4,s}=(v_{6,d},v_{0,d})$
group 0
chip 0, chip 1
$V_{0,s}=(v_{0,d},v_{1,d})$
group 1
chip 2, chip 3
$V_{1,s}=(v_{2,d},v_{3,d})$
group 2
chip 4, chip 5
$V_{2,s}=(v_{4,d},v_{5,d})$
group 3
chip 5, chip 6
$V_{3,s}=(v_{5,d},v_{6,d})$
-2,-2
-2,0
-2,2
0,-2
0,0
0,2
2,-2
2,0
2,2
$x_6(V_{3,s})$
$x_0(V_{4,s})$
$x_1(V_{0,s})$, $x_2(V_{0,s},V_{1,s})$
$x_3(V_{1,s})$, $x_4(V_{1,s},V_{2,s})$
$x_5(V_{2,s})$

Fig. 7. Folded state-transition trellis for multipath channel with 2 chips per state

As illustrated, the folded trellis for the multipath channel is still based on the transmitted discrete chip values. Another important point to note here is that the trellis remains exactly the same even if the coefficients of the multipath channel vary. The multipath effect only changes the transition values and not the number of states or transition paths. The transmitted signals are scaled version of the chip values in the state and remain the same regardless of the channel. Each transition path in the trellis represents one and only one transmitted sequence of L chips, even though it may arise from more than one bit-combination. For example, the transition path from $(-2, 0)$ in group 3 to $(0, -2)$ in group 4 corresponds to a transmitted sequence with the fifth and sixth chip values equal to $-2\sqrt{P}$ and 0 respectively. This gives rise to one and only one noiseless received chip signal regardless of the coefficients of the multipath channel. This noiseless multipath-corrupted received chip value formulated by eqn (10), $x_6(V_{3,s})$, is the transition value pertaining to this transition path.

V. Complexity analysis

In Figure 3, the number of states increases by a factor of 2^K at the junction of 2 bit intervals, while in Figure 7, the number of states does not increase, even though the number of transition paths increases. Hence, in a multipath situation, the number of states per chip is 4^K (exponential in K) and $(K+1)^L$ for the complete trellis and the folded trellis respectively. The complexity of the complete trellis chip-based MLSE is hence 4^K as a metric has to be calculated for each transition path out of each state. Similarly, the complexity of the bit-based MLSE is also 4^K as shown in eqn (20) that the transition metric is dependent on the 4^K possible bit-combinations of the previous and current bit intervals[1].

As the structure of the folded trellis depends on the nature of the spreading codes, the exact complexity varies with the code set. However, we can determine the lower and upper complexity bounds. The complexity depends on the number of metric computations. Since each transition path corresponds to a metric, the computational complexity is equivalent to the number of transition paths if the metrics in these paths are different. However, there are many transition paths with the same metric and recomputation is unnecessary. Consider the first multipath transition stage (from group 3 to group 4 in Figure 7 or from group $G-g(L-1)-1$ to group $G-g(L-1)$ in general). The $K+1$ transition values pertaining to the $K+1$ transition paths emerging out of each state are the same since each transition value, $x_6(V_{3,s})$, is only affected by $L=2$ chips, namely chip 5 and chip 6 (state value in group 3), which are the same for all these transition paths. Hence, the resulting Euclidean metrics of these transition paths, $(y_6 - x_6(V_{3,s}))^2$, are also the same. By using serial software implementation of the algorithm, the computation of these transition metrics does not have to be repeated. The computational complexity for this particular stage thus ranges from $K+1$ to $\min(2^K,(K+1)^L)$. The lower complexity bound of $K+1$ occurs if there are only $K+1$ active states (states from which transition paths enter and exit) in group 3. This is possible if the last 2 chip values of the 2 spreading codes are for example $\{$-1,-1$\}$ and $\{$1,1$\}$. This gives rise to only $K+1=3$ possible states, (-2,-2), (0,0) and (2,2). However, in other situations, the number of active states can be as high as 2^K (which is the case shown in Figure 7) or $(K+1)^L$ (which occurs if $2^K > (K+1)^L$ and paths originate from all the $(K+1)^L$ possible states) which gives rise to the complexity of 2^K or $(K+1)^L$ respectively. Thus, we see that the number of states ($(K+1)^L$) serves as a constraint on the complexity when 2^K becomes large.

[1] It is noted that this 4^K complexity could be reduced to 2^K through the introduction of a guard band or by ignoring the interchip interference for modest delay spreads.

Similarly, for transition paths in the next stage (between groups 4 and 0 in Figure 7 or all the transitions stages from group $G - g(L-1)$ to group 0 in general), we only have to perform the metric computation for one transition path emerging from each active state, the remaining paths out of the same state have the same metric. The computational complexity of this stage ranges from $(K+1)^2$ to $\min(4^K, (K+1)^L)$. The lower bound corresponds to the case when there are $(K+1)^2$ active states in group 4 due to $(K+1)^2$ transition paths in the previous stage terminating in different states of group 4 assuming that each of the $K+1$ (minimum) active states in group 3 has $K+1$ transition paths. For this stage, the transition values, $x_0(V_{4,s})$, are affected by one chip in the previous interval (chip 6) and one chip in the current interval (chip 0), hence there is a maximum of 4^K possible transition paths since there are 4^K possible bit-combinations, as we have shown in Figure 3. However, most of the bit-combinations have similar chip 6 and chip 0 values, resulting in a reduction in the number of transition paths and also all paths with similar transition values emerge from the same state. As only one metric needs to be calculated for all transition paths out of each state, the upper complexity bound is thus the number of available states which is again $(K+1)^L$. Hence, if $4^K > (K+1)^L$, the complexity is constrained by the number of states. For example in Figure 7, complexity is $(K+1)^L = 9$ since $4^K > 9$, this is a reduction in complexity compared with the optimum MLSE (4^K). For a value of $K = 6$ and $L = 3$, the optimum MLSE is about 12 times more complex than the upper complexity bound of the folded trellis. For larger values of K, the complexity reduction is even greater. We can view the second term of the upper complexity bound, $\min(4^K, (K+1)^L)$, as the factor that limits the complexity to a feasible level.

The remaining stages in the trellis are not affected by the symbols in the adjacent interval. As discussed earlier the minimum possible number of active states is $K+1$, thus the minimum number of transition paths possible is also $K+1$ with the assumption of one path per state. In this portion of the trellis, the chip values are not repeated in adjacent stage and the transition values are not only dependent on the state values in the group where the transition paths originate but also on the state values in the group where the paths terminate. Hence, even if the transition paths emerge from the same state, they give different metrics and so the computational complexity is equivalent to the number of transition paths. As the chip values are not repeated in adjacent stage, there is no limitation to the transition paths here (Note that due to the fact that there is only 1 new chip value per state in groups 3, 4 and 0, each state can only transit to a maximum of $K+1$ other states in the next group). Hence, each state can transit to a maximum of $(K+1)^L$ states, resulting in a total of $(K+1)^{2L}$ possible transition paths. However, the number of possible transition paths cannot be greater than the number of possible bit-combinations which is only 2^K since it is independent of bits in the adjacent interval. Thus the upper complexity bound here is $\min(2^K, (K+1)^{2L})$. In Figure 4, the number of users is only $K = 2$, therefore the maximum number of transition paths is 4 as shown in the figure. Note that we have repeated the chip values for groups 2 and 3 of Figure 7, as in the first portion of the trellis, the upper complexity bound here becomes $\min(2^K, (K+1)^L)$ which has reduced by a factor of $(K+1)^L$ from the former case. However, the repetition is not carried out for all the chips because the bottleneck of the complexity lies in the first portion of the trellis, since it can be as high as 4^K if it is not suppressed by the folded trellis. When $K = 11$ and $L = 3$, complexity of the multipath layers is $(K+1)^L$ which is then less than 2^K, hence in this situation it is a good idea to use repeated chip values throughout which is analogous to a folded trellis with a single chip per state.

