Development of a Multiuser Detection Testbed

Master's Thesis

Jordy Potman

University of Twente
Department of Electrical Engineering
Laboratory of Signals & Systems

Enschede, The Netherlands
University of Twente

Supervisors:	Prof. Dr. Ir. C. H. Slump
	Ir. F. W. Hoeksema
Date:	May 16, 2002
Report Code:	SAS 016N02

Abstract

A study on the practicality of DSP implementation of multiuser detection techniques for CDMA systems has been performed. This study was started with an evaluation of various multiuser detection techniques described in literature based on their bit-error-rate performance, required knowledge of received signal parameters and computational complexity. As a result of this evaluation the blind adaptive MMSE (Minimum Mean Square Error) detector and the PIC (Parallel Interference Cancellation) detector are found to be the most suitable detectors for implementation. Further analysis of these detectors lead to an adaptive algorithm for implementation of the blind adaptive MMSE detector and two methods for implementation of the PIC detector. The developed implementations for these detectors were tested for various CDMA system and detector parameters using simulations. Simulation results were also used to study the effect of detector parameters on detector performance and to compare the detectors to each other. The simulation results show that the bit-error-rate performance of the blind adaptive MMSE detector is superior to the conventional CDMA detector in perfect power control CDMA systems as well as CDMA systems that suffer from the near-far effect. The PIC detector has similar performance as the blind adaptive MMSE detector in perfect power control systems. In systems that suffer from the near-far effect the PIC detector does not perform well because of the amplitude estimation technique it uses. The blind adaptive MMSE detector has been chosen for DSP implementation, because of its better overall bit-error-rate performance and because no DSP implementation of the blind adaptive MMSE detector is known in literature. The developed DSP implementation of the blind adaptive MMSE detector has been used to evaluate the detected-bits-per-second performance that can be achieved by this detector on the current generation of DSPs.

Contents

List of Figures

List of Tables

Preface

When I started thinking about a subject for my master's assignment, the final assignment of my study electrical engineering at the University of Twente, I quickly knew that I would like to work on digital signal processing in cellular wireless mobile communications. While studying literature on cellular wireless mobile communications I became interested in multiuser detection techniques for code-division multiple access wireless communication systems. This thesis therefor describes the research on multiuser detection in cellular wireless mobile communications performed for my master's assignment at the Laboratory of Signals & Systems of the Faculty of Electrical Engineering of the University of Twente.

The work described in this thesis would not have been realized without the help of many others. First of all, I wish to thank my supervisors Prof. Dr. Ir. Kees Slump and Ir. Fokke Hoeksema for providing guidance, support and helpful comments during my master's assignment and for carefully reading this thesis. I would also like to thank Ing. Geert-Jan Laanstra for his help with getting the DSP to work. I wish to thank Hans Roelofs for his help on understanding the blind adaptive MMSE detector and for numerous discussions on multiuser detection. Further I would like to thank the staff and students of the Laboratory of Signals & Systems for providing such a nice working environment. Last but certainly not least, I wish to thank my family, friends and roommates for their support during my studies.

Enschede, May 16, 2002

Jordy Potman

Chapter 1

Introduction

Cellular wireless mobile telephony is an example of *multiple access communications*. In multiple access communications several users share a common communications channel over which they can send information simultaneously. *Multiple access techniques* are used to share the available channel resources with different users. In cellular wireless mobile telephony traditionally *Frequency-Division Multiple Access* (FDMA) and *Time-Division Multiple Access* (TDMA) multiple access techniques have been used. In these techniques frequency bands, respectively time slots are allocated to a user. The frequency band or time slot of this user is disjoint from the frequency band or time slot of all the other users. In this way the multiple access channel reduces to a multiplicity of single point-to-point channels.

With the introduction of the IS-95 system in the United States and some other countries and the rise of 3rd generation cellular wireless mobile telephony a third multiple access technique, *Code-Division Multiple Access* (CDMA), has gained importance. Instead of dividing the available channel resources over the different users by allocating frequency or time slots, this technique allocates all resources to all simultaneous users, controlling the power transmitted by each to the minimum required to maintain a given signal-to-noise ratio for the required level of performance. Each user employs a noiselike wideband signal occupying the entire frequency allocation for as long as it is needed. In this way, each user contributes to the interference affecting all the users, but to the least extent possible. The interference that a user experiences caused by the other users on a channel is called *multiple access interference* (MAI). The different users on a channel can be identified by the 'signature' of their noiselike wideband signal.

In the conventional method for detecting the information sent by a user in a CDMA system, multiple access interference is not taken into account. This method follows a *single-user detection* strategy in which each user is detected separately and the interference caused by the other users is treated as noise. Because of the nature of the multiple access interference, however,

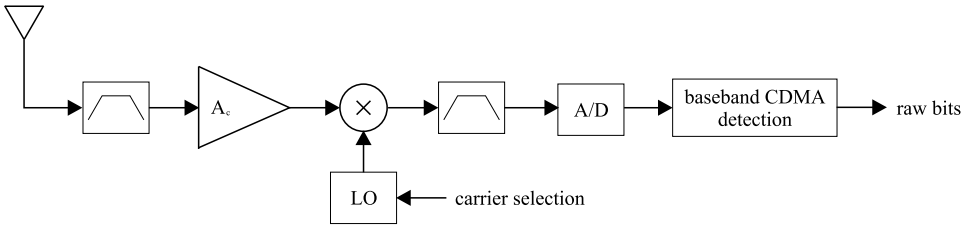


Figure 1.1: Basic CDMA receiver block diagram

a *multiuser detection* method is a better strategy. Here, information about the other users is used to improve detection of each individual user.

Over the years there has been a lot of research on the development of *multiuser detection techniques*. This has lead to a few optimal and a number of suboptimal multiuser detection techniques. Until recently however, there has not been much effort in the practical implementation of these techniques. A way to study the practical implementation of multiuser detection techniques is the development of a test-bed. The test-bed provides the environment in which these multiuser detection techniques would operate in a as practical system as possible and therefor allows development and testing of multiuser detection techniques in real world conditions.

1.1 Purpose and Constraints of Research

The purpose of research is to *develop a DSP-based multiuser detection test-bed that allows real-time evaluation of multiuser detection algorithms for CDMA systems*. To make this project feasible considering the time-frame of a Master's Thesis project and the available resources, a number of limitations is set on the test-bed.

The test-bed is limited to the baseband part of the CDMA receiver, after carrier-demodulation and before any error correction or other bit operations, see Figure 1.1. In the baseband part of the CDMA receiver the signals of the different users in the CDMA system are separated from each other. Therefor this is the part of a CDMA receiver that is of interest for multiuser detection.

The test-bed uses a very simple multiuser additive white Gaussian noise channel model. Real-world wireless mobile communication channels are a lot more complicated and are not only noisy but also have multipath and fading properties.

In a practical CDMA receiver synchronization techniques are used to achieve synchronization with the received signal. In the test-bed, however, it is assumed that the required synchronization information is available to the receiver and therefor synchronization techniques are not implemented.

In a large part of this thesis it is also assumed that all the users in the communications system are synchronized. In real-world wireless mobile

communications systems this is often not practical because the users in the system are located at different physical locations. In order to provide a more complete theoretical background asynchronous CDMA systems and detectors for asynchronous CDMA systems are still described in Chapters 2 and 3.

1.2 Organization of this Thesis

The remaining chapter of this thesis are organized as follows. Chapter 2 described CDMA systems and presents a way to represent these systems mathematically. Chapter 3 contains an overview and comparison of the main multiuser detection techniques for CDMA systems that have appeared in the literature over the years with as goal the selection of detection techniques that are most suitable for implementation. In Chapter 4 a more detailed analysis is given of the blind minimum mean square error detector and the parallel interference cancellation detector, two of the multiuser detection techniques presented in Chapter 3 that were chosen for further research. This analysis is needed to determine the best way in which these detectors can be implemented. Chapter 5 presents simulation results of the blind minimum adaptive mean square error and the parallel interference cancellation detector. The simulations are used to test if the developed algorithms for these detectors work correctly and to study the effects that the CDMA system parameters have on these detectors. In Chapter 6 implementation on a DSP of the blind adaptive minimum mean square error detector is discussed. This discussion is mainly focussed on determining the detected-bits-per-second performance that can be achieved for the blind adaptive minimum mean square error detector on the current generation of DSPs. Finally Chapter 7 presents the conclusions and recommendations that can be drawn from the research presented in this thesis.

Chapter 2

Code-Division Multiple Access

In this chapter Code-Division Multiple Access (CDMA) systems are described. First the principles of CDMA will be discussed. After that continuous- and discrete-time mathematical models will be given for synchronous as well as asynchronous code-division multiple access communication channels.

2.1 CDMA Principles

CDMA uses *direct-sequence spread-spectrum* techniques to achieve efficient multiple access communications. In a direct sequence spread spectrum transmitter each bit of a binary nonreturn-to-zero information signal is modulated by one period of a binary nonreturn-to-zero *pseudo-random sequence* to generate the transmitted signal, see Figure 2.1. This pseudo-random sequence is also referred to as *signature sequence*, *signature waveform*, or in an older terminology *code*, explaining the term code-division multiple access. The pseudo-random sequence is composed of elementary pulses of duration T_c commonly referred to as *chips*. Because the duration of a chip of the pseudo-random sequence is usually a factor between 31 and 128 smaller than the duration T of a bit of the information signal the modulated signal will be a wide-band signal with nearly the same spectrum as the pseudo-random sequence. The bandwidth expansion ratio T/T_c is also know as the *spreading gain*. In a CDMA system the particular signature sequence of a user identifies the particular point-to-point channel corresponding to that user.

CDMA receivers employ the signature sequence of a user as the key to recover the transmitted information. Detection of the transmitted data is accomplished with a correlation demodulator driven by a synchronized replica of the signature waveform used at the transmitter, see Figure 2.2. The by

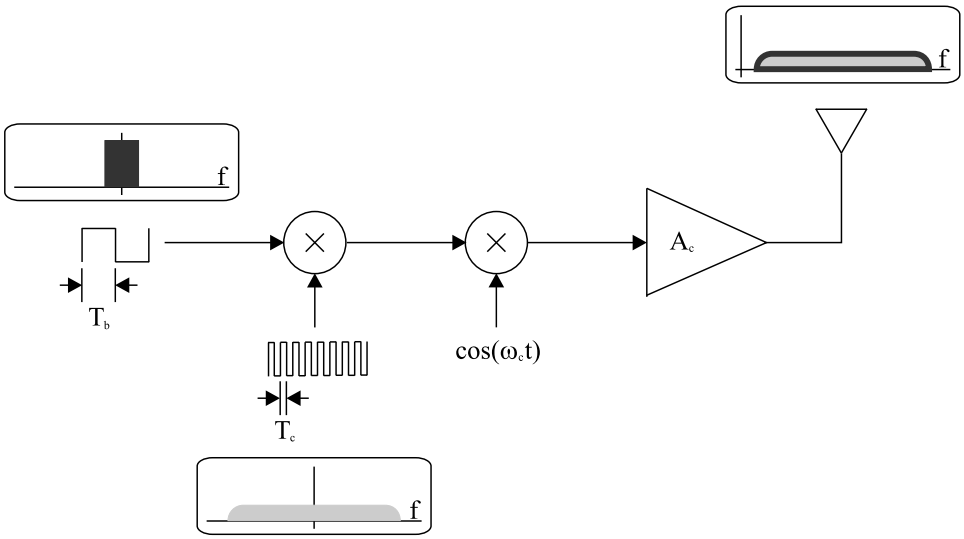


Figure 2.1: Direct-sequence spread-spectrum transmitter

design low correlation between the signature waveforms of the different users gives the CDMA system its multiple access properties.

In an ideal CDMA system orthogonal signature sequences would be used. A CDMA system using orthogonal signature sequences will cancel out all multiple access interference and yield single user performance. However, fully orthogonal systems are not practical for two reasons. First, for a given limited number of chips there only exist a limited number of orthogonal signature sequences. Secondly, and more importantly, if the users are not transmitting synchronously, the signature sequences are out of phase and loose their orthogonal property. CDMA systems normally use *shift-register sequences* or combinations of shift-register sequences for their signature se-

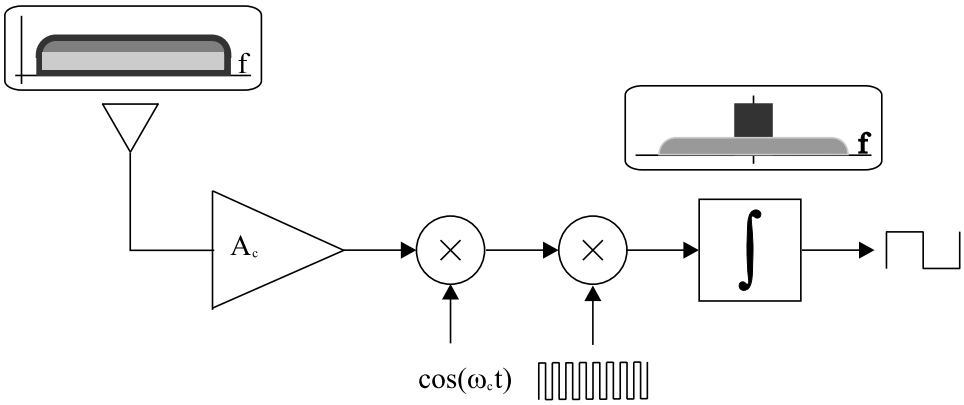


Figure 2.2: Direct-sequence spread-spectrum receiver

quences. It is not possible to obtain signature sequences for any pair of users that are orthogonal for all time offsets using this method.

Not having completely orthogonal signature sequences would not cause a dramatic performance decrease, if the received power of the interfering signal is smaller than the received power of the signal of the desired user. However, if the interfering signals are much stronger than the signal of the desired user, proper detection becomes impossible. This scenario occurs often in practical systems. For example consider the link from a cellular phone to a base station. If one cellular phone is transmitting from a position close to the base station and another cellular phone is transmitting from a position further away from the base station, then the received power of the signal of the first cellular phone will be higher than that of the distant cellular phone, assuming that their transmit powers are equal. Thus, the detection of the distant user will result in a severe increase in bit errors. This situation is generally called the *near-far problem*.

A natural solution to the near-far problem is the use of *power control*. The power control system compares the received power levels. It uses a control channel to transmit power status information to the cellular phones. The cellular phone will adapt its transmitting power in such a way that the received power at the base station is equal for all users. Power control alleviates the near-far problem at the expense of receiver complexity and increased bandwidth. In the next chapter it will be shown that the near-far problem can also be solved by using detection techniques that are less sensitive to differences between the received signal powers of the users.

The length is another property of the signature sequences that has to be taken into consideration. Traditionally signature sequences with a length N were used in CDMA so that $NT_c = T$, thus the period of the signature sequence is equal to the duration of a bit, so each bit is modulated by the entire signature sequence. These kind of signature sequences are so called *short codes*. The disadvantage of using signature sequences with a relatively short length is that the crosscorrelations between the signature sequences vary relatively strongly from each other. Since the crosscorrelations between the signature sequence of a channel and the signature sequence of the other channels determines the amount of interference on the channel, varying crosscorrelations will result in a bit-error-rate performance that varies between channels. To solve this so called *long codes* can be used.

Long codes are signature sequences with a period that is much longer than the duration of a bit, for example signature sequences with a period of $2^{42}T_c$. When these signature sequences are used to modulate a bit stream each bit will be modulated by a different section of the signature sequence. The amount of interference for a particular bit therefor depends on the crosscorrelations between the section of the signature sequence that is used to modulate that bit and the sections of the signature sequences that are used to modulate the bits that are transmitted at the same time in the

other channels. This results in an interference on each channel that varies from bit to bit, but averaged over the bits there will be the same amount of interference on each channel. Therefor the bit-error-rate performance on each channel will be the same.

The use of long codes is not the only solution to the problem of the varying bit-error-rate performance between channels of the traditional, short code, CDMA system. In the next chapter it will be shown that it is possible to develop detection techniques that remove or reduce multiple access interference on a channel and thus remove or reduce the influence that multiple access interference has on bit-error-rate performance of a channel. But first, in the next section, mathematical models for CDMA systems will be given that can be used to develop and analyze CDMA detection techniques.

2.2 Continuous-time CDMA Models

In this section continuous time models are given of a CDMA system with a total number of K users that transmit bits using binary antipodal modulation over a white Gaussian noise channel. These models are widely used in CDMA and multiuser detection literature. A good reference on this subject is the book *Multiuser Detection* by Sergio Verdú [15], on which this section is largely based. The notation used in the models assumes short code systems, but the models can be easily extended to long code systems.

2.2.1 Synchronous Channel

The basic synchronous K-user CDMA model describes the received signal of a CDMA system in which K synchronous bit streams antipodally modulate K signature waveforms which are transmitted over an *Additive White Gaussian Noise* (AWGN) channel. Both the bit streams and the signature waveforms are represented by non-return-to-zero (NRZ) signals. The received signal for one symbol period in such a system can be expressed as:

$$r(t) = \sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t). \tag{2.1}$$

Where the following notation is used:

- $s_k(t)$ is the deterministic signature waveform assigned to the kth user, normalized to have unit energy

  $$||s_k||^2 = \int_0^T s_k(t)dt = trianleq1. \tag{2.2}$$

  The signature waveforms are assumed to be zero outside the interval $[0, T]$, and therefore, in an AWGN channel, there is no intersymbol interference.

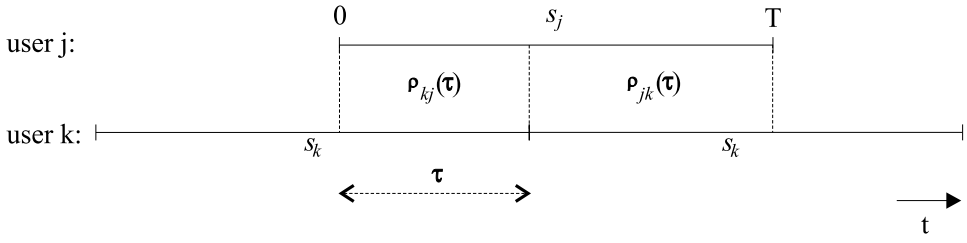


Figure 2.3: Definition of asynchronous crosscorrelations ($k < l$)

- A_k is the received amplitude of the kth user's signal. A_k^2 is referred to as the energy of the kth user.
- $b_k \in \{-1, +1\}$ is bit transmitted by the kth user.
- $n(t)$ is white Gaussian noise with zero mean and unit variance. It models thermal noise plus other noise sources unrelated to the transmitted signals. According to (2.1) the noise power in a frequency band with bandwidth B is $2\sigma^2 B$.[1]

The model above could for example describe the received signal on a forward link (base-station to mobile) in a CDMA cellular system, using a very basic channel assumption without fading or multipath.

In the next chapter it will be shown that the bit-error-rate performance of various demodulation strategies depends on the signal-to-noise ratios, A_k/σ, and on the similarity between the signature waveforms, quantified by their crosscorrelations defined as

$$\rho_{jk} \triangleq \langle s_j, s_k \rangle = \int_0^T s_j(t) s_k(t) dt. \tag{2.3}$$

2.2.2 Asynchronous Channel

In cellular systems the reverse link (mobile-to-base-station) is often not synchronized. Therefore offsets $\tau_k \in [0, T), k = 1, \ldots, K$ are introduced to model the lack of alignment of the bit epochs of the K different users at the receiver. The symbol-epoch offsets are defined with respect to an arbitrary origin. The received signal during one frame time in such a system can be expressed as:

$$r(t) = \sum_{k=1}^{K} \sum_{i=-M}^{M} A_k b_k[i] s_k(t - iT - \tau_k) + \sigma n(t). \tag{2.4}$$

Where the length of the frames transmitted by each user is assumed to be equal to $2M + 1$ bits.

[1] In the literature, the noise one-sided spectral level $2\sigma^2$ is frequently denoted by N_0.

The asynchronous model (2.4) can be viewed as a special case of the synchronous model (2.1), what often simplifies the probability of error analysis of detectors [10]. Each bit in (2.4) $\{b_k[i], k = 1, \ldots, K; i = -M, \ldots, M\}$ can be considered as coming from a different 'user' in a synchronous channel whose bit interval is $[-MT, MT+2T]$. In this view, the number of fictitious users is equal to $(2M + 1)K$.

As already stated in section 2.2.1, the performance of various demodulation strategies depends on the crosscorrelations between the signature waveforms. For asynchronous CDMA, two crosscorrelations between every pair of signature waveforms have to be defined, that depend on the offset between the signals. This can be seen in Figure 2.3. This figure shows that one symbol time for user j overlaps with two symbol times for user k. When the offset τ_j of user j is smaller than the offset τ_k of user k the two crosscorrelations can be defined as:

$$\begin{aligned} \rho_{jk}(\tau) &\triangleq \int_{\tau}^{T} s_j(t)s_k(t-\tau)dt, \\ \rho_{kj}(\tau) &\triangleq \int_{0}^{\tau} s_j(t)s_k(t+T-\tau)dt, \end{aligned} \tag{2.5}$$

where $\tau \in [0, T]$.

2.3 Discrete-time CDMA Models

Detectors commonly have a front-end whose objective is to obtain a discrete-time representation from the received continuous-time waveform $r(t)$. One way of converting the received waveform into a discrete-time representation is to pass it through a bank of matched filters, see Figure 2.4, each matched to the signature waveform of a different user. The outputs of the matched filters are than sampled at the end of each bit period. In other words, the matched filter bank correlates the received signal with the signature waveform of each individual user. To perform matched filtering for a user, knowledge of the signature waveform and the timing of that user is required.

In this section expressions are given for the sampled matched filter outputs for synchronous as well as asynchronous systems, these results are again mainly taken from Verdú [15].

2.3.1 Synchronous Channel

The matched filter bank correlates the received signal with the signature waveform of each individual user. The output of the matched filter for a user k for synchronous CDMA can therefor be expressed as:

$$y_k = \int_{0}^{T} r(t)s_k(t)dt \tag{2.6}$$

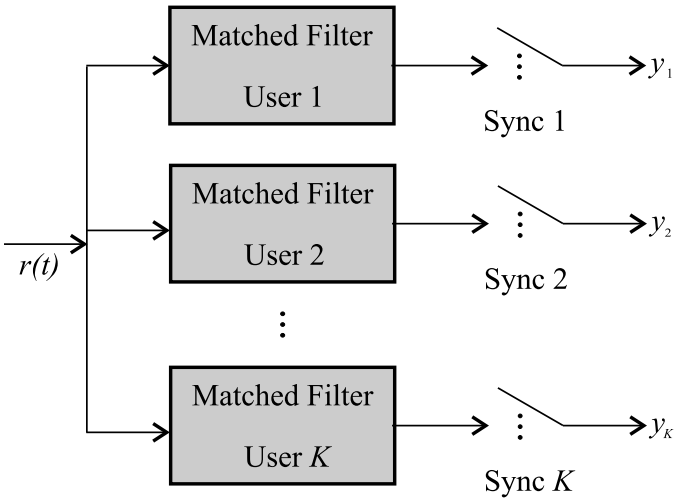


Figure 2.4: Bank of matched filters

$$= \int_0^T \left(\sum_{k=1}^{K} A_k b_k s_k(t) + \sigma n(t) \right) s_k(t) dt \tag{2.7}$$

$$= \int_0^T \left(A_k b_k[i] s_k(t) + \sum_{j \neq k} A_j b_j s_j(t) + \sigma n(t) \right) s_k(t) dt \tag{2.8}$$

$$= \int_0^T A_k b_k s_k(t) s_k(t) dt + \int_0^T \sum_{j \neq k} A_j b_j s_j(t) s_k(t) dt + \int_0^T \sigma n(t) s_k(t) dt \tag{2.9}$$

$$= A_k b_k \int_0^T s_k(t) s_k(t) dt + \sum_{j \neq k} A_j b_j \int_0^T s_j(t) s_k(t) dt + \sigma \int_0^T n(t) s_k(t) dt \tag{2.10}$$

Now by using (2.3) and the fact that $s_k(t)$ is normalized to have unit energy the matched filter outputs can be expressed as

$$y_k = A_k b_k + \sum_{j \neq k} A_j b_j \rho_{jk} + n_k, \tag{2.11}$$

where

$$n_k \triangleq \sigma \int_0^T n(t) s_k(t) dt \tag{2.12}$$

is a Gaussian random variable with zero mean and variance equal to σ^2.

Equation (2.11) can be written in vector form:

$$\mathbf{y} = \mathbf{RAb} + \mathbf{n}, \tag{2.13}$$

where

$$\begin{aligned} \mathbf{R} &= \{\rho_{jk}\}, && (2.14)\\ \mathbf{y} &= [y_1, \ldots, y_K]^T, && (2.15)\\ \mathbf{b} &= [b_1, \ldots, b_K]^T, && (2.16)\\ \mathbf{A} &= \mathrm{diag}\{A_1, \ldots, A_K\}, && (2.17)\\ \mathbf{n} &= [n_1, \ldots, n_K]^T && (2.18)\\ & && (2.19) \end{aligned}$$

So $\mathbf{n}$ is a zero-mean Gaussian random vector with covariance matrix equal to

$$E[\mathbf{n}\mathbf{n}^T] = \sigma^2\mathbf{R}. \tag{2.20}$$

2.3.2 Asynchronous Channel

Using the same reasoning as for the synchronous channel and equations (2.4) and (2.5) the matched filter outputs in the case of asynchronous CDMA can be expressed as

$$\begin{aligned} y_k[i] = A_k b_k[i] &+ \sum_{j<k} A_j b_j[i+1]\rho_{kj} + \sum_{j<k} A_j b_j[i]\rho_{jk} \\ &+ \sum_{j>k} A_j b_j[i]\rho_{kj} + \sum_{j>k} A_j b_j[i-1]\rho_{jk} + n_k[i], \end{aligned} \tag{2.21}$$

where

$$n_k[i] \triangleq \sigma \int_{\tau_k+iT}^{\tau_k+iT+T} n(t)s_k(t - iT - \tau_k)dt, \tag{2.22}$$

and it is assumed that $\tau_1 \leq \tau_2 \ldots \leq \tau_K$, that is the users are labelled chronologically, i.e. by their time of arrival.

Equation (2.21) can be written in matrix form:

$$\mathbf{y}[i] = \mathbf{R}^T[1]\mathbf{Ab}[i+1] + \mathbf{R}[0]\mathbf{Ab}[i] + \mathbf{R}[1]\mathbf{Ab}[i-1] + \mathbf{n}[i], \tag{2.23}$$

where the zero-mean Gaussian process $\mathbf{n}[i]$ has autocorrelation matrix

$$E[\mathbf{n}[i]\mathbf{n}^T[j]] = \begin{cases} \sigma^2\mathbf{R}^T[1], & \text{if } j = i+1; \\ \sigma^2\mathbf{R}[0], & \text{if } j = i; \\ \sigma^2\mathbf{R}[1], & \text{if } j = i-1; \\ \mathbf{0}, & \text{otherwise,} \end{cases} \tag{2.24}$$

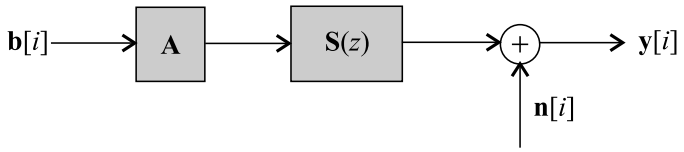


Figure 2.5: K-dimensional channel of matched filter outputs for asynchronous CDMA channel.

and the matrices $\mathbf{R}[0]$ and $\mathbf{R}[1]$ are defined by

$$R_{jk}[0] \triangleq \begin{cases} 1, & \text{if } j = k; \\ \rho_{jk}, & \text{if } j < k; \\ \rho_{kj}, & \text{if } j > k; \end{cases} \tag{2.25}$$

$$R_{jk}[1] \triangleq \begin{cases} 0, & \text{if } j \geq k; \\ \rho_{kj}, & \text{if } j < k. \end{cases} \tag{2.26}$$

The vector discrete-time model in (2.23) can be represented in the z-transform domain as

$$\mathbf{y}[i] = \mathbf{S}(z)\mathbf{A}\mathbf{b}[i] + \mathbf{n}[i], \tag{2.27}$$

where

$$\mathbf{S}(z) = \mathbf{R}^T[1]z + \mathbf{R}[0] + \mathbf{R}[1]z^{-1} \tag{2.28}$$

is the discrete time channel transfer function, see Figure 2.5.

Matched filter banks are often used as a front-end for CDMA detectors, as already stated in the beginning of this section. In the next chapter the models described in this chapter will be used for analytically evaluating the bit-error-rate performance of various CDMA detection techniques.

Chapter 3

Detection Techniques

Several detection techniques for the CDMA channel have been studied in the literature. One way of categorizing them is to divide them into single-user and multiuser methods [8], see Figure 3.1. A *single-user detector* is defined as a receiver structure that requires no information regarding the other (interfering) users present in the system and demodulates the data signal of one user only.

From the definition of a single-user detector it follows that a *multi-user detector* is as a receiver structure that does require information regarding the other (interfering) users present in the system and demodulates the data signal of all users. Multiuser detection techniques can be divided into joint detection and decision-driven techniques. In *joint detection* the front-end of the detector is traditionally (but not necessarily) a bank of matched filters followed by linear or nonlinear transformations on the matched filter outputs. *Decision-driven* techniques are characterized by the regeneration and subtraction of data estimates.

The goal of this chapter is to select the multiuser detection techniques that are best suited for implementation considering the performance level of the currently available hardware and the bit-error-rate performance levels that are currently requested from detection techniques for cellular wireless mobile telephony systems. Therefor the chapter starts with a discussion of the criteria for evaluation and comparison of the implementation related issues and performance levels of detection techniques. After that, several detection techniques for CDMA systems will be introduced. The first detector that will be discussed, the well-known conventional detector, is an example of a single-user detector. The maximum likelihood sequence estimator (MLSE) detector that is introduced subsequently is an example of a nonlinear joint detection technique. The decorrelating and minimum mean square error (MMSE) detector that are also discussed are examples of linear joint detection techniques. Examples of decision-driven multiuser detection techniques that are introduced in this chapter are the successive

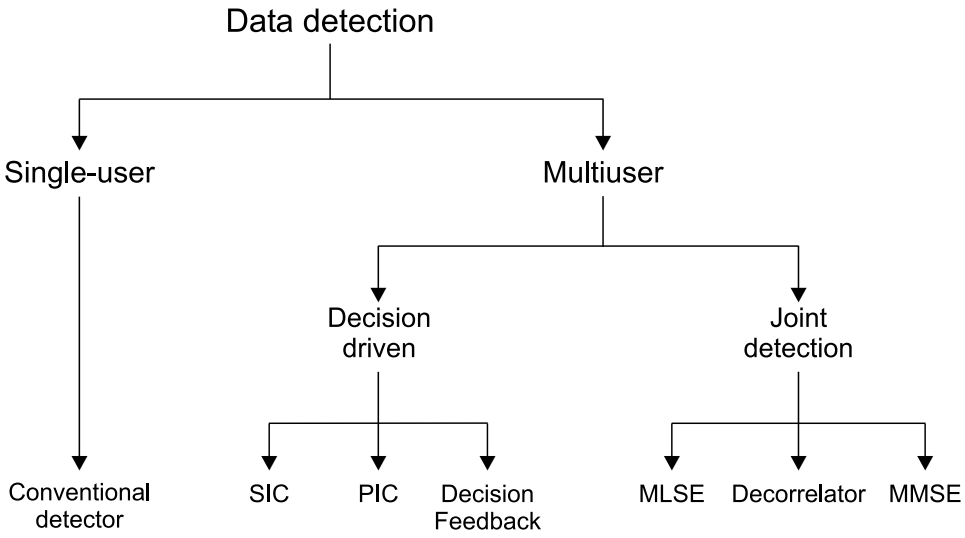


Figure 3.1: Classification of detection techniques.

interference cancellation (SIC), parallel interference cancellation (PIC) and decision-feedback detectors. In the second last section of the chapter a comparison of the introduced detectors will be given. The last section of this chapter summarizes the conclusions that can be drawn from the evaluation of the multiuser detection techniques performed in this chapter.

