Equalization of Co-Channel Interference in Future Mobile Communication Systems

Anders Nilsson-Stig and Henrik Perbeck

22nd January 1999

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
January 1999

Document Number
ISRN LUTFD2/TFRT--5610--SE

Author(s)
Anders Nilsson-Stig and Henrik Perbeck

Supervisors
Bengt Lindoff, Bo Bernhardsson, Jan Holst

Sponsoring organisation

Title and subtitle
Equalization of Co-Channel Interference in Future Mobile Communication Systems.

Abstract

This study shows that Joint Detection is a powerful method for eliminating co-channel interference in 8-PSK (Phase Shift Keying) modulation, provided a carrier SNR (Signal to Noise Ratio) greater than 15 dB. Joint Detection refers to decoding information from two parallel data sources receiving the same input signal.

A full-state Joint MLSE (Maximum Likelihood Sequence Estimation) algorithm was implemented, and showed good performance but great numerical complexity. For multi-path fading channels, a BER (Bit error Rate) less than 5% is achieved for any strength of the interferer, provided a carrier SNR of 25 dB or more.

To reduce complexity, a Conventional MLSE for the carrier including a built-in Decision Feedback algorithm for the interferer was implemented and evaluated. For multi-path fading channels, in most cases this MLSE/DF algorithm provides about 95% of the performance gain produced by Joint MLSE. Thus, for practical implementation, MLSE/DF may be an interesting solution.

The results are based on the assumption that carrier and interferer bursts are synchronized and that the training sequences of both carrier and interferer are known to the channel estimator. The need for further studies on Blind Detection of the interferer is pointed out, and a moment estimator for detection of a possible interferer is also discussed.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information
Also available as report 1999:E1 from the department of Mathematical Statistics

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
46

Recipient's notes

Security classification

Preface

This Master's thesis was produced at Ericsson Mobile Communications AB, Lund, in corporation with the Dept. of Automatic Control and the Dept. of Mathematical Statistics, Lund Institute of Technology. We would like to thank our competent and encouraging supervisors Bengt Lindoff (Ericsson), Bo Bernhardsson (Automatic Control) and Jan Holst (Mathematical Statistics). Our regular meetings left us with an insight to the joy of science, and with new spectacular problems for the world to solve after this study.

Finally, we would like to thank Bengt for considering passing this thesis, although he swore not to when beaten on the golf course.

Lund, January 1999

Anders Nilsson-Stig Henrik Perbeck

Contents

1 Introduction

The use of mobile communication systems has increased significantly over the last several years. Digital systems such as GSM have made it possible to improve the quality of speech and data transmission. The available bandwidth is limited and much effort is made to use it efficiently. The GSM system sends one bit of data per symbol. A future system, EDGE *(Enhanced Data rates for GSM Evolution)*, proposes sending three bits per symbol in the same GSM system as today. The modulation used by such a system becomes more sensitive to noise and other disturbances. This thesis focuses on one such disturbance, co-channel interference. Base stations covering adjacent cells cannot use the same frequencies since the cells overlap. To optimally utilize the bandwidth, not directly neighbouring cells are allowed to use the same frequencies. When an undesired signal *(interferer)* at the same frequency disturbs the main signal *(carrier)* it is referred to as co-channel interference. The structure of the interferer is assumed known and the aim of the thesis is to decode it in order to increase the quality of the carrier.

The underlying work of this report consists of developing equalizers that decode the carrier better by decoding and removing the interferer. First, a MLSE equalizer for the carrier only was designed using the Viterbi algorithm. It can be seen as a reference for what today's systems would achieve. Second, a full MLSE for both the carrier and the interferer was designed. Due to high numerical complexity it is probably not suitable for practical implementation. Finally, in order to reduce complexity, a MLSE equalizer for the carrier was combined with a decision feedback equalizer for the interferer.

This thesis analyzes the results of simulations using the different equalizers. It will be shown that large gains in performance can be achieved by decoding the interferer.

A Master's thesis made at Ericsson, see [2], examined possible gains of Joint Detection for binary GSM modulation, simulating a whole mobile communication system. This study focuses on the radio channel, and interference cancellation for the more sensitive EDGE system.

The report consists of four main parts. First, Section 2 provides general background and presents relevant underlying theory. In Section 3 the simulation environment, model assumptions and the design of the equalizers are explained. Section 4 presents the results of the simulations. Finally, Section 5 connects this work to related research in the field and considers areas of future interest.

2 Background

2.1 Mobile Communication Systems

To see the context of this essay it is necessary to understand the structure and use of mobile communication systems. In the last years there has been significant growth in the use and need for mobile communication, particularly following the introduction of digital systems like GSM. This has made it possible to increase the quality and rate of the information transmitted. So far the focus has been on speech, i.e. mobile phones. However, future mobile digital data communication will greatly increase the demands of the systems currently being designed.

A mobile communication system consists of the following basic units:

- mobile units, typically a mobile phone or a communication module for a portable computer,
- base stations, large radio transmitters/receivers distributed to cover a certain limited area (cell),
- communication network, the network to which the user wants to be connected e.g. a stationary telephone network or the Internet.

A mobile communication system is limited by the ability to transmit data only over a certain bandwidth. The available bandwidth is a finite resource and consequently calls for measures of efficient use. Figure 1 is an illustration of the components needed for digital data to be communicated through the air [1].

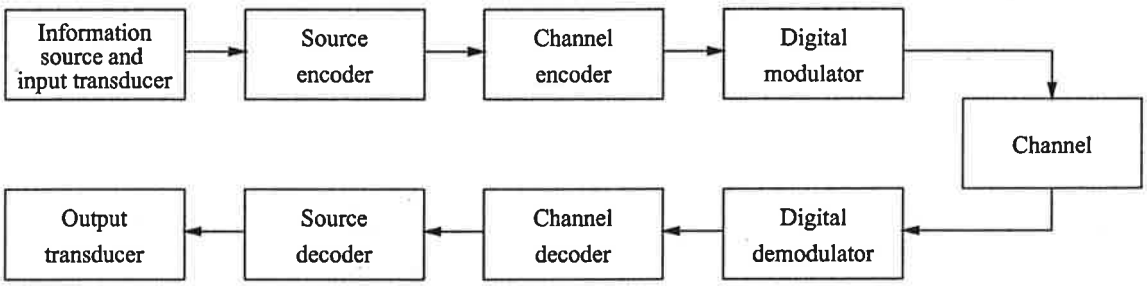


Figure 1: Elements of a digital communication system. The focus of the thesis is on the digital demodulator.

The main focus of this thesis is on what is called the digital demodulator and how it is being affected by the channel. To understand the importance of this interaction a brief discussion of the other components is useful.