Therefore, the dominating computational complexity lies in the expanded portion of the trellis, the lower and upper complexity bounds of the folded trellis for the multipath channel are thus $(K+1)^2$ and $\min(4^K, (K+1)^L)$ respectively. The complexity comparison between the folded trellis detector and the optimum MLSE detector is plotted in Figure 8. The figure shows that this scheme is best for a system with a small number of multipath components and a large number of users since its complexity grows exponentially with the number of multipath components and not with the number of users under these conditions. The complexity of the folded trellis is never greater than that of the original complete trellis MLSE according to the upper complexity bound. Furthermore, in most cases complexity is near the lower bound. For example, the complexity is usually $(K+1)^2 = 9$ (lower complexity bound) or $12 < \min(4^2, 3^3)$ for $K = 2$ and $L = 3$. Table II gives some codes with $G = 7$ that resulted in these two complexity values. If $L = 5$ and $K = 8$, $(K+1)^L < 4^K$, thus upper complexity bound is $(K+1)^L = 59049$, we expect transition paths emerging from all the possible states if the actual complexity is equal to the maximum complexity. By randomly generating a set of spreading codes, it was found that the folded trellis actually had a complexity of only 1424. Again, we note that the possible complexity reduction comes at the price of having the Viterbi algorithm run at the chip rate rather than the symbol rate. However, it is difficult to quantify this additional cost. In addition, the amount of reduction possible is intimately tied to the realization of the spreading codes.

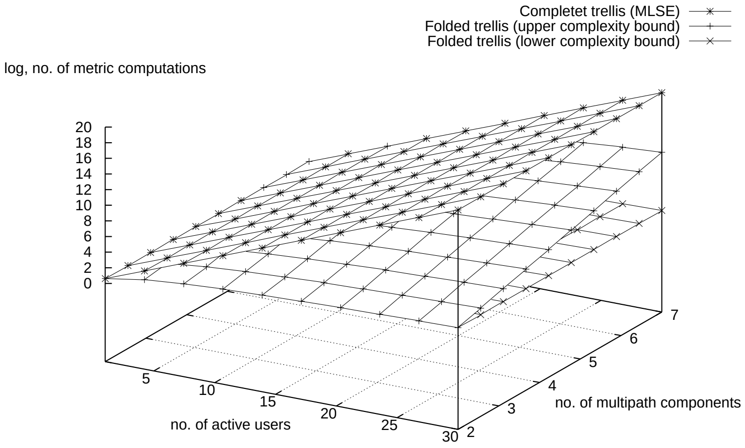

Fig. 8. Complexity comparison between the folded trellis detector and the optimum MLSE detector.

Complexity in multipath effected section (metric value calculations/chip)	Spreading codes
9	{-1 1 -1 -1 1 1 1} & {1 -1 -1 -1 1 -1 -1}
	{1 1 1 1 -1 -1 1} & {-1 -1 1 -1 1 -1 1}
	{-1 1 -1 -1 -1 1 1} & {-1 1 -1 1 1 1 1}
	{1 1 -1 -1 -1 1 -1} & {1 1 -1 -1 -1 1 1}
12	{1 -1 -1 -1 -1 -1 1} & {-1 -1 1 -1 1 1 -1}
	{1 -1 1 1 -1 -1 1} & {-1 -1 1 1 1 -1 -1}
	{-1 1 1 1 1 1 1} & {-1 -1 -1 1 1 1 1}
	{-1 -1 -1 1 -1 1 -1} & {-1 1 1 -1 -1 -1 1}

TABLE II

TABLE OF ACTUAL FOLDED TRELLIS COMPLEXITY FOR SOME PAIRS OF SPREADING CODES WITH $G=7$, $K=2$ AND $L=3$.

VI. Simulation Results

Monte-Carlo simulations were used to obtain the bit error rate (BER). The proposed scheme was simulated for multipath channels with equal-power users using random spreading codes. Random codes are chosen because it models the use of long spreading codes [13], allows for interferer diversity [14] such as different processing gains for different interferers, and the performance of a synchronous system using random codes is on average the same as that of an asynchronous system [15].

First, we consider a system with a processing gain of $G = 16$, E_b/N_0 of 7 dB and a channel impulse response of

$$H(z) = 0.227 + 0.466z^{-1} + 0.688z^{-2} + 0.466z^{-3} + 0.227z^{-4}. \tag{12}$$

The simulation was repeated using 200 frames with a frame size of 100000 bits and each frame uses a different set of random codes. Figure 9 shows the simulation results. Other MUDs such as the RAKE receiver and the minimum mean square estimator (MMSE) detector [16] were also simulated. The folded trellis uses $L = 5$ chips per state. Note that the MMSE detector in the multipath case has a filter length equal to $G + L - 1$ in order to capture all the signal energy that originated from the desired data bit [17]. The complete trellis scheme (MLSE) for the multipath case was also plotted. The folded trellis scheme performs very close to the complete trellis. Note that for the single-user case, the proposed scheme has the same performance as the optimum MLSE because the two possible paths never crossed. Note that the folded trellis and the complete trellis simulations were carried out for a Viterbi truncation length of 2 bit intervals in order to keep the length of the trellis at a feasible value.

Next, we consider a non-symmetric multipath environment with impulse response ($L = 3$)

$$H(z) = 0.802 + 0.535z^{-1} + 0.267z^{-2}. \tag{13}$$

This system uses a lower processing gain of $G = 7$ but with a higher signal-to-noise ratio (SNR) of $E_b/N_0 = 9$

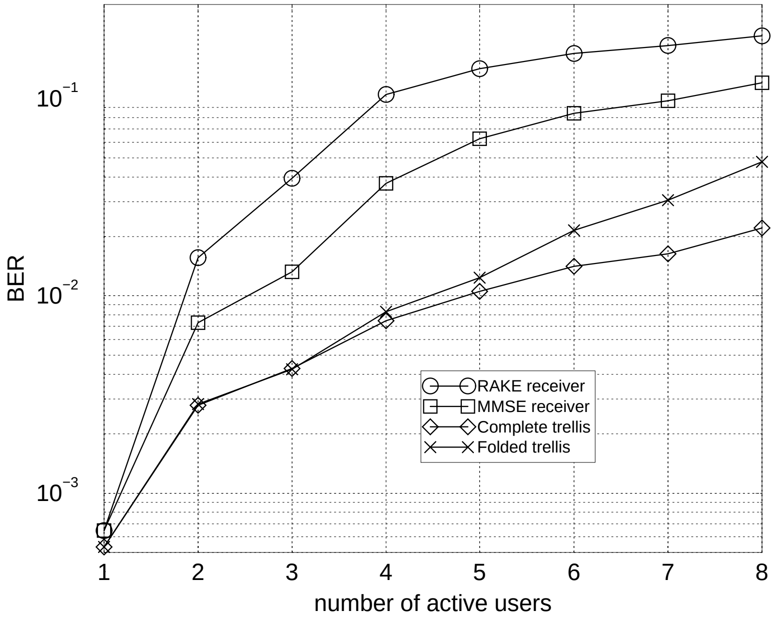

Fig. 9. BER against number of active users for $E_b/N_0 = 7$ dB and $G = 16$

dB. The simulation was repeated using 5000 frames with a frame size of 100 bits. The simulation results are plotted in Figure 10.

The figure shows that by increasing the SNR, the performance of the folded trellis is even closer to the MLSE detector. This is because the errors in the folded trellis are mainly due to noise and not due to MAI. The folding up of the original complete trellis does not remove any transition path from the trellis but merely combines paths that have similar transition values or noiseless received values (even in multipath scenario) which happen frequently due to equal-power transmission. The folded trellis also reduces the number of states and results in merging and divergence of transition paths. Due to the fact that no transition paths have been removed from the original trellis, all the possible hypothesis in each layer (or chip) are tested by the calculation of the square-law metrics, and errors are hence mainly due to noise.