3.1 Evaluation Criteria for Detection Techniques

The evaluation criteria for detection techniques can roughly be divided into two categories: *implementation related issues* and *performance measures*. The implementation related issues that will be considered for the detectors discussed in this chapter are the received signal- and system parameters that have to be estimated or known by the receiver and the computational complexity of the detector. The performance measures that will be used are bit-error-rate and asymptotic multiuser efficiency.

The *received signal- and system parameters* that have to be estimated or known by the receiver are the signature waveforms, timing and received amplitudes of each user and the noise level. The detectors discussed in this chapter may require all, or only a subset of this information for demodulation of a user, which will be indicated in the section describing the detector. A detector that requires more information will increase the complexity of the receiver, because the information often has to be extracted from the received signal.

The *computational complexity* of any detector can be quantified by its time complexity per bit, that is, the number of operations required by the

detector to demodulate the transmitted information divided by the total number of demodulated bits. For multiuser detection techniques the complexity of demodulating the transmitted information of an individual user usually increases with the total number of users K in the system. A time complexity per bit of $f(K)$ is written as $O(g(K))$ if there exists a constant $c > 0$ such that for large enough K, $f(K) \leq cg(K)$ [15]. In the sections describing the detectors, the computational complexity of each detector will be given for the synchronous as well as the asynchronous case.

The *bit-error-rate* or *probability of error* performance measure is the number of incorrectly detected bits relative to the total number of detected bits. For some detection techniques it is possible to derive an analytical expression for the bit-error-rate. However, for asynchronous systems this is often not straightforward and for some of the described detection techniques it is not possible at all. The only way to acquire a bit-error-rate for these detection techniques is from simulation results.

The *asymptotic multiuser efficiency* is an alternative performance measure that is generally easier to derive than the actual probability of error. It measures the slope with which the bit-error-rate for the kth user $\mathsf{P}_k(\sigma)$ goes to 0 (in logarithmic scale), as σ goes to 0. So in the high signal-to-noise ratio region. It indicates the performance of the detector when the interference caused by the other users is dominating over the noise. The asymptotic multiuser efficiency is defined as [15]

$$\eta_k \triangleq \lim_{\sigma \to 0} \frac{e_k(\sigma)}{A_k^2}, \tag{3.1}$$

where $e_k(\sigma)$ is the effective energy of user k, defined as the energy that user k would require to achieve a bit-error-rate equal to $\mathsf{P}_k(\sigma)$ in a single-user Gaussian channel with the same background noise level, that is,

$$\mathsf{P}_k(\sigma) = Q\left(\frac{\sqrt{e_k(\sigma)}}{\sigma}\right). \tag{3.2}$$

where Q is is the complementary Gaussian cumulative distribution function:

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \tag{3.3}$$

In the next sections equations will be given indicating the analytical performance of each detector using the performance measures described above. These equations will be used in the second last section of this chapter to compare the analytical performance of the different detection techniques. In most cases the equations are just presented as is, for their derivation the reader is referred to the literature, the book *Multiuser Detection* by Sergio Verdú [15] can serve as a good starting point.

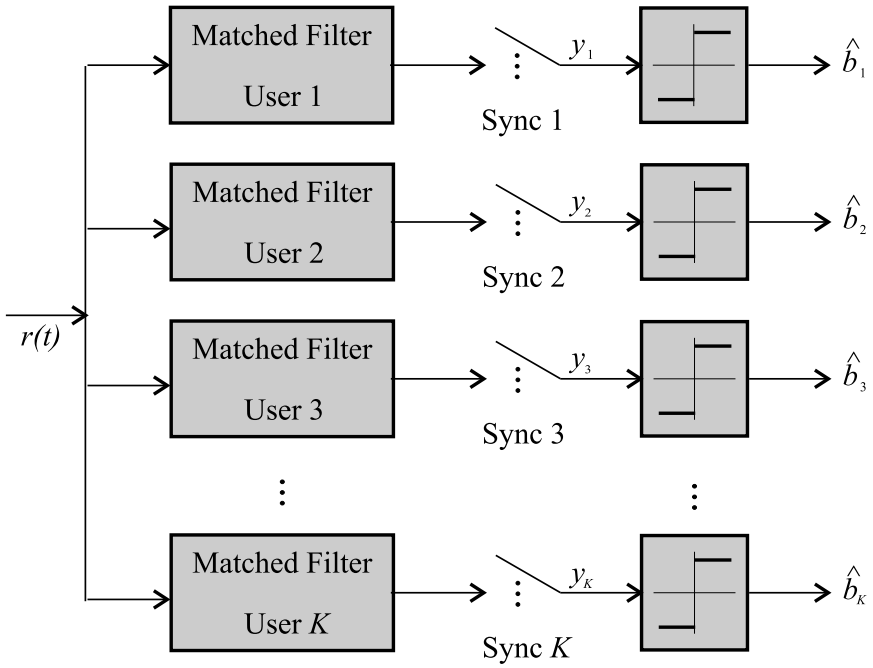


Figure 3.2: Conventional detector.

3.2 Conventional Detector

The conventional detector consists of the front-end with the bank of matched filters, used to obtain a discrete-time process from the received continuous-time waveform as described in section (2.3), followed by bit decisions directly based on the sign of the matched filter outputs

$$\hat{b}_k = \mathrm{sgn}(y_k), \tag{3.4}$$

see Figure 3.2. The signum function sgn is defined as

$$\mathrm{sgn(x)} \triangleq \begin{cases} 1 & \text{if } x >= 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{3.5}$$

The input of the signum function is often referred to as a *soft decision*, while the output of the signum function is often referred to as a *hard decision*.

Recall that y_k is defined in (2.11) as

$$y_k = A_k b_k + \sum_{j \neq k} A_j b_j \rho_{jk} + n_k, \tag{3.6}$$

for the synchronous case, or as

$$\begin{aligned} y_k[i] = A_k b_k[i] &+ \sum_{j<k} A_j b_j[i+1]\rho_{kj} + \sum_{j<k} A_j b_j[i]\rho_{jk} \\ &+ \sum_{j>k} A_j b_j[i]\rho_{kj} + \sum_{j>k} A_j b_j[i-1]\rho_{jk} + n_k[i], \end{aligned} \tag{3.7}$$

in (2.21) for the asynchronous case.

The only received signal- and system parameters required by the conventional detector to detect a user is knowledge of the signature waveform and timing of that user.

Since the detection of a user is independent of the other users in the system the number of computations needed for detection does not increase with the number of users. Therefor the time complexity per bit of the conventional detector is constant.

To start the analysis of the bit-error-rate of the conventional detector it is illustrative to consider the case where the signature waveform of the kth user is orthogonal to the signature waveforms of all other users. In that case ρ_{jk} equals zero when $j \neq k$ and the output of the matched filter for user k is reduced to that of the single user problem. The error probability for a single user in antipodal signaling is known to be

$$P(\sigma) = Q\left(\frac{A}{\sigma}\right). \tag{3.8}$$

Returning to the case of non-orthogonal codes it is illustrative to first analyze the two-user synchronous case before generalizing to the K-user case. To shorten the used notation $\rho_{12} = \rho$ is used for the crosscorrelation between the signature waveforms of the two users. The error probability of the conventional detector for user 1 in the two-user synchronous case is:

$$\begin{aligned} \mathsf{P}_1 &= P[b_1 \neq \hat{b}_1] \\ &= P[b_1 = +1]P[y_1 < 0 | b_1 = +1] + \\ &\quad P[b_1 = -1]P[y_1 > 0 | b_1 = -1]. \end{aligned} \tag{3.9}$$

In the non-orthogonal case the matched filter output y_1 is also dependent on b_2. So in order to find the error probability further conditioning on b_2 is needed:

$$\begin{aligned} P[y_1 > 0 | b_1 = -1] &= P[y_1 > 0 | b_1 = -1, b_2 = +1]P[b_2 = +1] + \\ &\quad P[y_1 > 0 | b_1 = -1, b_2 = -1]P[b_2 = -1] \\ &= P[n_1 > A_1 - A_2\rho]P[b_2 = +1] + \\ &\quad P[n_1 > A_1 + A_2\rho]P[b_2 = -1] \\ &= \frac{1}{2}Q\left(\frac{A_1 - A_2\rho}{\sigma}\right) + \frac{1}{2}Q\left(\frac{A_1 + A_2\rho}{\sigma}\right), \end{aligned} \tag{3.10}$$

where the independence of the data bits b_1, b_2 and the noise term n_1 is used. It is also assumed that the users transmit +1 and -1 with equal likelihood, setting $P[b = +1] = P[b = -1] = 0.5$ for both users. By symmetry the same expression is obtained for $P[y_1 < 0 | b_1 = +1]$. Therefore, the bit-error-rate of the conventional detector in the presence of one interfering user is

$$\mathsf{P}_1^c(\sigma) = \frac{1}{2}Q\left(\frac{A_1 - A_2\rho}{\sigma}\right) + \frac{1}{2}Q\left(\frac{A_1 + A_2\rho}{\sigma}\right), \tag{3.11}$$

where the superscript c is used to indicate the conventional detector. The error probability for user 2 is found in a similar way which leads to a similar equation in which the indices 1 and 2 are interchanged.

In Verdú [15] it is shown that the reasoning above can be generalized to an arbitrary number of users K and to the asynchronous case. The probability of error, or bit-error-rate of the conventional detector for synchronous CDMA can be expressed as (equation 3.90 in Verdú):

$$\begin{aligned}\mathsf{P}_k^c(\sigma) = \tfrac{1}{4^{K-1}} \textstyle\sum_{e_1\in\{-1,1\}} \cdots \sum_{e_j\in\{-1,1\},j\neq k} \cdots \\ \textstyle\sum_{e_K\in\{-1,1\}} Q\left(\frac{A_k}{\sigma} + \sum_{j\neq k} e_j \frac{A_j}{\sigma}\rho_{jk}\right).\end{aligned} \tag{3.12}$$

For asynchronous CDMA the probability of error function becomes (equation 3.104 in Verdú):

$$\begin{aligned}\mathsf{P}_k^c(\sigma) = \tfrac{1}{4^{K-1}} \textstyle\sum_{(e_1,d_1)\in\{-1,1\}^2} \cdots \sum_{(e_j,d_j)\in\{-1,1\}^2,j\neq k} \cdots \\ \textstyle\sum_{(e_K,d_K)\in\{-1,1\}^2} Q\left(\frac{A_k}{\sigma} + \sum_{j\neq k} \frac{A_j}{\sigma}(e_j\rho_{jk} + d_j\rho_{kj})\right).\end{aligned} \tag{3.13}$$

In these equations e_j and d_j represent the values of the bits on which the probability of error is conditioned.

The asymptotic multiuser efficiency of the conventional detector for synchronous CDMA is equal to (equation 3.122 in Verdú)

$$\eta_k^c = \left(\max\left\{0,\ 1 - \sum_{j\neq k}\frac{A_j}{A_k}|\rho_{jk}|\right\}\right)^2. \tag{3.14}$$

For asynchronous CDMA the asymptotic multiuser efficiency becomes (equation 3.124 in Verdú):

$$\eta_k^c = \left(\max\left\{0,\ 1 - \sum_{j\neq k}\frac{A_j}{A_k}(|\rho_{jk}| + |\rho_{kj}|)\right\}\right)^2. \tag{3.15}$$

Therefor the asymptotic multiuser efficiency of the conventional detector for the two-user synchronous case is equal to

$$\eta_k^c = \left(\max\left\{0,\ 1 - \frac{A_2}{A_1}|\rho|\right\}\right)^2. \tag{3.16}$$

This equation shows that the asymptotic multiuser efficiency is always equal to 1 in orthogonal systems because for orthogonal systems $\sigma = 0$. So in orthogonal systems the bit-error-rate plot of the conventional detector for high signal-to-noise ratios goes to 0 (in logarithmic scale) with slope 1. In non-orthogonal systems $\sigma \neq 0$ and the asymptotic multiuser efficiency for user 1 decreases when the amplitude of user 2 becomes larger than the

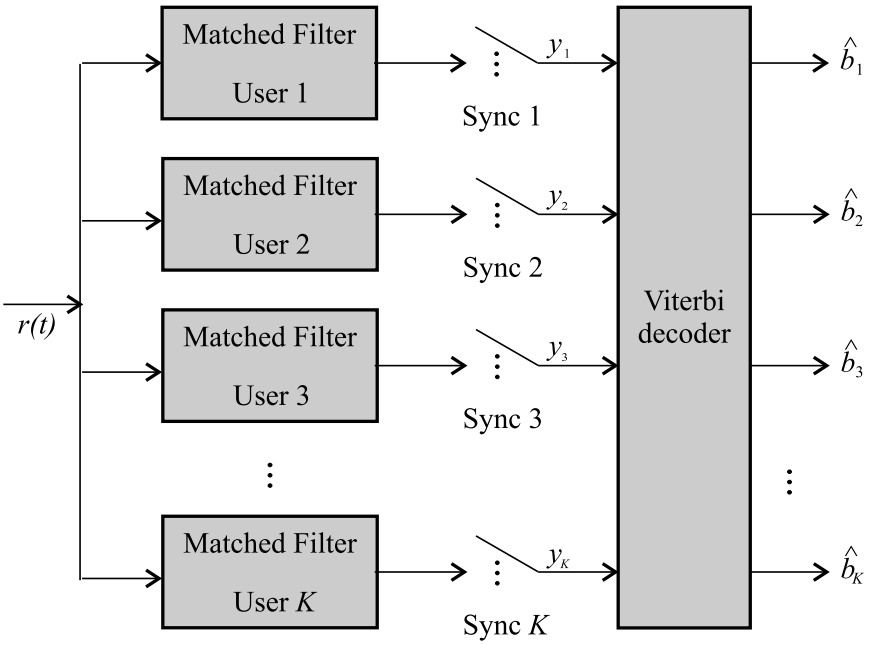


Figure 3.3: MLSE detector.

amplitude of user 1. When the amplitude of user 2 is huge compared to the amplitude of user 1 the asymptotic multiuser efficiency will become 0. This means that the bit-error-rate plot of the conventional detector for high signal-to-noise ratios goes to 0 (in logarithmic scale) with slope 0, or in other words the bit-error-rate of the detector does not improve when the signal-to-noise ratio improves.

3.3 Maximum Likelihood Sequence Estimator Detector

The maximum likelihood sequence estimator (MLSE) detector consists of a bank of matched filters, followed by an algorithm that chooses the input sequence $\mathbf{b}$ that maximizes the likelihood function of the matched filter outputs y. So for the synchronous case

$$\hat{\mathbf{b}} = \arg \max_{\mathbf{b} \in \{-1,+1\}^K} p(y|\mathbf{b}), \tag{3.17}$$

and for the asynchronous case

$$\hat{\mathbf{b}} = \arg \max_{\mathbf{b} \in \{-1,+1\}^{K(2M+1)}} p(y|\mathbf{b}), \tag{3.18}$$

when the asynchronous channel is considered as a $K(2M+1)$-user synchronous channel.

In the synchronous case this is equivalent to finding the hypothesis that minimizes the payoff function [11]:

$$\Omega(\mathbf{b}) = \int_0^T \left[r(t) - \sum_{k=1}^{K} A_k b_k s_k(t) \right]^2 dt. \tag{3.19}$$

If the asynchronous channel is considered as a $K(2M+1)$-user synchronous channel maximizing the likelihood function of the matched filter outputs is equivalent to finding the hypothesis that minimizes the payoff function [15]:

$$\Omega(\mathbf{b}) = \int_{-MT}^{MT+2T} \left[r(t) - S_t(\mathbf{b}) \right]^2 dt \tag{3.20}$$

for the asynchronous case, where

$$S_t(\mathbf{b}) = \sum_{k=1}^{K} \sum_{i=-M}^{M} A_k b_k[i] s_k(t - iT - \tau_k). \tag{3.21}$$

Therefor the maximum likelihood sequence estimator detector requires the following received signal- and system parameters at the receiver: the received amplitudes of all the users, the timing of the desired user, the timing of the interfering users and the signature waveforms of the desired and interfering users. In the synchronous case the requirement of knowledge of the timing of the interfering users can of course be dropped.

It can be shown [15] that generation of all the values of $\Omega(\mathbf{b})$ for the synchronous case can be done in a tree structure that takes $O(2^K)$ operations. The selection of the optimum $\mathbf{b}$ can then be done in $O(2^K)$ operations, so the time complexity per bit is $O(2^K/K)$. The payoff function for the asynchronous case (3.21) can be expressed as a sum of $(2M+1)K$ terms such that each term depends on K components of $\mathbf{b}$ and any consecutive terms share $K-1$ arguments. An expression of that type can be maximized in $O(2^K)$ operations [15] using a real-time version of dynamic programming called the *Viterbi Algorithm*, so the time complexity per bit for the asynchronous case is also $O(2^K/K)$. The maximum likelihood sequence estimator detector for the asynchronous case is shown in Figure 3.3.

It is not possible to derive a closed-form analytical expression for the bit-error-rate of the maximum likelihood sequence estimator detector. However it is possible to derive an equation for the asymptotic multiuser efficiency (equation 4.91 in Verdú):

$$\eta_k^l = \min_{\epsilon \in \{-1,0,1\}^K, \epsilon_k = 1} \frac{1}{A_k^2} \epsilon^T \mathbf{ARA} \epsilon, \tag{3.22}$$

where the superscript l is used to indicate the maximum likelihood sequence estimator detector. ϵ is an error vector, defined as the normalized difference

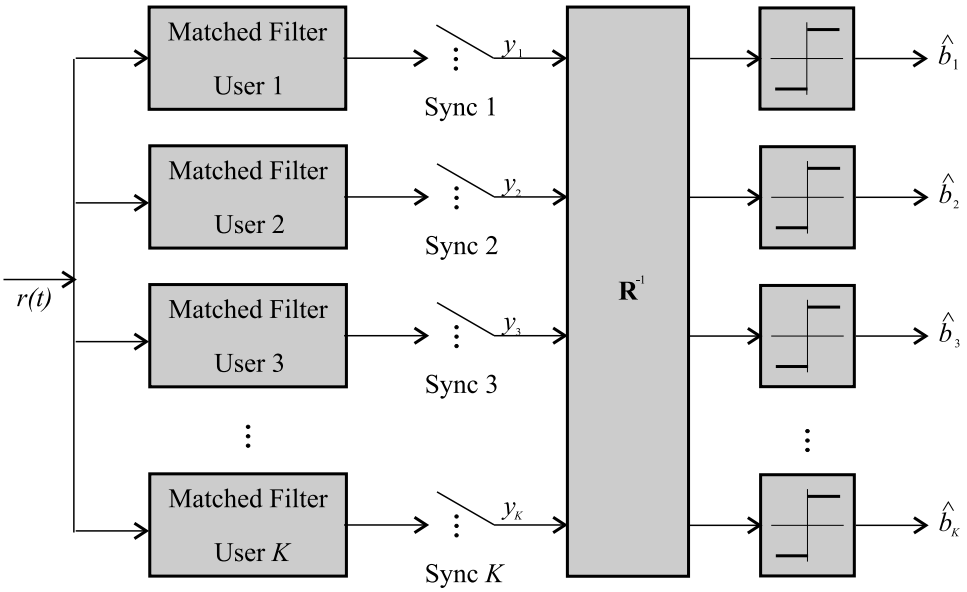


Figure 3.4: Decorrelating detector.

between any pair of distinct transmitted vectors. The set of error vectors that affects the kth user is

$$E_k = \{\epsilon \in \{-1, 0, 1\}^K, \epsilon_k \neq \mathbf{0}\}. \tag{3.23}$$

3.4 Decorrelating Detector

The decorrelating detector consists of a bank of matched filters followed by a linear transformation that multiplies the output of the matched filter bank with the inverse of the correlation matrix, see Figure 3.4. This removes all the interference for a user k, caused by any of the other users. The only source of interference left is the enhanced background noise equal to $\mathbf{R}^{-1}\mathbf{n}$. The bit decisions are then made based on the signs of the outputs of the linear transformation. So for the synchronous case:

$$\hat{b}_k = \text{sgn}((\mathbf{R}^{-1}\mathbf{y})_k), \tag{3.24}$$

For the asynchronous case $\mathbf{R}^{-1}$ is replaced with the inverse of the discrete-time channel transfer function (2.28)

$$\hat{b}_k = \text{sgn}((\mathbf{S}^{-1}(z)\mathbf{y})_k), \tag{3.25}$$

with

$$\mathbf{S}^{-1}(z) = [\mathbf{R}^T[1]z + \mathbf{R}[0] + \mathbf{R}[1]z^{-1}]^{-1}. \tag{3.26}$$

From these equations it can be seen that the decorrelating detector requires knowledge of the following received signal- and system parameters: the timing of the desired user, the timing of the interfering users and the signature waveforms of the desired and the interfering users.

The required number of operations needed for detection of a user increases linearly with the number of users K, therefor the time complexity per bit of the decorrelating detector is $O(K)$ for the synchronous as well as the asynchronous case. It can be shown [15] that the decorrelating detector can be implemented as a matched filter detector with modified filter coefficients. The computational complexity of that implementation is identical to that of the conventional detector. A problem of both implementations of the decorrelating detector is that if the crosscorrelations have to be calculated repeatedly, for example due to varying channel conditions, not only they have to be generated from the replicas of the received signature waveforms, but a matrix inversion of the correlation matrix has to be performed as well. Using Gaussian elimination this has a computational complexity of $O(K^3)$. A similar problem arises when long codes are used, because than the signature waveforms vary for each transmitted bit, also causing repeated calculation of the crosscorrelations and the inverse of the crosscorrelation matrix.

If the normalized crosscorrelations among all the signature waveforms are very small, the processing of the matched filter outputs with the matrix $\mathbf{R}^{-1}$ can be approximated by processing the matched filter outputs with $2\mathbf{I} - \mathbf{R}$. This detector is called the *approximate decorrelator*. An analog approximation can be used in the asynchronous case. If the crosscorrelations have to be calculated repeatedly, this approximation has the advantage that it does not require any processing of the crosscorrelations supplied by the crosscorrelators of the replicas of the signature waveforms. The approximate decorrelator is particularly advantageous in those asynchronous CDMA channels where the period of the signature waveform is much longer than the symbol period. In those cases, the crosscorrelations keep changing and it is cumbersome to repeatedly perform the matrix inversion required by the decorrelating detector.

The probability of error of the decorrelating detector for synchronous CDMA can be expressed as (equation 5.42 in Verdú):

$$\mathsf{P}_k^d(\sigma) = Q\left(\frac{A_k}{\sigma\sqrt{R_{kk}^+}}\right), \tag{3.27}$$

where $+$ is used as a shorthand for the inverse of a matrix, so R_{kk}^+ is a shorthand for element kk of the $\mathbf{R}^{-1}$ matrix, $(\mathbf{R}^{-1})_{kk}$ and the superscript d is used to indicate the decorrelating detector. The (asymptotic) multiuser efficiency of the decorrelating detector for synchronous CDMA is equal to

$$\eta_k^d = \frac{1}{R_{kk}^+}. \tag{3.28}$$

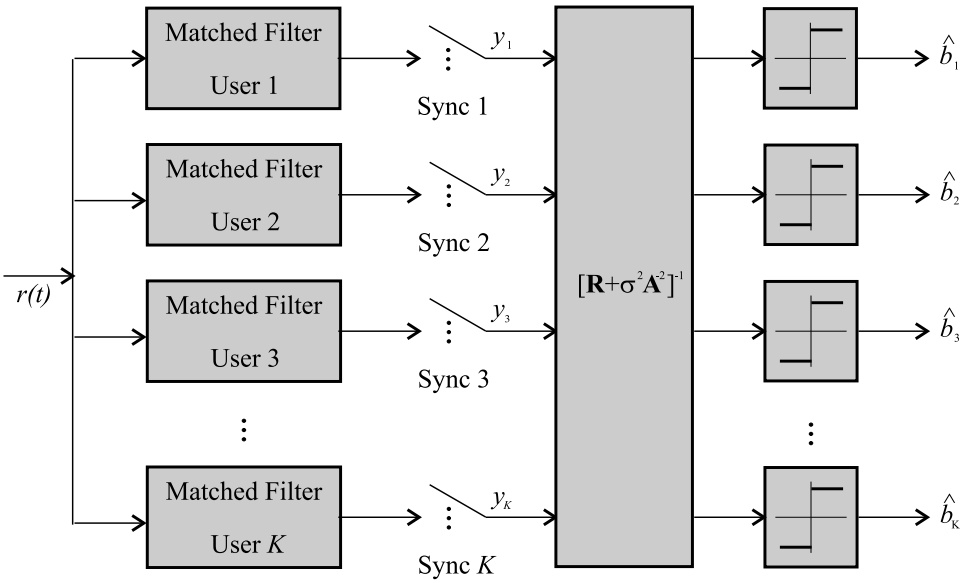


Figure 3.5: MMSE detector.

The bit-error-rate of the asynchronous decorrelating detector is equal to (equation 5.57 in Verdú)

$$\mathsf{P}_k^d(\sigma) = Q\left(\frac{A_k\sqrt{\eta_k^d}}{\sigma}\right), \tag{3.29}$$

where η_k^d is the (asymptotic) multiuser efficiency of the asynchronous decorrelating detector that is equal to

$$\eta_k^d = \left(\frac{1}{2\pi}\int_{-\pi}^{\pi}[\mathbf{R}^T[1]e^{j\omega} + \mathbf{R}[0] + \mathbf{R}[1]e^{-jw}]_{kk}^{+}d\omega\right)^{-1}. \tag{3.30}$$

3.5 Minimum Mean Square Error Detector

The minimum mean square error (MMSE) detector consists of a bank of matched filters followed by a linear transformation, see Figure 3.5, that maximizes the signal-to-interference ratio at the output of the linear transformation. For the synchronous case it can be shown [15] that this transformation has to be equal to:

$$\hat{b}_k = \mathrm{sgn}((\mathbf{My})_k) \quad \text{with} \quad \mathbf{M} = (\mathbf{R} + \sigma^2\mathbf{A}^{-2})^{-1}. \tag{3.31}$$

where

$$\sigma^2\mathbf{A}^{-2} = \mathrm{diag}\left\{\frac{\sigma^2}{A_1^2}, \ldots, \frac{\sigma^2}{A_K^2}\right\}. \tag{3.32}$$

For the asynchronous case, the MMSE linear detector is a K-input, K-output, linear time-invariant filter with transfer function

$$[\mathbf{R}^T[1]z + \mathbf{R}[0] + \sigma^2\mathbf{A}^{-2} + \mathbf{R}[1]z^{-1}]^{-1}. \tag{3.33}$$

The MMSE detector can be seen as a compromise solution between the conventional detector and the decorrelating detector. The conventional detector is optimized to fight the background noise exclusively, whereas the decorrelating detector eliminates the multiaccess interference disregarding the background noise. In contrast, the MMSE detector takes into account the relative importance of each interfering user and the background noise.

To construct the linear transformation the MMSE detector requires knowledge of the following received signal- and system parameters: the timing of the desired user, the timing of the interfering users, the noise level at the receiver, the received amplitudes and the signature waveforms of the desired as well as the interfering users.

Since the MMSE detector performs a similar transformation as the decorrelating detector its time complexity per bit is also $O(K)$ for the synchronous case as well as the asynchronous case. Just as the decorrelating detector, the MMSE detector can also be implemented as a matched filter detector with modified filter coefficients. The computational complexity of that implementation is identical to that of the conventional detector. The MMSE detector also has the disadvantage that it requires a matrix inversion, which has to be performed again and again when the crosscorrelations change because the channel conditions vary or because long codes are used.

The probability of error of the MMSE detector for synchronous CDMA is equal to (equation 3.124 in Verdú)

$$\begin{aligned}\mathsf{P}_k^m(\sigma) = 2^{1-K} \textstyle\sum_{e_1\in\{-1,1\}} \cdots \sum_{e_j\in\{-1,1\}, j\neq k} \cdots \sum_{e_K\in\{-1,1\}} \\ Q\left(\frac{A_k}{\sigma}\frac{(\mathbf{MR})_{kk}}{\sqrt{(\mathbf{MRM})_{kk}}}\left(1+\textstyle\sum_{j\neq k}\beta_j b_j\right)\right),\end{aligned} \tag{3.34}$$

where,

$$B_j = A_k(\mathbf{MR})_{kj}, \tag{3.35}$$

and

$$\beta_j = \frac{B_j}{B_k}. \tag{3.36}$$

The superscript m is used to indicate the MMSE detector. Equation 3.34 can also be used for asynchronous systems when an asynchronous system is viewed as a special case of a synchronous system as discussed in section 2.2.2. Of course the dimensions of the matrices in (3.34) change, since the equivalent synchronous system has $(2M+1)K$ fictitious users.

The MMSE detector converges to the decorrelating detector as $\sigma \to 0$ [15]. Therefor the asymptotic multiuser efficiency is identical to that of the

decorrelator:

$$\eta_k^m = \frac{1}{R_{kk}^+}. \tag{3.37}$$

in the synchronous case, and

$$\eta_k^m = \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} [\mathbf{R}^T[1]e^{j\omega} + \mathbf{R}[0] + \mathbf{R}[1]e^{-jw}]_{kk}^+ d\omega \right)^{-1}. \tag{3.38}$$

in the asynchronous case.

Compared to the decorrelating detector the MMSE will therefor only offer a slightly increased performance in low signal-to-noise ratio channels. In exchange for that slight increase in performance, the MMSE detector requires knowledge of the noise level at the receiver, which increases the complexity of the receiver. The main advantage of the MMSE detector is the ease with which it can be implemented adaptively. In literature, two types of the adaptive MMSE detector are described [15].

The first type of adaptive MMSE detectors uses *training sequences*. A training sequence is a string of data known to the receiver. When the *adaptive MMSE detector* receives the training sequence it uses an adaptive law to adjust its linear transformation. By doing so, the receiver learns the impulse response of the channel. If the impulse response of the channel changes over time, training sequences can be sent periodically to readjust the receiver. It can be shown [15] that the solution to which the adaptation law converges is the MMSE detector. The only knowledge needed by the adaptive MMSE detector is the timing of the desired user and the training sequence. When the adaptive MMSE detector is implemented as a matched filter detector with modified matched filter coefficients its time complexity per bit is constant, just as that of the conventional detector. Of course the matched filter coefficients still have to be calculated when the training sequence is received, but this requires far less operations than calculating and inverting the transformation matrix. Implementation of the adaptive MMSE detector for long code CDMA systems is not possible because in those systems the signature waveforms change for each bit, which would require adaptation of the detector for each bit and thus transmission of the training sequence for each bit. This causes of course way to much overhead for practical implementation.

The second type of adaptive MMSE detector is referred to as the *blind adaptive MMSE detector*, or the *minimum mean output energy (MMOE) detector*. This detector minimizes the variance at the output of the receiver with respect to the component orthogonal to the desired user's signature waveform, using a linear transformation in canonical form. In order to do so the detector requires knowledge of the timing and the signature waveform of the desired user. It can be shown that the solution to this problem is the MMSE linear transformation [15]. The blind adaptive MMSE detector can also be implemented as a matched filter detector with modified filter

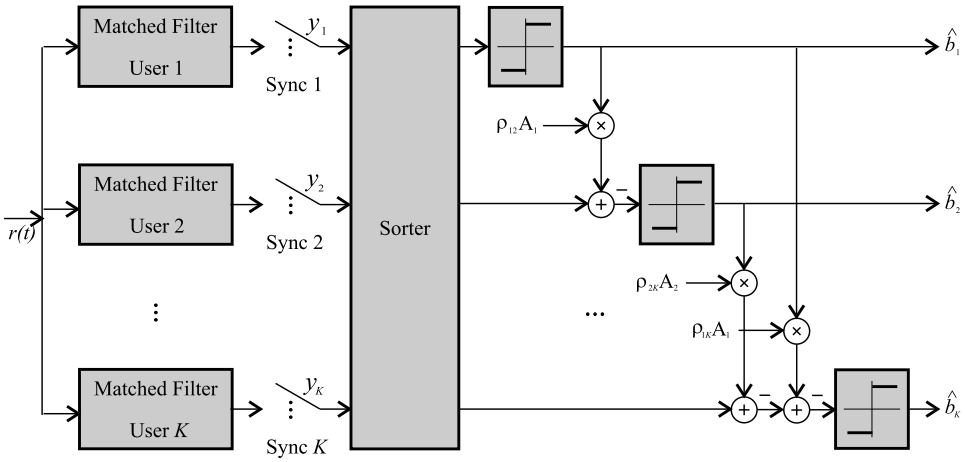


Figure 3.6: SIC detector.

coefficients, resulting in a constant time complexity per bit. Unfortunately the blind adaptive MMSE detector does not work for long code systems, because for each bit it would have to adapt to the section of the signature waveform that is used to modulate that bit.