The first element of a digital communication system is the *information source and input transducer*, i.e. the signal that is to be transmitted. This has to be converted into a sequence of binary digits by the *source encoder*. The *information sequence* is then passed to the *channel encoder* which will introduce redundancy to reduce information losses due to noise and disturbances. Another technique to achieve this is called *interleaving*. Consecutive data is portioned out and blocks to be transmitted are built so that loss of such a block will have limited consequences. This can be seen in more detail in Figure 2. The binary sequence ready to be transmitted is then passed to the *digital modulator* which is the interface to the channel. Here the binary sequence is mapped into signal waveforms which can be transmitted over the *communication channel*. The channel may be any physical medium but in the case of mobile systems mainly open air. The channel is always corrupted to some extent, due to thermal noise. Also, further disturbances can be caused by other sources such as automobile ignition noise, atmospheric noise and interference. A realistic model of the physical channel is very important to be able to design the system and in particular the *digital demodulator*. This component's job is to process the channel-corrupted transmitted waveform and transform it into a sequence of binary digits. These digits are ideally the same as those the digital modulator received in the first place, but in practice are always an approximation thereof. Finally the information sequences are reassembled by the *channel decoder* and converted back to its original signal form by the *source decoder*.

This thesis analyses different types of equalizers which are part of the digital demodulator in Figure 1. The role of the equalizer is better seen in Figure 2. Also the interleaver is here separated out from the channel encoder. The block interleaver takes blocks of data from different sequences and mixes them so that a whole sequence should not be lost. The binary sequence continues to the digital modulator, which maps the symbols onto the carrier frequency. The waveform signal is then transmitted over the channel. Further, the channel is *fading*. This means that due to reflection, many copies of the transmitted signal will arrive at the receiver with a slight time-delay. The phase shift of these signals can cause constructive or destructive interference which will affect the channel. Noise, $n(t)$, is also added to the signal. The received signal is band pass filtered, down converted to baseband, low pass filtered *(LPF)* and sampled at kT_c. At this stage the role of the equalizer is to decide which sequence was sent. It must use information about the channel and noise. The most likely sequence is passed to the *block deinterleaver* and further channel decoded.

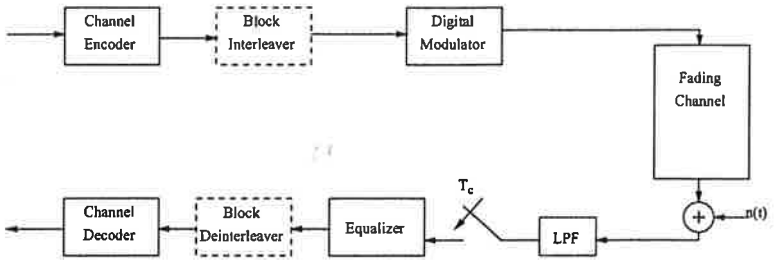


Figure 2: Block diagram of the digital demodulator showing interleaving and equalization.

TDMA and Burst Structure

Today's GSM system uses TDMA *(Time Division Multiple Access)* to efficiently use the available bandwidth of a frequency band, 200 kHz. The system's symbol rate is 271 kb/s, which gives a symbol time of 3.6 μs. Each frequency band is divided into eight time slots which are dedicated to different users. Consequently, the symbol rate for each user is 34 kb/s to be used for information and coding. During a time slot a burst is sent which includes data and a training sequence known to the receiver, see Figure 3. The time slot has length equivalent to 156.25 symbols, out of which 148 are used. The training sequence makes it possible to estimate the channel during the burst. For slow moving objects (< 100 km/h) the fading will not have an effect within a burst and the channel can be considered constant, i.e. no channel tracker is needed. The tails are used to get a final state for the Viterbi-coder. Sometimes the data sent are hand-over information which is then indicated by the flag. Details about the GSM system are found in [9].

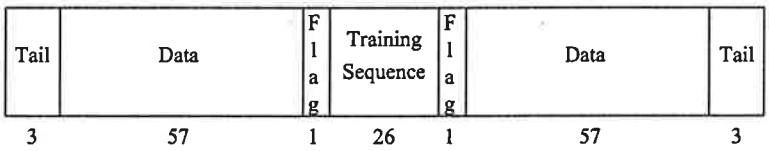


Figure 3: GSM burst structure.

Cell Planning and Co-Channel Interference

Cellular communication systems are based on the idea that each base station covers a certain area, a cell. These cells are mostly circular (directed antennas also exist). To be able to cover a larger area completely the cells need to overlap. The mobile unit picks the base station with the best signal

(the nearest) and changes base stations when needed. Clearly the neighbouring cells cannot use the same frequency as this would cause problems at the borders and high interference. However, base stations that are not direct neighbours can use the same frequency and can thus interfere with each other. The reason for allowing this is to increase the available bandwidth for each base station. The distance to an interfering base station is hopefully large enough that the co-channel signal is small compared to the main signal. Today co-channel interference (CCI) is treated as extra noise. In this thesis it will be shown that considerable improvements can be made if the channel of the interferer can be estimated and its signal removed.

EDGE and the Future

Future mobile communication systems will demand higher data rates and higher quality. To transmit computer data much lower bit error rates (BER) are required than for speech. One way to increase the data rate is to choose a different modulation. Instead of using only one bit for each transmitted symbol as GSM does today, the future GSM system EDGE *(Enhanced Data Rates for GSM Evolution)* suggests three bits per symbol, using 8-PSK modulation [4]. Types of modulation are discussed in Section 2.2 below. These signals, e.g. 8-PSK, will be more sensitive to noise and interference and the performance would consequently increase substantially if the CCI could be reduced. Other systems with very high capacity, such as WCDMA *(Wideband Code Division Multiple Access)* [11], requires completely new technology and infrastructure. The advantage of EDGE is that it could be incorporated into the GSM system. When the system no longer is able to function with EDGE-quality it can switch to GSM as its fall-back mode.