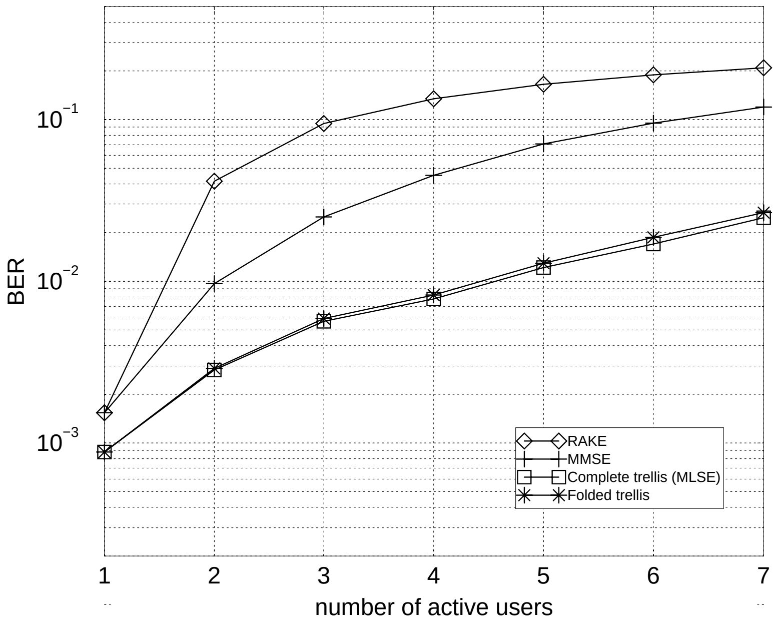

Fig. 10. BER against number of active users for $E_b/N_0 = 9$ dB and $G = 7$

VII. Conclusions

In this paper, a MUD for the downlink of the DS-CDMA systems is proposed. It has been shown that the bit-based MLSE can also be operated at the chip level using a trellis diagram. The proposed detector folds up the original trellis such that the number of states involved is reduced from 4^K to $(K+1)^L$ for the multipath channel and consequently the resultant complexity is $\min(4^K, (K+1)^L)$ versus 4^K for the original algorithm. The folded trellis has been illustrated for the multipath scenario. The new detector uses all the possible transmitted discrete chip values as the states instead of all the possible bit-combinations. Simulation results of the proposed detector were compared with the optimum MLSE. It is found that the performance of the folded trellis detector is close to the MLSE especially for large SNR, since the proposed detector is mainly limited by noise and not MAI.

This paper suggests a technique of reducing the number of states but the decoding procedure which is in terms of the occurrences of the bit-combinations may not be optimum. Hence, future work includes other decoding procedure such as giving different weights to the decisions made at different stages in the trellis.

APPENDIX

I. IMPLEMENTING THE BIT-BASED MLSE WITH VITERBI ALGORITHM

The following proof shows the implementation of the MLSE using the Viterbi algorithm in the downlink of a DS-CDMA system. Eqn (6) can be rewritten as

$$\hat{\mathbf{B}}(z) = \arg \max_{\mathbf{B}(z) \in \{-1,+1\}^{MK}} \left[2\langle Y(z), X(z)\rangle - \langle X(z), X(z)\rangle \right], \tag{14}$$

where

$$\langle Y(z), X(z)\rangle = \sum_i y_i x_i. \tag{15}$$

First, consider

$$\begin{aligned} \langle Y(z), X(z)\rangle &= \langle Y(z), \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z)\rangle \\ &= \sum_{m=0}^{M-1} \langle Y(z), \mathbf{R}^T(z) z^{-Gm}\rangle \mathbf{P}\mathbf{b}(m), \end{aligned} \tag{16}$$

where $\langle Y(z), \mathbf{R}^T(z) z^{-Gm}\rangle$ is actually the output of the RAKE matched filter bank in the m-th message interval. Even though each value of $\langle Y(z), \mathbf{R}^T(z) z^{-Gm}\rangle$ is not a sufficient statistics for the detection of $\mathbf{b}(m)$, the entire sequence of outputs of the RAKE receiver is a sufficient statistics for the selection of the most likely sequence, $\mathbf{B}(z)$.

Next, the second term is expanded as,

$$\begin{aligned}\langle X(z), X(z)\rangle &= \langle \mathbf{B}^T(z)\mathbf{P}\mathbf{R}(z), \mathbf{R}^T(z)\mathbf{P}\mathbf{B}(z)\rangle \\ &= \sum_{m=0}^{M-1}\left[\mathbf{b}^T(m)\mathbf{P}\langle\mathbf{R}(z),\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}(m)\right. \\ &\quad \left.+2\mathbf{b}^T(m-1)\mathbf{P}\langle\mathbf{R}(z),\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}\mathbf{b}(m)\right], \qquad (17)\end{aligned}$$

making use of the fact that $\mathbf{b}(m) = \mathbf{0}$ for $m < 0$ and

$$\langle\mathbf{R}(z)z^{-Gj}, \mathbf{R}^T(z)z^{-Gm}\rangle = \begin{cases}\langle\mathbf{R}(z),\mathbf{R}^T(z)\rangle & \text{if } j = m \\ 0 & \text{if } |j-m| > 1\end{cases}. \qquad (18)$$

The second condition is due to the fact that all the components of $\mathbf{R}(z)$ are of length $G + L - 1 < 2G$, or simply, the bits that are more than 1 bit interval away from each other do not interfere with each other.

Hence, the decision rule can be expressed as a sum of metrics,

$$\hat{\mathbf{B}}(z) = \arg\max_{\mathbf{B}(z)\in\{-1,+1\}^{MK}} \sum_{m=0}^{M-1}\lambda_m(\mathbf{b}(m-1),\mathbf{b}(m)), \qquad (19)$$

with the transition metric given as

$$\begin{aligned}\lambda_m(\mathbf{b}(m-1),\mathbf{b}(m)) &= 2\langle Y(z),\mathbf{R}^T(z)z^{-Gm}\rangle\mathbf{P}\mathbf{b}(m) \\ &\quad -\mathbf{b}^T(m)\mathbf{P}\langle\mathbf{R}(z),\mathbf{R}^T(z)\rangle\mathbf{P}\mathbf{b}(m) \\ &\quad -2\mathbf{b}^T(m-1)\mathbf{P}\langle\mathbf{R}(z),\mathbf{R}^T(z)z^{-G}\rangle\mathbf{P}\mathbf{b}(m). \qquad (20)\end{aligned}$$

Each metric can be computed using the current RAKE receiver output, a subset of control sequences, $\mathbf{b}(m)$, and the set of transmitted sequences, $\mathbf{b}(m-1)$. Therefore, this is a discrete time deterministic control problem with additive cost and finite input and state spaces, so the Viterbi algorithm can be readily applied.

In an AWGN channel, $R_k(z) = S_k(z)\,\forall(k = 1, \ldots, K)$ and the RAKE receiver reduces to the conventional matched filter bank. Since $S_k(z)$ is of length G, there is no interference from the other bit intervals. The last term in eqn (20) becomes 0 since $\langle \mathbf{R}(z), \mathbf{R}^T(z)z^{-G} \rangle = 0$. Hence, the transition metric, $\lambda_m(.)$, does not involve any other bit-interval and a decision can be made for the bit interval in question based on these metrics.

References

[1] G. D. Forney, Jr., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[2] S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. IT-32, pp. 85–96, January 1986.

[3] A. J. Paulraj and C. B. Papadias, "Space-Time Processing for Wireless Communications," *IEEE Signal Processing Magazine*, pp. 49–83, November 1997.

[4] V. M. DaSilva, E. S. Sousa, and V. Jovanovic, "Effect of Multipath Propagation on the Forward Link of a CDMA Cellular System," *Wireless Personal Communications*, pp. 33–41, 1994.

[5] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[6] A. Duel-Hallen, J. Holtzman, and Z. Zvonar, "Multiuser Detection for CDMA Systems," *IEEE Personal Communications*, pp. 46–58, April 1995.

[7] P. Jung and P. D. Alexander, "A Unified Approach to Multiuser Detectors for CDMA and their Geometrical Interpretations," *IEEE Journal on Selected Areas in Communication*, vol. 14, pp. 1595–1601, October 1996.

[8] S. Verdu, "Computational Complexity of Optimum Multiuser Detection," *Algorithmica*, vol. 4, pp. 303–312, 1989.