3.6 Successive Interference Cancellation Detector

The successive interference cancellation (SIC) detector is a detector that belongs to the class of decision-driven detectors. The idea behind decision-driven detectors is simple: if a decision has been made about an interfering user's bit, then that interfering signal can be subtracted from the received waveform. This will cancel the interfering signal, provided that the decision was correct; otherwise it will double the contribution of the interferer.

The simplest form of the SIC detector consists of a bank of matched filters followed by a sorter, see Figure 3.6. In a lot of implementations the sorter sorts the signals of the different users in order of decreasing received energies. However, this is not necessarily best since it fails to take into account the crosscorrelations among users. A sensible alternative is to order users according to the signal-to-noise ratios computable using

$$E\left[\left(\int_0^T r(t)s_k(t)dt\right)^2\right] = \sigma^2 + A_k^2 + \sum_{j\neq k} A_j^2\rho_{jk}^2, \tag{3.39}$$

which can be estimated easily from the matched filter outputs. Interference cancellation for a user k is achieved by subtracting from the received signal of user k an estimate of the multiple access interference generated by users with higher received energies or higher signal-to-noise ratios than that of

user k. For the synchronous case this results in the following decision rule

$$\hat{b}_k = \operatorname{sgn}\left(y_k - \sum_{j=1}^{k-1} A_j \rho_{jk} \hat{b}_j \right), \tag{3.40}$$

where it is assumed that the users are numbered in order of decreasing received energies or signal-to-noise ratios, so the user with the highest received energy or signal-to-noise ratio has the lowest number.

Up to now it was assumed that the users in the asynchronous channel are numbered so that their offsets are increasing, and, thus, the jth user's bit that overlaps with $b_k[i]$ on the right side is $b_j[i]$ if $j > k$. However, as already indicated for the synchronous case, for successive interference cancellation it is convenient to number the users in the (inverse) order they are cancelled, which is, normally, dictated by their relative received powers or matched filter outputs. So in the asynchronous case it is also assumed that the users are numbered in order of decreasing received energies or signal-to-noise ratios, so the user with the highest received energy or signal-to-noise ratio has the lowest number. In order to still be able describe the successive interference cancellation detector in the asynchronous case it is useful to introduce the following notation:

$$\delta_{kj} = 1\{\tau_k < \tau_j\}, \tag{3.41}$$

so δ_{kj} equals 1 when $\tau_k < \tau_j$ and 0 otherwise. Then $b_k[i]$ overlaps on the right side with $b_j[i - \delta_{kj} + 1]$ and overlaps on the left side with $b_j[i - \delta_{kj}]$. It is now possible to generalize (3.40) to

$$\hat{b}_k[i] = \operatorname{sgn}\left(y_k[i] - \sum_{j=1}^{k-1} A_j (\rho_{jk} \hat{b}_j[i - \delta_{kj}] + \rho_{kj} \hat{b}_j[i - \delta_{kj} + 1]) \right). \tag{3.42}$$

The SIC detector tries to remove the multiaccess interference. Therefor it is best compared to the decorrelating detector. But whereas the decorrelating detector truly removes the multiple access interference, the SIC detector may actually double the interference. Once such an error is made in the SIC detector, it is likely to accumulate more errors. Furthermore, the demodulation of the first user will never be done in absence of multiple access interference. Especially in the case of a large number of users, this may decrease the performance rapidly. For the same reason, the SIC detector operates best in an unbalanced power environment. Thus if the received powers of the users vary heavily, the SIC detector has a better probability of demodulating the users correctly. Another problem of the successive interference cancellation detector is that the signals of the different users are demodulated sequentially, which causes the delay in demodulating the signals to grow with the number of users.

The following knowledge about the received signal- and system parameters is required at the receiver by the SIC detector: the timing of the desired user, the timing of the interfering users, the received amplitudes and the signature waveforms of the desired and interfering users.

To detect a user the successive interference cancellation detector requires a number of operations that increases linearly with the number of users K. Therefor the time complexity per bit of the SIC detector is $O(K)$. The decision rule (3.40) indicates that the SIC detector requires the crosscorrelations between the signature waveforms. This causes some problems when the SIC detector has to be used in long code CDMA systems, because in that case the crosscorrelations will have to be calculated for each transmitted bit. Fortunately this operation is a lot less complex than the matrix inversion that would be required for each bit in the long code version of the decorrelating or MMSE detector, or the adaptation that would have to be performed for each bit in the blind adaptive MMSE detector. Therefor using the SIC detector is still feasible for long code CDMA systems.

Using (2.11) and (3.40) the following expanded decision rule can be obtained

$$\hat{b}_k = \operatorname{sgn}\left(A_k b_k + n_k + \sum_{j=1}^{k-1} A_j \rho_{jk} (b_j - \hat{b}_j) + \sum_{j=k+1}^{K} A_j \rho_{jk} b_j \right). \qquad (3.43)$$

Since the cancellation residuals depend on the other random variables inside the sign function, an analytical exact evaluation of bit-error-rate is difficult and only possible for very small numbers of users K. Unfortunately it is not possible to obtain an analytical expression for the multiuser efficiency either. Therefor performance measures for the successive interference cancellation detector can only be obtained by simulations or trough analytical approximations.

3.7 Parallel Interference Cancellation Detector

The parallel interference cancellation (PIC) detector is a detector that also belongs to the class of decision-driven detectors. It alleviates the problem of the SIC detector that the performance of successive cancellation for a particular user is greatly affected by the order in which the users are cancelled.

The parallel interference cancellation detector is based on a technique that employs multiple iterations in detecting the data bits and cancelling the interference, therefor it is also referred to as multistage interference cancellation (MIC) detector. In its simplest form the parallel interference cancellation detector consists of two stages. The first stage consists of a conventional bank of matched filters. The second stage performs, for each user, reconstruction and subtraction of an estimate of the interference from

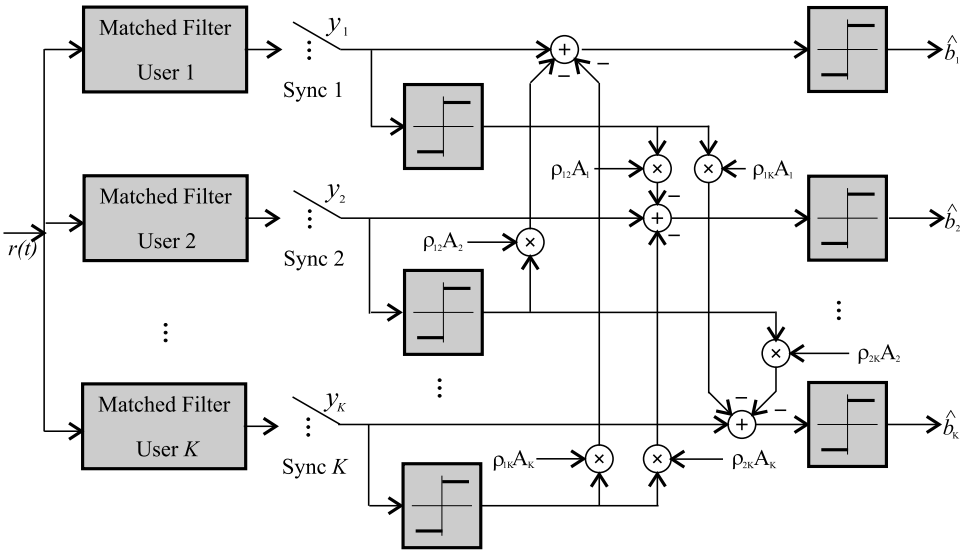


Figure 3.7: Two stage PIC detector

all other users, see Figure 3.7. So the second stage in fact performs successive cancellation for all users. This leads to the following decision rules:

$$\hat{b}_k = \operatorname{sgn}\left(y_k - \sum_{j \neq k} A_j \rho_{jk} \hat{b}_j \right) \tag{3.44}$$

for synchronous CDMA and

$$\begin{aligned} \hat{b}_k \quad = \quad & \operatorname{sgn}\left(y_k - \sum_{j<k} A_j (\rho_{kj} \hat{b}_j[i+1] + A_j \rho_{jk} \hat{b}_j[i]) \right. \\ & \left. - \sum_{j>k} A_j (\rho_{kj} \hat{b}_j[i] + A_j \rho_{jk} \hat{b}_j[i-1]) \right) \end{aligned} \tag{3.45}$$

for asynchronous CDMA.

The motivation for such a detector is given by comparing with the successive interference cancellation detector. In the SIC detector, the demodulation of user k is carried out in the presence of the multiple access interference of all users that have a weaker received power. After all, the SIC detector demodulates the received signals in order of decreasing received power. By using the PIC detector, the interference caused by all interfering users can be estimated trough the bank of matched filters and is then cancelled out. Another advantage of the PIC detector over the SIC detector is that the PIC detector demodulates the signals of the individual users in parallel, therefor

the delay in demodulating signals in the PIC detector does not increase with the number of users.

The parallel interference cancellation detector requires the knowledge of the same received signal- and system parameters at the receiver as the successive interference cancellation detector. So it requires information about: the timing of the desired user, the timing of the interfering users, the received amplitudes and the signature waveforms of the desired and interfering users. The time complexity per bit of the PIC detector is comparable to the time complexity per bit of the SIC detector, it also increases linearly with the number of users and is therefor $O(K)$. The PIC detector also requires the crosscorrelations between the signature waveforms, just as the SIC detector. This of course causes the problem that these crosscorrelations have to be calculated for each transmitted bit in long code CDMA systems. However, this is not such a complex operation that implementation of PIC detection in long code CDMA systems becomes infeasible. Obtaining analytical results for the bit-error-rate or asymptotic multiuser efficiency of the PIC detector is unfortunately not possible, the only way to obtain these performance measures is again trough simulation results.

A very useful property of the parallel interference cancellation detector is that it can be used for combining different detection techniques. For example, the first stage of the PIC detector can be replaced by a decorrelating detector, thus improving performance, but of course also increasing computational complexity. Another way of expanding the parallel interference cancellation detector is adding more stages. This will however not always increase performance and in some cases adding another stage may actually hurt performance.

3.8 Decorrelating Decision-Feedback Detector

In this section the decorrelating decision-feedback detector is discussed. Decision-feedback detectors are decision-driven multiuser detectors that combine several of the features of the decision-driven detectors that were discussed in the previous sections:

- As in successive cancellation, the intermediate decisions used in decision-feedback detectors are final (output) decisions.
- Decision-feedback detectors operate sequentially, demodulating one bit at a time.
- As in multistage detection with decorrelating first stage, both linear and nonlinear methods are used to combat multiuser interference.

The term *decision-feedback* is borrowed from an approach extensively used for the demodulation of single-user channels subject to intersymbol inter-

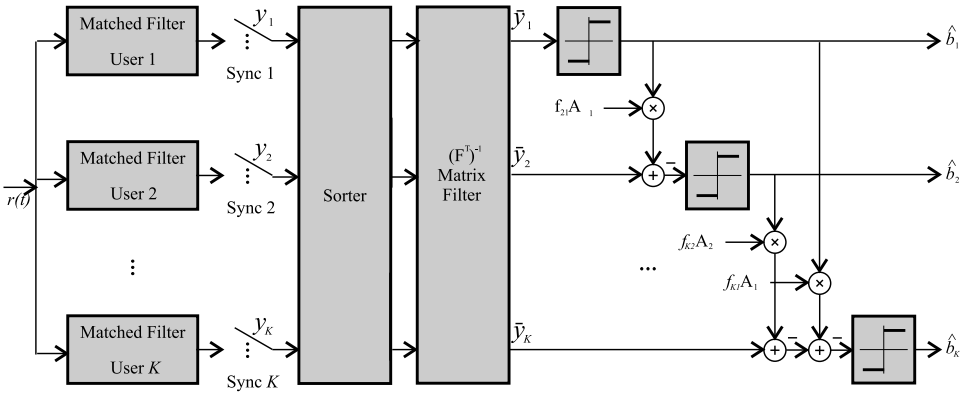


Figure 3.8: Synchronous decorrelating decision-feedback detector.

ference, whereby previous output ("final") decisions are used as a surrogate transmitted data to cancel intersymbol interference.

The synchronous decorrelating decision-feedback detector consists of a bank of matched filers followed by a sorter and two transformations, see Figure 3.8. The sorter sorts the signals of the different users in order of decreasing received energies. In order to obtain the two transformations a Cholesky decomposition $\mathbf{R} = \mathbf{F}^T\mathbf{F}$ ($\mathbf{F}$ is lower triangular) of the correlation matrix is used [15].

The first transformation is a forward filter $(\mathbf{F}^T)^{-1}$ that eliminates multiuser interference, yielding the whitened matched filter outputs

$$\bar{\mathbf{y}} = \mathbf{F}^{-T}\mathbf{y} = \mathbf{FAb} + \bar{\mathbf{n}}, \tag{3.46}$$

where $\bar{\mathbf{n}}$ is a Gaussian K-vector with independent components, each with variance σ^2. Since $\mathbf{F}$ is lower triangular, the first whitened matched filter output is

$$\bar{y}_1 = F_{11}A_1b_1 + \bar{n}_1. \tag{3.47}$$

Thus, $\bar{y}_1$ contains no interference from other users. For $k > 1$, $\bar{y}_k$ does contain interference from lower-numbered users:

$$\bar{y}_k = F_{kk}A_kb_k + \sum_{j=1}^{k-1} F_{kj}A_jb_j + \bar{n}_k. \tag{3.48}$$

Following the philosophy of successive cancellation the users can be demodulated sequentially by

$$\hat{b}_k = \operatorname{sgn}\left(\bar{y}_k - \sum_{j=1}^{k-1} F_{kj}A_j\hat{b}_j\right). \tag{3.49}$$

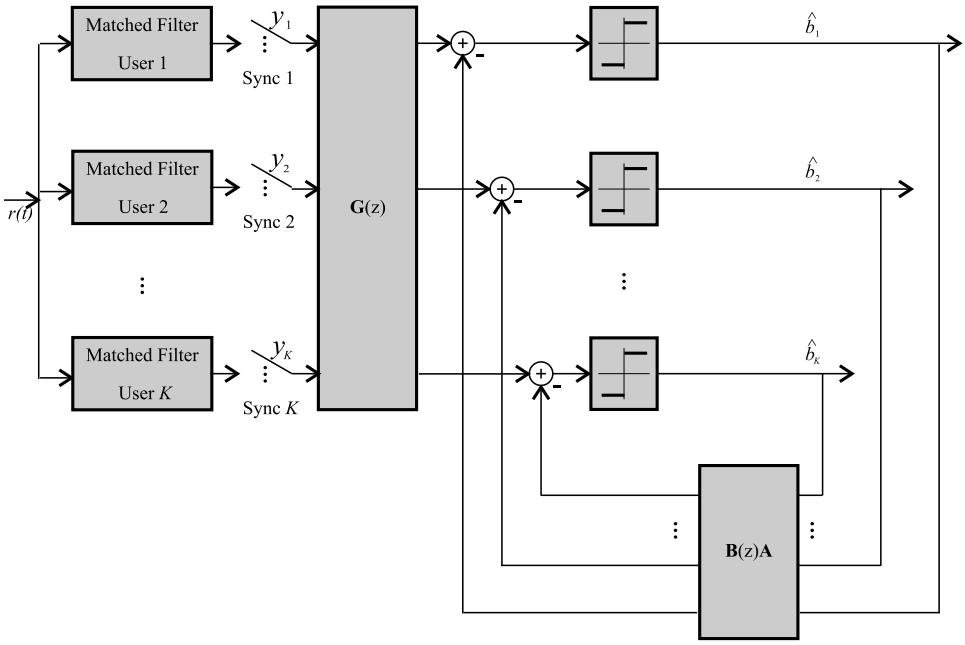


Figure 3.9: Asynchronous decorrelating decision-feedback detector.

The reconstruction of the multiaccess interference from the detected bits can be written in matrix form as $(\mathbf{F} - \text{diag}\{\mathbf{F}\})\mathbf{A}\hat{\mathbf{b}}$, which is the second transformation in Figure 3.8 that performs a feedback filter operation . Using this expression the decision rule 3.49 can be written in matrix notation as:

$$\hat{\mathbf{b}} = \text{sgn}(\mathbf{F}^{-T}\mathbf{y} - (\mathbf{F} - \text{diag}\{\mathbf{F}\})\mathbf{A}\hat{\mathbf{b}}). \tag{3.50}$$

For asynchronous CDMA the feed-forward and feedback filters of the decorrelating decision-feedback detector (figure 3.9) have $K \times K$ transfer functions [15]:

$$\mathbf{G}(z) = [\mathbf{F}[0] + \mathbf{F}[1]z]^{-T}, \tag{3.51}$$

$$\mathbf{B}(z) = \mathbf{F}[0] - \text{diag}\{\mathbf{F}\}[0] + \mathbf{F}[1]z^{-1}. \tag{3.52}$$

Where $\mathbf{F}[0]$ is a lower triangular matrix and $\mathbf{F}[1]$ is an upper triangular matrix with zero diagonal which can be found by factoring the discrete time channel transfer function (2.28) as

$$\mathbf{S}(z) = [\mathbf{F}[0] + \mathbf{F}[1]z]^{T}[\mathbf{F}[0] + \mathbf{F}[1]z^{-1}]. \tag{3.53}$$

The decorrelating decision-feedback requires the same knowledge about the received signal- and system parameters at the receiver as the other decision driven detectors. So it requires information about: the timing of the desired user, the timing of the interfering users, the received amplitudes and the signature waveforms of the desired and interfering users.

Implementation of the feed forward and feedback filtering of a decorrelating decision-feedback detector requires a number of operations that increases linearly with the number of users. This results in a time complexity per bit of $O(K)$. However the other operations involved in the decorrelating decision-feedback detector like sorting of the users according to their received energies, generation of the forward filter using a Cholesky decomposition and inversion of the resulting triangular matrix, make the decorrelating decision-feedback detector more computationally intensive than the other decision-driven detectors. The decorrelating decision-feedback detector also has the disadvantage that these operations have to be performed again and again when the channel conditions vary or when long codes are used, making implementation of the decorrelating decision-feedback detector less feasible in these systems.

The decorrelating decision-feedback detector is the only non-linear multiuser detector for which it is possible to obtain a closed-form expression for the probability of error for an arbitrary number of users and arbitrary cross-correlation matrices. This expression is equal to (equation 7.68 in Verdú)

$$\begin{aligned} \mathsf{P}_k^{ddf}(\sigma) &= 2^{1-k} \sum_{(b_1,\ldots,b_{k-1})\in\{-1,1\}^{k-1}} \sum_{(\hat{b}_1,\ldots,\hat{b}_{k-1})\in\{-1,1\}^{k-1}} \\ &\quad \pi_k(b_1-\hat{b}_1,\ldots,b_{k-1}-\hat{b}_{k-1}) \\ &\quad \times \prod_{j-1}^{k-1} [1\{b_j=\hat{b}_j\} + (1-2\ 1\{b_j=\hat{(b)}_j\}) \\ &\qquad \times \pi_j(b_1-\hat{b}_1,\ldots,b_{j-1}-\hat{b}_{j-1})], \end{aligned} \tag{3.54}$$

where

$$\pi_k(e_1,\ldots,e_{k-1}) = Q\left(F_{kk}\frac{A_k}{\sigma} + \sum_{l=1}^{k-1} F_{kl}\frac{A_l}{\sigma}e_l\right). \tag{3.55}$$

Here the superscript *ddf* is used to indicate the decorrelating decision feedback detector.

3.9 Comparison of Detection Techniques

An important characteristic of a detection technique is its bit-error-rate performance. However as became clear in the previous sections, it is unfortunately not always possible to derive analytical expressions for the bit-error-rate of a detector. Therefor a lot of research effort has been put into simulating multiuser detectors. A recent paper in this field is A Simulation Comparison of Multiuser Receivers for Cellular CDMA [2], by Buehrer and others. This paper compares all of the detection techniques described in the previous sections, with exception of the maximum likelihood sequence

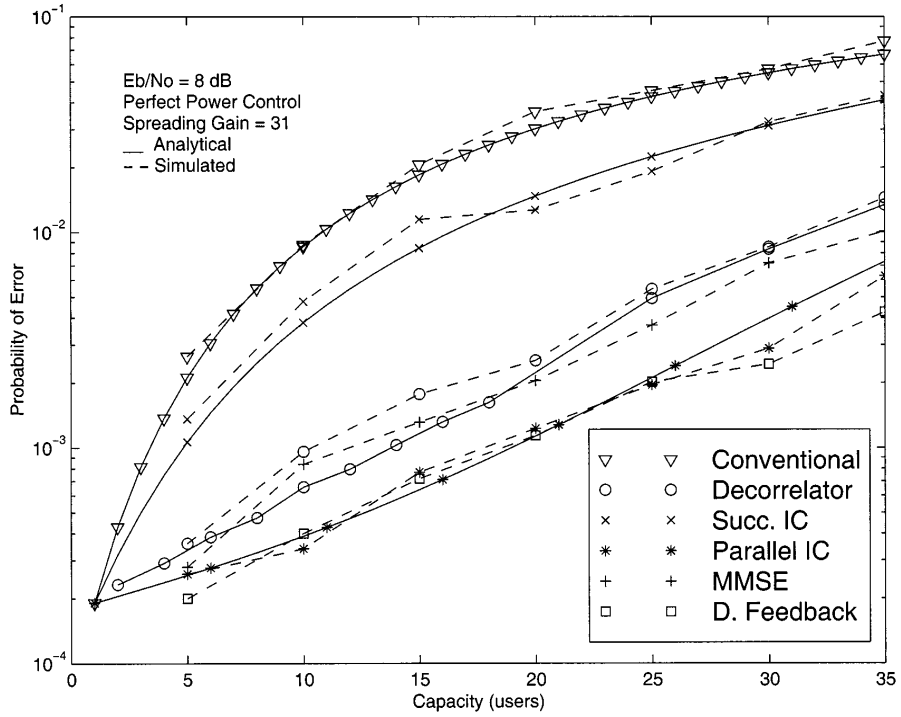


Figure 3.10: Capacity curves for perfect power control (E_b/N_0 = 8dB and processing gain = 31) [2] .

estimator (MLSE) detector, on the basis of common assumptions. The results published in this paper will be used here to compare the bit-error-rate performance of the different multiuser detectors.

Figure 3.10 contains a plot of the capacity curves taken from [2] for a CDMA system with signal-to-noise ratio E_b/N_0 = 8-dB, spreading gain $N = 31$, so 31 chips per symbol and perfect power control and therefor equal received energies. Where E_b is the energy of a bit equal to A_k^2 and N_0 is the one sided spectral level equal to $2\sigma^2$. The simulated Parallel IC detector has a conventional first stage followed by two cancellation stages.

The results show that the detectors can be divided into three performance groups. The first group contains two nonlinear detectors, the PIC detector and the decorrelating decision-feedback detector, these detectors can handle the most users for a given bit-error-rate. The second group contains two linear detectors, the decorrelating detector and the MMSE detector, which can handle slightly less users as the detectors in the first group for the same bit-error-rate. The third group contains the conventional detector and the SIC detector which can handle even less users as the detectors in the second group for the same bit-error-rate. The detectors in group one and two are of the most interest because they offer the highest performance.

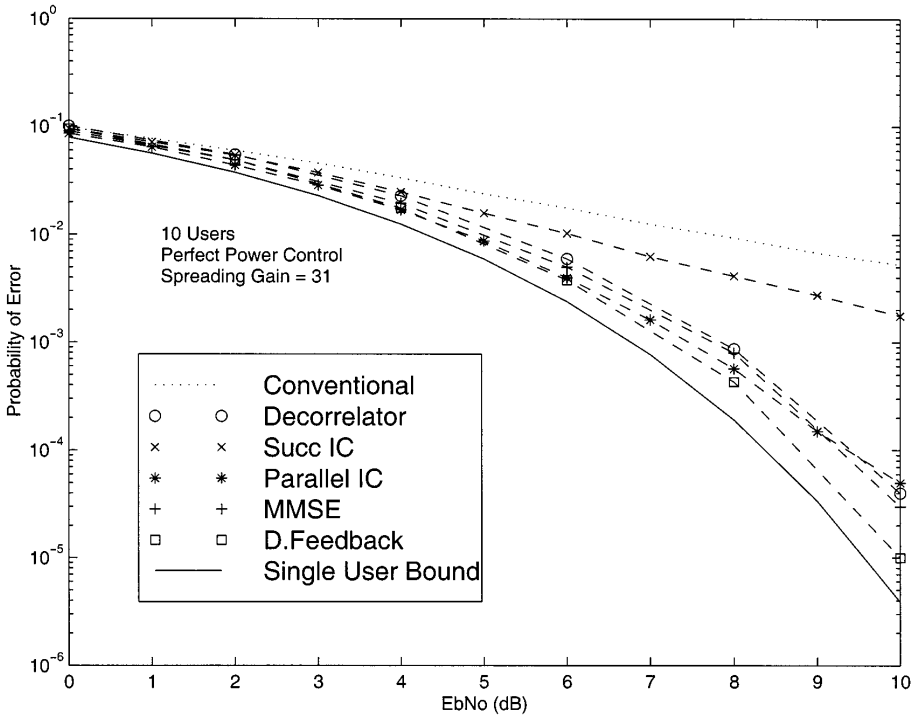


Figure 3.11: BER versus E_b/N_0 with perfect power control (ten users and processing gain = 31) [2].

The linear detectors in group two roughly support three times the number of users of the conventional detector for a given probability of error, while the two nonlinear detectors in group one support four times the number of users of the conventional detector for a given bit-error-rate. The performance of the successive interference cancellation (SIC) detector is significantly poorer due to lack of variance in the received signal powers.

Figure 3.11 shows the bit-error-rate performance plotted versus the signal-to-noise ratio E_b/N_0 for $K = 10$ users and perfect power control. Again the decorrelating detector, MMSE detector, PIC detector and decorrelating decision-feedback detector provide significant improvement over the conventional detector with each providing bit-error-rate improvements of over an order of a magnitude compared to the conventional detector at 10dB signal to noise ratio, while the SIC detector only provides a small improvement. In this case again the performance of the nonlinear PIC and decorrelating decision-feedback detectors is better than the performance of the linear decorrelating and MMSE detectors. Thus, in additive white gaussian noise channels nonlinear detectors provide superior performance compared to linear detectors although the difference is minor at low signal-to-noise ratio levels.

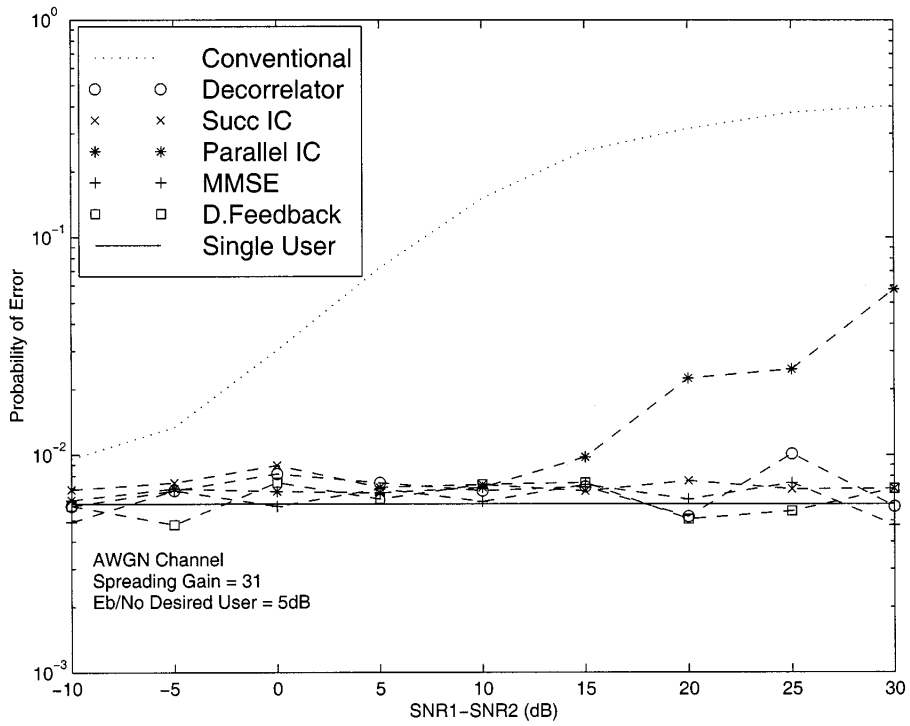


Figure 3.12: Performance degradation in near-far channels (E_b/N_0 = 5dB and processing gain = 31) [2].

Figure 3.12 presents the bit-error-rate performance of the receivers in the presence of two interfering users, one with equal power to the desired user and one with a power which varies from 10-dB below the desired user to 30-dB above the desired user. Therefor this figure will give an indication of the near-far resistance of the different detection techniques. As expected, the conventional detector degrades quickly in the presence of strong interference. The SIC detector and the decorrelating decision-feedback detector which benefit from diverse powers are found to be robust to strong interferers, as is the decorrelating detector which has a performance which is independent of the user energies. The MMSE detector having a theoretical near-far resistance identical to the decorrelating detector also displays robustness. The PIC detector is not as robust and shows slow degradation for high interference power. This is caused by the fact that estimation of the amplitude of the weak user is inaccurate in the first stage of cancellation due to the dominating interference, resulting in inaccurate cancellation of the weak user's signal from the strong users signal. This results in an unreliable estimate of the signal of the strong user, causing problems when the strong user's signal is cancelled from the weak user's signal in the second stage. One way of improving the parallel interference cancellation receiver in such situations would be to avoid cancelling the weak user since its amplitude estimation is unreliable.

Asymptotic multiuser efficiency plots for multiuser detectors are found less frequently in literature. Fortunately the asymptotic multiuser efficiency can, at least for 2 user systems, be determined analytically for all discussed multiuser detectors [15].

Figure 3.13 contains a plot of the analytical asymptotic multiuser efficiency of user 1 of all the discussed detectors for a 2 user synchronous system with a relatively high crosscorrelation of 0.6. This plot shows that, as expected, the MLSE detector has the best asymptotic multiuser efficiency performance. The performance of the conventional detector decreases rapidly when the amplitude of the interferer increases, showing that the conventional detector is really bad at combatting multiple access interference. The decorrelating and MMSE detector have equal asymptotic multiuser efficiency performance because the MMSE detector converges to the decorrelating detector for high signal-to-noise ratios. In this particular case the performance of both these detectors is relatively bad because of the high crosscorrelations. The two-user asymptotic multiuser efficiency of the two stage PIC detector is equal to that of the SIC detector in this particular situation, since the decisions for user 1 are identical for both detectors. The figure clearly shows that the performance of these detectors is worse than that of the conventional detector for small interferer energies, whereas if the received interferer energy is sufficiently high, then the asymptotic efficiency of these detectors equals 1. This is caused by the fact that the decisions made about the transmitted bit of the interferer are inaccurate when the interferers received

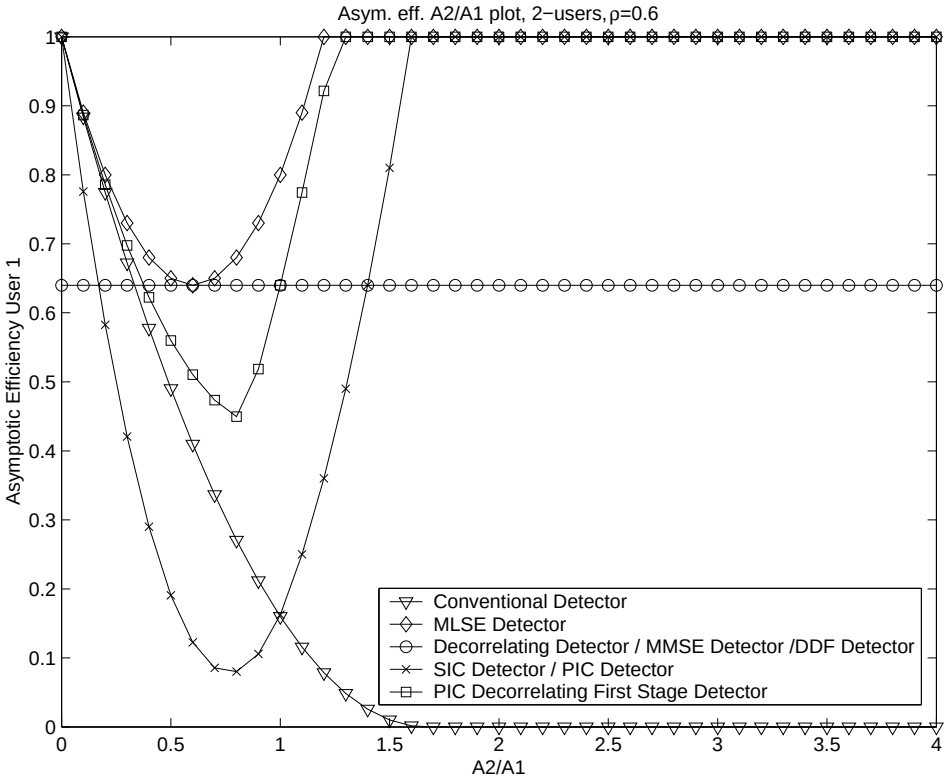


Figure 3.13: Asymptotic efficiency for a 2 user synchronous system with $\rho = 0.6$.