2.2 Modulation of Digital Signals

To transmit a sequence of digital data over a communication channel the data must be transformed into an analog waveform. This is done in the digital modulator, see Figure 1. The sinusoidal waveform has a certain carrier frequency, f_c. The three basic types of modulation are amplitude modulation (AM), phase modulation (PM) and frequency modulation (FM). They represent the different parts of a waveform that can transmit information. A PM signal typically looks like

$$s(t) = \cos(2\pi f_c t + \theta(t)),$$

where the information is represented by $\theta(t)$. Although abrupt changes of the phase are possible, they demand increased bandwidth of the signal. In-

stead, a method called *continuous phase modulation*, CPM, is often used [1]. The phase changes during the symbol duration time. This also introduces the notion of memory. How to interpret the information from the phase depends on which symbol was sent before. Today's GSM system uses CPM where $\theta(t)$ is convoluted with a smoother Gaussian function, $\tilde{g}(t)$, such that

$$s(t) = \cos(2\pi f_c t + \theta(t) * \tilde{g}(t)),$$

This is called GMSK *(Gaussian Minimum Shift Keying)*, see [9]. This is a non linear modulation. However, its linear approximation is a PSK-signal *(Phase Shift Keying)* represented as

$$s(t) = g(t)\cos(2\pi f_c t + \theta(t)),$$

where $g(t)$ is related to $\tilde{g}(t)$ in a certain sense. The structure of an M-PSK, *(M-ary Phase Shift Keying)*, signal will now be analyzed further. As mentioned, EGDE uses 8-PSK and GSM uses BPSK *(Binary Phase Shift Key, $M = 2$)*. Let $g(t)$ be the pulse shape of the signal. Assume M different signal waveforms are to be transmitted, then $\theta(t) = 2\pi(m-1)/M, m = 1, 2, \ldots, M$ and

$$\begin{aligned} s(t) &= g(t)\cos\left[2\pi f_c t + \frac{2\pi}{M}(m-1)\right] \\ &= g(t)\cos\frac{2\pi}{M}(m-1)\cos 2\pi f_c t - g(t)\sin\frac{2\pi}{M}(m-1)\sin 2\pi f_c t. \end{aligned}$$

Consequently, the signal waveforms can be represented as a linear combination of two orthonormal signal waveforms, $f_1(t)$ and $f_2(t)$. Thus,

$$s(t) = s_{m1} f_1(t) + s_{m2} f_2(t),$$

where

$$\begin{aligned} f_1(t) &= \sqrt{\frac{2}{\mathcal{E}_g}} g(t)\cos 2\pi f_c t \\ f_2(t) &= -\sqrt{\frac{2}{\mathcal{E}_g}} g(t)\sin 2\pi f_c t. \end{aligned}$$

Here $\mathcal{E}_g$ denotes the energy in the pulse $g(t)$. Now the two dimensional vectors $\mathbf{s} = \begin{bmatrix} s_{m1} & s_{m2} \end{bmatrix}$ are given by

$$\mathbf{s} = \begin{bmatrix} \sqrt{\frac{\mathcal{E}_g}{2}}\cos\frac{2\pi}{M}(m-1) & \sqrt{\frac{\mathcal{E}_g}{2}}\sin\frac{2\pi}{M}(m-1) \end{bmatrix}, \quad m = 1, 2, \ldots, M$$

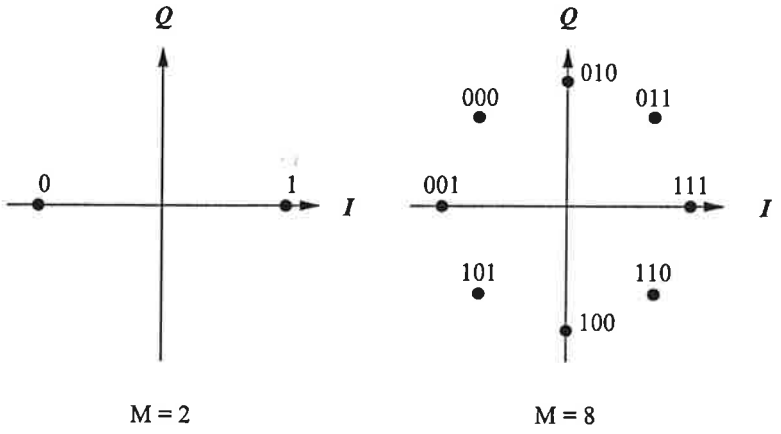


Figure 4: Signal space diagrams for BPSK and 8-PSK signals.

This vector representation can be illustrated in IQ-space and is called *signal space diagram.* In Figure 4 this is shown for BPSK and 8-PSK.

When a signal is received the equalizer must decide which of the signal waveforms was sent. In the case of BPSK it is clear that a signal with a positive x-value in the signal space diagram will be considered a 1 and negative x-value a 0. Also, in the case of 8-PSK the decision is based on the shortest distance to a waveform. When an error is made, often one of the neighbouring waveforms is erroneously chosen. Since every symbol actually contains three bits of data it is desirable that as few bits as possible are wrong. As seen in Figure 4 the different symbols are interpreted as bit groups in such a way that only one bit changes between neighbouring waveforms. This is called *Gray encoding* [4].

2.3 Equalization

The process of equalization deals with the issue of extracting the initial signal sent from the received signal. The received signal has passed through the channel and been disturbed by noise. An effective equalizer should therefore make use of information about the structure of the transmitted signal, the radio channel, noise and other disturbances. In Section 2.3.1 the effects of interference are treated. There are two main types of interference. *Inter Symbol Interference* (ISI) occurs when reflected signals with longer path are delayed into the next sampling interval and interfere with subsequent waveforms. The other type is *Co-Channel Interference* (CCI). This is a result of signals from another transmitter, or cell, leaking into and consequently interfering with the desired signal. To incorporate this kind of information

the equalizer must estimate the channel and may use the training sequence to do this, see Section 2.3.2. Thereafter, the equalizer must decide which waveform actually was sent. The optimal way to do this is by using a *Maximum Likelihood Sequence Estimator* (MLSE), which can be implemented using the Viterbi-algorithm [1]. The Viterbi-decoder is central in the work of this thesis and is described in Section 2.3.3.

2.3.1 Interference

Fading

The constantly changing characteristics of the media makes radio channels time-variant. Reflections by the ground or objects close to the receiver cause signals to arrive slightly out of phase due to different path lengths. The interference pattern of these waves is sometimes constructive but sometimes destructive. The result is that the radio channel changes constantly, a phenomenon called *fading*. The impulse response can be modelled as a zero-mean complex valued Gaussian process with a *Rayleigh distributed* envelope. The Rayleigh probability density function is described by

$$p_R(r) = \frac{r}{\sigma^2} e^{-r^2/2\sigma^2}, \quad r \geq 0.$$

Fading has been taken into account in the simulations in Section 4.2.2. For slow moving objects the channel can be considered constant during one burst in the GSM and EDGE systems. For greater speeds the receiver could move into an area with a different channel within a single burst and a channel tracker would be needed.

Intersymbol Interference

Depending on the surroundings, the channel response can sometimes be longer than the symbol time. The consequence is that previously sent signals interfere with the latest symbol, *intersymbol interference* (ISI). This typically occurs when the signal is reflected, refracted or diffracted by larger objects, e.g. mountains and large buildings. The model of the received signal, down converted to baseband and sampled at symbol rate, y_t becomes,

$$y_t = h_0 u_t + h_1 u_{t-1} + \ldots + h_L u_{t-L} + \varepsilon_t = Hu + \varepsilon_t$$

where u_t is the signal sent at time t and h_i is the coefficient denoting the influence of signal u_{t-i}, see [1]. The noise ε_t is here assumed to be white, although it is actually not as will be shown in Section 2.3.1. L is said to be

the *memory* of a channel with tap length $L+1$. In the GSM system the components are designed to handle a full energy five-tap channel. Since such a channel is not a very realistic surrounding, other typical surroundings are presented in Table 1. For more detailed propagation models, see [10].