[9] H.-L. Lou, "Implementing the Viterbi Algorithm," *IEEE Signal Processing Magazine*, pp. 42–52, September 1995.

[10] H. K. Sim and D. G. M. Cruickshank, "A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis," *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, pp. 846–850, May 1999.

[11] W. V. Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276–283, February 1976.

[12] Z. Zvonar and D. Brady, "Optimum Detection in Asynchronous Multiple-Access Multipath Rayleigh Fading Channels," in *The Proceedings of the 1992 Conference on Information Science and Systems, Princeton, NJ*, March 1992.

[13] TIA/EIA IS-95A, *Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular system*. Telecommunication Industry Association, 1993.

[14] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel, "Iterative Multiuser Interference Reduction: Turbo CDMA," *IEEE Transactions on Communications*, vol. 47, pp. 1008–1014, July 1999.

[15] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, "Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance," *IEEE Transactions on Communications*, vol. 46, pp. 1693–1699, December 1998.

[16] U. Madhow and M. L. Honig, “MMSE Interference Suppression for Direct-sequence Spread-Spectrum CDMA,” *IEEE Transactions on Communications*, vol. 42, pp. 3178–3188, December 1994.

[17] D. G. M. Cruickshank, “Radial basis function receivers for DS-CDMA,” *IEE Electronics Letters*, vol. 32, pp. 188–190, February 1996.

A Sub-Optimum MLSE Detector With A Folded State-Transition Trellis Preselection Stage

Hak K. Sim and David G. M. Cruickshank

Department of Electronics and Electrical Engineering
The University of Edinburgh
The King’s Buildings, Mayfield Rd
Edinburgh EH9 3JL
PH: (+44) 131 650 5655, (+44) 131 650 5591
FAX: (+44) 131 650 6554
hks@ee.ed.ac.uk, dgmc@ee.ed.ac.uk

Abstract

The maximum-likelihood sequence estimation (MLSE) multiuser detector has very good performance but at the expense of the exponentially increasing number of states with the number of users. A scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved has been proposed in [1]. In this paper, the folded trellis detector is combined with the MLSE. The main idea here is to use the folded trellis which has a linear complexity to preselect a subset of the bit-combinations and then apply MLSE to this much smaller set.

1 Introduction

In a direct-sequence code-division multiple-access (DS-CDMA) system, users are distinguished by using unique spreading codes. However, the multipath fading effect in the channel causes intersymbol interference. This destroys any orthogonality of the spreading codes and causes multiple access interference (MAI) from the other users. To overcome the MAI, multiuser detectors (MUDs) have been suggested where the users are decoded jointly for their mutual benefits.

The maximum likelihood sequence estimator (MLSE) [2], which finds the optimum data bit vector that results in minimum joint probability, has exponential complexity growth with the number of users. The various bit-combinations in the MLSE are taken to be the different states in the decoding trellis. A scheme that exploits the feature of power-equality in the downlink to reduce the number of states involved has been proposed in [1]. There is of course a tradeoff between the performance and the complexity. The complexity of this scheme depends on the number of chips used per state. Small number of chips per state is favourable due to its low complexity but at the expense of performance. In this paper, we suggest a better utilisation of the resources, since similar performance to MLSE receiver can be achieved with a lower complexity. The basic idea here is to use the original folded trellis which has 1 chip per state to select those bit-combinations that are more likely to be transmitted, after which the MLSE criteria is performed on this much smaller subset of bit-combinations. If the transmitted bit-combination is always among those selected bit-combinations, the performance would be that of the MLSE. As the folded trellis has discarded most of the unlikely bit-combinations, the MLSE is only applied to a few bit-combinations, thereby limiting the complexity to a reasonable level.

The next section describes the downlink channel model. Section 3 recaps the MLSE and the folded trellis. The performance-complexity tradeoff of the folded trellis is explained with the help of a state-transition trellis diagram. The following section looks at the new detector which comprises a folded trellis detector and a MLSE detector. The complexity of the proposed detector is compared with the optimum MLSE in Section 5. Simulation results are presented in Section 6. Finally, in Section 7 some conclusions are drawn.

2 The DS-CDMA Downlink Channel Model

The equivalent discrete-time mathematical model for the downlink of a DS-CDMA system will be discussed in this section. We consider a bit- and chip-synchronous AWGN channel with K users sharing the same bandwidth. Each user in the system has a pre-assigned spreading waveform,

$$S_k(z) = \sum_{i=0}^{G-1} c_{i,k} z^{-i}. \qquad (k = 1, \ldots, K) \tag{1}$$

where G is the processing gain of the spreading waveform, $c_{i,k} \in \{-1, +1\}$ is the i-th chip value of the k-th user. It is assumed that the received samples are taken at the chip rate. We consider a BPSK modulated system. The input bit sequences for different users are identically independently distributed (i.i.d) and equiprobable. From the perspective of each individual mobile, signals from all the users pass through the same channel in the downlink.

Assume that each user transmits with constant power. In the synchronous AWGN downlink channel, each bit interval is independent of the other bit intervals without intersymbol interference. Hence, we can simply consider a bit interval. The assumption of an AWGN channel is for the ease of explanation and is not a limitation of the proposed detector [3]. The noiseless received signal in a bit interval can be expressed as

$$X(z) = \sum_{k=1}^{K} S_k(z)\sqrt{P_k} b_k = \sum_{i=0}^{G-1} x_i(\mathbf{b}) z^{-i}, \tag{2}$$

where $\mathbf{b} = [b_1, \ldots, b_K]^T$ is the input bit vector, $b_k \in \{-1, +1\}$ is the input bit symbol of the k-th user, P_k represents the received power of the k-th user, and $x_i(\mathbf{b})$ is the i-th noiseless received chip signal in a bit interval as a result of transmitting the data vector $\mathbf{b}$.

The received signal, is $Y(z) = X(z) + N(z)$, where $N(z)$ is the AWGN signal, which has zero mean and two-sided power spectral density equal to $N_o/2$. We have made the assumptions that the receiver knows exactly the received power and the spreading codes of the users, and the channel attenuation is generalised to be real for analysing coherent methods.

3 The MLSE and the Folded Trellis Detectors

3.1 The MLSE detector

The MLSE detector is optimum in the sense that it finds the input bit sequence which maximises the conditional probability, or likelihood of the observation sequence. Mathematically, the input bit vector for one interval is decoded as

$$\begin{aligned} \hat{\mathbf{b}} &= \arg \min_{\mathbf{b}\in\{-1,+1\}^K} \sum_{i=0}^{G-1} [y_i - x_i(\mathbf{b})]^2 && (3) \\ &= \arg \max_{\mathbf{b}\in\{-1,+1\}^K} \left\{ 2\sum_{k=1}^{K} \sqrt{P_k} b_k \sum_{i=0}^{G-1} y_i c_{i,k} - \sum_{i=0}^{G-1} x_i^2(\mathbf{b}) \right\}. && (4) \end{aligned}$$

where y_i is the coefficient of z^{-i} in $Y(z)$ and $\sum_{i=0}^{G-1} y_i c_{i,k}$ is the output of the k-th user's matched filter [2]. However, this implies an exhaustive search over the 2^K possible sequences.

The MLSE operates at the chip level if eqn (3) is used and at the bit level if eqn (4) is used instead. We call the respective detectors the chip-based MLSE (CBMLSE) and the bit-based MLSE (BBMLSE). Both implementations yield exactly the same performance thereotically, since no approximation has been made.

3.2 The folded trellis detector

The folded trellis [1] operates at the chip level using all the possible unit-power transmitted chip values as the states in the state-transition trellis, instead of using the bit-combinations. This reduces the number of states from 2^K (exponential) to $K+1$ (linear) and folds up the state-transition trellis, resulting in a reduction in complexity. For example, a 2-user case with spreading codes {-1,-1,1,-1,-1,-1,1} and {-1,1,1,-1,-1,1,1} results in a single-chip folded trellis as shown in Figure 1.