	Conv. Detector	MLSE	Decorr. Detector	MMSE	Adapt. MMSE	Blind MMSE	Decision Driven
Signature waveform desired user	•	•	•	•		•	•
Timing desired user	•	•	•	•	•	•	•
Received amplitudes		•		•			•
Noise level				•			
Signature waveforms interfering users		•	•	•			•
Timing interfering users		•	•	•			•
Training sequence desired user					•		
Synchronous computational complexity	constant	$O(2^K)$	$O(K)$	$O(K)$	$O(K)$	$O(K)$	$O(K)$
Asynchronous computational complexity	constant	$O(2^K)$	$O(K^3)$	$O(K^3)$	$O(K)$	$O(K)$	$O(K)^1$ $O(K^3)^2$

[1]For the SIC/MIC detector
[2]For the DDF detector

Table 3.1: Summary of requirements and computational complexity of multiuser detection techniques.

energy is small, resulting in inaccurate cancellation of the interferers signal. When the received energy of the interferer increases, the decisions about the interferers transmitted bit become more reliable and the accuracy of the cancellation improves. The asymptotic multiuser efficiency for user 1 of the two-user decorrelating decision feedback (DDF) detector is equal to that of the decorrelating detector. For user 2 the asymptotic multiuser efficiency of the decorrelating decision feedback detector is equal to that of the multistage interference cancellation detector with decorrelating first stage. Therefor this asymptotic multiuser efficiency is plotted in the figure as well. The plot shows that on average the decorrelating decision feedback detector will perform better in high interference situations than the decorrelating as well as the PIC detector.

Table 3.1 summarizes the the received signal- and system parameters that have to be estimated or known at the receiver and the computational complexity of the different detection techniques that have been discussed in this chapter. All the discussed decision driven detectors (SIC, PIC and

Decorrelating Decision Feedback) are summarized in the decision driven column, since they all require the same information about the received signal- and system parameters at the receiver and have the same computational complexity.

A detector that requires knowledge of more received signal- and system parameters at the receiver will increase the complexity of the receiver, because the information will have to be extracted from the received signal. The table shows that all multiuser detection techniques require knowledge about more received signal- and system parameters at the receiver than the conventional detector and thus require a more complex receiver, except for the adaptive and blind MMSE detectors. The adaptive MMSE detector however uses training sequences whose transmission requires bandwidth.

All detectors in the table have higher computational complexity requirements than the conventional detector. The MLSE detector has a computational complexity that increases exponentially in the number of users, which will make practical implementation impossible for larger numbers of users. The linear dependency in the number of users of the computational complexity of the decorrelating and MMSE detector, is only valid for the synchronous case in short code CDMA systems. In the asynchronous case or in long code CDMA systems the inverse of the transformation matrix has to be calculated repeatedly, this is an $O(K^3)$ operation. The main advantage of the adaptive MMSE and blind MMSE detector over the normal MMSE detector therefor is that these detectors have a computational complexity linear in the number of users, also for the asynchronous case, while they achieve similar bit-error-rate performance as the normal MMSE detector. The table indicates that the decision driven detectors all have a linear computational complexity. For the SIC detector and PIC detector this is true for the synchronous as well as the asynchronous case. These detectors do require more operations for long code CDMA systems, but implementation remains possible for these systems. The decorrelating decision-feedback detector however requires a matrix inversion, which has to be performed repeatedly in the asynchronous case or in long code CDMA systems, causing a strongly increased computational complexity.

Looking at the performance and the requirements of the different multiuser detection techniques summarized in this section, some detection techniques show properties that make them very attractive for further research. The blind adaptive MMSE detector requires knowledge of the same parameters of the received signal as the conventional detector, while its bit-error-rate performance will equal that of the MMSE detector, provided that the adaptation algorithm has converged. The only disadvantage of the blind adaptive MMSE detector is that it cannot be implemented for long code CDMA systems. Of the decision-driven detectors the parallel interference cancellation (PIC) detector shows good bit-error-rate performance results, while it does not require the complex transformations needed by the decor-

relating decision-feedback detector. The PIC detector can also be implemented for long code CDMA systems with a relatively low increase in computational complexity. Unfortunately the PIC detector is not as near-far resistent as some of the other detectors, but this can be improved by avoiding cancellation of weak users or by implementing some form of power control. This last option of course removes one of the advantages of using multiuser detection, operation without power control, but the improved bit-error-rate performance compared to the conventional detector remains.

3.10 Conclusions

- The MMSE detector has good bit-error-rate performance compared to the conventional detector, but requires knowledge of a lot of received signal parameters.
- The blind adaptive MMSE detector, when the adaptation algorithm has converged, will have the same bit-error-rate performance as the MMSE detector, but only requires knowledge of the same received signal parameters as the conventional detector.
- The blind adaptive MMSE detector cannot be implemented for long code CDMA systems.
- The PIC detector has good bit-error-rate performance compared to the conventional detector in non near-far systems at the expense of increased required knowledge of received signal parameters and increased computational complexity.
- The PIC detector can be implemented for long code CDMA systems without a large increase in computational complexity.

Because of these properties the blind adaptive MMSE detector and PIC detector are the most promising for implementation and are therefor studied in the rest of this thesis, along with the conventional detector for comparison purposes. This study will start with a more detailed analysis of the blind adaptive MMSE and PIC detector in the next chapter.

Chapter 4

Blind Adaptive MMSE and PIC Detector

In the previous chapter the blind adaptive minimum mean square error and parallel interference cancellation detector were chosen as the most promising detectors for implementation. The description of the blind adaptive MMSE detector in that chapter however does not indicate how this detector can be implemented. Therefor in this chapter an algorithm for implementation of the blind adaptive MMSE detector will be derived. Chapter 3 briefly described one implementation for the PIC detector. In this chapter that implementation and another implementation for the PIC detector will be described in more detail. This chapter also indicates how the received amplitudes of the individual users, that are needed by the PIC detector, can be estimated from the received signal. Since in simulation and implementation, for which the study of the detectors in this chapter is meant, only synchronous CDMA systems are considered, the analysis in this chapter is also limited to the synchronous case. Extension to asynchronous systems should however be straightforward.

4.1 Blind Adaptive MMSE Detector

In this section the blind adaptive MMSE detector will be mathematically analyzed in more detail. This analysis is mostly taken from Hans Roelofs' Masters Thesis [11]. First a general notation for linear multiuser detectors, the class of multiuser detectors to which the blind adaptive MMSE detector belongs, will be developed. After that it will be shown that the blind adaptive MMSE detector, which minimizes the mean output energy, also minimizes the mean square error. Finally, an adaptive algorithm will be derived for the implementation of the blind adaptive MMSE detector. For convenience in this section it will be assumed that the desired user is user 1, but the same reasoning can of course be applied to all users in the system.

4.1.1 Linear Multiuser Detectors

The blind adaptive MMSE detector is an example of a *linear multiuser detector*. Linear multiuser detectors apply a linear transformation to the outputs of the matched filter bank to produce a new set of outputs, which hopefully provide better performance when used for estimation. Other examples of linear multiuser detectors that were described in Chapter 3 are the conventional, decorrelating and MMSE detector. In this section a common notation for these linear detectors will be developed.

Since matched filtering is also a linear operation, the matched filter bank followed by a linear transformation used in linear multiuser detection can be seen as a matched filter bank with modified sequences. So the signature sequence s is replaced by a modified signature sequence c. A linear multiuser detector for user 1 can be characterized by the modified sequence c_1, which is the sum of two orthogonal components. One of these components is the signature sequence of user 1, s_1. The other component is denoted as x_1 and will be referred to as the *x sequence*, so

$$c_1 = s_1 + x_1, \tag{4.1}$$

with $c_1, s_1, x_1 \in \mathbb{R}^N$, where N is the number of chips per symbol and

$$\langle s_1, x_1 \rangle = 0. \tag{4.2}$$

Since x_1 is orthogonal to s_1, any x_1 can be chosen to minimize the correlation between the multiple access interference and c_1, while the correlation with user 1 remains constant. Thus

$$\langle s_1, c_1 \rangle = ||s_1||^2 \triangleq 1. \tag{4.3}$$

The linear detector makes its decision for user 1 based on the sign of the output of the matched filter for user 1 with modified sequence, so

$$\hat{b}_1 = \text{sgn}(\langle r, c_1 \rangle). \tag{4.4}$$

Every linear multiuser detector can be written in this form, so it is a *canonical representation* [15] for linear multiuser detectors. For a conventional detector the x_1 component of the modified signature sequence is zero, so $c_1 = s_1$ and the notation used in Chapter 2 returns. For the decorrelating detector in addition to being orthogonal to s_1, x_1, as a result of the decorrelating property, must satisfy

$$\langle s_k, x_1 \rangle = -\rho_{1k} \tag{4.5}$$

for all $K - 1$ other users k in the system. It can be shown [15] that the MMSE detector can also be expressed using this canonical representation.

The output of the matched filter with modified sequence for user 1 is equal to

$$\begin{aligned} y_1 &= \langle r, c_1 \rangle \\ &= \langle r, s_1 + x_1 \rangle \\ &= \langle A_1 b_1 s_1 + \sum_{k=2}^{K} A_k b_k s_k + \sigma n, s_1 + x_1 \rangle \\ &= A_1 b_1 \langle s_1, s_1 + x_1 \rangle + \sum_{k=2}^{K} A_k b_k \langle s_k, s_1 + x_1 \rangle + \sigma \langle n, s_1 + x_1 \rangle. \end{aligned} \quad (4.6)$$

Using (4.3) and the fact that $\langle s_1, s_k \rangle = \rho_{1k}$ gives

$$y_1 = \langle r, c_1 \rangle = A_1 b_1 + \sum_{k=2}^{K} A_k b_k (\rho_{1k} + \langle s_k, x_1 \rangle) + \sigma \langle n, s_1 + x_1 \rangle. \quad (4.7)$$

With $\tilde{n}_1$ as a shorthand notation for $\langle n, s_1 + x_1 \rangle$ this can also be written as

$$y_1 = \langle r, c_1 \rangle = A_1 b_1 + \sum_{k=2}^{K} A_k b_k (\rho_{1k} + \langle s_k, x_1 \rangle) + \sigma \tilde{n}_1. \quad (4.8)$$

The bit-error-rate of the linear detector for user 1 can be easily derived from the probability of error of the conventional detector (3.12) and is equal to [6]:

$$\begin{aligned} \mathsf{P}_1 &= \frac{1}{2^{K-1}} \sum_{e_2 \in \{-1,1\}} \cdots \sum_{e_K \in \{-1,1\}} \\ &\quad Q\left(\frac{A_1 \langle s_1, c_1 \rangle + \sum_{k=2}^{K} A_k e_k \langle s_k, c_1 \rangle}{\sigma \|c_1\|} \right), \end{aligned} \quad (4.9)$$

where e_k represents the bit value for user k, which is equally likely to be +1 or -1.

4.1.2 Minimizing Mean Output Energy

The blind adaptive MMSE detector in fact minimizes the mean output energy (MOE), as was stated in section 3.5. In this section it will be shown that by minimizing the mean output energy, the mean square error (MSE) is also minimized.

The *mean output energy* of a linear multiuser detector for user 1 is defined as:

$$\text{MOE} \triangleq E[(\langle r, c_1 \rangle)^2] \quad (4.10)$$

The trivial solution to minimizing this equation is setting $c_1 = 0$. However, since c_1 is defined as the sum of s_1 and x_1, this solution is eliminated.

It can be expected intuitively that minimizing the output energy of the linear detector is a sensible approach. This is because the energy at the output of the detector can be written as the sum of the energy due to the desired signal plus the energy due to the interference (background noise and multiple access interference). Any x_1, as long as x_1 is orthogonal to s_1, can be chosen to minimize the interference, but it will not influence the energy of the desired signal. The x_1 that minimizes the mean output energy also minimizes the mean square error as the following reasoning shows. The mean output energy and the *mean square error* of the linear detector for user 1 can be written as, respectively,

$$\mathrm{MOE}(x_1) = E[(\langle r, s_1 + x_1\rangle)^2] \tag{4.11}$$

and

$$\begin{aligned} \mathrm{MSE}(x_1) &= E[(A_1 b_1 - \langle r, s_1 + x_1\rangle)^2] \\ &= E[(A_1 b_1)^2 - 2A_1 b_1 \langle r, s_1 + x_1\rangle + \\ &\quad (\langle r, s_1 + x_1\rangle)^2]. \end{aligned} \tag{4.12}$$

Then the fact is used that the received signal r can be decomposed in a desired signal portion $A_1 b_1 s_1$ and a residue term R. The residue consists of multiple access interference and white Gaussian noise.

$$\begin{aligned} \mathrm{MSE}(x_1) &= E[(A_1 b_1)^2] - E[2A_1 b_1 \langle A_1 b_1 s_1 + R, s_1 + x_1\rangle] + \\ &\quad E[(\langle r, s_1 + x_1\rangle)^2]. \end{aligned} \tag{4.13}$$

The last term of equation 4.13 is equal to the mean output energy $\mathrm{MOE}(x_1)$. Since $b_1 \in \{-1, +1\}$, $(A_1 b_1)^2 = {A_1}^2$ and $E[(A_1 b_1)^2] = {A_1}^2$, equation 4.13 can be written as

$$\begin{aligned} \mathrm{MSE}(x_1) &= {A_1}^2 - E[2(A_1 b_1)^2 \langle s_1, s_1 + x_1\rangle] - \\ &\quad E[2A_1 b_1 \langle R, s_1 + x_1\rangle] + \mathrm{MOE}(x_1) \end{aligned} \tag{4.14}$$

Further it is assumed that bit b_1 is independent from the other bits $b_k, k \neq 1$ and that b_1 is independent the white Gaussian noise. From these two assumptions, it follows that b_1 is uncorrelated with the multiple access interference. Finally, by also using the obvious fact that b_1 is independent from c_1 the term $E[2A_1 b_1 \langle R, s_1 + x_1\rangle]$ can be written as $E[2A_1 b_1]E[\langle R, s_1 + x_1\rangle]$, so

$$\begin{aligned} \mathrm{MSE}(x_1) &= {A_1}^2 - E[2(A_1 b_1)^2 \langle s_1, s_1 + x_1\rangle] - \\ &\quad E[2A_1 b_1]E[\langle R, s_1 + x_1\rangle] + \mathrm{MOE}(x_1) \end{aligned} \tag{4.15}$$

Since $E[2A_1 b_1] = 0$, because b_1 is equally likely to be 1 or -1, the term $E[2A_1 b_1]E[\langle R, s_1 + x_1\rangle]$ can be eliminated, resulting in

$$\begin{aligned} \mathrm{MSE}(x_1) &= {A_1}^2 - 2{A_1}^2 \langle s_1, s_1 + x_1\rangle + \mathrm{MOE}(x_1) \\ &= {A_1}^2 - 2{A_1}^2 \|s_1\|^2 + \mathrm{MOE}(x_1) \\ &= \mathrm{MOE}(x_1) - {A_1}^2. \end{aligned} \tag{4.16}$$

So the mean square error and the mean output energy differ by only a constant and the arguments that minimize both functions are the same. To implement the MSE function in an algorithm, knowledge of the data bits for user 1 is needed. For an implementation of the MOE function this knowledge is not needed, which means that an algorithm based on the MOE function does not require training sequences. It can be shown that the mean output function MOE(x_1) is strictly convex over the set of signals orthogonal to s_1. Therefor, the output energy has no local minima other than the unique global minimum. With this property the stochastic gradient descent method can be used to adaptively implement the blind adaptive MMSE detector [6].

4.1.3 Stochastic Gradient Decent Method

The stochastic gradient descent method is based on the gradient decent method. The *gradient descent method* is used to find the parameter θ_{min} that minimizes the following function:

$$\Xi(\theta) = E[g(X, \theta)]. \tag{4.17}$$

Where X is a random variable and $g(\cdot)$ is a function. If the function Ξ is convex, then for any initial condition θ_0, the gradient descent algorithm converges to the minimum of Ξ. The algorithm follows the direction of steepest descent (i.e., the direction opposite to the gradient $\nabla\Xi$):

$$\theta_i = \theta_{i-1} - \mu\nabla\Xi(\theta_{i-1}), \tag{4.18}$$

where the subscript i is used to indicate the iteration number of the algorithm. If the *step size* μ is arbitrarily small, then eventually θ_i will be as close to θ_{min} as desired. In practice, the step size can be progressively decreased as the algorithm converges. According to (4.17) the probability distribution of X has to be known in order to compute the gradient. Though this knowledge may be available, the use of these distributions can be avoided by using the *stochastic gradient descent method*.

The stochastic version of the algorithm replaces the unknown term $\nabla\Xi(\theta_{i-1}) = \nabla E[g(X, \theta_{i-1})]$ by the unaveraged $\nabla g(X_i, \theta_{i-1})$. With X_i the realization of the random variable X for iteration i. This results in the following stochastic gradient descent algorithm:

$$\theta_i = \theta_{i-1} - \mu\nabla g(X_i, \theta_{i-1}). \tag{4.19}$$

where the subscript i again is used to indicate the iteration number.

4.1.4 Adaptive Implementation

The stochastic gradient decent method can be used to find the x sequence x_{opt} that minimizes the mean output energy. The MOE function, given as

$\text{MOE}(x_1) = E\left[(\langle r, s_1 + x_1\rangle)^2\right]$, is then the equivalent of the $\Xi(\theta)$ function, the x sequence is the equivalent of θ and the received signal r is the equivalent of X. To minimize the mean output energy the x sequence is adapted each bit period using the stochastic gradient descent algorithm (4.19). Since subscripts are already used to indicate users the iteration number is indicated with an index $[i]$. So the stochastic gradient descent algorithm for adaptation of the x sequence can be written as

$$x_1[i] = x_1[i-1] - \mu\nabla(\langle r, s_1 + x_1[i-1]\rangle)^2. \tag{4.20}$$

Here $r[i]$ indicates the received signal for the bit period of the ith bit in the bit stream.[1] $x[i-1]$ is the value of the x sequence obtained during the previous iteration of the algorithm from the previous received signal $r[i-1]$. Note that s_1 is the same for all bit periods (short codes).

Equation (4.20) requires the gradient of $(\langle r[i], s_1 + x_1[i-1]\rangle)^2$ for the *ith* bit interval with respect to x_1. This gradient is

$$\nabla_{x_1}(\langle r[i], s_1 + x_1[i-1]\rangle)^2 = 2\langle r[i], s_1 + x_1[i-1]\rangle r[i]. \tag{4.21}$$

Equation (4.21) states that the gradient of the mean output energy is equal to a scaled version of the received signal $r[i]$. After all, $2\langle r[i], s_1 + x_1[i-1]\rangle$, is only a constant.

At this stage, the gradient descent algorithm for the x sequence can be modified so that it satisfies the orthogonality condition $\langle s_1, x_1\rangle = 0$. This is done by replacing the received signal $r[i]$, by the component of $r[i]$ which is orthogonal to s_1. The orthogonal component $r_{ort}[i]$ is written as:

$$r_{ort}[i] = r[i] - \langle r[i], s_1\rangle s_1. \tag{4.22}$$

By using the stochastic gradient decent algorithm for the x sequence (4.20), the expression for the gradient (4.21) and the orthogonal component of the received signal to s_1 (4.22) the algorithm for minimizing the mean output energy can be found:

$$\begin{aligned} x_1[i] &= x_1[i-1] - 2\mu\langle r[i], s_1 + x_1[i-1]\rangle r_{ort}[i] \\ &= x_1[i-1] - 2\mu\langle r[i], s_1 + x_1[i-1]\rangle(r[i] - \\ &\quad \langle r[i], s_1\rangle s_1) \end{aligned} \tag{4.23}$$

The expression $\langle r[i], s_1\rangle$ in equation (4.23) is a normal *matched filter* operation for user 1, as used in the matched filter bank described in Chapter 2. The expression $\langle r[i], s_1 + x_1[i-1]\rangle$ in equation (4.23) is a matched filter for user 1 with modified sequence (4.6). Since the x_1 sequence of this filter is adapted by the stochastic gradient descent rule this filter is referred to as

[1] $r[i]$ is a vector of N samples of the received signal, sampled at the chip times, for the ith bit period. Where N is the number of chips per symbol.

the *adaptive filter*. The output of the matched filter for user 1 for the ith bit period is written as:

$$Z_{mf1}[i] = \langle r[i], s_1 \rangle. \tag{4.24}$$

Analogously, the output of the adaptive filter for user 1 for the ith bit period is written as:

$$Z_1[i] = \langle r[i], s_1 + x_1[i-1] \rangle. \tag{4.25}$$

Substituting (4.24) and (4.25) in (4.23), the adaptation rule for the x sequence can be written as:

$$x_1[i] = x_1[i-1] - 2\mu Z_1[i](r[i] - Z_{mf1}[i]s_1). \tag{4.26}$$

The output of the adaptive filter $Z_1[i]$ is used as the decision statistic of the blind adaptive MMSE detector for user 1:

$$\hat{b}_1[i] = \text{sgn}(Z_1[i]) = \text{sgn}\langle r[i], s_1 + x_1[i-1] \rangle. \tag{4.27}$$

The output energy of the adaptive filter will be minimal when the x_1 sequence has converged to $x_{1,opt}$, the x sequence that minimizes the mean output energy for user 1. In subsection 4.1.2 it was shown that the mean square error is than also minimized. So when the x_1 sequence has converged to $x_{1,opt}$ the decision statistic for user 1 of the blind adaptive MMSE detector is equivalent to decision statistic for user 1 of the MMSE detector. Figure 4.1 gives a graphical representation of the implementation of the blind adaptive MMSE detector.

The natural choice for initialization of the x sequence is $x_1[0] = 0$. Whether the algorithm is stable depends on the value for the step size μ. Using some approximations, it was found in [6] that the condition on μ for stability is:

$$0 \leq \mu < \frac{2}{A_{max}^2 + \sigma^2}, \tag{4.28}$$

where A_{max} is the maximum amplitude among all users. A smaller step size will result in a longer adaptation time. On the other hand, a smaller step size will also result in an $x_1[i]$ which is closer to $x_{1,opt}$, where $x_{1,opt}$ is the orthogonal sequence component that results in a global minimum of the mean output energy. So the best value for μ is a trade-off between adaptation time and accuracy.

4.2 PIC Detector

In this section the parallel interference cancellation detector will be analyzed in more detail. First the different ways in which parallel interference cancellation can be implemented are discussed. After that the decision rule of the PIC detector is mathematically analyzed and the results of this analysis are used to further refine the PIC implementation.

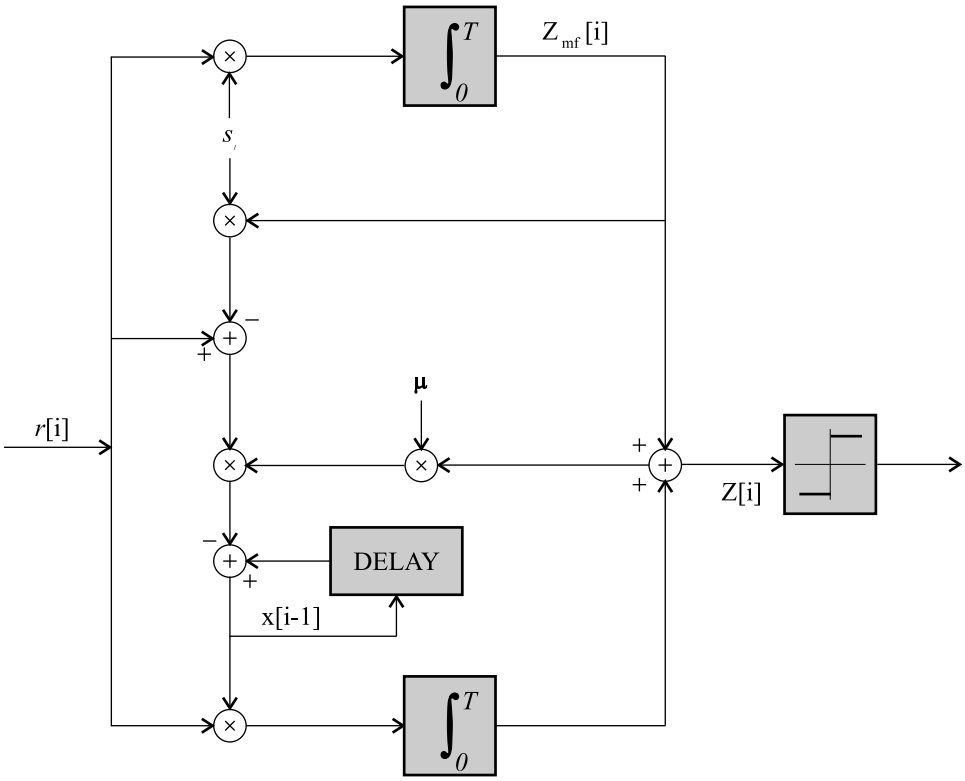


Figure 4.1: Blind Adaptive MMSE detector.

4.2.1 PIC Implementation

There are two implementations for parallel interference cancellation [2]. The implementation that was used to describe the parallel interference cancellation detector in Chapter 3 is the so-called *narrowband implementation*. In this implementation interference is cancelled from the narrowband outputs (matched filter outputs or outputs of previous stages) by using the estimates of the data symbols and channel gains as well as the known cross-correlations between users. This can be clearly seen from Figure 3.7 and equations (3.44) and (3.45). The decision rule of the two-stage narrowband PIC detector for synchronous CDMA given in Chapter 3 can be easily extended to an S-stage narrowband PIC detector. For stage $s+1$ the decision rule of the S-stage narrowband PIC detector can be expressed as

$$\hat{b}_k^{(s+1)} = \operatorname{sgn}\left(Z_k^{(s+1)}\right) \tag{4.29}$$

with

$$Z_k^{(s+1)} = y_k - \sum_{j \neq k} A_j \rho_{jk} \hat{b}_j^{(s)} \tag{4.30}$$

as the decision statistic for stages $s > 1$ and

$$Z_k^{(1)} = y_k \tag{4.31}$$

as the decision statistic for stage $s = 1$. A schematic representation of an S-stage narrowband PIC detector is given in figure 4.2.

The second implementation for parallel interference cancellation requires the estimation, regeneration, and cancellation of the signal of each interferer from the signal of each of the desired users, see Figure 4.3. Since each of the wideband signals has to be regenerated this implementation is referred to as the *wideband implementation*. The decision rule for stage $s+1$ of the S-stage wideband PIC detector can be expressed as

$$\hat{b}_k^{(s+1)} = \operatorname{sgn}\left(Z_k^{(s+1)}\right) \tag{4.32}$$

with

$$Z_k^{(s+1)} = \int_0^T \hat{r}_k^{(s)}(t) s_k(t) dt \tag{4.33}$$

as the decision statistic, where the received signal $\hat{r}_k^{(s)}$ for user k at stage s is estimated according to

$$\hat{r}_k^{(s)}(t) = r(t) - \sum_{j=1}^{K} \hat{u}_j^{(s)}(t) \tag{4.34}$$

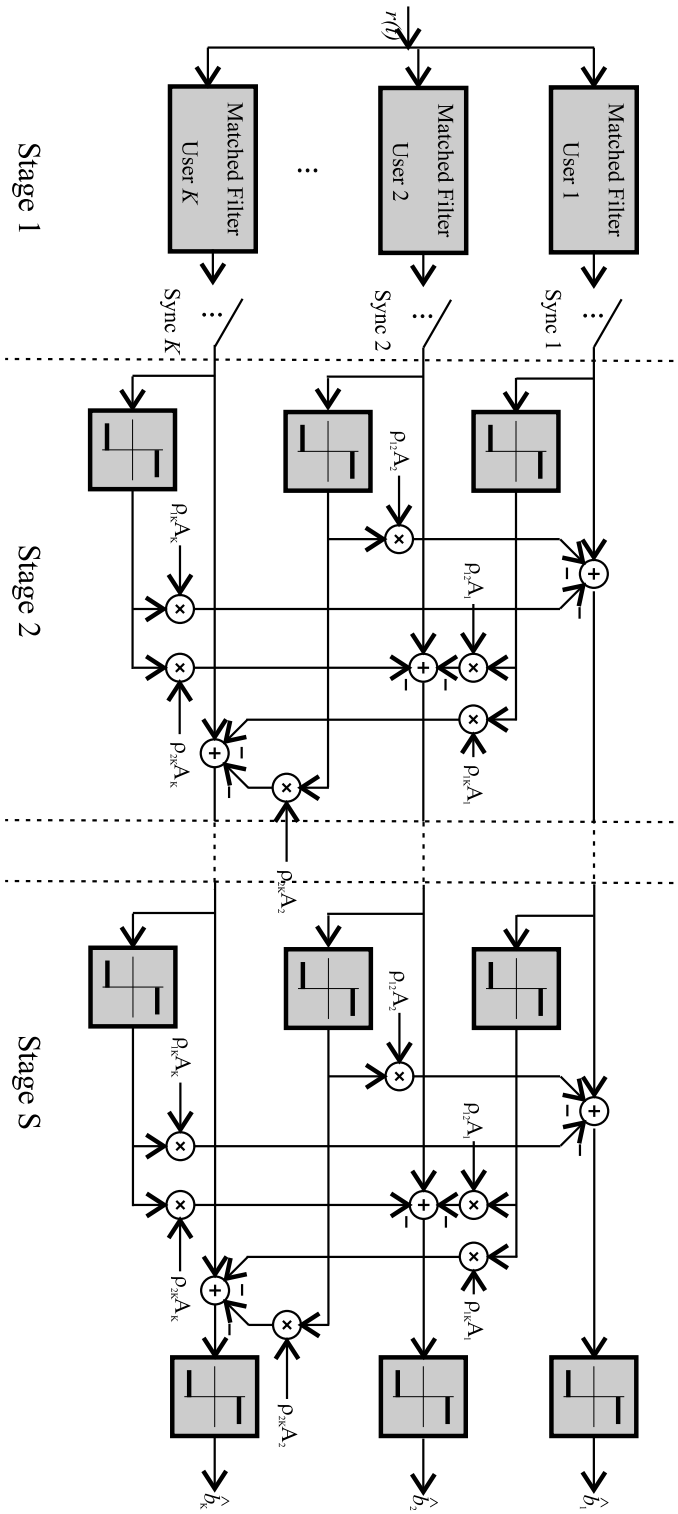


Figure 4.2: S stage narrowband PIC detector.