Surrounding	*Channel (H)*
Rural area	(1)
Urban area	(1 0.1)
Hilly terrain	(1 0 0 0.2 0.1)
Equalization test	(1 1 1 1 1)

Table 1: Typical GSM radio channels with ISI.

Not only do the physical surroundings affect the channel H experienced by the equalizer. The transmitter and receiver filters must also be taken into account to fully describe the incoming signal, see Section 3.1.2.

Co-Channel Interference

The type of interference focused on in this thesis is *co-channel interference*, CCI. In areas where base stations are situated closely, e.g. in cities, signals from other base stations may interfere with the main signal. As mentioned in Section 2.1 directly adjacent cells are not allowed to use the same frequency, but cells further away are. Due to the distance the interfering signals will always be reasonably small. However, since the EDGE system is designed to work in areas with high signal to noise ratio and closely situated base stations, the problem of CCI will be more dominant than today. Classically, CCI has been treated as part of the white noise. However, the structure of the CCI signal is known and if its channel can be found then the model can be refined as

$$\begin{aligned} y_t &= h_0 u_t + h_1 u_{t-1} + \ldots + h_L u_{t-L} + \varepsilon_t = Hu + \varepsilon_t \\ &= h_0 u_t + h_1 u_{t-1} + \ldots + h_L u_{t-L} + g_0 v_t + g_1 v_{t-1} + \ldots + g_K v_{t-K} + e_t \\ &= Hu + Gv + e_t, \end{aligned}$$

where u_t and h_i are defined as above. Similarly, v_t and g_i are the signal and coefficients of the interfering transmitter, K its memory and e_t is white noise. In practice it is far more complicated than just extending the model. The signals are most likely not synchronized. Estimation of the co-channel, G, is difficult if its training sequence is not known. Further, since the desired and unwanted signals are not synchronized, the corrupting signal might only be switched on during a part of the burst.

2.3.2 Channel Estimation

Effective equalization requires accurate estimation of the channel. In radio transmission the channels are time-varying, making it necessary to follow changes in the channel. In the GSM system every burst contains a training sequence (see Figure 3), which is used to estimate the channel for that burst. Estimation is done by least squares (LS), see [3], of the form

$$\hat{H} = (U^T U)^{-1} U^T Y$$

Not all 26 symbols of the training sequence are used. The reason is that symbols sent before the training sequence should not effect the estimate due to the memory of the channel ($L_{max} = 4$). If the training sequence is the first 26 symbols sent,

$$U = \begin{pmatrix} u(6+L) & u(5+L) & \dots & u(6) \\ u(7+L) & u(6+L) & \dots & u(7) \\ \vdots & \vdots & \ddots & \vdots \\ u(21+L) & u(20+L) & \dots & u(21) \end{pmatrix}, \quad Y = \begin{pmatrix} y(6+L) \\ y(7+L) \\ \vdots \\ y(21+L) \end{pmatrix}$$

The LS-estimate for this model is always consistent, providing the model and reality coincide. This holds even when the noise is not white, see [3]. The most computationally expensive step in estimating H is calculating $(U^T U)^{-1}$. However, the predefined training sequences that GSM uses are designed such that $(U^T U)^{-1}$ is always diagonal for the mid-16 symbols. The computation therefore requires only a straightforward multiplication with a normalizing factor.

Estimating the co-channel is much more complicated. One method is to use blind detection which has the advantage that the training sequence does not have to be known. This is further discussed in Section 5.2. Since the primary aim of this thesis is not to investigate estimation techniques, it is assumed that the training sequences from the two signals are synchronized and known. This is not a realistic assumption but it is convenient for showing the performance gain of co-channel decoding . Further, synchronization methods exist to find the training sequence of an unknown signal by correlation, see [1]. By assuming that the training sequences are known and synchronized the LS-estimation above can be extended to

$$\begin{pmatrix} \hat{H} \\ \hat{G} \end{pmatrix} = (U^T U)^{-1} U^T Y,$$

where Y is the same as before and

$$U = \begin{pmatrix} u(6+L) & u(5+L) & \dots & u(6) & v(6+K) & v(5+K) & \dots & v(6) \\ u(7+L) & u(6+L) & \dots & u(7) & v(7+K) & v(6+K) & \dots & v(7) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u(21+L) & u(20+L) & \dots & u(21) & v(21+K) & v(20+K) & \dots & v(21) \end{pmatrix}.$$

It is important to note that for both H and G, assumptions need to be made concerning the number of taps. In Section 4.4 the effects of choosing the wrong channel model will be examined.

2.3.3 Data Decoding

In this section two types of equalization methods will be discussed. The first type, used in all the simulations in this thesis is based on the maximum-likelihood sequence detection criterion. It is referred to as the Viterbi algorithm or *maximum likelihood sequence estimation* (MLSE) and is optimal for minimizing the probability of error. The second equalization method is called *decision feedback equalization* (DF). It uses previously detected symbols to compensate for ISI in the present symbol. For a detailed analysis of these methods, see [1].

MLSE

Consider a discrete time model where the n-th received signal is denoted by y_n. Let $\tilde{y}_n = h_0 u_n + h_1 u_{n-1} + \ldots + h_L u_{n-L}$ be the undistorted signal produced by the last symbols $(u_n, \ldots, u_{n-L})$ sent. Assuming $u_i, i = 1 \ldots n$, are iid (independent and identically distributed) and the noise, $\varepsilon \in N(0, \sigma^2)$. The channel, symbols and noise are all complex. The probability density of $y_n = \tilde{y}_n + \varepsilon_n$ given that $\tilde{y}_n$ was sent is

$$p(y_n \mid \tilde{y}_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-|y_n - \tilde{y}_n|^2/2\sigma^2}.$$

The aim is to find the symbol sequence $(u_n, \ldots, u_{n-L})$ that maximizes this probability, i.e. for a N-sequence of symbols received, $\mathbf{y}_N$, the sequence U_N that maximizes

$$p(\mathbf{y}_N \mid \tilde{\mathbf{y}}_N) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N exp\left(-\frac{1}{2\sigma}\sum_{k=1}^{N} |y_k - \tilde{y}_k|^2\right).$$

This is equivalent to maximizing the logarithm and discarding any terms independent of y_k and $\tilde{y}_k$, i.e. maximizing

$$-\sum_{k=1}^{N} |y_k - \tilde{y}_k|^2$$

or minimizing

$$\sum_{k=1}^{N} |y_k - \tilde{y}_k|^2.$$

Using the vector representation in Figure 4, this can be seen as the sum of the Euclidean distance between y_k and $\tilde{y}_k$. Call this the *metric*, PM_i, of the sequence with i symbols and

$$PM_1(u_2, u_1) = \sum_{k=1}^{2} |y_k - \tilde{y}_k|^2$$

$$PM_2(u_3, u_2, u_1) = PM_1(u_2, u_1) + |y_3 - \tilde{y}_3|^2.$$

One way to find the sequence with the lowest metric would be to calculate the metric for all possible sequences. However, for a sequence of length N with M symbols in the sending alphabet, there are M^N possible sequences, making such a calculation difficult for large N due to high complexity. One way to sidestep this problem is to the Viterbi algorithm.