Let

$$U_i(d_i) = 2d_i - K \qquad d_i = 0, \ldots, K \tag{5}$$

represents the $K+1$ discrete values at the i-th chip and $\mathbf{U} = [U_0(d_0), \ldots, U_{G-1}(d_{G-1})]^T$ be the vector

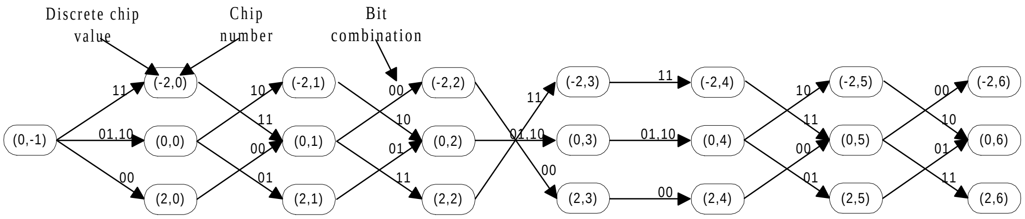

Figure. 1. *Single-chip folded state-transition trellis*

that represents a path in the trellis. If $\mathbb{U}_1$ is the set that consists of all the 2^K possible paths that correspond to the 2^K bit-combinations, then the minimum metric sum is given as

$$M_{\text{min}} \;=\; \min_{\mathbf{U}\in\mathbb{U}_1} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} U_i(d_i) \right]^2 . \tag{6}$$

The metric sum measures the Euclidean distance between the received signal vector, $Y(z)$, and one of the 2^K possible noise-free received signal vectors, $X(z)$. The signal vector, $X(z)$ or $\mathbf{U}$, closest (minimum metric sum) to the observation $Y(z)$ has the highest likelihood of being the transmitted signal.

Up to this point, the folded trellis operates exactly as the CBMLSE. However, as a result of the crossings in the folded trellis, there are more possible paths and not limited to 2^K, as observed in Figure 1. Let $\mathbb{U}_2$ be the set that consists of all the other paths, and $\mathbb{U} = \mathbb{U}_1 \cup \mathbb{U}_2$ be the joint set. If $\mathbb{U}_2$ is included into the metric sums calculation, then we have

$$M_{\text{min}} \;=\; \min_{\mathbf{U}\in\mathbb{U}} \sum_{i=0}^{G-1} \left[y_i - \sqrt{P} U_i(d_i) \right]^2 . \tag{7}$$

Initially, it seems that by doing this, the computation increases. Fortunately, the Viterbi algorithm [4] can be used to carry out eqn (7). Consequently, this reduces complexity from $O(2^K)$ to $O(K+1)$ (to be discussed in Section 5).

After the Viterbi algorithm is applied to the trellis, the survival path with the minimum metric sum is selected. As each transition state is associated with 1 or more bit-combinations (see Figure 1), the total number of occurrences of each bit-combination is determined for the entire path. The bit-combination with the highest frequency of occurrence at the end of the interval is selected to be the decoded bits. If 2 or more bit-combinations have the same highest frequency of occurrence, then the complete trellis search (MLSE) is performed on this subset.

Multiple chips can be grouped together and taken as the states too. Consider the same 2-user example, Figure 2 illustrates how this can be achieved. The states used are all the possible combinations of n_c of the $K+1$ discrete values $U_i(d_i)$ if there are n_c chips per state. The noiseless received chip signals or the transition values can be expressed in terms of the discrete values as $x_i(U_i(d_i)) = \sqrt{P}U_i(d_i)$. Using multiple chips per state reduces the number of discarded paths, and hence improves the performance since information is lost whenever a decision is made. Obviously, if all the chips are grouped together as 1 state, then the performance would approach that of the MLSE since there would be no decision-making until the end of the bit-interval. Therefore, by varying the number of chips used per state, the performance lies between that of the MLSE and the single-chip folded trellis. However, the increase in complexity from $K+1$ (single-chip) to $\min(2^K, (K+1)^{n_c})$ (multi-chip) is exponential.

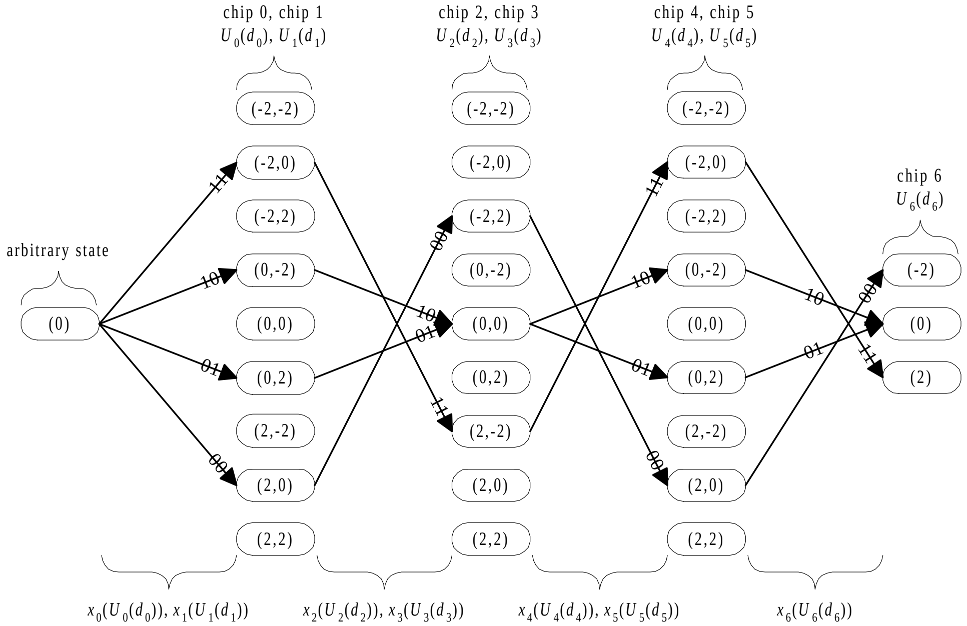

Figure. 2. *Folded state-transition trellis with 2 chips per state*

4 The Sub-optimum MLSE

Figure 3 shows the block diagram of the proposed sub-optimum MLSE detector. It consists of a folded trellis first stage which uses the reliability information inherent in the trellis, followed by the CBMLSE second stage with reduced number of states. The preselected n_s bit-combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$, are passed to the MLSE which makes used of this information to decode the received signal.

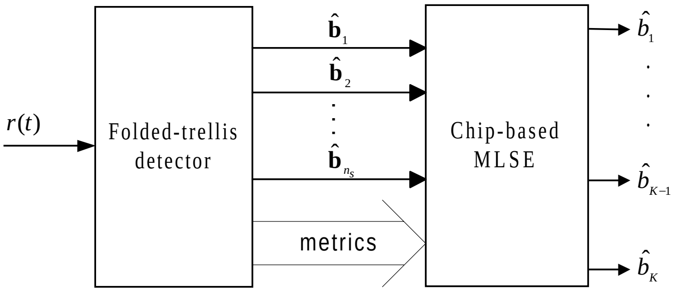

Figure. 3. *Block diagram of the sub-optimum MLSE detector*

4.1 Reliability information

We denote the probability density of the observation $Y(z)$ under the hypothesis that $X_l(z)$ is the noise-free received signal as $p_l(Y(z))$. Assuming that each noise free received signal, $X_l(z), l = 1, \ldots, 2^K$, is independent and equally likely to be received, the likelihood function is

$$p_l(Y(z)) = \frac{1}{(\pi N_0)^{G/2}} \exp\left(-\frac{\sum_{i=0}^{G-1}(y_i - x_{l,i})^2}{N_0}\right). \qquad (l = 1, \ldots, 2^K) \tag{8}$$

Let

$$M_l = \sum_{i=0}^{G-1}(y_i - x_{l,i})^2 \qquad (l = 1, \ldots, 2^K) \tag{9}$$

represents the 2^K metric sums obtained at the end of the complete trellis. The probability of making a right decision if the MLSE selects $X_j(z)$ is thus

$$P_j(c) = \frac{\exp(-M_j/N_0)}{\sum_{l=1}^{2^K}\exp(-M_l/N_0)}. \qquad (j = 1, \ldots, 2^K) \tag{10}$$