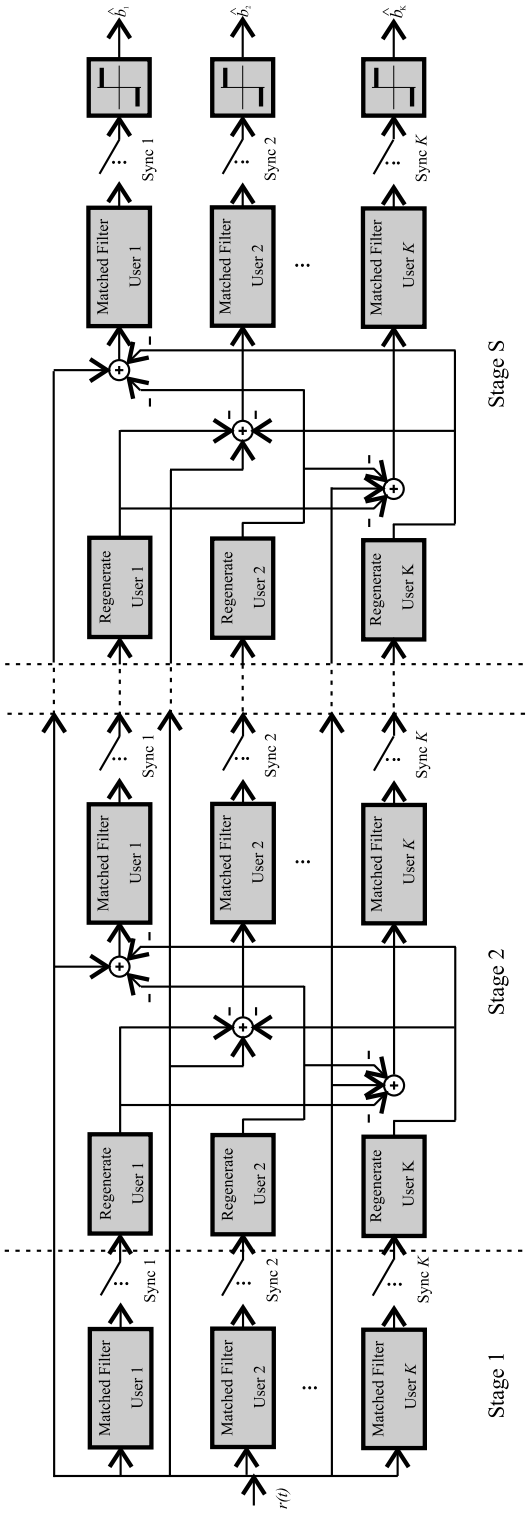


Figure 4.3: S stage wideband PIC detector.

and the signal $\hat{u}_j^{(s)}(t)$ corresponds to the reconstructed signal of user j at stage s. This signal is reconstructed according to

$$\hat{u}_k^{(s)}(t) = A_k \hat{b}_k^{(s)} s_k(t). \tag{4.35}$$

At stage $s = 1$ there has not been any interference cancellation, therefor

$$\hat{r}_k^{(1)}(t) = r(t). \tag{4.36}$$

Both implementations of the parallel interference cancellation detector have the same theoretical bit-error-rate performance [2]. For short-code CDMA systems the narrowband implementation requires less computations than the wideband implementation, because it avoids regeneration of the wideband signal. For long-code systems this advantage disappears because in that case the crosscorrelations have to be recalculated for each symbol. In both cases the narrowband implementation requires more memory because the crosscorrelations ρ_{jk} between the signature sequences of all the users have to be stored.

4.2.2 Amplitude Estimation

The equations for the decision rules of the narrowband implementation (4.30) and the wideband implementation (4.35) show that both implementations require knowledge of the received amplitudes of the signals of all the users. Since this information is not directly available at the receiver, the received amplitudes have to be estimated. A common way to do this is to use the matched filter outputs or outputs of a previous stage, which are both referred to as soft decisions, as a joint estimation of the detected bits and the received signal amplitudes, avoiding the use of separate channel estimation algorithms that increase the overall complexity of the receiver. In this case the decision statistic of the narrowband PIC detector for stages $s > 1$ (4.30) can be rewritten as

$$Z_k^{(s+1)} = y_k - \sum_{j \neq k} Z_j^{(s)} \rho_{jk} \tag{4.37}$$

and the equation for the reconstructed signal of the wideband PIC detector for stages $s > 1$ (4.35) can be rewritten as

$$\hat{u}_k^{(s)}(t) = Z_k^{(s)} s_k(t). \tag{4.38}$$

Unfortunately the amplitude estimates derived from the output of the matched filter bank are, especially in the first stages, not very accurate because they still contain a lot of interference. Therefor enhanced performance can be achieved at the cost of increased receiver complexity by using separate adaptive channel estimation algorithms. Nevertheless is will be assumed in the rest of this thesis that the amplitude estimates are obtained using the matched filter bank outputs.

4.2.3 PIC Decision Rule Analysis

In Correal et al. [3] it is shown that straightforward implementation of wideband parallel interference cancellation based on *complete subtraction* of the interference estimates results in a *biased decision statistic*. The bias has its strongest effect in the first stage of interference cancellation, in the subsequent stages its effect diminishes. However if the bias leads to incorrect cancellation at the first stage the effects of these errors may be observed at the next stages. In this section it is shown that the decision statistic of a straightforward implementation of narrowband parallel interference cancellation based on complete subtraction of the interference is also biased.

Following the same reasoning as Correal et al. analysis is started with a two user synchronous system. The decision statistic at stage $s = 1$ (i.e. before any interference cancellation) for b_1, a bit of user 1, is the output of the matched filter for user 1:

$$Z_1^{(s=1)} = A_1 b_1 + A_2 b_2 \rho + n_1 \tag{4.39}$$

where

$$n_1 = \sigma \int_0^T n(t) s_1(t) dt \tag{4.40}$$

and ρ is the crosscorrelation between the signature sequences of user 1 and user 2. Similarly, for user 2 at stage $s = 1$, the decision statistic is given by

$$Z_2^{(s=1)} = A_2 b_2 + A_1 b_1 \rho + n_2. \tag{4.41}$$

At stage 2, the interference caused by user 2 is subtracted from the matched filter output for user 1 to form a new soft decision for user 1 at stage 2. Using (4.37) the decision statistic for user 1 at stage 2 can be written as

$$Z_1^{(s=2)} = Z_1^{(s=1)} - Z_2^{(s=1)} \rho \tag{4.42}$$

Substitution of the decision statistics of users 1 and 2 from (4.39) and (4.41), and cancelling common terms results in

$$Z_1^{(s=2)} = A_1 b_1 - A_1 b_1 \rho^2 + n_1 - n_2 \rho \tag{4.43}$$

Conditioning on b_1 and taking the expected value of the decision statistic yields

$$E[Z_1^{(s=2)} | b_1] = A_1 b_1 - A_1 b_1 E[\rho^2], \tag{4.44}$$

as $E[n_1] = E[n_2] = 0$ because n is white Gaussian noise with 0 mean and unit variance. According to Verdú [15]

$$E[\rho^2] = \frac{1}{N} \tag{4.45}$$

where N is the processing gain, so (4.44) becomes

$$E[Z_1^{(s=2)}|b_1] = A_1 b_1 \left(1 - \frac{1}{N}\right). \tag{4.46}$$

Since the estimate of the interference caused by user 2 is correlated with user 1's amplitude and bit value, a bias is produced when it is used to remove the interference. The bias in the mean of the decision statistic is evident in (4.46).

Following a similar approach this result can be extended to a K-user system, resulting in

$$E[Z_1^{(s=2)}|b_1] = A_1 b_1 \left(1 - \frac{K-1}{N}\right). \tag{4.47}$$

So the bias in the mean increases linearly with the system load (number of users K) and is inversely propertional to the processing gain N.

In Correal et al. [3] a simple method to mitigate the effect of the bias and improve the performance of parallel multistage interference cancellation is proposed. This method is based on multiplying the amplitude estimates with a *partial-cancellation factor* $0 \leq C_K^{(s)} \leq 1$ that varies with the stage of cancellation s and system load K. This multiplication has to be be performed before the amplitude estimates are used to subtract the interference in case of the narrowband implementation, or before the amplitude estimates are used to reconstruct the signal in case of the wideband implementation. This can be interpreted as modifying equation (4.30) and (4.34) to include a partial-cancellation factor $C_K^{(s)}$ resulting in respectively

$$Z_k^{(s+1)} = y_k - \sum_{j \neq k} C_K^{(s)} A_j \rho_{jk} \hat{b}_j^{(s)} \tag{4.48}$$

and

$$\hat{r}_k^{(s)}(t) = r(t) - C_K^{(s)} \sum_{j=1}^{K} \hat{u}_j^{(s)}(t). \tag{4.49}$$

4.3 Conclusions

- By minimizing the mean output energy the blind adaptive MMSE detector also minimizes the minimum mean square error.
- The derivation of an adaptive algorithm for implementation of the blind adaptive MMSE detector has been shown.
- The adaptation time, stability and accuracy of the the adaptive algorithm all depend on the step size μ.

- There are two possible implementations for the PIC detector, a narrowband implementation and a wideband implementation, that have the same bit-error-rate performance, but different computational complexity.
- The amplitude estimates needed by the PIC detector can be obtained by using the soft decisions of the previous stage of the PIC detector as joint estimation of the detected bit and the received amplitude.
- Both implementations of the PIC detector have a bias in their decision statistic.
- The effect of this bias can be reduced by the introduction of a partial cancellation factor.

In the next chapter the adaptive implementation of the blind MMSE detector will be simulated and the influence of the step size μ on adaptation time, stability and accuracy will be studied. In the next chapter also both the narrowband and the wideband implementation of the PIC detector will be simulated and the effects of amplitude estimation, the bias in the decision statistics and the partial-cancellation factor on the bit-error-rate performance will be studied.

Chapter 5

Simulation

In this chapter first the development of a simulator for the conventional, blind adaptive MMSE and parallel interference cancellation detectors will be described. After that, simulation results for these detectors obtained with this simulator will be presented, analyzed and compared with the simulation results that were used in Chapter 3 to select the algorithms of interest for implementation. Finally the conclusions that can be drawn from the obtained simulation results will be summarized.

5.1 Simulator

Nowadays a simulator usually is an implementation of a system model on a digital computer. The simulator described in this section implements the synchronous CDMA model as described in Chapter 2 and the conventional, blind adaptive MMSE and parallel interference cancellation detectors that were desctribed in Chapter 3 and further analyzed in Chapter 4. Since the model is implemented on a digital computer it has to be converted to a discrete-time model. In the simulator the received signal model is therefor evaluated every chip time and the detectors operate on this sampled model. This is different from the discrete-time model in Chapter 2 that is sampled at the symbol time, because in the simulator information about the signals at the chip level is needed.

The CDMA model as described in Chapter 2 basically consists of two major parts, the transmitter and the channel, both described in subsection 5.1.1. The detectors that are supported by the simulator are described in subsection 5.1.2. The CDMA model has as input the data streams that the different users want to transmit and the amplitudes they use to transmit that data. The contents of the data streams and the amplitudes the different users use for transmission depend on the kind of simulation that has to be performed. The different simulation modes that the simulator supports are described in subsection 5.1.3. Subsection 5.1.4 describes how the simula-

tion modes and simulation parameters can be specified to the simulator. It also describes the format the simulator uses to return the simulation results. Finally subsection 5.1.5 describes the choices regarding implementation related issues that were made during the implementation process.

5.1.1 Transmitter and Channel

The transmitter part of the simulator generates the transmitted signal for one symbol period from the bits that the different users in the system want to transmit and the amplitudes they use for transmission, according to the first part of equation (2.1):

$$r(t) = \sum_{k=1}^{K} A_k b_k s_k(t). \tag{5.1}$$

Since the simulator operates on chip level this equation has to be sampled at the chip times:

$$r(pT_c) = \sum_{k=1}^{K} A_k b_k s_k(pT_c), \tag{5.2}$$

where p is the chip index. Defining $\mathbf{s}_k$ as a vector representation of the signature sequence of user k sampled at the chip times and defining $\mathbf{v}$ as a vector representation of the transmitted signal sampled at the chip times this can also be written as:

$$\mathbf{v} = \sum_{k=1}^{K} A_k b_k \mathbf{s}_k. \tag{5.3}$$

Since the CDMA model described in Chapter 2 assumes an additive white gaussian noise (AWGN) channel the simulator has to add noise to the sampled transmitted signal to generate the received signal. The standard deviation σ of the noise is calculated from the signal-to-noise ratio (SNR) and the amplitude A of a user k as:

$$\sigma = \sqrt{\frac{A_k^2}{2\ 10^{\frac{SNR_k}{10}}}}. \tag{5.4}$$

The 2 appears into the denumerator of this equation because the signal-to-noise is expressed in the signal energy for one bit period $E_b = {A_k}^2$ and the noise one sided spectral density N_0, which is equal to $2\sigma^2$. Noise is than added for each sample of the transmitted signal by adding the output of a normally distributed random number generator with mean 0 and standard deviation 1 (Gaussian distribution) multiplied with the calculated standard

deviation. This results in the following equation for the received signal for one symbol period.

$$\mathbf{r} = \sum_{k=1}^{K} A_k b_k \mathbf{s}_k + \sigma \mathbf{n}. \tag{5.5}$$

The normally distributed random number generator uses the boxmuller [1] algorithm to generate a normally distributed random number from the output of a uniformly distributed random number generator. As a uniformly distributed random number generator the Mersenne Twister [9] is used.

5.1.2 Detectors

The simulator can simulate the conventional detector, the blind adaptive MMSE detector and the narrowband and wideband implementations of the PIC detector. Each detector takes as input the received signal for one symbol period and the desired user (the user whose bits have to be detected) and outputs the detected bit of the desired user.

The simulated conventional detector correlates the received signal with the signature sequence of the desired user as indicated by equation (3.6). It than outputs the sign of the correlator output as the detected bit as indicated by (3.4).

The simulated blind adaptive MMSE detector implements equations (4.24), (4.25), (4.26) and (4.27). The implementation has two operation modes. The first mode is a 'training mode' in which the x sequence is updated according to the adaptation rule (4.26). The second mode is a 'detection' mode in which the x sequence is not updated and the detector basically operates as a conventional detector that uses a signature sequence modified by the addition of the x sequence.

The narrowband implementation of the PIC detector in the simulator is implemented according to equations (4.29) and (4.48). The wideband implementation is implemented according to the equivalent equations for the wideband case: (4.32), (4.33), (4.49) and (4.35). Since the detectors in the simulator both are based on the equations that include a partial-cancellation factor, the simulator can be used to study the influence of the bias and the partial-cancellation factor on detector performance. Both implementations of the detector also have the option to use either actual amplitudes, or amplitudes estimated from the matched filter outputs or previous stages, so the influence of amplitude estimation on detector performance can be studied as well.

5.1.3 Simulation Modes

The simulator supports six *simulation modes*. The simulation mode determines the bit streams that the different users transmit and the parameter of the synchronous CDMA model that is varied during the simulation.

The *data* simulation mode simulates a communications system in which the desired user is instructed to transmit either only ones, or only minus ones, or random ones and minus ones. The other users can be instructed independently to transmit either only ones, or only minus ones, or random ones and minus ones as well. The simulator outputs the number of transmitted bits, the number of incorrectly detected bits and the bit-error-rate. In this simulation mode all the synchronous CDMA model parameters are kept constant during simulation.

The *file* simulation mode simulates the transmission of a file by the desired user. So the bit stream of the desired user consists of the bits of the file that has to be transmitted. The other users in the system again transmit either only ones, or only minus ones, or random ones and minus ones. The simulator outputs again the number of transmitted bits, the number of incorrectly detected bits and the bit-error-rate, but it also writes the detected bits to a file. This way the received file can be compared to the file that was transmitted. Again all the synchronous CDMA model parameters are kept constant during simulation.

The next four simulation modes all generate bit-error-rate (BER) estimates as a function of a series of values for a synchronous CDMA system model or detector parameter. The probability of a detection error for a bit of the desired user depends on the bits of the other users in the system that were transmitted at the same time, because those bits determine the interference present in the system. Therefor the simulator has to simulate all the possible combinations of bits for the other users in the system in order to get a good estimate of the bit-error-rate. The bit that is transmitted by the desired user however is not important and is kept 1 all the time. This is similar to the conditioning on the bits of the other users that was needed in Chapter 3 to obtain an analytical expression for the probability of error. Because the occurrence of a detection error is a rare event a large number of transmitted and detected bits may have to be simulated in order to obtain enough detection errors to reliably estimate the bit-error-rate. These four simulation modes therefor repeat the simulation of all the possible combinations of bits for the other users in the system until a certain number of detection errors is reached. This whole procedure is repeated for each value of the series of values of the CDMA model or detector parameter that is varied. Apart from the bit-error-rate these simulation modes all also output the number of detection errors and the number of transmitted bits that were simulated, for each value of the series of values of the CDMA model or detector parameter that is varied.

The *BER/SNR* simulation mode can be used to obtain estimates of the bit-error-rate for different values of the signal-to-noise ratio (SNR). The other CDMA model parameters are kept constant during simulation.

The *BER/users* simulation mode can be used to obtain estimates of the bit-error-rate for different values of the number of active users in the system.

The other CDMA model parameters are kept constant during simulation.

The *BER/NFR* simulation mode can be used to obtain estimates of the bit-error-rate for different values of the received energy of one of the interfering users in the system. This simulates the near-far effect. The other CDMA model parameters are kept constant during simulation.

The *BER/I* simulation mode can be used to obtain estimates of the bit-error-rate of the blind adaptive MMSE detector as a function of the number of iterations of the adaptation rule. So after each iteration of the adaptation rule the bit-error-rate is determined. The CDMA model parameters are kept constant during simulation.

5.1.4 Simulator Input and Output

Simulator input and output is performed trough files. The simulator reads the properties of the system that has to be simulated and the simulation mode that has to be used from a file and writes the simulation results to a file. This has the advantage that both the simulation results and the system properties that were used to generate those results are saved.

Matlab functions are provided to generate plots from the simulation results of the *BER/SNR*, *BER/users*, *BER/NFR* and *BER/I* simulation modes. These functions can plot the confidence intervals of the estimated bit-error-rate. A confidence interval gives the probability that, given a probability distribution function $P(x)$ with mean μ, a measurement falls within an interval x centered around μ, so

$$CI(x) = CI(x - \mu, x + \mu) = \int_{\mu - x}^{\mu + x} P(x)dx. \tag{5.6}$$

Since the simulated transmitter, channel and detectors are all memoryless, detection errors occur as a sequence of independent random events in time and can therefor be modelled as a Poisson process. Therefor in this case $P(x)$ is the probability distribution function of the Poisson process. The number of detection errors obtained from the simulator is then used as a mean for creation of a Poisson distribution. The inverse Poisson probability distribution function in Matlab is now used to calculate the lower- and upper bounds on the number of detection errors from the created Poisson distribution, for a specified percentage of confidence. These bounds, together with the simulated number of bits, are then used to plot the confidence intervals.

5.1.5 Implementation

The simulator is implemented as a command line program, written in the C programming language. Command line programs have the advantage that they can easily be called from within other applications, like for example

Matlab. This allows the use of Matlab scripts for automated execution of a number of simulations with different simulation parameters.

The C programming language was chosen because it generates fast running programs compared to for example Matlab, which is important considering the large number of transmitted and detected bits that have to be simulated to obtain reliable results. C also has the advantage that it is supported by most DSP platforms, so algorithms implemented for the simulator can be ported relatively easily to a DSP. A disadvantage of using C compared to Matlab is that C does not provide all the library functions that Matlab provides for simulating communications systems.

All synchronous CDMA model and detector parameters in the simulator are represented as double precision floating point numbers for the most accurate representation. This comes at the cost of increased simulation times because double precision floating point operations on most processors require more clock cycles to execute than their single precision equivalents.

5.2 Simulation Results

In this section the simulation results, obtained with the simulator described in the previous section, for the conventional, blind adaptive MMSE and PIC detectors will be presented.

In all simulations a maximal length sequence with length 31 is used to generate the spreading sequences for the different users. Since short codes are used the processing gain in the simulations is also 31. A maximal length sequence with length 31 can generate only 31 different spreading sequences, therefor the simulated systems supports 31 users maximum. Maximal length sequences can be generated using a linear feedback shift register [5]. To generate the sequences used for the simulations in this section a linear feedback shift register with 5 states and feedback connections for bit 2 and 5 is used.

5.2.1 Conventional Detector

Figure 5.1 shows the bit-error-rate of the conventional detector as a function of the signal-to-noise ratio (SNR) of the desired user in a ten user CDMA system with perfect power control. Plotted are the simulation result and the analytical expression for the bit-error-rate of the conventional detector (3.12) that was obtained in Chapter 3. The 95% confidence intervals of the simulation result are plotted as well. From the figure it can be clearly seen that the plot of the simulation result and the plot of the analytical expression are almost identical. The plot of the analytical expression for the bit-error-rate of the conventional detector lies within the 95% confidence intervals of the simulation result. This indicates that the simulator gives reliable simulation results for the bit-error-rate of the conventional detector.

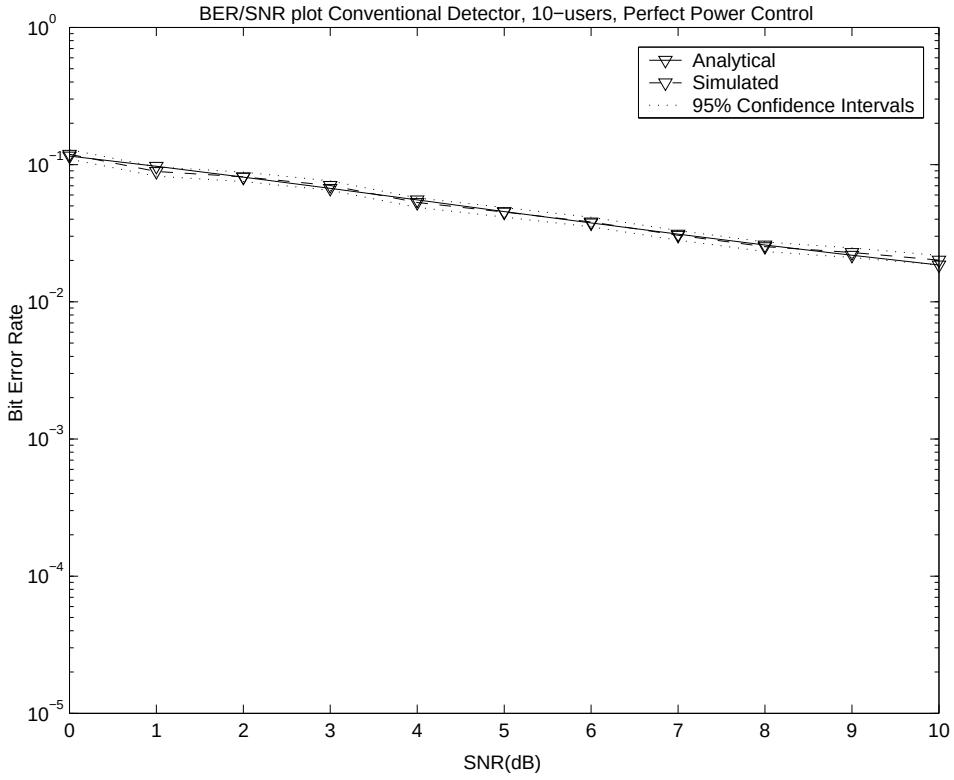


Figure 5.1: BER/SNR plot conventional detector (ten users, processing gain = 31, perfect power control)

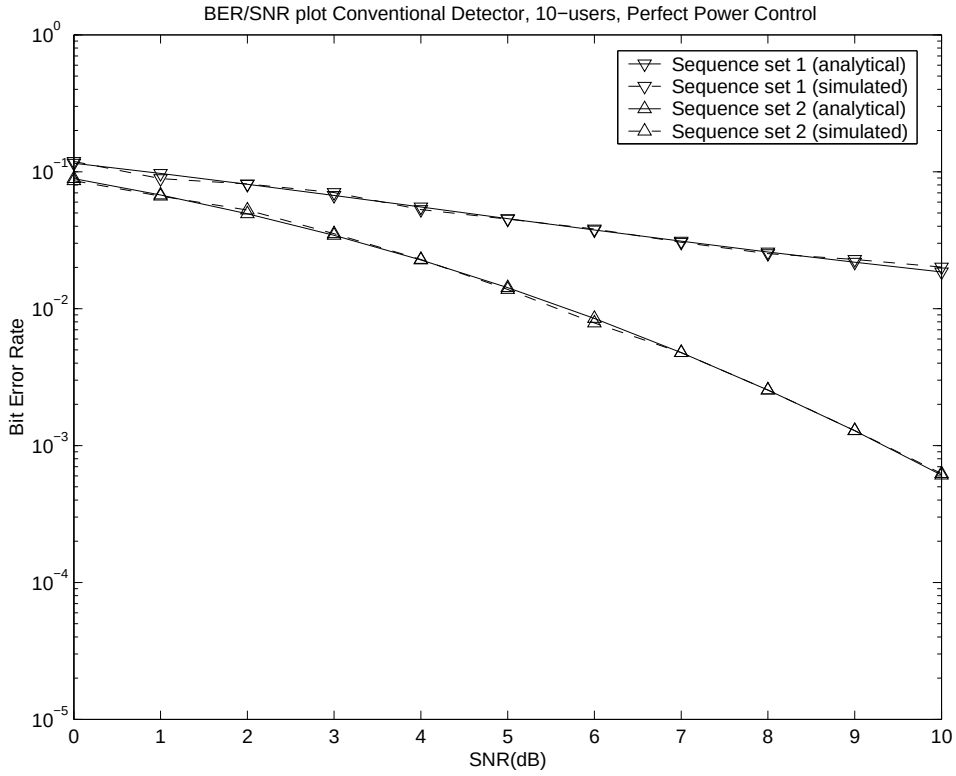


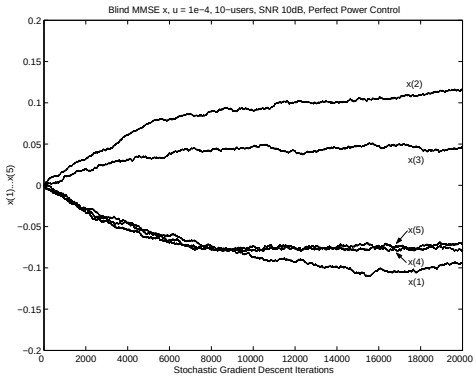
Figure 5.2: BER/SNR plot conventional detector for two different sets of signature sequences (ten users, processing gain = 31, perfect power control)

When Figure 5.1 is compared to Figure 3.11 it can be seen that the simulated conventional detector delivers a similar level of bit-error-rate performance as the conventional detector in Figure 3.11. The difference can be explained by the fact that the exact signature sequences that were used by Buehrer and others [2] to obtain Figure 3.11 are not known. The simulation results in this Chapter are therefor obtained using (probably) different signature sequences. That the used signature sequences can cause a relatively large difference in bit-error-rate can be clearly seen from Figure 5.2. This figure again shows the bit-error-rate of the conventional detector as a function of the signal-to-noise ratio, but this time for two different sets of signature sequences. Sequence set 1 is the set of signature sequences that was used to obtain Figure 5.1 and is used for all the other simulations described in this section. Sequence set 2 is a set of signature sequences with length 31 that can be generated by a 5 state linear feedback shift register with feedback connections for bit 1, 2 and 5. Although sequence set 2 appears to deliver better bit-error-rate performance, the other simulations described in this section will all use sequence set 1, because it was already used to generate a lot of the simulation results before the performance of sequence set 2 was studied.

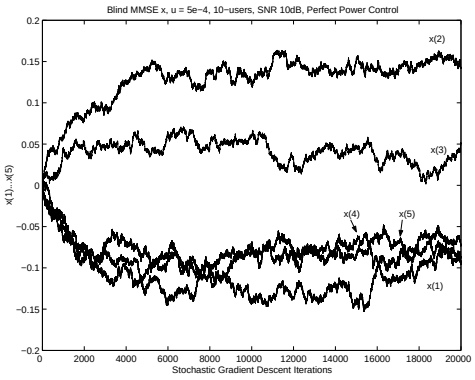
5.2.2 Blind Adaptive MMSE Detector

Before the blind adaptive MMSE detector can be simulated a proper step size μ has to be found that results in an x sequence that is close to x_{opt}, where x_{opt} is the x sequence that results in a global minimum of the mean output energy and thus the minimum mean square error. According to equation (4.28) the maximum value for μ is determined by the maximum amplitude among all users in the system and the noise level. In the case of perfect power control with the amplitudes of all the users in the system equal to 1 for 1dB of signal-to-noise ratio equation (4.28) indicates that blind adaptive MMSE algorithm will run stable for a step size $\mu = 1.4$ or smaller. However, when the blind adaptive MMSE detector is simulated with this step size, all amplitudes equal to 1 and a signal-to-noise ratio of 1dB, the algorithm turns out to be unstable. So equation (4.28) appears, at least for this particular case, to be incorrect. In [6] it is stated that the approximations that were used to derive equation (4.28) become exact when the crosscorrelations between the signature sequences in the system go to zero. So equation (4.28) may not give a reliable upper limit for the step size μ because the crosscorrelations between the used signature sequences are too high. However this assumption has not been verified.

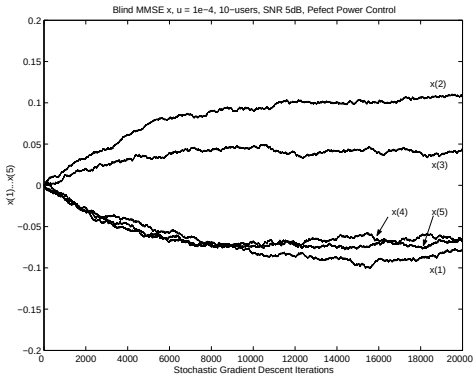
By starting at a step-size $\mu = 1$ and gradually reducing the step size it was found that the blind adaptive MMSE algorithm is stable and will generate an x sequence that is close to x_{opt} for a 10 user 10dB signal-to-noise ratio, perfect power control system with a step size of $\mu = 10^{-4}$.



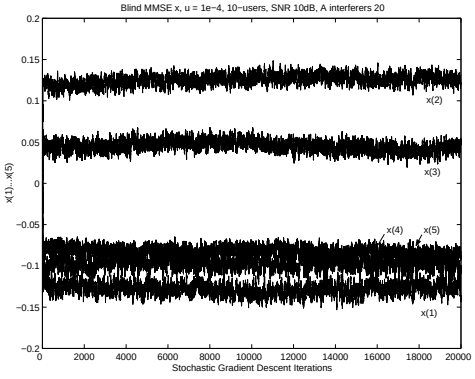
(a) Default parameters



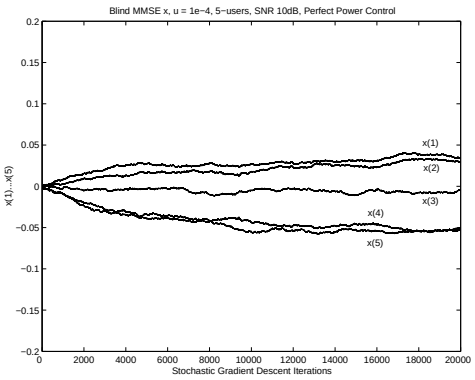
(b) $\mu = 5 \cdot 10^{-4}$



(c) 5dB SNR



(d) A interferers 20



(e) 5 users

Figure 5.3: Convergence for 5 elements of the x sequence for varying CDMA model parameters. Default parameters: $\mu = 10^{-4}$, 10 users, 10dB SNR, Perfect Power Control

In Figure 5.3 the values of the first 5 elements of the 31 element long x sequence of the blind adaptive MMSE detector are plotted as a function of the number of iterations of the adaptation algorithm for the x sequence 4.26). In each of the subfigures one of the CDMA model parameters or the blind adaptive MMSE step size μ is changed. The adaptation algorithm for the x sequence performs one iteration each symbol time. Since the the simulator uses short code signature sequences with a length of 31 chips one iteration has a duration of 31 chip times.