The recursive representation of the metric implies that many sequences will have the same metric up to a certain point. If such a metric is calculated and is the lowest up to that point, its path is stored and only the new branches need to be investigated. The number of branches that needs to be investigated depends on the memory of the channel, L. Clearly, only symbols which can influence $\tilde{y}_k$, will have an effect in minimizing the metric.

The Viterbi-algorithm is best illustrated by an example. Let $H = \begin{pmatrix} 1 & 0.5 \end{pmatrix}$ and the signal be 4-PSK, i.e. $u_k = [\pm 1, \pm i]$. Figure 5 shows the tree structure for this example.

Since the channel has a memory, $L = 1$, the first time, all of the branches survive to the next step, where the first metrics are calculated for all branches having different $U = (u_2, u_1)$.

$$\begin{aligned} PM_1(u_2, u_1) &= \sum_{k=1}^{2} |y_k - HU|^2 \\ &= |y_1 - u_1|^2 + |y_2 - (u_2 + 0.5u_1)|^2 \end{aligned}$$

Now, the metrics of the branches ending with the same u_2 are compared and only the one with the lowest metric needs to be built further. Had the memory been longer than one, the paths with equal ending sequences of length L would have been compared. Consequently, M^L branches will survive to the next step (here $M^L = 4$). These surviving branches are extended and comparisons are made M times among M^L metrics to again

find the M^L best metrics. This gives a complexity of $\mathcal{O}(M^{L+1})$. Hence,

$$\begin{aligned} PM_2(u_3, u_2, u_1) &= PM_1(u_2, u_1) + |y_3 - HU|^2 \\ &= PM_1(u_2, u_1) + |y_3 - (u_3 + 0.5u_2)|^2, \end{aligned}$$

which is dependent on u_1 only through the previous metric, chosen earlier as the best one. By this way of identifying and discarding sequences which can never be optimal, the Viterbi algorithm reduces the complexity of MLSE from $\mathcal{O}(M^N)$ to $\mathcal{O}(NM^{L+1})$.

Decision Feedback

The decision feedback equalizer consists of two filters and one detector, as seen in Figure 6. The input from the receiver, y_k, is passed through a *feedforward filter* in order to concentrate the energy to the first taps. ISI is reduced by removing previously detected symbols after passing through a *feedback filter*. An approximation of the last sent symbol, $\hat{u}_k$, is passed to the *detector*. The detector chooses the symbol most likely sent from the alphabet, $\tilde{u}_k$, which is the output data. This symbol is also used again by the feedback filter in order to reduce ISI in the next symbol. Clearly, the advantage of DF is its low complexity, $\mathcal{O}(NM)$.

If the detected symbols are correctly identified this is a very effective way of removing ISI. However, as soon as a decision error is made it will be used, incorrectly, to remove ISI in the next symbol. That symbol is now more likely to be detected wrongly. This phenomenon is called *error propagation* and is the disadvantage of decision feedback equalization as opposed to MLSE (see [1], pp. 622–626). In MLSE all possible combinations of symbols that can affect the last symbol are compared.

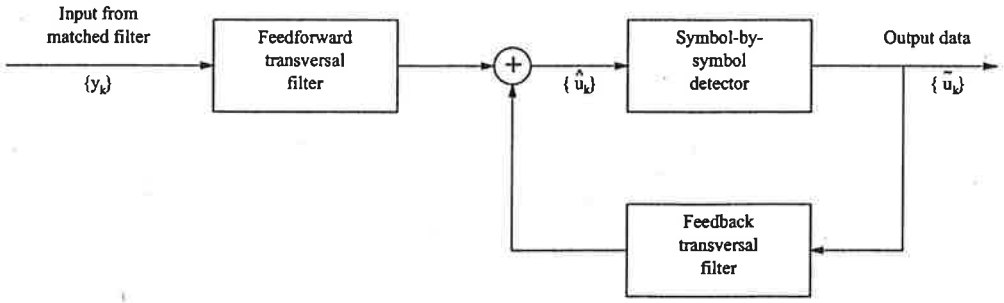


Figure 6: Block diagram of a decision feedback equalizer.

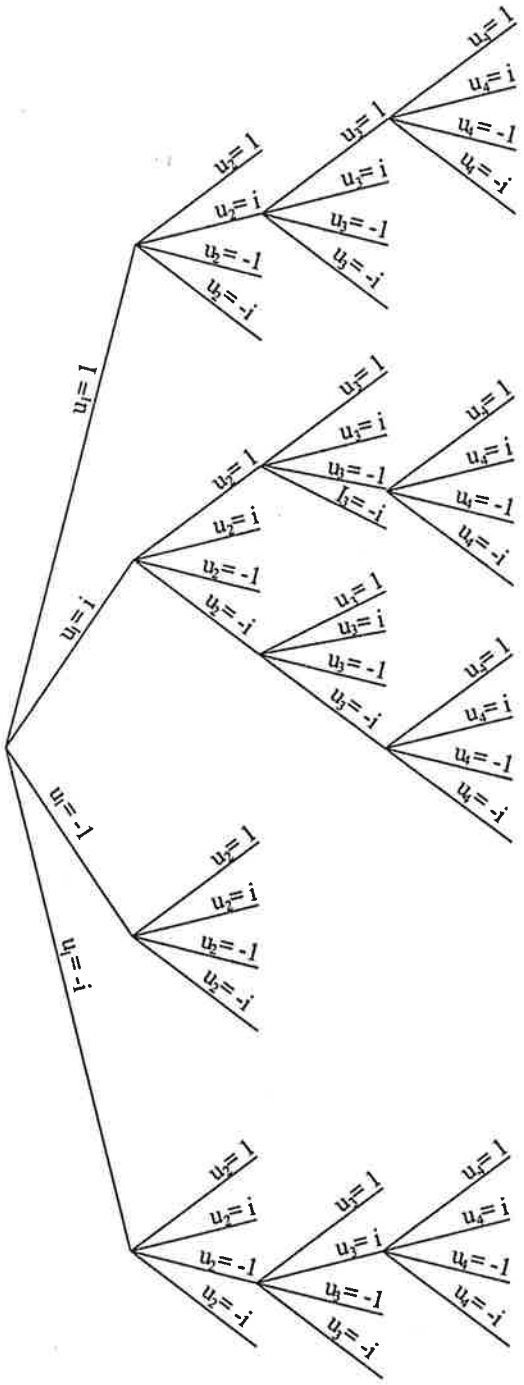


Figure 5: Tree diagram for Viterbi decoding, $M = 4$ and $L = 1$. All branches survive the first step. Subsequently, branches with equal ending symbol are compared and the ones with the lowest metric survive.