Therefore, the probability of making a wrong choice if the MLSE selects $X_j(z)$ is

$$\begin{aligned} P_j(e) &= 1 - P_j(c) \\ &= 1 - \frac{1}{1 + \sum_{l=1, l\neq j}^{2^K}\exp(-\Delta_{l,j})}. \qquad (j = 1, \ldots, 2^K) \end{aligned} \tag{11}$$

where $\Delta_{l,j} = (M_l - M_j)/N_0$ denotes the scaled difference between the metric sums. The probability $P_j(e)$ can be used as the reliability information for selecting $X_j(z)$. Since $\exp(-\Delta_{l,j})$ decreases exponentially, the sum of the exponentials, $\sum_{l=1,l\neq j}^{2^K}\exp(-\Delta_{l,j})$, is dominated by the smallest $\Delta_{l,j}$. Hence, $\sum_{l=1,l\neq j}^{2^K}\exp(-\Delta_{l,j}) \approx \exp(-\Delta_j)$ where Δ_j is the smallest difference. We can simply use

$$P_j(e) \approx 1 - \frac{1}{1+\exp(-\Delta_j)} \tag{12}$$

as a factor to determine how reliable the selected metric sum is. If $\Delta_j \to 0$, then $P_j(e) \to 0.5$, since a small difference in the metric sums is more likely to cause an error in the selection of the metric sum. On the other hand, if $\Delta_j \to \infty$, then $P_j(e) \to 0$, thus minimising the errors in the selection of the metric sum. Interpreting it in the signal space, as the other metric sums get further away from the minimum metric sum, the less likely it is that the minimum metric sum is incorrect.

We can define a sphere with radius r and centre at the signal vector, V_0, which gives the minimum metric sum. The degree of reliability of V_0 is dependent on the number of other signal vectors that falls within this sphere. If there are no other vectors within this sphere, then V_0 is very reliable. On the other hand, if there are other vectors within this circle, then V_0 is unreliable. The parameter r is thus a threshold determining whether to consider other signal vectors or not.

4.2 Folded trellis first stage

Instead of selecting the smallest metric sum at the end of the trellis and tracing back along the trellis, the new scheme selects the n_m smallest metric sums and then trace back from all the nodes associated with these metric sums. For each trace-back, the frequencies of occurrences of the various bit-combinations are again determined. The frequencies of all these bit-combinations are then compared and ranked in a decreasing order. The first n_s unique bit-combinations are chosen and passed to the CBMLSE.

The following describes the method of determining n_m. First preassign a value to the radius r and the maximum number of metric sums to be n_M. Let N_m be the number of metric sums within the sphere of radius r including M_j. The value of n_m is defined as

$$n_m = \begin{cases} N_m & 1 \le N_m < n_M \\ n_M & N_m \ge n_M \end{cases} . \tag{13}$$

If $\Delta_{l,j} > r$ for all l, provided M_j is smallest, then there are no other metric sums within the sphere ($N_m = 1$) and M_j is reliable. If $N_m < n_M$, then $n_m = N_m$ and all the vectors within the sphere of radius

r are checked by the CBMLSE. However, if $N_M > n_M$, then n_m is restricted to n_M and only the $n_M - 1$ metric sums closest to M_j are investigated along with M_j. By setting a small value for r, most of the other metric sums will be outside the sphere, hence complexity is low but performance is not good since most metric sums are not considered. On the other hand, by setting a large value for r, most metric sums fall within the sphere. If trace-back performs on all of these, complexity would increase, hence n_M is used to limit the number of metric sums considered. However, setting n_M to a small value and $r = \infty$, does not always result in the best performance (shown in Section 6).

There are $K + 1$ metric sums, $M_0, \ldots, M_K$, at the end of the trellis corresponding to the $K + 1$ chip states. Without loss of generality, assume that M_0 is the smallest metric sum and the other metric sums are ranked in the order of increasing distance. Figure 4 shows the vector representation of all the metric sums. $M_1, \ldots, M_4$ are the other metric sums within the sphere of radius r centred at M_0. Metric sums, $M_5, \ldots, M_7$, fall outside the sphere. Hence, $N_m = 5$ and if $n_M = 4$, then $n_m = n_M = 4$ and we trace back along the trellis starting from the states associated with $M_0, \ldots, M_{n_m - 1}$. Assume that there are N_i bit-combinations associated with the trace-back path of metric sum M_i $(i = 0, \ldots, n_m - 1)$. Out of the $N_0 + \ldots + N_{n_m - 1}$ bit-combinations, the n_s unique and most frequently occurring ones are selected and passed to the CBMLSE stage. The value of n_s affects the complexity of the CBMLSE stage and must be kept small. However, it sho

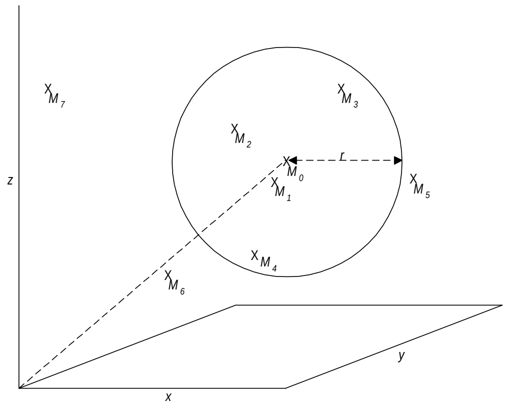

Figure. 4. *A 3-dimensional signal vector space*

4.3 CBMLSE second stage

The MLSE criteria is applied to the n_s most frequently occurring bit-combinations from the n_m smallest metric sums. The inputs of the CBMLSE are the metrics and the n_s preselected bit-combinations from the previous stage. Let $\mathbb{B}$ be the set that consists of these n_s bit-combinations, $\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{n_s}$. Hence, the decoding rule becomes

$$\hat{\mathbf{b}} \quad = \quad \arg\min_{\mathbf{b}\in\mathbb{B}} \sum_{i=0}^{G-1} \left[y_i - x_i(\mathbf{b})\right]^2 , \tag{14}$$

instead of eqn (3). The CBMLSE selects from among the much smaller set of bit-combinations, the one with the smallest Euclidean distance. By limiting the values of n_m and n_s to reasonable values the complexity is maintained at a practical level.

Furthermore, we recommend that the second stage is implemented with the CBMLSE instead of the BBMLSE, since all the metrics, $[y_i - x_i(\mathbf{b})]^2$ $(i = 0, \ldots, G-1, \mathbf{b} \in \mathbb{B})$, have already been computed in the previous stage.

5 Complexity

In this section, the complexities of the proposed detector and the optimum BBMLSE detector are compared. We determine the number of floating-point operations (FLOP) required per information bit transmitted.

As the nature of the transitions in the folded trellis depends on the spreading codes of the users, the exact structure of the folded trellis also differs from codes to codes. However, the computational complexity can still be determined without the knowledge of the actual transitions in each stage. For each transition path, the Euclidean metric, $(y_i - \sqrt{P}U_i(d_i))^2$ $(i = 0, \ldots, G-1)$, is dependent on $U_i(d_i)$ which is the value of the state where the path terminates. Hence, it is clear that this metric is the same regardless of the state where the transition path emerges. Therefore, all transitions entering a particular state have the same Euclidean metric and recomputation is unnecessary. In other words, only one metric computation is required for each state.

For every metric, the square of the difference between the observation and the hypothesis is calculated. Hence, the complexity of the metric sums calculation is $(3G-1)(K+1)$. Comparing the $K+1$ metric sums at the end of the trellis and selecting the n_m smallest sums requires $K + \ldots + (K - n_m + 1) = n_m(2K - n_m + 1)/2$ FLOP. Tracing back the survival paths and selecting the n_s most frequently occurring bit-combinations only requires fixed-point operations.