In Figure 5.3(a) the values of the first 5 elements of the 31 element x sequence of an blind adaptive MMSE detector with $\mu = 10^{-4}$ are plotted for a 10 user, perfect power control CDMA system where the amplitudes of all the users are equal to 1 and the signal-to-noise ratio is 10dB. It can be seen that the values of the plotted elements of the x sequence have converged to the values of the optimal x sequence after about 12000 iterations of the adaptation rule. This can be seen because the values hardly change anymore after that number of iterations, indicating that the minimum of the output energy has been reached. Since one iteration corresponds with one symbol time, 12000 iterations correspond with 1.25 seconds when a symbol rate of 9600bps is used.

For Figure 5.3(b) the step size of the blind adaptive MMSE detector is increased to $\mu = 5 \cdot 10^{-4}$, all the other parameters are kept the same as in 5.3(a). The figure clearly shows that the values of the elements of the x sequence converge after fewer iterations as in 5.3(a). The estimated values for the elements of the x sequence however continue to vary for each iteration after convergence has been reached. Therefor the obtained estimate for the optimal x signature at any iteration number after the x signature has converged is less accurate than in Figure 5.3(a).

In Figure 5.3(c) the step size is changed back to $\mu = 10^{-4}$ and the signal-to-noise ratio is decreased to 5dB. If this figure is compared to Figure 5.3(a) it appears that the signal-to-noise ratio does not have a lot of influence on the convergence or accuracy of the values of the x sequence.

For Figure 5.3(d) the signal-to-noise ratio is changed back to 10dB and the amplitudes of the interfering users in the system are changed to 20. The elements of the x sequence converge to their optimal value very fast, but, as in Figure 5.3(b), the obtained estimates for the optimal values of the elements of the x sequence at any iteration number after the x sequence has converged is less accurate. For this particular case it would be beneficial to choose a smaller step size μ to trade a larger number of iterations needed for convergence for a more accurate estimate of the optimal x sequence.

Finally in Figure 5.3(e) the total number of users in the system is decreased to 5 and the amplitudes of the interfering users are changed back to 1, so the situation of perfect power control returns. In this situation the x sequence reaches convergence after fewer iterations than in Figure 5.3(a). The values that the elements of the x sequence converge to are also smaller

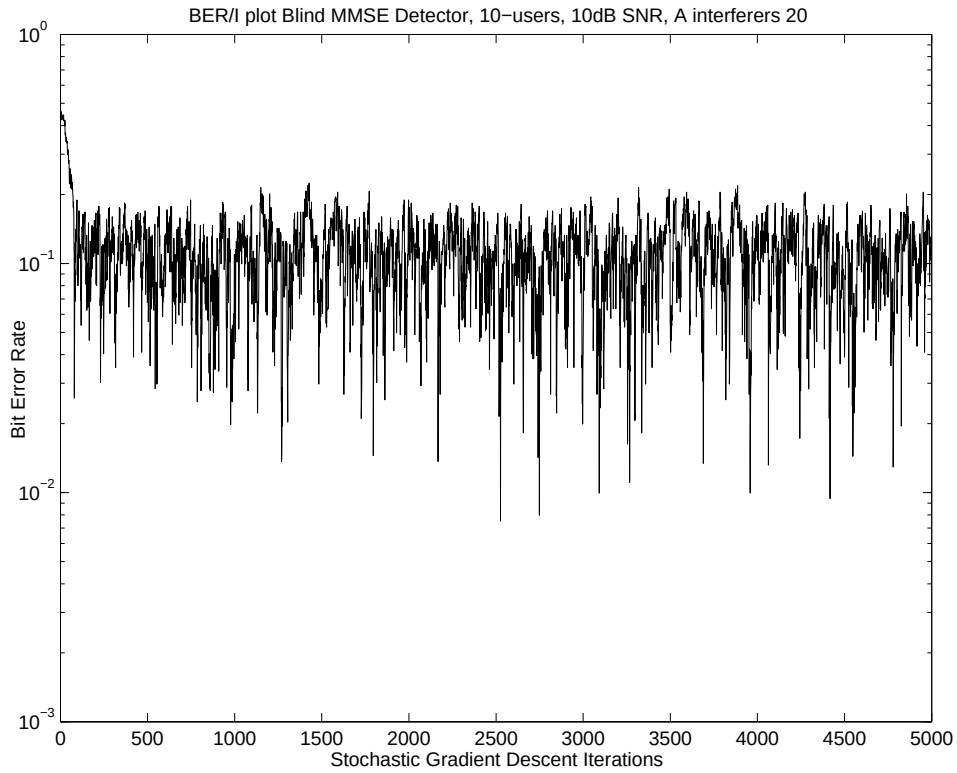


Figure 5.4: BER/I plot. Blind adaptive MMSE detector ($\mu = 10^{-4}$, ten users, processing gain = 31, A interferers 20)

than the values to which the elements converge in 5.3(a). Since the system contains less users, there are less possible combinations for the interference and therefor the optimal value for the x sequence is found sooner. Also because the system contains less users, more elements of the x sequence can be used to cancel the interference caused by one interfering user and therefor the individual components of the x sequence can remain smaller.

To show the influence of the x sequence on the bit-error-rate in Figure 5.4 the bit-error-rate of the blind adaptive MMSE detector with $\mu = 10^{-4}$ is plotted as a function of the number of iterations of the adaptation rule for the x sequence. So after each iteration of the adaptation rule the bit-error-rate is determined. The system that is simulated is a ten user CDMA system with interferer amplitudes 20. This system was also simulated to obtain Figure 5.3(d). Figure 5.4 shows that the x sequence is converged after about 100 iterations, after which the bit-error-rate begins to vibrate. From the low bit-error-rate performance that the detector achieves it can be concluded that the obtained estimate of the optimal x sequence is not very accurate.

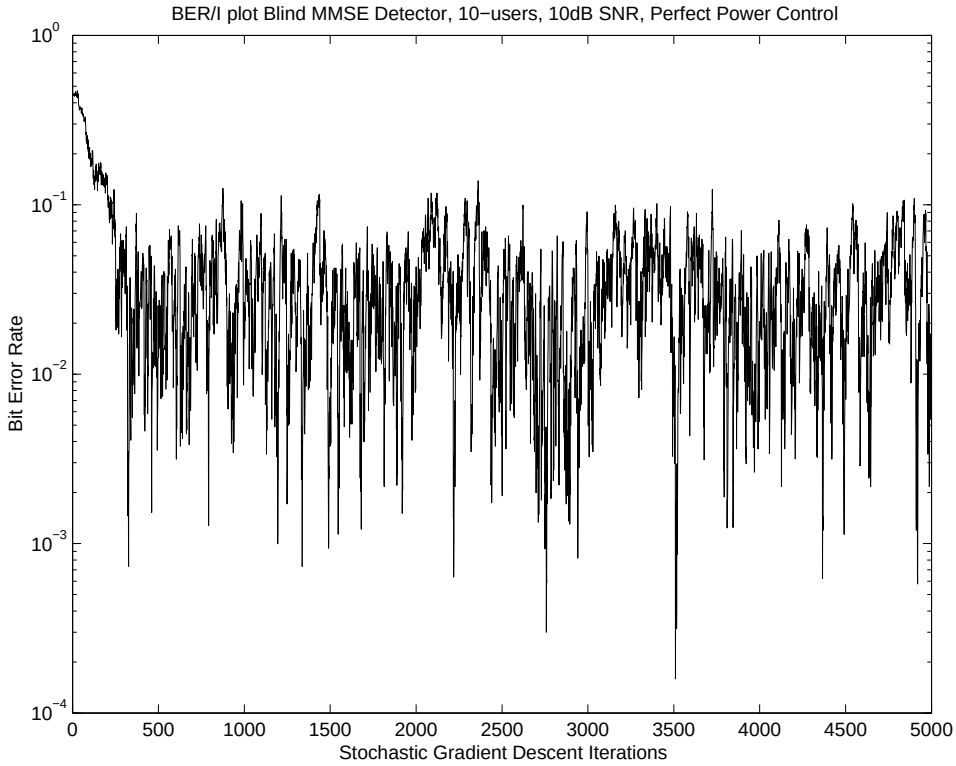


Figure 5.5: BER/I plot. Blind adaptive MMSE detector ($\mu = 5 \cdot 10^{-5}$, ten users, processing gain = 31, A interfers 20)

In the discussion of Figure 5.3(d) it was suggested that the accuracy of the estimate of the optimal x sequence in a system with large interferer amplitudes can be improved by using a smaller step size μ. This is done in Figure 5.5. For this figure the same simulation is performed as for Figure 5.4, but this time the blind adaptive MMSE detector has a step size of $\mu = 5 \cdot 10^{-5}$. When Figure 5.5 is compared to Figure 5.4 it clearly shows that convergence of the x sequence to its optimal value takes more iterations, but the bit-error-rate performance that the detector achieves after the x sequence is converged is also better.

From the simulations described above it can be concluded that the x sequence of the blind adaptive MMSE detector has to converge to the optimal x sequence x_{opt} in order for the detector to achieve its optimal bit-error-rate performance. It can also be concluded that the values for the elements of the x sequence that are obtained when adaptation of the x sequence is stopped after it has converged are not necessarily the most accurate estimation of the values of the elements of the x_{opt} sequence, because the values of the x sequence continue to 'vibrate' around the optimal values. So the x sequence basically goes trough two stages: during the first stage it converges to the optimal sequence x_{opt} and during the second stage the values of x 'vibrate' around the optimal values.

The blind adaptive MMSE detector in the simulator can operate in two modes, a 'training' mode and a 'detection' mode, as described in subsection 5.1.2. Only during the 'training' mode the x sequence of the detector is updated. During the 'detection' mode the x sequence is kept constant. The 'training' mode of the simulated detector is used when the values of the x sequence have not yet converged to the values of the optimal x sequence and the detector therefor is not adapted to the simulated system. The 'detection' mode is always used after the training mode, so when the detector has adapted to the simulated system. Since the x sequence goes trough two stages, the 'training' mode of the blind adaptive MMSE detector consists of two stages as well. In the first stage the x sequence is just updated according to the adaptation rule for a certain number of iterations. It is assumed that this number of iterations is enough for the x sequence to converge, so this stage can be called the 'convergence' stage. In the second stage the x sequence is also updated for a certain number of iterations, but the average x sequence over this stage is calculated as well. At the end of the stage the x sequence of the detector is replaced by the calculated averaged x sequence, so this stage can be called the 'averaging' stage. This way an averaged estimate of the values of the optimal x sequence is obtained that are hopefully closer to the actual values of the optimal x sequence than a single estimate. The number of iterations for each of these stages can be specified by the user of the simulator. In order to be able to chose the number of iterations for the

'convergence' stage the user should have an idea about how many iterations of the adaptation rule for the x sequence are required for the x sequence to converge to the optimal value for the system that is simulated. The number of iterations that is chosen for the 'averaging stage' is less important, but should be relatively large in order to get a proper average.

For each simulation mode that is described in section 5.1.3 the simulator always starts the blind adaptive MMSE detector in the 'training' mode. In this mode all users in the system, including the desired user, transmit random bits for the number of iterations that the user of the simulator specified for the 'convergence' and 'averaging' stages. These random bits are used to adapt the detector to the system that is simulated. After that the detector is switched to the 'detection' mode and the users start to transmit the bit streams as indicated in the description of the simulation modes in section 5.1.3. In case of a simulation mode in which one parameter of the synchronous CDMA model or the detector is varied this whole procedure is repeated for each value of the series of values for this parameter. When a BER/SNR simulation for a signal-to-noise ratio varying from 0 to 10dB with 1dB increments is used as an example this means that the detector is started in the 'training' mode. The 'convergence' and 'averaging' stages are than executed to adapt the detector to the system with a signal-to-noise ratio of 0dB. When the user specified number of iterations for 'convergence' and 'averaging' stages are executed the detector is switched to the 'detection' mode and the bit-error-rate of the system with a signal-to-noise ratio of 0dB is determined. The simulation for a signal-to-noise ratio of 0dB is now complete, so the detector is switched back to the 'training' mode. Again the 'convergence' and 'averaging' stages are executed, but this time to adapt the detector to the system with a signal-to-noise ratio of 1dB. After that the detector is switched to 'detection' mode again and the bit-error-rate for 1dB signal-to-noise ratio is determined. This process is continued for the whole range of signal-to-noise ratios.

It is not practical to include a 'training' mode in implementations of the blind adaptive MMSE detector for use in actual CDMA systems, because than the detector would have to be switched to 'training' mode each time when one of the CDMA system parameters changes. This would interrupt the normal detection process which is of course not possible. So the blind adaptive MMSE detector in actual CDMA systems has to adapt its x sequence during normal detection. This results of course in a lot of detection errors when the detector is just turned on and has not yet adapted. This will have to be solved by using error detection or error correction. However after a certain number of detected bits the detector will have reached convergence and achieve its optimal bit-error-rate for the system. Provided that the CDMA system parameters change gradually from now on the detector will not drift far from convergence, because the adaptation algorithm can quickly adapt to small changes in the CDMA system parameters.

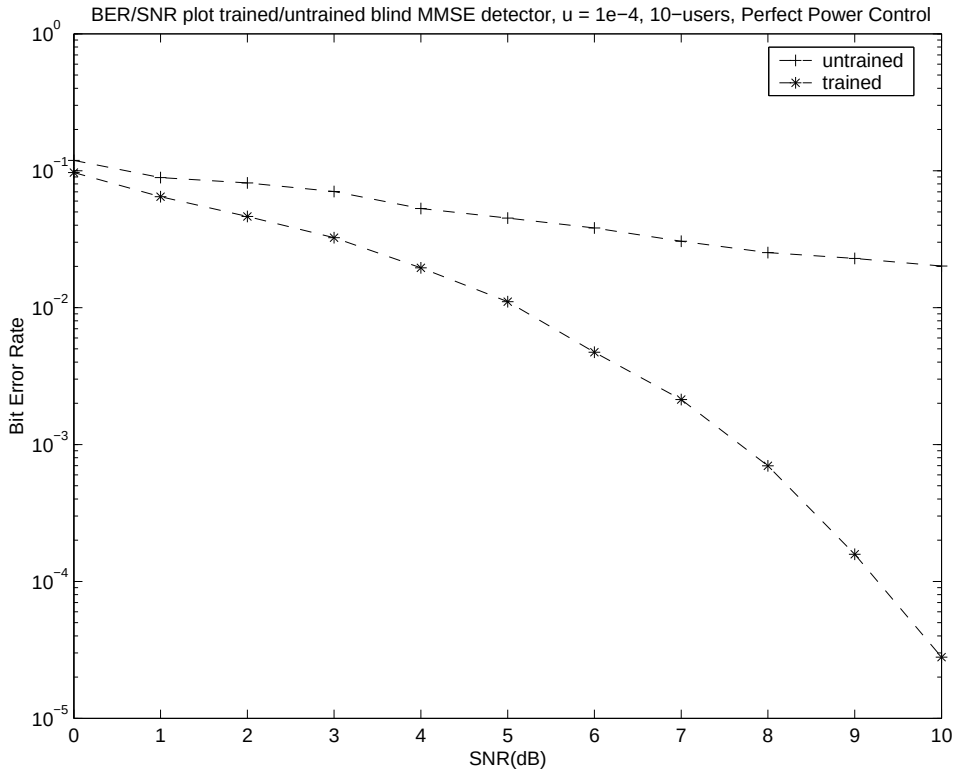


Figure 5.6: BER/SNR plot for 'trained' and 'untrained' blind adaptive MMSE detector ($\mu = 10^{-4}$, ten users, processing gain = 31, perfect power control)

That the 'training' mode is very important for the bit-error-rate performance of the blind adaptive MMSE detector can be clearly seen from Figure 5.6. In this figure two BER/SNR simulations of the blind adaptive MMSE detector with step size $\mu = 10^{-4}$ are plotted for a ten user, perfect power control CDMA system. In the first BER/SNR simulation the number of iterations for the 'convergence' and 'averaging' stages are both set to 0, so the detector is not adapted to the system when the detector is switched to 'detection' mode and simulation to determine the bit-error-rate is started. Contrary to a 'normal' BER/SNR simulation of the blind adaptive MMSE detector the x sequence of the detector now *is* updated during the simulation to determine the bit-error-rate. The second BER/SNR simulation is a 'normal' BER/SNR simulation of the blind adaptive MMSE detector in which the detector is first adapted to the system in the 'convergence' and 'averaging' stages of the 'training' mode before the bit-error-rate is determined. In this particular case 25000 iterations are chosen for the 'convergence' stage and 10000 iterations are chosen for the 'averaging' stage. The minimum number of iterations required for the 'convergence' stage can be determined from Figure 5.3(a) in which the components of the x sequence are plotted as a function of the number of iterations of the adaptation rule for the x sequence. The number of iterations for the 'convergence' stage that is used here is chosen a widely larger than the number of iterations needed for convergence of the x sequence as shown in Figure 5.3(a) just to be certain that the x sequence has converged. Figure 5.6 clearly shows for each value of the signal-to-noise ratio that has been simulated that the blind adaptive MMSE detector that is adapted to the system has a better bit-error-rate performance than the detector that has not had the chance to adapt to the system.

The answer to the question if it is realistic to assume that the blind adaptive MMSE detector is always adapted to the CDMA system and thus achieves its optimal bit-error-rate performance depends on the system the detector is used in. If for example the number of active users in the system changes often it is probably not realistic to assume that the blind adaptive MMSE detector has enough time to converge its x sequence to the optimal x sequence before the system changes again. When simulating those kinds of systems it is not realistic to have a training mode that is long enough for complete convergence of the x sequence. How long it takes to converge the x sequence depends on the data rate used in the system, since the x sequence is adapted for each received symbol. So in high data rate systems convergence is reached faster than in low data rate systems. The systems simulated in this thesis are assumed to change relatively slowly, so in those systems the blind adaptive MMSE detector will have enough time to converge. Therefor in all simulations of the blind adaptive MMSE detector in this thesis the training mode is long enough for complete convergence of the x sequence

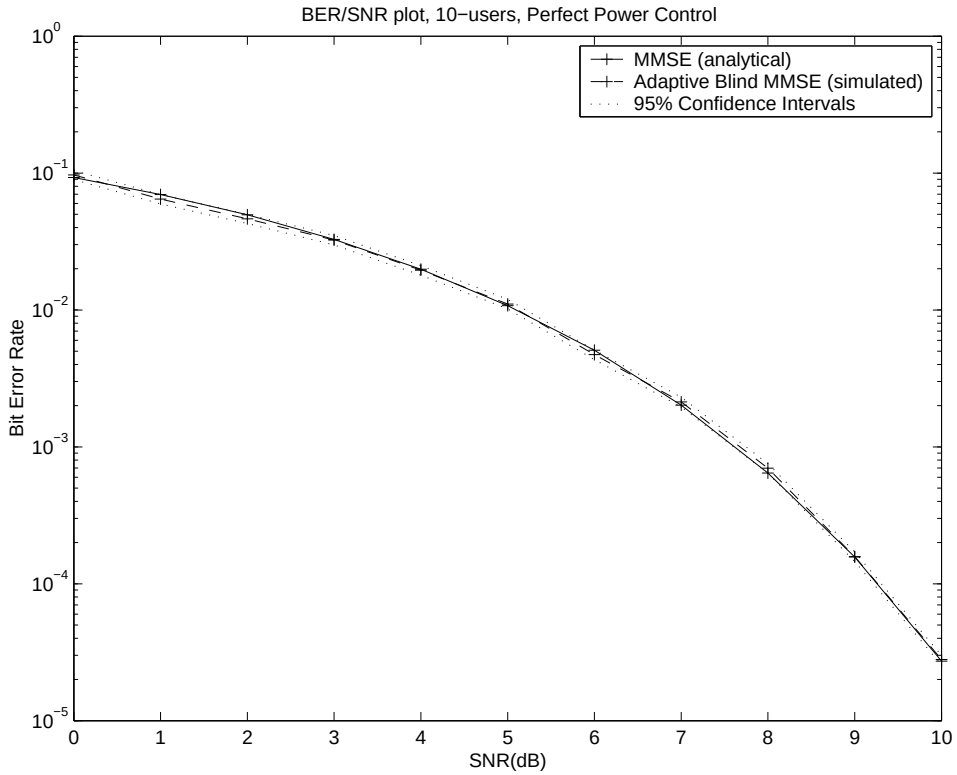


Figure 5.7: BER/SNR plot. MMSE detector and blind adaptive MMSE detector ($\mu = 10^{-4}$, ten users, processing gain = 31, perfect power control)

unless it is stated otherwise.

In Figure 5.7 the bit-error-rate of the simulated blind adaptive MMSE detector is compared with the analytical bit-error-rate of the MMSE detector (3.34) for a number of signal-to-noise ratios. The figure shows that the blind adaptive MMSE detector achieves the same bit-error-rate performance as the MMSE detector when the x sequence of the blind adaptive MMSE detector has converged to x_{opt}, the value of the x sequence that results in a global minimum of the output energy and thus in a global minimum of the mean square error.

5.2.3 PIC Detector

In this subsection the statements made about the PIC detector in section 4.2 will be verified with simulation results. One of the first statements made about the PIC detector in section 4.2 is that both implementations of the detector, the narrowband implementation and the wideband implementation, have the same theoretical performance. Since both implementations of the PIC detector are available in the simulator, this statement can be easily

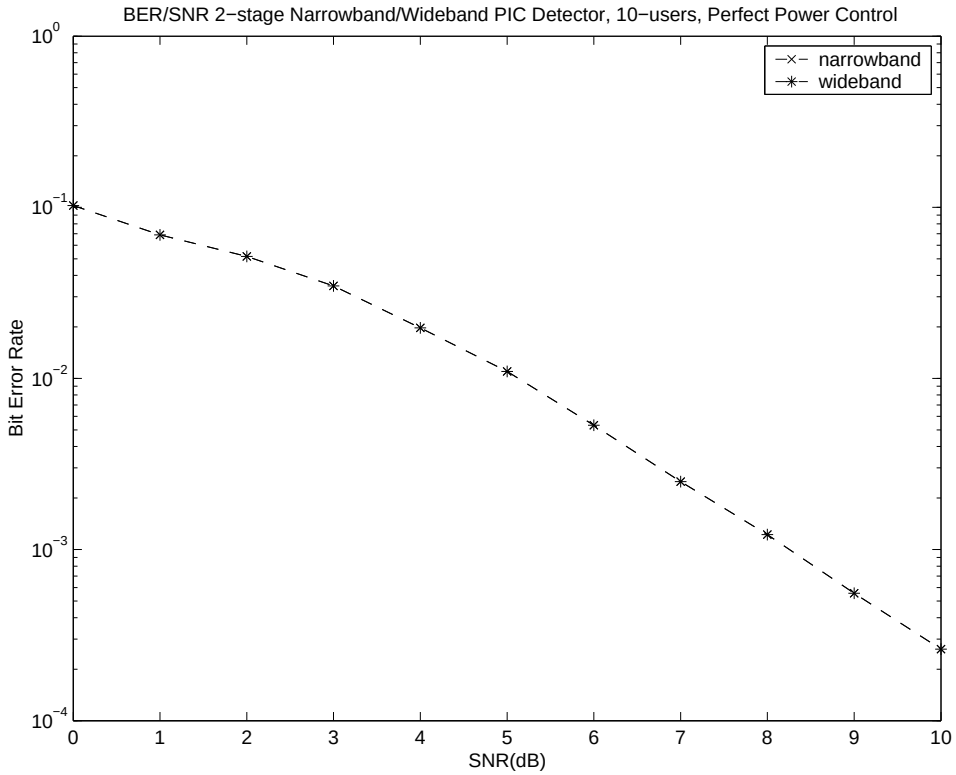


Figure 5.8: BER/SNR plot. Narrowband / wideband PIC detector (ten users, processing gain = 31, perfect power control)

verified with a simulation.

In Figure 5.8 BER/SNR simulations are shown for a 2-stage narrowband and a 2-stage wideband PIC detector used in a ten user CDMA system with perfect power control. The figure clearly shows that both detectors have exactly the same bit-error-rate performance. Since the narrowband implementation of the PIC detector requires less computations for detection of a bit than the wideband implementation, for the short code systems considered in this thesis, it is used for the rest of the simulations of the PIC detector in this section.

In Figure 5.9 the influence of the number of stages of the PIC detector on its bit-error-rate performance is studied. The figure shows the bit-error-rate as a function of the signal-to-noise ratio of a 2-stage and a 3-stage narrowband PIC detector in a ten user CDMA system with perfect power control. It can be seen that the 3-stage PIC detector achieves a lower bit-error-rate in the high signal-to-noise ratio region. This can be explained by the fact that the 3-stage PIC detector can produce more reliable estimates for the multiple access interference and therefor also more reliably cancels the multiple access interference. Since the influence of the multiple access

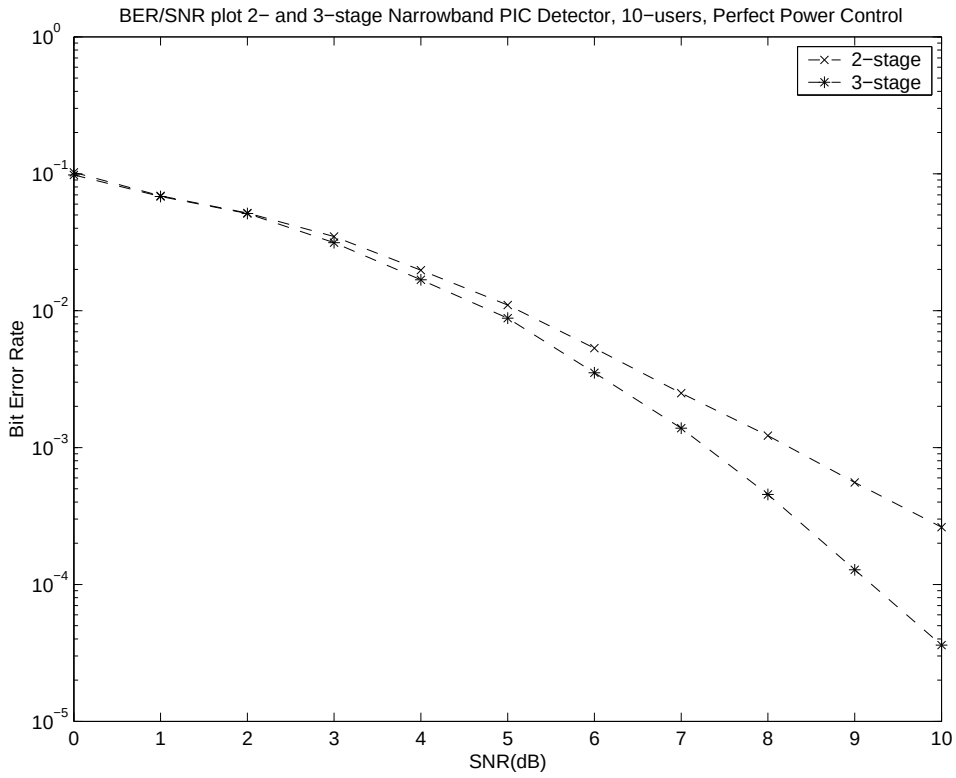


Figure 5.9: BER/SNR plot. 2-stage / 3-stage narrowband PIC detector (ten users, processing gain = 31, perfect power control)

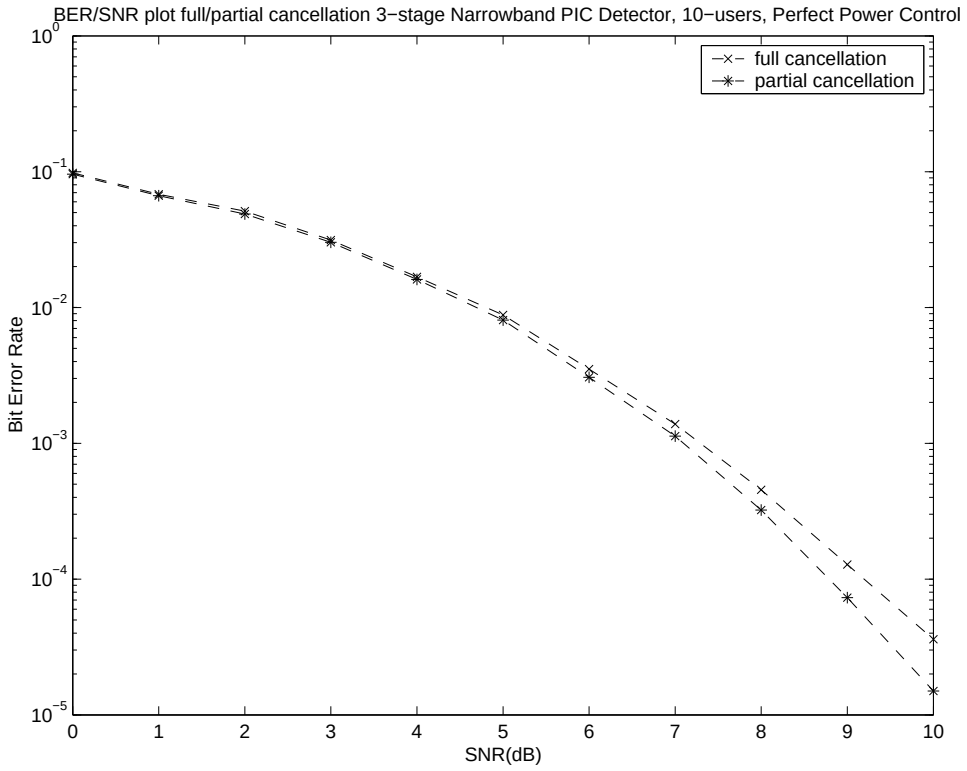


Figure 5.10: BER/SNR plot. Full-cancellation / partial-cancellation 3-stage narrowband PIC detector (ten users, processing gain = 31, perfect power control)

interference on the bit-error-rate becomes more important in the high signal-to-noise ratio region, the advantages of the 3-stage PIC detector mainly become apparent for higher signal-to-noise ratios.

In section 4.2 it was shown that the decision statistic of the PIC detector is biased, which will have a negative effect on the bit-error-rate performance. To improve this performance it was suggested to introduce partial cancellation factors into the cancellation stages of the PIC detector as proposed by Correal et al. [3]. In Figure 5.10 BER/SNR simulations are shown for two 3-stage narrowband PIC detectors in a 10 user CDMA system with perfect power control. The partial cancellation factors of both the first and the second stage of cancellation of the first detector are set to 1. So this detector in fact performs full cancellation in both of the cancellation stages. The partial cancellation factors of the first stage of cancellation of the second detector are set to 0.5. The partial cancellation factors of the second stage of cancellation of this detector are set to 1. So this detector performs partial cancellation in its first cancellation stage and full cancellation in its second cancellation stage. Figure 4.2 shows that the detector that uses par-

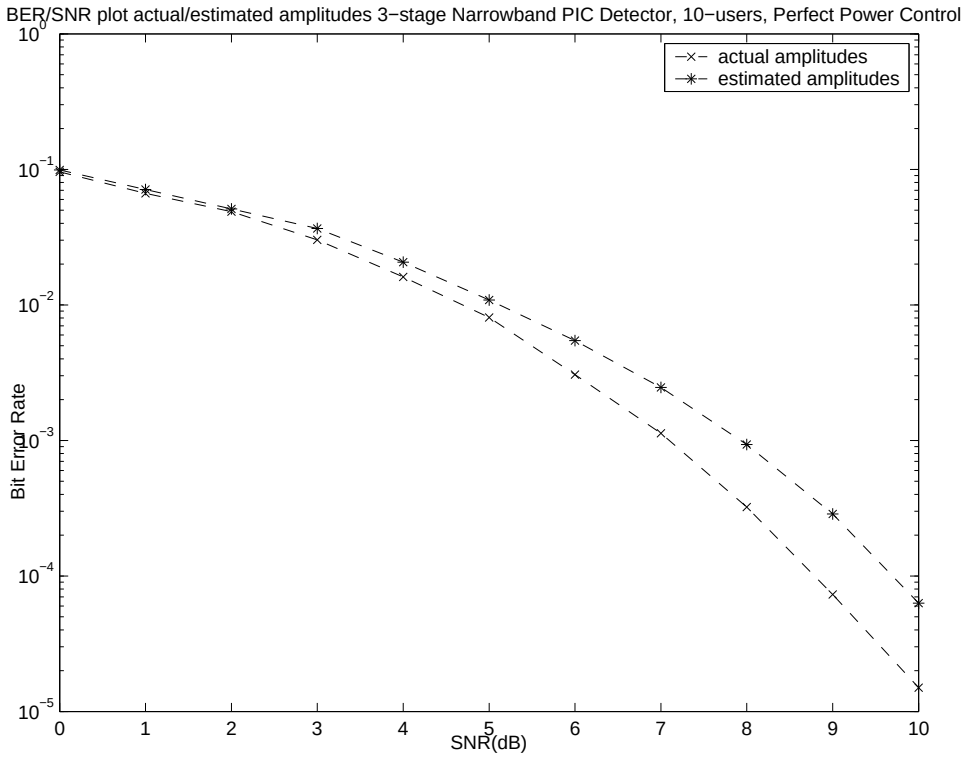


Figure 5.11: BER/SNR plot. Actual amplitudes / estimated amplitudes 3-stage partial cancellation narrowband PIC detector (ten users, processing gain = 31, perfect power control)

tial cancellation has a better performance in the high signal-to-noise ratio region. Again the fact that this improvement mainly shows in the high signal-to-noise ratio region can be explained by the fact that in that region multiple access interference becomes of more importance for the bit-error-rate. So when a detector is better capable of cancelling this multiple access interference, this will mainly show in the high signal-to-noise ratio region.