3 Simulation Environment

Implementation and evaluation of the algorithms were simulated using Matlab. The main reason for the choice of Matlab was the ease of implementation and numerical reliability. Possibly, some of the results could be verified analytically (see [1], pp. 593–601).

3.1 General Setup

The main task was to evaluate the idea of reducing CCI by decoding the interfering data. The main environment for the work was Matlab, in which both decoding algorithms and models for carrying out simulations were implemented. The simulations were carried out in IQ signal space. In this section, the Matlab implementation of the simulations is discussed. In Section 3.2, the decoding algorithms are described in detail.

3.1.1 Definitions

In the simulations, the channels are modelled as

$$y_t = h_0 u_t + h_1 u_{t-1} + \ldots + h_L u_{t-L} + g_0 v_t + g_1 v_{t-1} + \ldots + g_K v_{t-K} + e_t$$

or

$$y_t = \sum_{i=0}^{L} h_i u_{t-i} + \sum_{i=0}^{K} g_i v_{t-i} + e_t = Hu + Gv + e_t,$$

where H and G are row vectors representing *channel filters* for the carrier and interferer signals respectively, and u and v are column vectors representing sent symbols. The quantity y_t is the received signal at a time t and e_t is Gaussian noise. The sent symbols u_t, v_t are assumed to be independent and identically distributed complex 8-PSK symbols with $E|u_t|^2 = E|v_t|^2 = 1$.

The Signal to Noise Ratio, SNR, of a channel is defined as

$$SNR = \frac{E|Hu|^2}{E|e_t|^2} = 10 \log_{10} \frac{\sum_{i=0}^{L} h_i^2}{V_e},$$

where h_i are the elements of the H carrier channel vector and V_e is the variance of the white Gaussian noise, i.e. $V_e = Var\{e_t\}$.

The Carrier to Interferer ratio, C/I, is defined as

$$C/I = \frac{E|Hu|^2}{E|Gv|^2} = 10 \log_{10} \frac{\sum_{i=0}^{L} h_i^2}{\sum_{i=0}^{K} g_i^2},$$

where g_i are the elements of the G interfering channel vector.

It is natural to define the Interferer to Noise Ratio, I/N, as

$$I/N = 10 \log_{10} \frac{\sum_{i=0}^{L} g_i^2}{V_e}.$$

3.1.2 Choice of H and G Channels

The choice of the carrier channel, H, in the simulation models has been based on real GSM conditions. The signal is affected by the transmitter filter, H_{trm}, the propagation path, H_{path}, and the receiver filter, H_{rec}. A reasonable discrete model of the transmitter and receiver filters is

$$H_{trm} = H_{rec} = \begin{pmatrix} 0.3 & 1 & 0.3 \end{pmatrix}.$$

This is a linear approximation of the GMSK pulse shaping filter, sampled at a symbol rate (see [4]). In this model, H_{path} is used to describe the effects of multipath propagation. For simulations, the "Urban Area" channel from Table 1 was chosen, since urban areas, where CCI is often an important limiting factor, are the main destination for the new EDGE system. This channel can be approximated by

$$H_{path} = \begin{pmatrix} 1 & 0.1 \end{pmatrix}.$$

Thus, H is defined as the convolution of H_{trm}, H_{path} and H_{rec}. Hence, a scaled 3-tap approximation of the channel, which has been used in most of our simulations, is

$$H = \begin{pmatrix} 0.4 & 1 & 0.5 \end{pmatrix}.$$

The interfering signal channel G was modelled under the assumption that it is remote and the desired signal is dominant. Hence, we make the assumption that the receiver is only reached by a single-tap width interfering wave. Since the desired and interfering signals are unsynchronized, every interfering symbol is likely to affect two sample intervals. The interfering channel was modelled using two taps equal in energy:

$$G = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

When simulating fading channels, the same H and G as for constant channels have been used. However, in each burst each tap is multiplied by a $N(0,1)$-distributed number. Since these numbers are independent, the energy of the signal reaching the receiver will be *on average* the same as for a constant channel. Hence SNR and C/I should in the case of fading channels be interpreted as average values.

3.1.3 Transmitting Data

Throughout the simulations 8-PSK signal modulation has been used, since 8-PSK is likely to be part of the new EDGE standard definition for future GSM systems. While the change from today's two symbol alphabet to an eight symbol alphabet aims to improve the data rate, it makes the signal more sensitive to interference, since the distance between the data points in signal space is reduced. In addition, it brings a dramatic increase in complexity to the Viterbi decoder algorithm — from $\mathcal{O}(2^{L+1})$ to $\mathcal{O}(8^{L+1})$, where $L+1$ is the number of taps in the ISI filters.

Data were sent in bursts, similar to those in today's GSM system. Each burst consists of a 26-symbol *training sequence*, followed by a 57-symbol *data sequence*. The training sequences used were selected from a set of sequences used in the GSM system. The middle 16 out of the 26 symbols in these sequences are specially designed to reduce numerical complexity when estimating the radio channel. Thus, only 16 of the 26 symbols are used for channel estimation. The data sequence was constituted by 57 random symbols $0 \ldots 7$ from a rectangular distribution. *Gray encoding* (see Figure 4), where symbol values with adjacent positions in symbol space only differ by one bit, was used to reduce bit error rates.

The received and decoded data sequences were compared to the transmitted true sequences, and the Bit Error Rate, *BER*, was calculated. Throughout the results, the BER will be used as the measure of decoding quality. Actually, *raw Bit Error Rate* is a more exact terminology, since BER often refers to signals that have passed through the whole communication system, including channel encoding and interleaving. Notice that the MLSE algorithm is designed to minimize a *metric*, not the BER. Still, BER is the commonly used measure of decoder performance.

When calculating the BER, the three last data symbols in each burst were omitted, since these are not decoded using the full strength of the Viterbi algorithm. This is because a proper MLSE decision on a symbol has to be made using information from the L succeeding symbols, due to ISI. Hence, only 54 symbols per burst were used for analysis. To achieve full decoding of all symbols, a tail sequence, known to the decoder, must be transmitted succeeding the data sequence. As can be seen in Figure 3, this is done in today's GSM.