The CBMLSE only has to carry out the summation over the G chips for the n_s bit-combinations, since all the metrics have already been computed in the previous stage. This results in a complexity of $(G-1)n_s$ FLOP. The selection of the minimum metric sum out of the n_s metric sums requires $n_s - 1$ FLOP. Thus, the second stage complexity is small as compared with the first stage. Note that the optimum CBMLSE would require to sum up the metrics that correspond to all the 2^K bit-combinations. Thus the reduction here is by a factor of $2^K/n_s$.

The total run-time complexity of the optimum BBMLSE is $K(2^K + G)$ FLOP, which includes the K matched filters' $K(G-1)$ FLOP. The remaining operations are the calculation of the metric sums. An additional $2^K - 1$ FLOP are needed for the selection of the minimum metric sum. If the BBMLSE is used as the second stage in the proposed detector, then the additional complexity would be $(K+1)n_s + G(K-n_s)$, even without considering the additional pre-calculations needed.

The complexity is summarised in Table 1. As n_m and n_s are small, the dominating factors in the proposed detector and the BBMLSE are $3GK$ and $K2^K$ respectively. Hence, there is significant reduction in complexity.

	Complexity (FLOP per information bit)
Folded trellis first stage:	
metric sums computations metric sums selection	$(3G-1)(K+1)$ $n_m(2K - n_m + 1)/2$
CBMLSE second stage:	
metric sums computations metric sum selection	$(G-1)n_s$ $n_s - 1$
Optimum BBMLSE:	
metric sums computations metric sum selection	$K(2^K + G)$ $2^K - 1$

Table. 1. *Complexity comparison between the proposed detector and the BBMLSE detector in a AWGN channel.*

6 Simulation Results

Monte-Carlo simulations were used to obtain the bit error rate (BER). The proposed scheme was simulated for a system with equal-power users using random spreading codes and $E_b/N_0 = 7$ dB. These codes, which are randomly generated for every trial in the simulation, are considered because it models the use of long spreading codes [5] or when properties of the sequences have been corrupted due to inter-chip interference arising from the presence of a multipath channel. Furthermore, the performance of a synchronous system using random codes is on average the same as that of an asynchronous system [6] and it allows for interferer diversity [7] such as different processing gains. Figure 5 shows the average BER against the number of active users for $G = 16$, $r = \infty$ and $n_M = 3$. Graphs of the sub-optimum MLSE are plotted for $n_s = 1, \ldots, 4$. Other MUDs such as the MMSE [8], the optimum MLSE, 1-chip original folded trellis were also simulated. The sub-optimum MLSE scheme performs better than the MMSE and very close to the optimum MLSE for $n_s > 1$. As n_s increases from 1 to 3, the performance improves. Thereafter, no further improvement is observed. Note the great improvement over the 1-chip original folded trellis (which is in fact the sub-optimum MLSE with $n_M = 1$ and $n_s = 1$), when the MLSE second stage is added. When $n_s = 1$, the undesirable performance shows that the most frequently occurring state is not necessary the correct one, especially when metric sums, other than the smallest one are considered. As r is set to infinity, even highly unreliable metric sums are considered. The effect is most detrimental when the number of users is two. The reason is that all the metric sums (total number of metric sums equals to 3 when $K = 2$) including unreliable ones are considered. If only $n_s = 1$ state is selected, the possibility of making an error is high. Fortunately, this can be resolved by considering the reliability of the metric sums (shown in the last simulation results of this section). As K increases, the number of metric sums considered is still fixed at $n_M = 3$ even though the total number of metric sums exceeds 3. Hence, some of the less reliable sums are not considered, and performance gradually improves. Furthermore, by selecting just 1 bit-combination, the CBMLSE stage becomes redundant. It is simply the original folded trellis detector taking into consideration other unreliable metric sums. Therefore, we can conclude that increasing n_M without increasing n_s has an adverse effect on the performance and n_s should be kept close to n_M.

The conditions used in Figure 6 are the same as in Figure 5 except that here n_s is fixed at 2 and n_M varies from 1 to 4. It is observed that the performance improves as n_M is increased up to 3. For $n_M = 4$, there is no significant improvement. For this case, setting $n_M = 3$ and $n_s = 2$ is the optimal trade-off with respect to performance and complexity. Even for 8 users, when there are 256 bit-combinations, the sub-optimum MLSE only has to consider 2 bit-combinations to give near optimum performance.

In Figure 7, we investigate the effect of “overloading” the system. Here the processing gain, G, is set to be 7 and the number of users is increased up to 10 which is greater than G. From the figure, it is clear that with just $n_M = 2$ and $n_s = 2$, the performance of the sub-optimum MLSE has already reached that of the MLSE even under excessive loading. For a BER of 0.05, the proposed detector can have 2.5 users more than the MMSE and 6 users more than the conventional matched filter receiver [9]. Using the complexity table, it is found that for 10 users, the complexity of the MLSE is 45 times greater than the proposed MUD. The

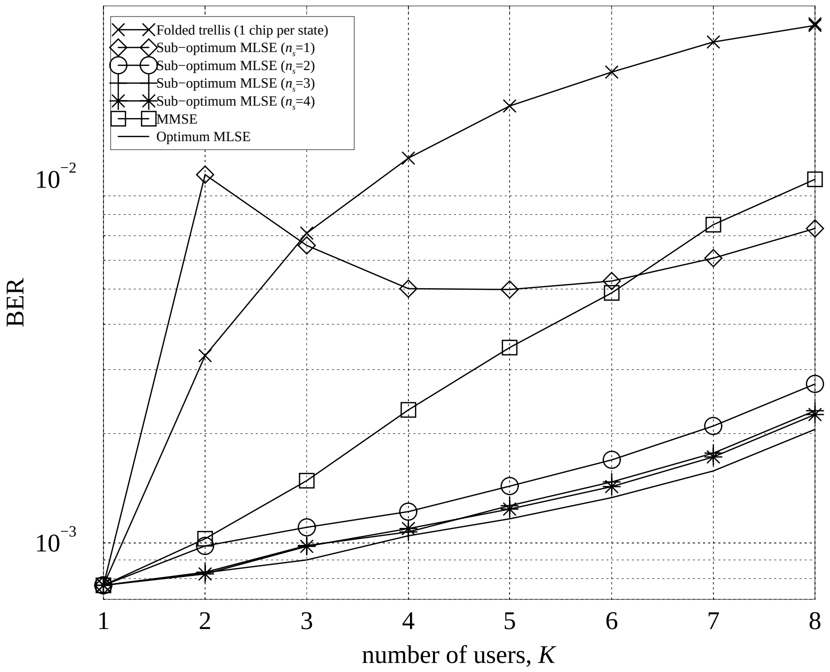

Figure. 5. *BER against number of active users for* $E_b/N_0 = 7$ *dB,* $G = 16$*,* $r = \infty$ *and* $n_M = 3$*.*

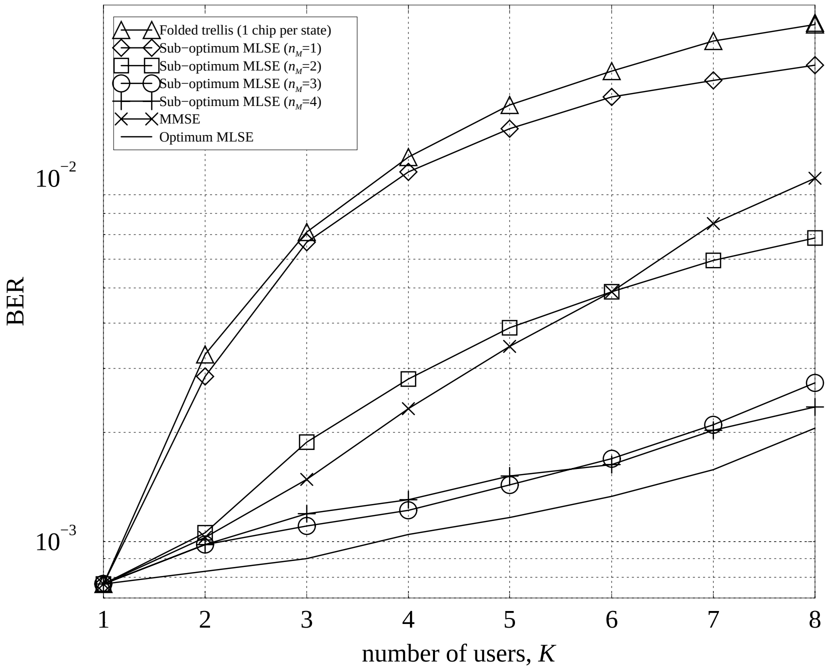

Figure. 6. *BER against number of active users for* $E_b/N_0 = 7$ *dB,* $G = 16$*,* $r = \infty$ *and* $n_s = 2$*.*

graph for $n_M = 1$ and $n_s = 1$, which is equivalent to the original folded trellis without the CBMLSE stage, is also plotted. Here we observe the great performance gain over the MMSE especially for high loading, without even exploiting the CBMLSE stage.