So far all the PIC detectors simulated in this section have had perfect knowledge of the received amplitudes of the signals of all the users in the system. As was already stated in section 4.2 this information is normally not directly available to the detector, but has to be estimated from the received signal. One way of doing this is using the soft decisions for each user in the system as an estimate for the received amplitude of that user. In Figure 5.11 the effect on the bit-error-rate of using estimated amplitudes instead of actual amplitudes is shown. Figure 5.11 displays the BER/SNR curves of two 3-stage narrowband PIC detectors with a partial cancellation factor of 0.5 for the first cancellation stage in a 10 user CDMA system with perfect power control. One of the detectors has perfect knowledge of the received

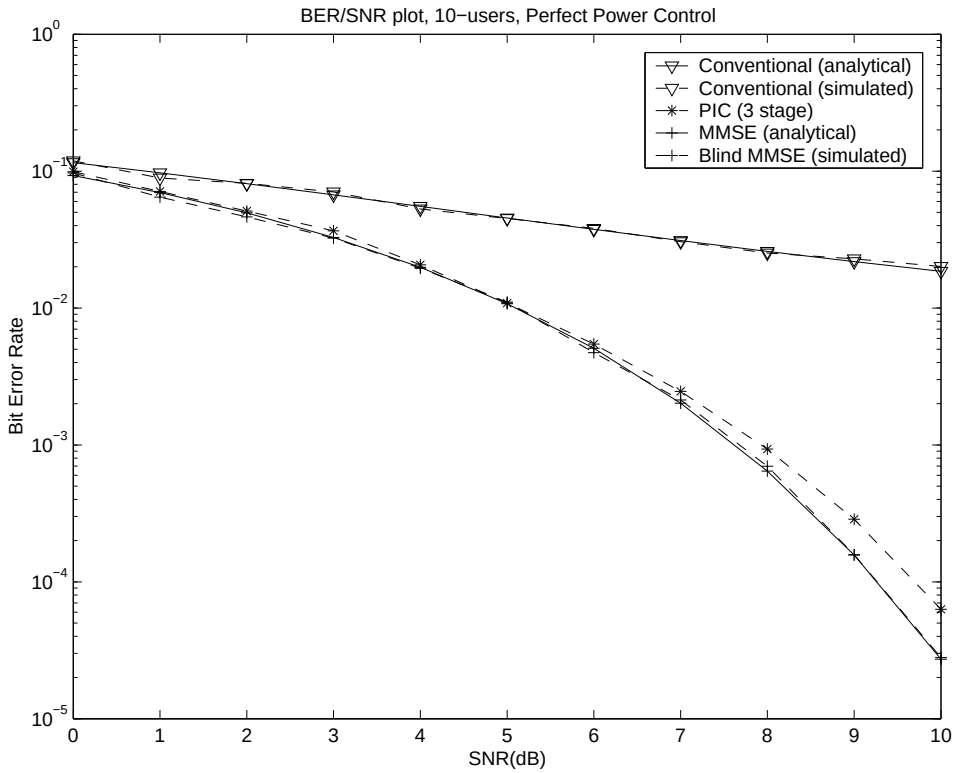


Figure 5.12: BER/SNR detector comparison with perfect power control (ten users, processing gain = 31)

amplitudes of all the users in the system, the other detector estimates the received amplitudes from the soft decisions. In this figure it can be clearly seen that using estimated amplitudes results in less reliable cancellation of multiple access interference, since the detector using estimated amplitudes has higher bit-error-rates in the high signal-to-noise ratio region than the detector using the actual amplitudes. This is caused by the fact that the soft decisions are not very reliable estimates of the received amplitudes. The estimates improve however after more stage of cancellation have been performed.

5.2.4 Detector Comparison

In this subsection the performance of the conventional, the blind adaptive MMSE and the PIC detectors will be be compared for a number of CDMA systems using simulations.

In Figure 5.12 the bit-error-rate simulation results of the conventional, blind adaptive MMSE and PIC detector arecompared for a range of signal-to-noise ratios for a 10 user CDMA system with perfect power control. For

comparison purposes the analytical bit-error-rates for the conventional and the MMSE detector are shown as well. Before the bit-error-rate is determined the simulated blind adaptive MMSE detector is adapted to the system in 'training' mode during 25000 'convergence' stage iterations and 10000 'averaging' stage iterations, using a step size $\mu = 10^{-4}$ for every value in the range of signal-to-noise ratios. The simulated PIC detector is a 3-stage narrowband PIC detector using estimated amplitudes with partial cancellation factor 0.5 for the first stage of cancellation and partial cancellation factor 1 for the second stage of cancellation. The figure clearly shows that the multiuser PIC and blind adaptive MMSE detectors achieve a far lower bit-error-rate than the conventional detector for the same signal-to-noise ratio. So using multiuser detection is even advantageous in a perfect power control system, in which the conventional detector performs a lot better than in systems without power control. When Figure 5.12 is compared to Figure 3.11, which is taken from Buehrer and others [2], it can be seen that the conventional, blind adaptive MMSE and PIC detector simulated in this section achieve similar performance as the conventional, MMSE and PIC detector simulated by Buehrer and others [2]. The small differences that are found can be explained by the fact that the exact signature sequences used by Buehrer are not known.

In Figure 5.13 the simulations of Figure 5.12 are repeated for a CDMA system without power control, in which the amplitudes of the interfering users are 20 times the amplitude of the desired user. The bit-error-rate performance of the conventional as well as the PIC detector reduces to about 0.5 for this system, while the performance of the blind adaptive MMSE detector is not affected at all. This result could already be expected from studying Figure 3.12, which is also taken from Buehrer and others [2]. Figure 3.12 shows the bit-error-rate for the desired user of a number of detectors as a function of the received energy ratio between the desired user and an interfering user. The figure clearly shows that the bit-error-rate of the conventional and the PIC detector worsens when the received energy of the interfering user becomes larger than the received energy of the desired user.

For the conventional detector this can be explained by the fact that it treats multiple access interference as noise, so when the received energy of the interfering user increases it has the same effect as an increased noise level in the system. The PIC detector however is supposed to cancel multiple access interference and should therefor not show such a bad increase in bit-error-rate when the multiple access interference increases. Unfortunately the PIC detector has to know the amplitudes of the components of the individual users in the received signal to perform cancellation. Since these amplitudes are not known at the detector they have to be estimated from the received signal. These estimates are not very reliable for the weak users in the first stage of cancellation, which can still be noticed in the later stages of cancellation and therefor also in the detection results, as already explained

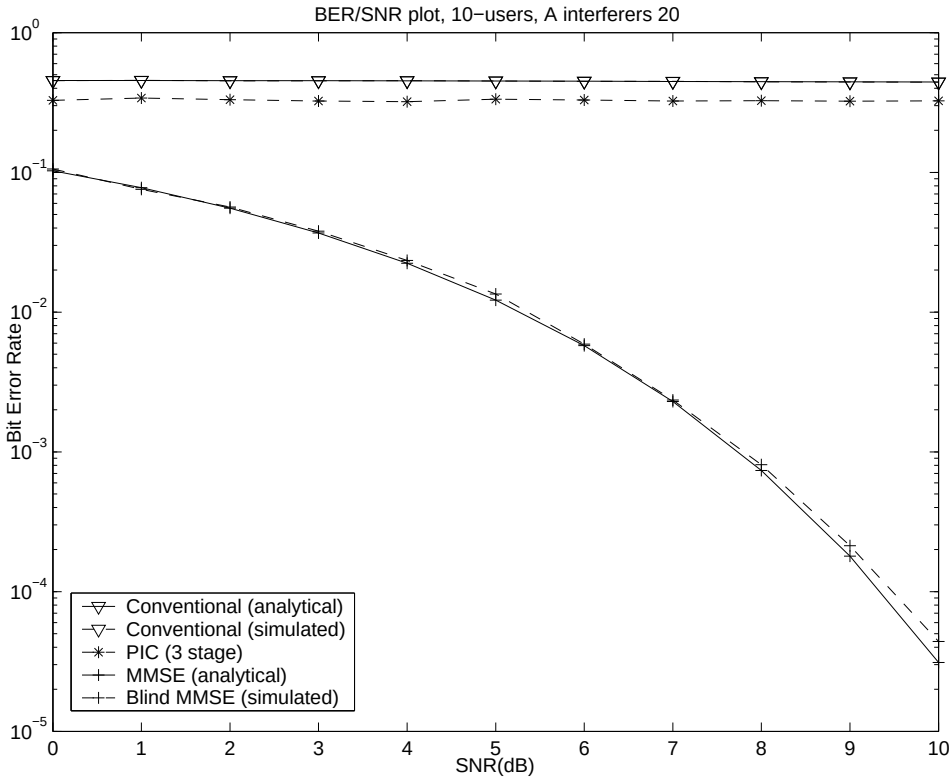


Figure 5.13: BER/SNR detector comparison with A interferers 20 (ten users, processing gain = 31)

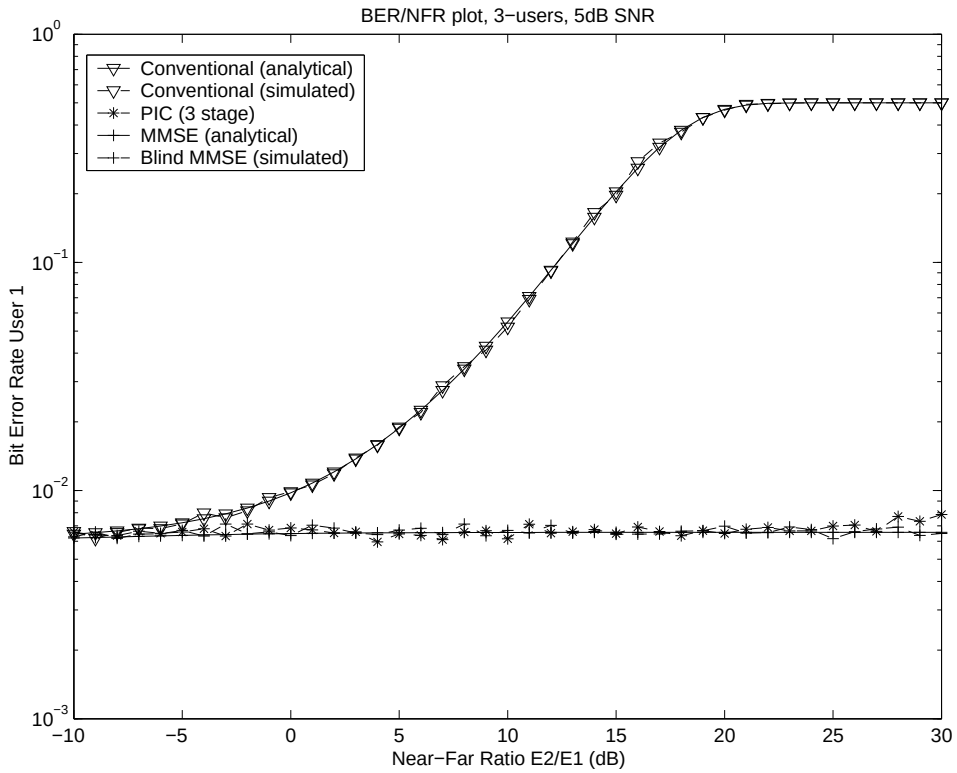


Figure 5.14: Performance degradation in near-far channels detector comparison (three users, SNR = 5dB, processing gain = 31)

in Chapter 3. This explains the high bit-error-rates of the simulated PIC detector for large interferer amplitudes. When the PIC detector in some way can obtain reliable amplitude estimates, for the small as well as the large amplitude users, the bit-error-rate of the PIC detector actually improves when the the interferer amplitudes increase [14].

Figure 5.14 shows the bit-error-rate for the desired user of the conventional, blind adaptive MMSE and PIC detector as a function of the received energy ratio between the desired user and one of the interfering users in a three user CDMA system. The received energy of the other interfering user is equal to the received energy of the desired user. So this figure is similar to Figure 3.12 and shows the performance of the detectors in near-far situations. The simulated blind adaptive MMSE detector is adapted to the system in 'training' mode for every value in the range of received energy ratios before the bit-error-rate for that received energy ratio is determined, using the same 'training' mode parameters as before. The simulated PIC detector again is a 3-stage narrowband PIC detector using estimated amplitudes with partial cancellation factor 0.5 for the first stage of cancellation and partial cancellation factor 1 for the second stage of cancellation.

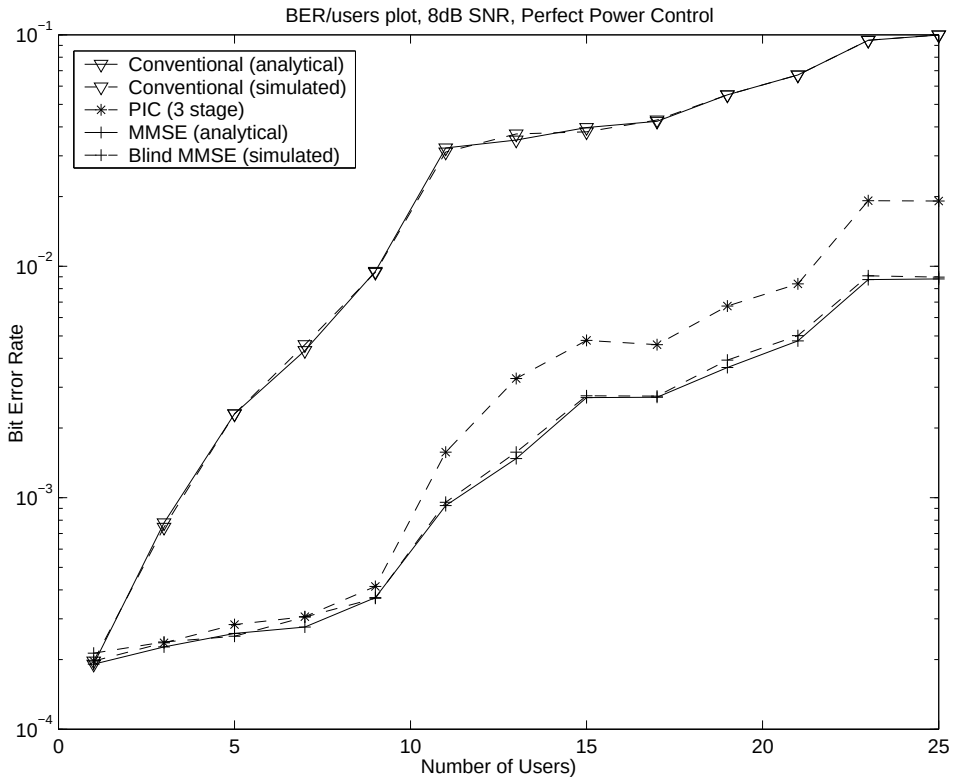


Figure 5.15: User capacity detector comparison, (SNR = 8dB, processing gain = 31, perfect power control)

Figure 5.14 shows a similar degradation in bit-error-rate for the conventional detector as Figure 3.12 for increasing interference. The degradation in bit-error-rate for the PIC detector in Figure 5.14 however is by far not as strong as in Figure 3.12.

It is assumed that this is caused by the fact that the crosscorrelation of the signature sequence of the desired user with the signature sequence of the strong interferer used to obtain Figure 3.12 is larger than the crosscorrelation of the sequences of these users used for Figure 5.14. In case of a larger crosscorrelation, more interference from the strong interferer will show up in the soft decisions of the desired user. Since the exact signature sequences used by Buehrer are not known, this can unfortunately not be proved.

Figure 5.15 shows a plot of the bit-error-rate of the simulated conventional, blind adaptive MMSE and PIC detectors as a function of the number of users in the CDMA system. The signal-to-noise ratio is set to 8dB and it is assumed that the CDMA system has perfect power control. The simulated blind adaptive MMSE detector is adapted to the system in 'training' mode for every number of users before the bit-error-rate for that number of

users is determined, using the same 'training' mode parameters as before. The simulated PIC detector again is a 3-stage narrowband PIC detector using estimated amplitudes with partial cancellation factor 0.5 for the first stage of cancellation and partial cancellation factor 1 for the second stage of cancellation. The plotted curves show that the bit-error-rates of all three detectors increase when the number of active users in the system and thus the multiple access interference increases. The plotted curves also show that the blind adaptive MMSE detector and the PIC detector (in this perfect power control situation) can better deal with the increased multiple access interference than the conventional detector. When Figure 5.15 is compared to 3.10 which is a similar plot for a similar system from Buehrer and others [2] it can be seen that both figures are quite similar up to about ten users. For larger numbers of users the bit-error-rates in Figure 5.15 increase more rapidly than those in Figure 3.10. This may be explained by the fact that the set of signature sequences used by Buehrer has lower crosscorrelations for the signature sequences of the higher numbered users.

To get an idea about how big the differences in bit-error-rate between the different detectors really are Figure 5.16 shows the results of a simulation of an image transmission by one user in a ten user CDMA system without power control. The received amplitudes of the other users are 20 times the amplitude of the user that is transmitting the image. The image is a 267 by 400 pixel uncompressed image with 24 bit RGB data per pixel, so the total image size is 2563200 bits. In Figure 5.16(a) the original image is shown.

Figure 5.16(b) shows the reconstructed image when the conventional detector is used to detect the bits of the image transmission in the CDMA system described above. The simulated conventional detector achieves a bit-error-rate of 0.44 (1135278 errors in 2563200 detected bits).

Figure 5.16(c) shows the reconstructed image when an blind adaptive MMSE detector that is not yet adapted to the system is used to detect the bits of the image transmission. The simulated blind adaptive MMSE detector does adapt its x sequence with a step size $\mu = 10^{-4}$ while the image bits are received and achieves a bit-error-rate of 0.11 (281887 errors in 2563200 detected bits).

Finally Figure 5.16(d) shows the reconstructed image when an blind adaptive MMSE detector that is adapted to the system is used to detect the bits of the image transmission. The simulated blind adaptive MMSE detector is adapted in training mode during a 25000 iterations 'convergence' stage and a 10000 iterations 'averaging' stage with a step size $\mu = 10^{-4}$ and achieves a bit-error-rate of 0.000032 (82 errors in 2563200 detected bits).

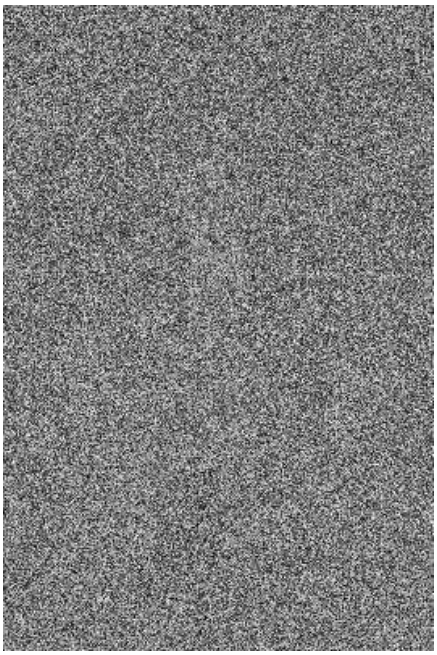
A PIC detector that uses amplitudes estimated from the soft decisions achieves a similar bit-error-rate as the conventional detector for the high interference CDMA system that is considered here. So the reconstructed image for the PIC detector would be similar to the reconstructed image of

the conventional detector and is therefor not shown in Figure 5.16.

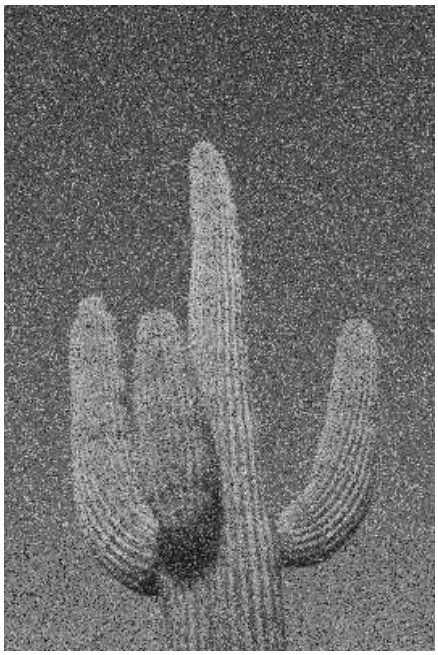
5.3 Conclusions

- A simulator for synchronous CDMA systems using the conventional, blind adaptive MMSE, or narrowband or wideband PIC detectors has been developed.
- The developed simulator can simulate the supported detectors for a variety of synchronous CDMA system and detector parameters.
- Comparison of the simulated bit-error-rate results of the conventional detector with analytically obtained bit-error-rate results shows that the simulator works correctly.
- Simulation of the conventional detector with two different sets of signature sequences shows that the bit-error-rate quite heavily depends on the used signature sequences.
- The equation for the maximum value of the step size μ for which the blind adaptive MMSE detector is stable, that was mentioned in Chapter 4, appears to be incorrect.
- The x sequence of the blind adaptive MMSE detector has to converge to the optimal x sequence x_{opt} before the detector will achieve its optimal bit-error-rate performance.
- The values for the elements of the x sequence that are obtained when adaptation of the x sequence is stopped after it has converged are not necessarily the most accurate estimation of the values of the elements of the x_{opt} sequence, because the values of the x sequence continue to 'vibrate' around the optimal values after convergence has been reached.
- In order to achieve the optimal bit-error-rate performance with the blind adaptive MMSE detector the simulator uses a two stage 'training' mode to adapt to the CDMA system before the actual simulation is performed.
- It has been shown that this 'training' mode has a large influence on the bit-error-rate simulation results that are obtained for the blind adaptive MMSE detector.
- When the blind adaptive MMSE detector is adapted to the CDMA system it achieves the same bit-error-rate performance as the MMSE detector.

(a) Original image



(b) Conventional detector



(c) 'Untrained' blind adaptive MMSE detector

(d) 'Trained' blind adaptive MMSE detector

Figure 5.16: Simulation of transmission of an image (ten users, SNR = 10dB, A interferers 20).

- The simulation results show that the narrowband and the wideband implementation of the PIC detector achieve exactly the same bit-error-rate performance.
- The bit-error-rate performance of the PIC detector can be improved by adding additional cancellation stages.
- Experiments with a partial cancellation factor in the first cancellation stage of the PIC detector show that the bit-error-rate performance can be improved by using a partial cancellation factor.
- Simulation results show that amplitude estimation from the soft decisions stages has a quite large negative influence on the bit-error-rate performance of the PIC detector. So there is room for improvement by using better amplitude estimation techniques.
- Using the blind adaptive MMSE and PIC multiuser detectors is even advantageous in perfect power control CDMA systems in which the conventional detector achieves its optimal performance.
- The performance of the blind adaptive MMSE detector is hardly influenced by different received amplitudes.
- The PIC detector however shows disappointing performance in case of different received amplitudes, because of the fact that the amplitude estimation technique it uses becomes unreliable in case of strongly differing received amplitudes. This effect so far has not been studied extensively in literature.
- The bit-error-rate of all three of the detectors increases when the number of active users in a perfect power control CDMA system increases. The blind adaptive MMSE and PIC detector however can support a larger number of active users in the system while still achieving the same bit-error-rate as the conventional detector for a smaller number of active users. This shows that the PIC detector handles multiple access interference better than the conventional detector as long as the received amplitudes do not differ strongly.

The converged blind adaptive MMSE detector and the PIC detector achieve almost similar bit-error-rate performance in perfect power control CDMA systems. In systems with different received amplitudes the bit-error-rate performance of the converged blind adaptive MMSE detector is far better than the bit-error-rate performance of the PIC detector. This is the first reason for choosing the blind adaptive MMSE detector for DSP implementation. The second reason for choosing the blind adaptive MMSE detector for DSP implementation is that no existing DSP implementations of this detector are known, while DSP implementations of the PIC do exist.

Chapter 6

Blind Adaptive MMSE DSP Implementation

In this chapter implementation of the blind adaptive MMSE detector on the Texas Instruments 'C6711 DSP starter kit is described. The blind adaptive MMSE detector is chosen for DSP implementation for two reasons, as already stated in the previous chapter. The first reason is that in systems with different received amplitudes the bit-error-rate performance of the converged blind adaptive MMSE detector is far better than the bit-error-rate performance of the PIC detector. The second reason for choosing the blind adaptive MMSE detector for DSP implementation is that no existing DSP implementations of this detector are known, while DSP implementations of the PIC do exist.

Considering the time-frame of the project it was not possible to create a real-time implementation of the blind adaptive MMSE detector that operates on real-time generated received signals. Instead it was chosen to study the detected-bits-per-second performance of the blind adaptive MMSE detector that can be achieved on a 'C6711 DSP. This performance test consists of detection on a received CDMA signal that is stored in memory on the DSP board. During the detection process the blind adaptive MMSE detector code can be timed to obtain the detected-bits-per-second performance. The detected bits are also stored in memory on the DSP board so that they can be compared with, for example, the detected bits of the simulator for the same received CDMA signal. This way it can be ensured that the DSP implementation of the blind adaptive MMSE detector is functionally correct.

In the next section the 'C6711 DSP starter kit will be described briefly. After that the overall implementation of the blind adaptive MMSE performance test is discussed. Then the architecture of the 'C6711 DSP is explained in order to understand how the blind adaptive MMSE detector algorithm can be optimized for the 'C6711 DSP. Optimization of the blind adaptive MMSE detector algorithm for the 'C6711 DSP is discussed after

that. The last section of this chapter presents the conclusions that can be drawn from the DSP implementation of the blind adaptive MMSE detector.

6.1 C6711 DSP Starter Kit

The Texas Instruments 'C6711 DSP starter kit is a DSP board containing an 150 MHz Texas Instruments TMS320C6711 floating point digital signal processor (DSP), 16 MB SDRAM, 128 KB Flash memory and a Texas Instruments TLC320AD535 16-bit data converter. The DSP starter kit can be connected to a PC trough a parallel port cable.

The 'C6711 DSP starter kit is accompanied by a number of development tools consisting of a C compiler, assembler, linker and an integrated development environment containing a visual debugger and profiler. The provided C compiler provides useful feedback about how C code can be further optimized and can achieve performances exceeding 70% of the performance of similar code written in hand optimized assembly [4].

The 'C6711 DSP Starter Kit was chosen because it is one of the DSP platforms available at the Laboratory of Signals & Systems that satisfies the two main requirements for implementation of the blind adaptive MMSE performance test: a floating point DSP and a relatively large memory. Implementation of the blind adaptive MMSE detector does not require a floating point processor, but it simplifies and speeds up the development process because fixed point arithmetic precision does not have to be taken into account. A fixed point implementation of the algorithm can than always be derived from the working floating point implementation. The DSP platform requires a relatively large memory because the received signal and the detected bits have to be stored.

6.2 Blind Adaptive MMSE Performance Test Implementation

The blind adaptive MMSE performance test consists of two C programs, one running on the DSP and one running on the PC to which the DSP is connected. These two programs communicate trough real-time data exchange (RTDX) [7]. RTDX enables transmission and reception of data between a host computer (PC) and a DSP target system using so called RTDX channels without stopping the applications running on the PC and the DSP.

The task of the PC program is to read received CDMA signal data from disk and transfer it to the DSP and to retrieve the detected bits from the DSP and store them in a file on the PC. The received CDMA signal data is generated by the simulator described in the previous chapter. The simulator code has been modified to include an option to save the received signal to disk as a binary stream of floating point numbers. In addition the

simulator saves the originally transmitted bits and the detected bits of the simulated detector. In this way, the detected bits of the DSP detector and the simulated detector can be compared to each other and to the originally transmitted bits.

The DSP program takes care of the DSP side of the RTDX communication process, stores the received signal in memory and of course performs blind adaptive MMSE detection. The implementation of the blind adaptive MMSE detector on the DSP always operates in the mode in which the x sequence is updated, this in contrast to the simulator (see subsection 5.1.2). This mode of the blind adaptive MMSE detector is the most computationally intensive and therefor limits the detected-bits-per-second performance. So, for a fair comparison of the detection results, the DSP blind adaptive MMSE detector has to be compared with a simulated blind adaptive MMSE detector that has not been 'trained' and updates its x sequence during detection.

To simplify the code of the blind adaptive MMSE detector for the DSP, it is limited to using length 31 spreading sequences. By default the detector uses the first spreading sequence that is generated by the linear shift feedback register that was used to generate the spreading sequences for the simulations described in Chapter 5. This spreading sequence is hard-coded into the C program that is running on the DSP. So, in order to change the used spreading sequence the DSP program has to be recompiled.

Using RTDX turned out to be quite cumbersome because the RTDX channels have limitations for the amount of data that can be transferred in one RTDX data transfer. Since the received signal of, for example, a bit-error-rate simulation for a number of signal-to-noise ratios, is a few megabytes large, the received signal has to be transferred to the DSP in a number of RTDX transfers. Since the overhead of an RTDX data transfer appears to be quite large, transferring the received signal to the DSP over the parallel port interface takes quite some time. RTDX also limits the possibilities of profiling the code on the DSP, since it appears to use some of the same resources as the profiler. So either RTDX or the profiler can be active. Even with these limitations RTDX still appeared to be the only way to transfer a relatively large amount of data from the PC to the memory on the DSP board and back when the DSP board is connected to the PC using a parallel port or JTAG interface. When an internal PCI DSP board is used there are probably other and better ways of transferring data between the PC and the DSP.

6.3 'C6711 Architecture

In this section the 'C6711 architecture and instructions are summarized. This summary is limited to the information that is needed for implementa-

tion and optimization of the blind adaptive MMSE detector algorithm for the 'C6711 DSP. More information about the architecture of the TMS320C6000 series of DSPs to which the 'C6711 belongs can be found in [12] or in [4].

As with conventional processors, the 'C6711 is composed of three main parts: the Central Processing Unit (CPU), memories and peripherals, all connected by internal buses. In addition the 'C6711 has an External Memory Interface (EMIF) for connection to common memory devices and also a Host Port Interface (HPI).

The CPU, which is the heart of the DSP, is on its turn composed of four elements: the program control unit, two data paths, control registers and test, emulation, control and interrupt logic. The program control unit fetches, dispatches and decodes instructions. The two data paths are then used to execute these instructions. The control registers are used for interrupt control and to support floating point operations.

The two data paths in the CPU are known as data path 1 and data path 2. Each data path has four execution units known as .L, .M, .S, and .D, a register file containing 16 32-bit general-purpose registers and multiple busses for data communication between each data path and memory, data communication within each data path and data communication between the two data paths. See Figure 6.1. Since the register files of data path 1 and 2 are known as respectively register file A and register file B, data path 1 is also referred to as data path A and data path 2 is also referred to as data path B. These names for the data paths are used in Figure 6.1.