3.1.4 Significance of Simulations Results

Assume the Bit Error Rate under certain circumstances is equal to p. Let X be the sum of n independent observations of BER. Then X will follow a

binomial distribution $X \in Bin(n,p)$, with $E(X) = np$ and $V(X) = np(1-p)$. The BER is estimated as $\hat{p} = \frac{X}{n}$, and $E(\hat{p}) = p$, $D(\hat{p}) = \sqrt{\frac{p(1-p)}{n}}$. See [12] for more information. If BER is to be estimated with a relative accuracy of $\varepsilon = \frac{D(\hat{p})}{\hat{p}}$ then the number of samples n should satisfy the expression

$$\varepsilon = \frac{\sqrt{np(1-p)}}{np}.$$

Using the approximation $(1-p) = 1$, which is reasonable since the BERs are small, the relation can be expressed

$$np = \frac{1}{\varepsilon^2}.$$

Note that np is the total number of error bits. If simulations proceed until 100 error bits are detected, the BER will be estimated with an accuracy of about $\pm 10\%$.

In the simulations 10% accuracy in BER is provided down to approximately the $3 \cdot 10^{-3}$ BER level. Lower BERs will be less accurate, and in some figures a cut has been made at the 10^{-4} BER level.

3.2 Implementation of Decoders

The implementation of Viterbi MLSE decoders had two major purposes. The first was to design the MLSE/Decision Feedback and the Joint MLSE algorithms and evaluate their performance for 8-PSK signals. In addition, Ericsson needed general Matlab implementations of the decoders for future simulation use. Thus, much effort was made to implement decoders as generally as possible. All the Matlab decoder functions will process

- any memoryless signal modulation type
- any ISI channel vectors H and G (when present)
- any data sequence length
- information concerning known symbols preceding the data sequence.

To the user they will return

- the decoded data sequence
- the decoded interfering data sequence (when present)
- a metric of the decoding quality.

Note that there are practical limitations for the input parameters. Algorithm complexity grows significantly when increasing the number of possible values per symbol, M, or the length of H or G.

3.2.1 General Algorithm Design

The MLSE design is based on a *Survivors* array, containing all surviving states at a certain time. A Conventional MLSE contains M^L states, while a Joint MLSE contains M^{L+K} states, where L and K refer to the ISI memory length of H and G, respectively. Each surviving state represents a certain path through the L latest symbol positions, as well as the K latest interfering positions for the Joint MLSE algorithm. The number of states is equal to the number of possible paths, which implies that two states cannot represent the same path. This fact is used for optimization, and for each state $S = 0, 1, \ldots$ in the Survivors array, the number S itself represents the specific latest L symbols' path. This is encoded by interpreting each state S as a M-ary number, letting the M-ary digits represent the L latest symbols, as shown by the example in Table 2.

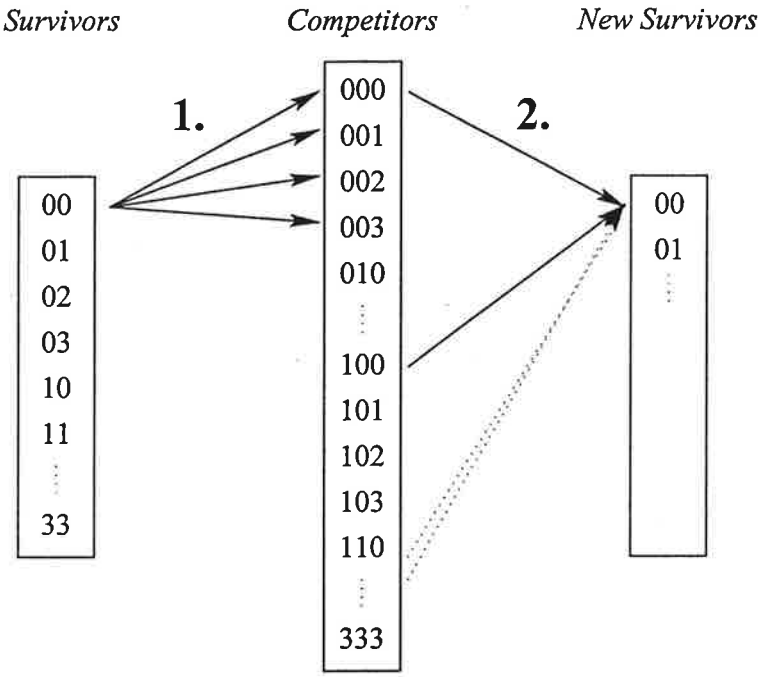
State S	*4-ary representation*	*Latest L path*
0	000	0, 0, 0
1	001	0, 0, 1
2	002	0, 0, 2
3	003	0, 0, 3
4	010	0, 1, 0
5	011	0, 1, 1
⋮	⋮	⋮
63	333	3, 3, 3

Table 2: Latest path encoding for each state in the Survivors array. In this example the modulation $M = 4$ and the memory length $L = 3$.

In addition to representing a path, each state in the array contains the historical path preceding the L latest symbols, and a *Euclidean distance metric*, which is the accumulated Euclidean distance between the path signal and the actual received signal.

First in each cycle of the algorithms, a *Competitors* array is created. It consists of states representing all surviving paths from the previous cycle, which are copied and extended to include all possible values of the symbol received during the present cycle. The metrics of all these states are calculated, and states with identical latest L symbols' paths, including the present cycle symbol, are compared. In these comparisons, only the state with the smallest metric value for each L symbols' latest path is passed on to the new Survivors' array. See Figure 7 for an example.

A main advantage of this path encoding system is that *no search is needed*



1. *A Competitors array is created from the Survivors array. Each state in the Survivors array results in four states in the Competitors array.*

2. *The (four) states whose latest 2 symbols are equal are compared to each other, and the one containing the smallest metric value wins. The winner is stored in the new Survivors array, at the correct position encoding the latest 2 symbols. For example, suppose the 16th Competitors state, encoded* 100 *in the figure, was the winner. Then the symbol* 1 *will be added to the historical path belonging to the state* 00 *in the new Survivors array.*

Figure 7: A step in the Viterbi algorithm. $M = 4$ *and* $L = 2$.

when comparing adequate states to each other, since the array position itself represents the path. The positions in the Competitors array are encoded similarly to the Survivors' array. This informs which positions should be compared to each other and into which position in the Survivors' array the surviving state should be stored. Another advantage is that no time or memory is used for saving the L latest symbols for each state.

3.2.2 Technical Details

The algorithm as implemented is not able to process H or G vectors with less than two positions. If shorter vectors are inputed, the program will automatically add a zero element, which may cause greater complexity than expected by the user. Note that if H or G contains only a single element, there is no intersymbol interference present, and consequently there is no need for an MLSE decoder.