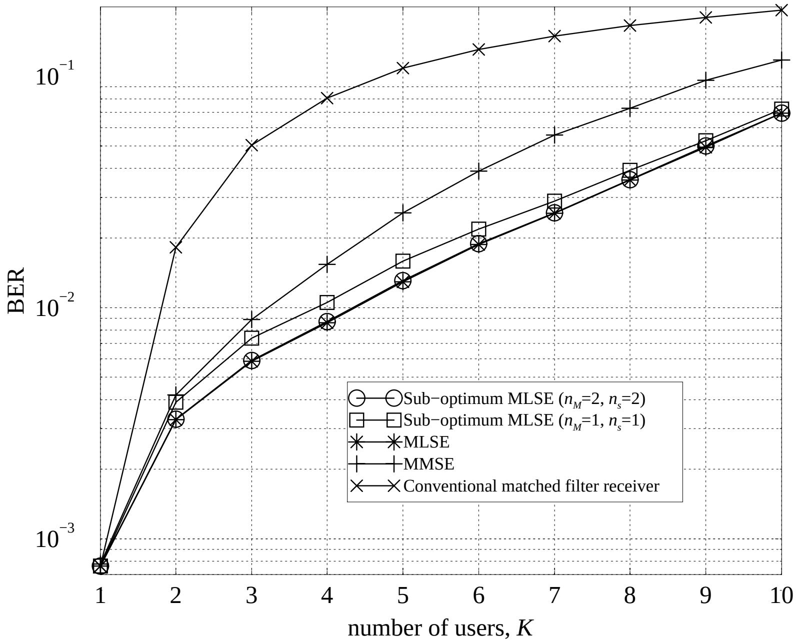

Figure. 7. *BER against the number of active users for* $E_b/N_0 = 7$ *dB,* $G = 7$ *and* $r = \infty$.

In Figure 8, we investigate the effect of varying r. The conditions are the same as in the simulation results shown in Figure 5. The difference is that here the BER against r is plotted for a 2-user system using $n_s = 1$ and $n_s = 2$. It is observed that for $n_s = 1$, as r increases, the BER decreases initially. However, on increasing r further, performance degrades rapidly. This explains the unsatisfactory performance of the sub-optimum MLSE if $n_s = 1$ and $K = 2$ as seen in Figure 5. For $n_s = 2$, performance gradually improves as r increases. Note that if $n_s = 1$, the second stage is unnecessary, but simulation results show that if $n_m > 1$ and $n_s = 1$, performance can be worse than the original folded trellis detector. Hence, it is recommended that $n_s = 1$ is avoided. As long as $n_s \neq 1$, $n_m \neq 1$, and n_s is kept closed to n_m, r does not have to be considered, or simply sets to infinity.

7 Conclusions

In this paper, a MUD for the equal-power downlink of the DS-CDMA systems is proposed. It has been shown that by using the folded trellis as preselection stage for the MLSE, the performance of the sub-

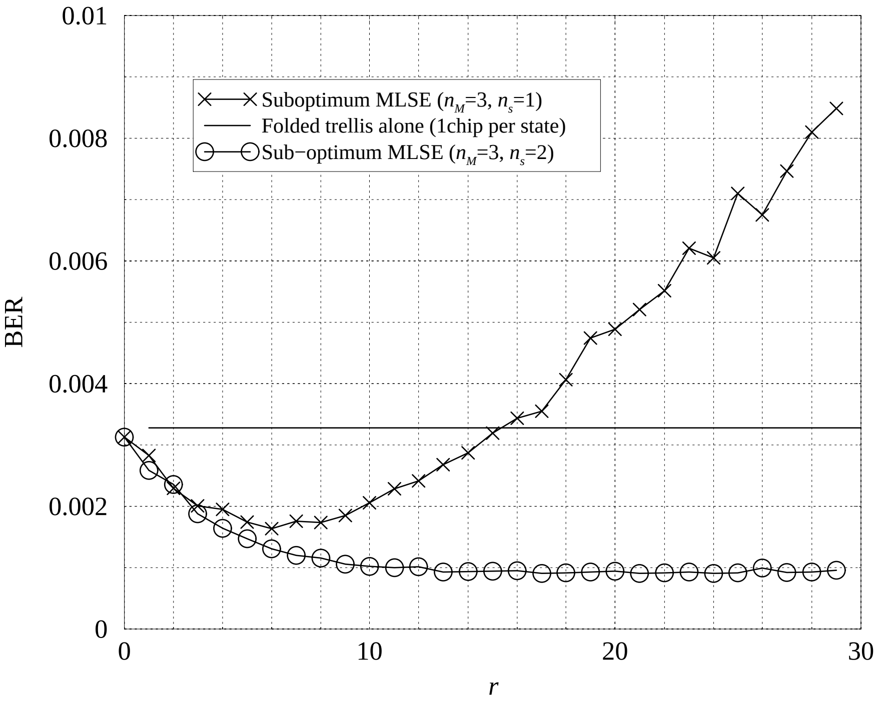

Figure. 8. *BER against the radius of sphere for* $E_b/N_0 = 7$ *dB,* $G = 16$ *and 2 users*

optimum MUD can be near the optimum MLSE. The complexity of the MLSE is greatly reduced due to the small subset of bit-combinations preselected by the folded trellis. In short, we have exploited the low complexity of the folded trellis and the high performance of the MLSE using the preselection concept, resulting in a better trade-off between complexity and performance.

References

[1] SIM, H. K., AND CRUICKSHANK, D. G. M.:, ’A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis,’ *Proceedings of the 1999 49th IEEE Vehicular Technology Conference*, May 1999, pp. 846–850.

[2] VERDU, S.:, ’Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels,’ *IEEE Transactions on Information Theory*, January 1986, **IT-32**, pp. 85–96.

[3] SIM, H. K., AND CRUICKSHANK, D. G. M.:, ‘A Chip Based Multiuser Detector for the Downlink of a DS-CDMA System using a Folded State-Transition Trellis.’ To appear in *IEEE Transactions on Communications*.

[4] FORNEY, JR., G. D.:, ’The Viterbi Algorithm,’ *Proceedings of the IEEE*, March 1973, **61**, pp. 268–278.

[5] TIA/EIA IS-95A:, Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular system. Telecommunication Industry Association, 1993.

[6] REED, M. C., SCHLEGEL, C. B., ALEXANDER, P. D., AND ASENSTORFER, J. A.:, ’Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance,’ *IEEE Transactions on Communications*, December 1998, **46**, pp. 1693–1699.

[7] ALEXANDER, P. D., REED, M. C., ASENSTORFER, J. A., AND SCHLEGEL, C. B.:, ’Iterative Multiuser Interference Reduction: Turbo CDMA,’ *IEEE Transactions on Communications*, July 1999, **47**, pp. 1008–1014.

[8] MADHOW, U., AND HONIG, M. L.:, ’MMSE Interference Suppression for Direct-sequence Spread-Spectrum CDMA,’ *IEEE Transactions on Communications*, December 1994, **42**, pp. 3178–3188.

[9] PROAKIS, J. G.:, Digital Communications. McGraw-Hill, 1995.