In the execution units of each data path the instructions are executed. Execution of an instruction is pipelined on a pipeline with ten phases. Different types of instruction require a different number of these phases and thus a different number of clock cycles to complete their execution. It is however possible to schedule instructions so that execution of an instruction on an execution unit starts just one cycle after execution of the previous instruction on the same execution unit has started. This way each execution unit can output a result every cycle as long as there are enough instructions that can be scheduled in such a way that the pipelines of all the execution units remain completely filled.

All four execution units in a data path operate on 32-bit operands and execute instructions simultaneously. However, the .L and .S units can also operate on 40-bit operands. Each unit executes a specific set of operations, which can be seen as a limitation because, for example, only one multiplication per cycle can be executed on a data path.

The .L units (.L1 for datapath 1 and .L2 for datapath 2) are 40-bit integer and 32-bit floating point Arithmetic and Logic Units (ALUs). These two units can be used for:

- 32/40-bit integer arithmetic and compare operations

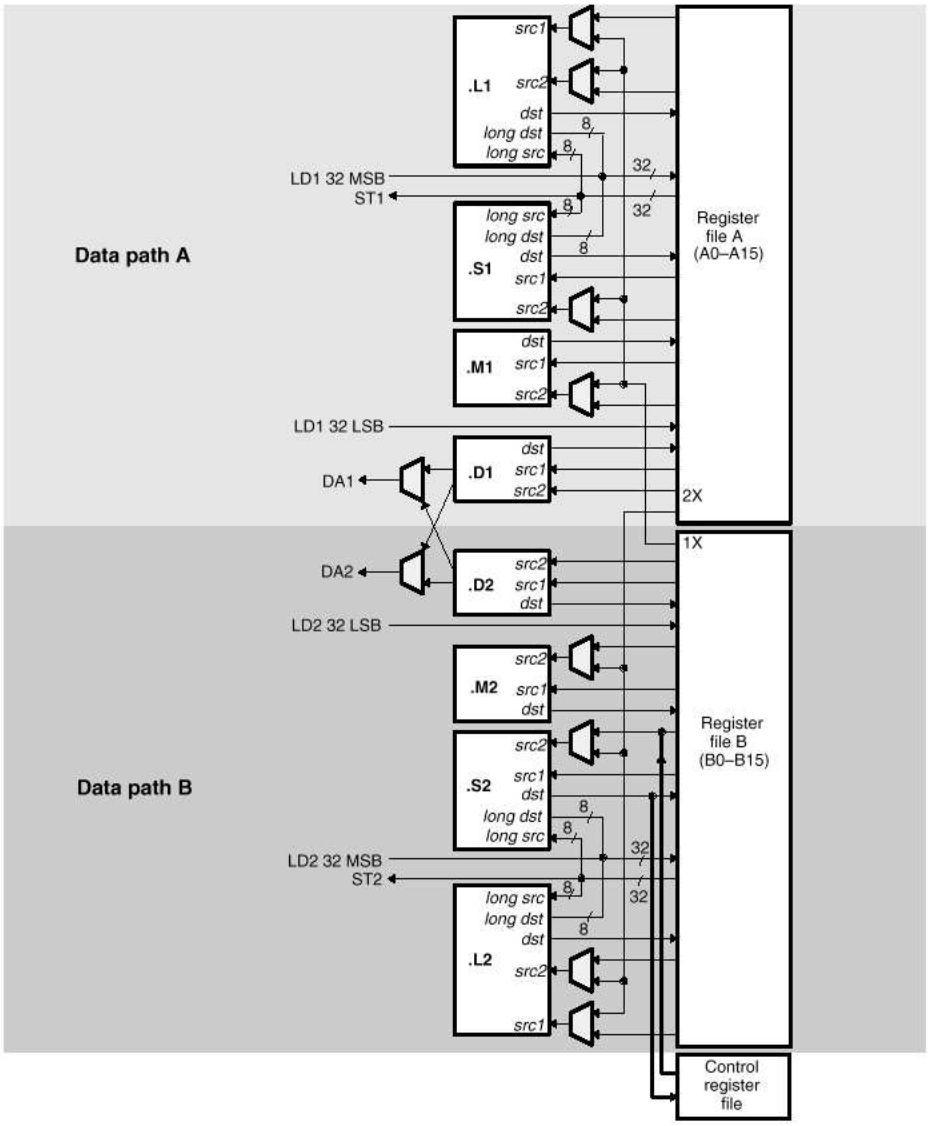


Figure 6.1: 'C67xx CPU Data Paths

- single/double precision floating point arithmetic operations
- integer to floating point and floating point to integer conversions
- 32-bit logical operations
- normalization and bit count operations
- saturated arithmetic for 32/40-bit operations.

The .M units (.M1 and .M2) are two hardware multiplier units capable of performing 16-bit by 16-bit signed and unsigned integer multiplications and single and double precision floating point multiplications.

The .S units (.S1 and .S2) contain 32-bit integer ALUs and 40-bit shifters. These units can be used for:

- 32-bit integer arithmetic, logic and bit field operations
- single/double precision floating point compare operations
- single/double precision floating point absolute value operations
- single/double precision floating point reciprocal/square root reciprocal estimation
- 32/40-bit shifts
- branches (.S2 only when using a register)
- register transfers to and from the control registers (.S2 only)
- Constant generation.

The data units (.D1 and .D2) can be used for the following operations:

- Load and store with 5-bit constant offset
- Load and store with 15-bit constant offset (.D2 only)
- Load double word (64-bit data) with 5-bit constant offset
- 32-bit additions/subtractions
- Linear and circular address calculations.

Each data path contains a register file composed of 16 32-bit general purpose registers (A0-A15 for data path 1 and B0-B15 for data path 2). These registers can support 32- and 40-bit integer or fixed point data and 64-bit double precision floating point data. To create 40- or 64-bit operands two registers have to be concatenated; these registers must be from the same data path and ordered as odd (MSB) first and even (LSB) second. The general-purpose registers can be used for:

Instruction	Cycles	.L	.M	.S	.D
MPYSP	4		√		
ADDSP	4	√			
SUBSP	4	√			
SUB	1	√		√	
B	6			√	
LDW	5				√
LDDW	5				√
STW	1				√
CMPLTSP	1			√	

Table 6.1: Overview of clock cycles and execution units for the instructions used by the blind adaptive MMSE detector algorithm.

- data
- data address pointers
- conditional registers

The blind adaptive MMSE detector algorithm uses the floating point multiply and add operations to calculate the matched filter and adaptive filter outputs. To calculate the updated x sequence in addition the floating point subtract operation is needed. In contrast to the simulator the blind adaptive MMSE performance test uses single precision floating point numbers, therefor for all these operations the single precision variant is used. So the blind adaptive MMSE detector implementation used by the blind adaptive MMSE performance test uses the following floating point arithmetic instructions: MPYSP, ADDSP and SUBSP.

Calculation of the matched filter outputs, the adaptive filter outputs and the updated x sequence all require loops that execute a fixed number of times. These loops can be implemented using the integer subtract instruction (SUB) and the branch instruction (B).

To load the received signal and the filter coefficients from memory the load word instruction (LDW) is used. On the 'C6711 DSP it is possible to load two single-precision floating point numbers at once using the load double word instruction (LDDW). To store the updated x sequence the store word instruction (STW) is used.

The detected bit that the blind adaptive MMSE detector returns depends on the sign of the output value of the adaptive filter. This can for example be implemented by using the single-precision floating point compare-for-less-than instruction (CMPLTSP).

Table 6.1 indicates for each of these instructions the number of clock cycles they take to complete and the execution units they can be executed

on. This table can be used as an easy reference during optimization of the blind adaptive MMSE detector algorithm in the next section.

6.4 Blind Adaptive MMSE Detector Optimization

In this section optimization of the blind adaptive MMSE detector algorithm for the 'C6711 DSP is described. All C code in this section is compiled using the C compiler that accompanies the 'C6711 DSP starter kit with optimizations enabled. After each optimization step the blind adaptive MMSE detector algorithm is tested by comparing the detected bits that the algorithm generates with the detected bits generated by the simulator to ensure that the optimized blind adaptive MMSE detector algorithm still functions correctly.

The blind adaptive MMSE detector algorithm is described by equations (4.24), (4.25), (4.26) and (4.27) in Chapter 4. A direct translation of these equations into C code leads to the following code:

```
int sgn(float a) {
    if (a >= 0)
        return 1;
    else
        return -1;
}

int blind_adaptive_mmse(float *r, float *s, float *x) {
    float   Zmf, Z;
    int     c;

    Zmf = 0;
    Z = 0;
    for (c = 0; c < NUM_CHIPS; c++)
        Zmf += r[c] * s[c];
    for (c = 0; c < NUM_CHIPS; c++)
        Z += r[c] * (s[c] + x[c]);
    for (c = 0; c < NUM_CHIPS; c++)
        x[c] = x[c] - 2 * MU * Z * (r[c] - Zmf * s[c]);
    return sgn(Z);
}
```

The `r` pointer points to an array of 31 floats, containing samples of the received signal, `s` points to an array of 31 floats containing the signature sequence and `x` points to an array of 31 floats in which the x sequence is stored. The variable `Zmf` is used to store the output of the matched filter and the variable `Z` is used to store the output of the blind adaptive MMSE

Instruction	Cycles
function overhead + prolog loop 1	19
loop 1	9 * 7 = 63
epilog loop 1 + prolog loop 2	39
loop 2	8 * 5 = 40
epilog loop 2 + prolog loop 3	60
loop 3	6 * 10 = 60
epilog loop 3 + function overhead	47
Total	328

Table 6.2: Clock cycles used by different parts of the blind adaptive MMSE detector algorithm.

filter. `NUM_CHIPS` is a constant with value 31, the number of chips of the used signature sequence. `MU` is a constant with value 10^{-4}, the step size of the blind adaptive MMSE detector.

The number of clock cycles that this implementation of the blind adaptive MMSE detector algorithm takes to execute on the 'C6711 can be found by counting the number of cycles used by the `blind_adaptive_mmse` function in the assembly code that the C compiler outputs. To obtain the correct number of total used cycles the number of cycles used in loops have of course to be multiplied by the number of times the loop is executed. Since the C compiler often unrolls loops the number of times the assembly loop is executed is often smaller than the number of times the C loop is executed. The number of cycles used by the `blind_adaptive_mmse` function above is split out over the different parts of the function in table 6.2.

The code above is not optimal for a number of reasons. A first optimization can be seen by realizing that the second loop can also be written as:

```
for (c = 0; c < NUM_CHIPS; c++)
    Z += r[c] * x[c];
Z += Zmf;
```

This greatly reduces the number of additions required to calculate `Z`. A second optimization is to combine the first and the second loop into one loop. This is possible since the first and second loop both run over the number of chips and the second loop is not dependent on the result of the first loop. Combining the first and second loop into one loop reduces the loop overhead and thus makes the code more efficient. A third optimization can be realized by reducing the number of calculation performed in the third loop by moving the calculation of `2 * MU * Z` out of this loop. This is possible because both `MU` and `Z` are independent of the loop variable `c`. A fourth and last optimization can be realized by replacing the call to the `sgn` function with the code of this function, which removes the call overhead. These last

two optimizations are probably performed as well by the C compiler when optimizations are turned on. Implementing all these optimizations results in the following code:

```
int blind_adaptive_mmse(const float * restrict r,
                        const float * restrict s,
                        float * restrict x) {
    float   Zmf, Z, MU2Z;
    int     c;

    Zmf = 0;
    Z = 0;
    for (c = 0; c < NUM_CHIPS; c++)
    {
        Zmf += r[c] * s[c];
        Z += r[c] * x[c];
    }
    Z += Zmf;
    MU2Z = 2 * MU * Z;
    for (c = 0; c < NUM_CHIPS; c++)
        x[c] -= MU2Z * (r[c] - Zmf * s[c]);
    if (Z >= 0)
        return 1;
    else
        return -1;
}
```

The `const` keywords in the function declaration indicate to the C compiler that the values pointed to by the `r` and `s` pointers are not modified by the function. The `restrict` keywords in the function declaration indicate to the C compiler that the `r`, `s` and `x` pointers point to different objects in memory that do not overlap. This helps the compiler to optimize memory access, because it now knows that instructions that are accessing memory trough different pointers are not accessing the same memory location and can therefor be executed in parallel.

The number of cycles used by the optimized `blind_adaptive_mmse` function is split out over the different parts of the function in table 6.3. The table shows that the second loop in the optimized C code uses the same number of cycles as the third loop in the original C code indicating that the compiler did indeed already perform the two optimizations for the third loop of the original code. The first loop of the optimized C code however only takes 63 cycles to execute, while it performs the work of the first and the second loop of the original code.

To further optimize the blind adaptive MMSE detector algorithm for the 'C6711 DSP the lengths of the arrays used to store the samples of the

Instruction	Cycles
function overhead + prolog loop 1	18
loop 1	9 * 7 = 63
epilog loop 1 + prolog loop 2	53
loop 2	6 * 10 = 60
epilog loop 2 + function overhead	47
Total	241

Table 6.3: Clock cycles used by different parts of the optimized blind adaptive MMSE detector algorithm.

received signal, the signature sequence and the x sequence all have to be increased by one. Since these arrays are now an even number of floats long (32 to be exact) the LDDW instruction of the 'C6711 DSP can now be used to read two floating point numbers from these arrays at once. To make the C compiler produce code that uses the LDDW instruction the code of the blind adaptive MMSE algorithm has to be modified as follows:

```
int blind_adaptive_mmse(const double * restrict r,
                        const double * restrict s,
                        double * restrict x) {
    float   Zmf0, Z0, Zmf1, Z1, MU2Z;
    int     c;

    Zmf0 = 0;
    Z0 = 0;
    Zmf1 = 0;
    Z1 = 0;
    for (c = 0; c < ((NUM_CHIPS + 1) / 2); c++)
    {
        Zmf0 += _itof(_hi(r[c])) * _itof(_hi(s[c]));
        Zmf1 += _itof(_lo(r[c])) * _itof(_lo(s[c]));
        Z0 += _itof(_hi(r[c])) * _itof(_hi(x[c]));
        Z1 += _itof(_lo(r[c])) * _itof(_lo(x[c]));
    }
    Zmf0 += Zmf1;
    Z0 += Z1;
    Z0 += Zmf0;
    MU2Z = MU2 * Z0;
    for (c = 0; c < ((NUM_CHIPS + 1) / 2); c++)
    {
        x[c] = _itod(_ftoi(_itof(_hi(x[c])) - MU2Z *
            (_itof(_hi(r[c])) - Zmf0 * _itof(_hi(s[c])))),
            _ftoi(_itof(_lo(x[c])) - MU2Z *
```

Instruction	Cycles
function overhead + prolog loop 1	13
loop 1	7 * 16 = 112
epilog loop 1 + prolog loop 2	27
loop 2	8 * 8 = 64
epilog loop 2 + function overhead	41
Total	257

Table 6.4: Clock cycles used by different parts of the optimized blind adaptive MMSE detector algorithm using double reads.

```
            (_itof(_lo(r[c])) - Zmf0 * _itof(_lo(s[c]))));
    }
    if (Z0 >= 0)
        return 1;
    else
        return -1;
}
```

The pointers in the function declaration are changed to double pointers to read two float values as one double from the received signal, signature sequence and x sequence arrays. Since in the loop two float values are read at once, the loop counters have to be divided by two and the operation in the loop has to be performed twice. The code in the loop uses the `_hi()`, `_lo()` and `_itof()` intrinsics to access the individual float values in the read doubles. The `_ftoi()` and `_itod()` intrinsics are used to store two float values in one double to update the x sequence. Intrinsics usually are special functions that map directly to inlined 'C6711 instructions, but the specific intrinsics used here do not map to any instructions but provide direct access to the two individual 32-bit components in a 64-bit value and circumvent the C type system. More information about intrinsics can be found in the TMS320C6000 Programmer's Guide [13].

The number of cycles used by the optimized `blind_adaptive_mmse` function using double reads is split out over the different parts of the function in table 6.4. The table shows that the number of cycles used by the algorithm has increased compared to the previous version. This is caused by the fact that the first loop almost uses double the number of cycles compared to the previous version. The second loop together with its epilog actually uses less cycles as the second loop plus epilog in the previous version.

Studying the information provided in the assembly output of the first loop of the code shows why the loop takes so many cycles. The first reason is that the compiler for some reason has not unrolled the loop. The second reason is that the compiler did not optimally divide the instructions over the available execution units:

```
;*      Resource Partition:
;*                                  A-side   B-side
;*      .L units                      3*       1
;*      .S units                      0        1
;*      .D units                      2        1
;*      .M units                      3*       1
;*      .X cross paths                1        1
;*      .T address paths              2        1
```

The execution units on data path A are used more than the execution units on data path B. This is mainly caused by the uneven number of instructions in the loops that have to be executed on the .D units with the result that the .D units are used an uneven number of times.

The TMS320C6000 Programmer's Guide [13] suggest unrolling the loop as a solution for uneven used resources. Unrolling the first loop results in the following code for the unrolled first loop:

```
Zmf0 = 0;
Z0 = 0;
Zmf1 = 0;
Z1 = 0;
Zmf2 = 0;
Z2 = 0;
Zmf3 = 0;
Z3 = 0;
for (c = 0; c < ((NUM_CHIPS + 1) / 4); c+=2)
{
    Zmf0 += _itof(_hi(r[c])) * _itof(_hi(s[c]));
    Zmf1 += _itof(_lo(r[c])) * _itof(_lo(s[c]));
    Z0 += _itof(_hi(r[c])) * _itof(_hi(x[c]));
    Z1 += _itof(_lo(r[c])) * _itof(_lo(x[c]));
    Zmf2 += _itof(_hi(r[c + ((NUM_CHIPS + 1) / 4)])) *
        _itof(_hi(s[c + ((NUM_CHIPS + 1) / 4)]));
    Zmf3 += _itof(_lo(r[c + ((NUM_CHIPS + 1) / 4)])) *
        _itof(_lo(s[c + ((NUM_CHIPS + 1) / 4)]));
    Z2 += _itof(_hi(r[c + ((NUM_CHIPS + 1) / 4)])) *
        _itof(_hi(x[c + ((NUM_CHIPS + 1) / 4)]));
    Z3 += _itof(_lo(r[c + ((NUM_CHIPS + 1) / 4)])) *
        _itof(_lo(x[c + ((NUM_CHIPS + 1) / 4)]));
}
Zmf0 = Zmf0 + Zmf1 + Zmf2 + Zmf3;
Z0 = Z0 + Z1 + Z2 + Z3;
Z0 += Zmf0;
```

The rest of the code remains the same as the code in the optimized

Instruction	Cycles
function overhead + prolog loop 1	21
loop 1	9 * 4 = 36
epilog loop 1 + prolog loop 2	36
loop 2	8 * 8 = 64
epilog loop 2 + function overhead	41
Total	198

Table 6.5: Clock cycles used by the optimized blind adaptive MMSE detector algorithm using double reads with unrolled first loop

version of the blind adaptive MMSE algorithm that uses double reads. The number of cycles used by the optimized `blind_adaptive_mmse` function using double reads with an unrolled first loop is given in table 6.5. So manually unrolling the first loop results in the loop taking only 36 cycles to execute. The table actually shows that the compiler unrolled loop 1 one more time, since the loop code is only executed 4 times.

The first loop of the optimized blind adaptive MMSE algorithm in total has to perform 64 multiplications (2 multiplications for each of the 32 chips in the signature sequence). The 'C6711 contains two hardware multipliers that can deliver one multiplication result per cycle each when their pipelines are kept full constantly. So the theoretical minimum number of cycles in which the first loop of the optimized blind adaptive MMSE detector can execute is 32 cycles. The obtained 36 cycles for this loop is quite close to this optimum and further optimization of this loop is not useful.

There might still be some room for improvement for the second loop of the double read version of the optimized blind adaptive MMSE algorithm. The information for this loop provided by the compiler in the assembly output shows that the used resources are not divided equally over data path A and data path B. There are however no C optimization tricks left that might improve the resource allocation. It is possible to rewrite the loop in assembly which makes manual resource allocation possible and might result in a further optimized loop. The time constraints of the project this thesis describes unfortunately made further exploration in this direction impossible.

6.5 Detected Bits Per Second Performance

Time constraints and difficulties with profiling severely limited the detected bits per second performance analysis that could be performed on the blind adaptive MMSE algorithm optimized for the 'C6711 DSP. In this section the few results for the detected bits per second performance that could be obtained will be discussed.

The number of clock cycles used by the 'C6711 optimized blind adaptive MMSE algorithm, as obtained in the previous section, gives an indication for the theoretical minimum number of cycles that is required by the detection algorithm for detection of a bit. It is a minimum because the required number of cycles can increase because of, for example, pipeline stalls caused by slow memory access. Since these effects cannot be seen in the generated code, the only way to obtain a realistic value for the number of cycles that the implemented detector requires for detection of a bit is trough profiling. From the profile results that have been obtained it appeared that the actual number of cycles required for detection of a bit is a lot higher than the number of cycles that execution of the `blind_adaptive_mmse` theoretically requires. Not enough profile results have been obtained however to make a reliable statement about this. Therefor the number of clock cycles required by the 'C6711 optimized blind adaptive MMSE algorithm for detection of a bit that was obtained in the previous section will be used for some calculations about the *theoretical* detected bits per second performance that can be achieved by an 'C6711 implementation of the blind adaptive MMSE detector.

The last version of the optimized blind adaptive MMSE detector algorithm described in the previous section required 198 clock cycles for detection of a bit. For easier calculations this number is rounded to 200 cycles. The 'C6711 DSP has a 150MHz clock which means that the detector can detect 750000 bits per second. In Table 6.6 the number of channels a single 'C6711 DSP can detect and the number of DSPs necessary to detect a single channel for a number of often used data rates in wireless communications is shown. One channel in this table corresponds with one user. So in case of a base station with a single DSP for a 9600bps CDMA system, 77 users can be active simultaneously in the cell of this base station. In the calculation of the Channel/DSP numbers in Table 6.6 the time it takes the DSP to switch from detection of one user to detection of another user is not taken into account. In the calculation of the DSPs/Channel numbers in Table 6.6 the overhead of inter-DSP communication is not taken into account. For all calculations in Table 6.6 it is assumed that the DSP performs no other tasks than blind adaptive MMSE detection.

Time constraints did not allow for further research on the hardware that is currently used in cellular wireless telephony base stations. Therefor it is unfortunately not possible to draw any conclusions about the practicality of using blind adaptive MMSE detectors in, for example, cellular wireless telephony base stations from these results.

Data rate (bps)	Channels/DSP	DSPs/Channel
9600 (GSM)	77	0.01
21400 (GPRS 1 time slot)	35	0.03
171200 (GPRS 8 time slots)	4	0.23
144000 (UMTS moving vehicle)	5	0.19
2000000 (UMTS indoor)	0	2.67

Table 6.6: Number of channels supported by a 'C6711 implementation of the blind adaptive MMSE detector for often used data rates

6.6 Conclusions

- The blind adaptive MMSE detector has been implemented on a Texas Instruments TMS320C6711 DSP to determine the detected bits-per-second performance of the blind adaptive MMSE detector on the current generation of DSPs.
- The implemented blind adaptive MMSE detector has been optimized for the 'C6711 architecture.
- The optimized blind adaptive MMSE detector can theoretically detect 750000 bits/s on a 150MHz TMS320C6711 DSP.
- In case of a base station with a single DSP for a 9600bps CDMA system using blind adaptive MMSE detection, 77 users can be active simultaneously in the cell of this base station.
- Actual measurements of the demodulated bits per second performance of the optimized blind adaptive MMSE detector could not be performed because of time constraints.

Chapter 7

Conclusions and Recommendations

In this chapter the conclusions that can be drawn from the research on multiuser detection described in this thesis are presented. Recommendations for further research based on the work described in this thesis are given as well.

7.1 Conclusions

Several multiuser detection techniques for CDMA systems have been described. These detection techniques have been evaluated on their bit-error-rate performance, required knowledge of received signal parameters and computational complexity. As a result of this evaluation, the blind adaptive minimum mean square error detector and the parallel interference cancellation detector have been chosen as the most promising multiuser detectors for implementation. Both detectors achieve a significantly improved bit-error-rate, performance compared to the conventional CDMA detector at the cost of an increased computational complexity. In case of the PIC detector, when compared to the conventional and blind adaptive MMSE detector, additional required knowledge of received signal parameters is necessary.

Further analysis of the blind adaptive MMSE detector showed that, by minimizing the mean output energy, the blind adaptive MMSE detector also minimizes the mean square error. Therefor the blind adaptive MMSE detector will achieve the same bit-error-rate performance as the normal MMSE detector. Also, an algorithm for implementation of the blind adaptive MMSE detector has been derived.

Further literature study of the PIC detector showed that it can be implemented using either a narrowband implementation or a wideband implementation. Both implementations achieve the same bit-error-rate performance,

but have different computational complexities. Also a method for estimating the received amplitudes, that are required by the PIC detector but not by the blind adaptive MMSE detector, is found. The study also showed that both implementations have a bias in their decision statistic. The effect of this bias can be reduced by introduction of a partial cancellation factor.

In order to get more experience with the detection algorithms and to verify the simulation results found in literature, a simulator for the conventional, the blind adaptive MMSE and the PIC detector in synchronous CDMA systems has been developed. Simulation results obtained with this simulator confirm the simulation results found in literature.

From simulation of the blind adaptive MMSE detector it appears that the upper limit on the step size for stability of the adaptation algorithm of this detector, that is given in literature [6], is incorrect. Simulation of the blind adaptive MMSE detector further showed that the x sequence of this detector has to converge to the x sequence that minimizes the mean square error in order for this detector to obtain its optimal bit-error-rate performance. To reach this convergence the adaptation algorithm of the blind adaptive MMSE detector needs a certain amount of time that depends on the CDMA system parameters and the step size of the adaptation algorithm. When the CDMA system parameters change, the x sequence that minimizes the mean square error changes as well and the x sequence of the detector has to converge again. The blind adaptive MMSE detector will therefor not reach its optimum bit-error-rate performance in CDMA systems where the system parameters change within the convergence time of the adaptation algorithm.

Simulation of the PIC detector showed that introduction of a partial cancellation factor has a positive influence on the bit-error-rate performance of this detector. The simulation results also showed that the amplitude estimation that is used in the PIC detector has a negative influence on the bit-error-rate performance.

Comparison of the simulation results of the conventional, the blind adaptive MMSE and the PIC detector shows that both multiuser detectors achieve a much better bit-error-rate performance than the conventional detector in a perfect power control synchronous CDMA system. Without power control (and therefor different received amplitudes) the bit-error-rate performance of the conventional and the PIC detectors deteriorates strongly. For the conventional detector this is expected because it completely relies on power control to obtain reasonable bit-error-rate performance. For the PIC detector the strong deterioration in bit-error-rate performance is caused by the fact that its amplitude estimation technique is not very reliable with different received amplitudes. Until now, this effect of the received amplitudes on the bit-error-rate of the PIC detector using amplitude estimation has not been studied in literature very extensively. When the number of

active users in the CDMA system increases the bit-error-rate performance of all three detectors deteriorates. However the blind adaptive MMSE and the PIC detector continue to have a better bit-error-rate performance than the conventional detector. This also shows that the PIC detector handles multiple access interference better than the conventional detector as long as the received amplitudes do not vary strongly.

A DSP implementation of the blind adaptive MMSE detector has been developed because this detector performs better in CDMA systems with different received amplitudes and DSP implementation of this detector is not known to have been shown in literature before. The experience obtained from simulation of the blind adaptive MMSE detector have been used for implementation of this detector on the DSP. The main goal of the DSP implementation of the blind adaptive MMSE detector is to determine the detected-bits-per-second performance that can be achieved by this detector on the current generation of digital signal processors. In order to obtain the maximum level of performance the blind adaptive minimum mean square error algorithm has been optimized for the architecture of the Texas Instruments 'C6711 DSP. The optimized blind adaptive MMSE detector can theoretically detect one user of a 750000 bits per second synchronous CDMA system on a 150MHz DSP. Actual measurements of the detected-bits-per-second performance could not be obtained because of time constraints.

In order to make a statement about the practicality of the use of a DSP implementation of the blind adaptive MMSE detector in CDMA base stations two research steps will still have to be performed. First the data rate, number of supported users and other requirements on the base station have to be studied in order to determine the processing power requirements of the base station. After that a single DSP test-bed can be used to do detected-bits-per-second measurements of the DSP version of the blind adaptive MMSE detector to determine if and at what cost the processing power requirements of the base station can be met.

The DSP implementation of the blind adaptive minimum mean square error detector is not the *DSP-based multiuser detection test-bed that allows real-time evaluation of multiuser detection algorithms for CDMA systems* that was envisioned when the project that this thesis describes started. Nevertheless the developed simulator has been used to study a lot of the aspects of multiuser detection. The developed DSP implementation of the blind adaptive MMSE detector gives an indication of detected-bits-per-second performance that can be achieved by a DSP implementation of this detector. One problem that at least needs to be solved for development of a real-time version is the data transfer between the DSP and the host PC.

7.2 Recommendations

- Actual measurements of the detected bits per second performance of the DSP implementation of the blind adaptive MMSE detector have to be performed.
- The processing power of the signal processing other than detection that has to be performed in CDMA receivers has to be studied to determine the processing power required by the complete CDMA receiver.
- The processing power currently available in CDMA base stations has to be studied in order to make a statement about the practicality of the blind adaptive MMSE detector in these base stations.
- Implementation of the blind adaptive MMSE detector in programmable hardware, as for example an FPGA, could be studied to determine if a DSP is actually the most suitable hardware for implementation of such a detector.
- A DSP implementation of the PIC detector could be developed in order to compare the detected bits per second performance of this detector with that of the blind adaptive MMSE detector.
- The use of other amplitude estimation techniques in the PIC detector could be studied.
- The influence of the partial cancellation factor on the bit-error-rate of the PIC detector for large interferer amplitudes could be studied.
- The stability condition for the blind adaptive MMSE detector can be studied more thoroughly, starting with an analysis of the validity of the stability condition given in literature, e.g. [6].
- Testing of DSP implementations of multiuser detectors could be simplified by using PCI based DSP boards, because this simplifies data exchange between the DSP and the host PC.

Bibliography

[1] G.E.P. Box and M.E. Muller. A note on the generation of random normal deviates. *Annals Math. Stat*, 29:610–611, 1958.

[2] R. M. Buehrer, N. S. Correal-Mendoza, and B. D. Woerner. A Simulation Comparision of Multiuser Receivers for Cellular CDMA. *IEEE Transactions on Vehicular Technology*, 49(4):1065–1085, July 2000.

[3] N. S. Correal, R. M. Buehrer, and B. D. Woerner. A DSP-Based DS-CDMA Multiuser Receiver Employing Partial Parallel Interference Cancellation. *IEEE Journal on Selected Areas in Communications*, 17(4):613–630, April 1999.

[4] N. Dahnoun. *Digital Signal Processing Implementation*. Prentice Hall, 2000.

[5] S. Haykin. *Communication Systems*. John Wiley & Sons, 1994.

[6] M. Honig, U. Madhow, and S. Verdú. Blind Adaptive Multiuser Detection. *IEEE Transactions on Information Theory*, 41(4):944–960, July 1995.

[7] D. Keil. Real-Time Data Exchange. Whitepaper SPRY012, Texas Instruments Incorporated, February 1988.

[8] D. Koulakiotis and A. H Aghvami. Data Detection Techniques for DS/CDMA Mobile Systems: A Review. *IEEE Personal Communications*, 7(3):24–34, June 2000.

[9] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, January 1998.

[10] H. V. Poor and S. Verdú. Probability of Error in MMSE Multiuser Detection. *IEEE Transactions on Information Theory*, 43(3):858–871, May 1997.

[11] H. B. Roelofs. Performance Analysis of Wireless Communication Systems Using Fast Simulation. Master's thesis, University of Twente, December 2001.

[12] Texas Instruments Incorporated. *TMS320C6000 CPU and Instruction Set Reference Guide*, February 2000.

[13] Texas Instruments Incorporated. *TMS320C6000 Programmer's Guide*, February 2001.

[14] M. K. Varanasi and B. Aazhang. Near-optimum detection in synchronous code-division multiple-access systems. *IEEE Transactions on Communications*, 39(5):725–736, May 1991.

[15] S. Verdú. *Multiuser detection*. Cambridge University Press, 1998.