The user may input an *initializing sequence* containing the symbols immediately preceding the data sequence. This would be the case, e.g. in the simulation bursts, where the data sequence is preceded by a known training sequence. In this case, the program first compensates for ISI influence from the initializing sequence on the received sequence, and then starts the decoding algorithm assuming that the data sequence was preceded by zeros only.

Before entering the "stationary" phase described in Figure 7, the algorithm needs to build a startup Survivors' array. To accomplish this a necessary number of the first steps in the data sequence are used (see Section 2.3.3).

3.2.3 Overview of the decoders

Decoder	*Input parameters*	*Complexity*
MLSE	Signal type, H, Received sequence, Sequence preceding u	$\mathcal{O}(M^{L+1})$
MLSE / Decision Feedback	Signal type, H, G, Received sequence, Sequence preceding u, Sequence preceding v	$\mathcal{O}(M^{L+2})$
Joint MLSE	Signal type, H, G, Received sequence, Sequence preceding u, Sequence preceding v	$\mathcal{O}(M^{L+K+2})$

Table 3: A brief overview of the decoders implemented in Matlab.

Table 3 shows a compilation of input parameters and algorithm complexity for the implemented decoders. L refers to the ISI memory length of H, i.e. the number of elements in H minus one. Analogously, K is the number of elements in G minus one. Notice that complexity grows by a factor of M as Decision Feedback is applied for the v sequence, and by a further factor of M^K as Joint MLSE is applied. M refers to the number of possible values for each symbol, defined by $M = 2^n$, where n is the number of bits per symbol.

Similarly, the signal type inputed to the program must contain M symbol values and their corresponding positions in signal space.

4 Simulation Results

The channels described in Section 3.1.2 were used in all simulations, i.e. $H = (\ 0.4\ 1\ 0.5\ )$ and $G = (\ 1\ 1\ )$.

4.1 Conventional MLSE in a CCI Environment

The first results show a Conventional MLSE performing in a CCI environment. The Conventional MLSE uses the model

$$y_t = Hu + e_t,$$

while the real channel can be described as

$$y_t = Hu + Gv + \tilde{e}_t,$$

including an interfering channel G.

The conventional MLSE will treat the interfering signal Gv as white noise. The performance for the MLSE for different SNR and C/I values is illustrated in Figure 8. Note the symmetry around the line C/I = SNR, which shows that, in this case, co-channel interference is approximately equivalent to white noise interference.

4.2 Performance of MLSE/DF and Joint MLSE

In a CCI environment a decoder uses the "real" model

$$y_t = Hu + Gv + e_t$$

and will be far more successful than a Conventional MLSE. In the following sections the simulation results for MLSE/DF and Joint MLSE are presented.

Estimation of H and G

For channels H and G it is assumed that

- bursts, including training sequences, are synchronized.
- training sequences of H and G are known to the decoder.

H and G are estimated from the synchronized training sequences using LS (see Section 2.3.2). In [2], it is shown that the performance of the estimation is strongly dependent on the cross-correlation between the training sequences used by the carrier and the interferer. However, today's 8 existing GSM training sequences are not designed to minimize cross-correlations. In

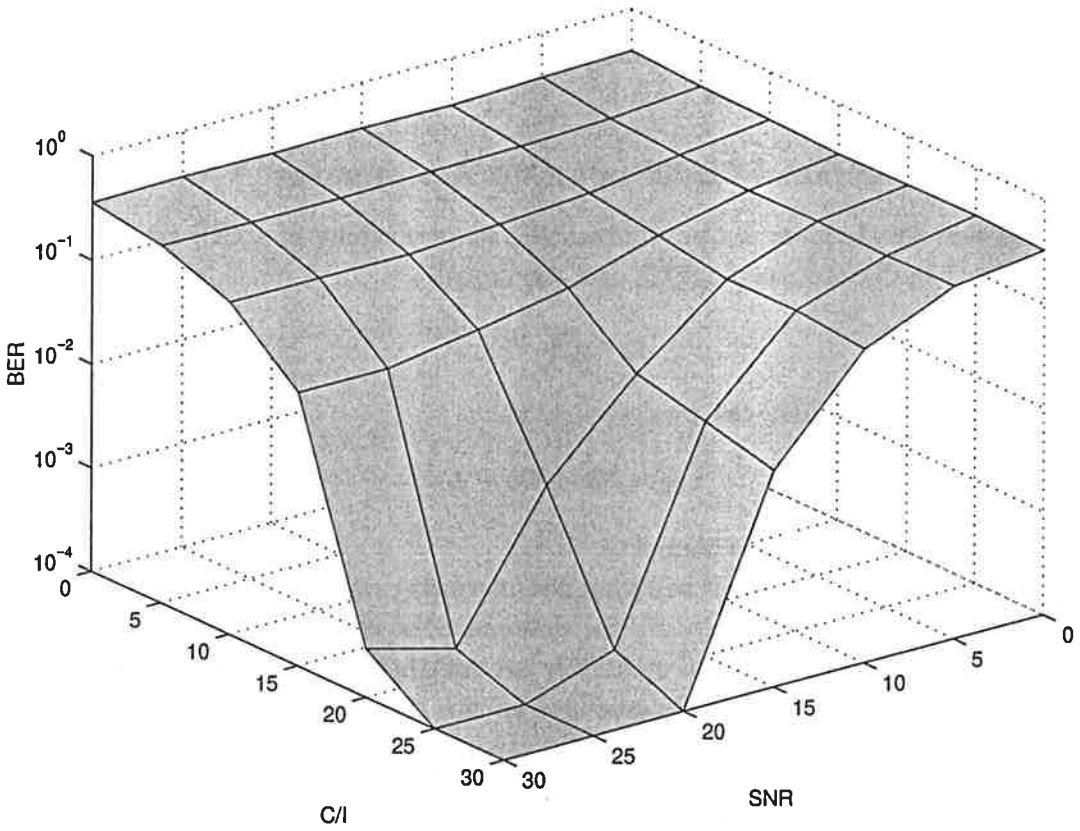


Figure 8: Performance of Conventional MLSE in an AWGN and CCI environment. Channels are estimated from known training sequences. Note the approximate symmetry around C/I=SNR, implying that performance degradation caused by co-channel interference is equivalent to degradation caused by white noise of similar strength.

the simulations of this report, training sequence 0 has been used for the carrier and training sequence 1 for the interferer. According to [2], the best combination of training sequences is 0/2 and the worst is 4/5. The combination 0/1 is a bit worse than average in performance, implying that the results presented below could be further improved by using better combinations or improved design of the training sequences.

4.2.1 Constant Channels

In Figure 9, the performance of the Joint MLSE detector is shown. When compared to Figure 8, it is obvious that performance is drastically improved

when the interferer is stronger than the white noise, i.e. I/N $\gg$ 0 dB. This implies that for SNRs up to about 15 dB no big improvement can be expected.

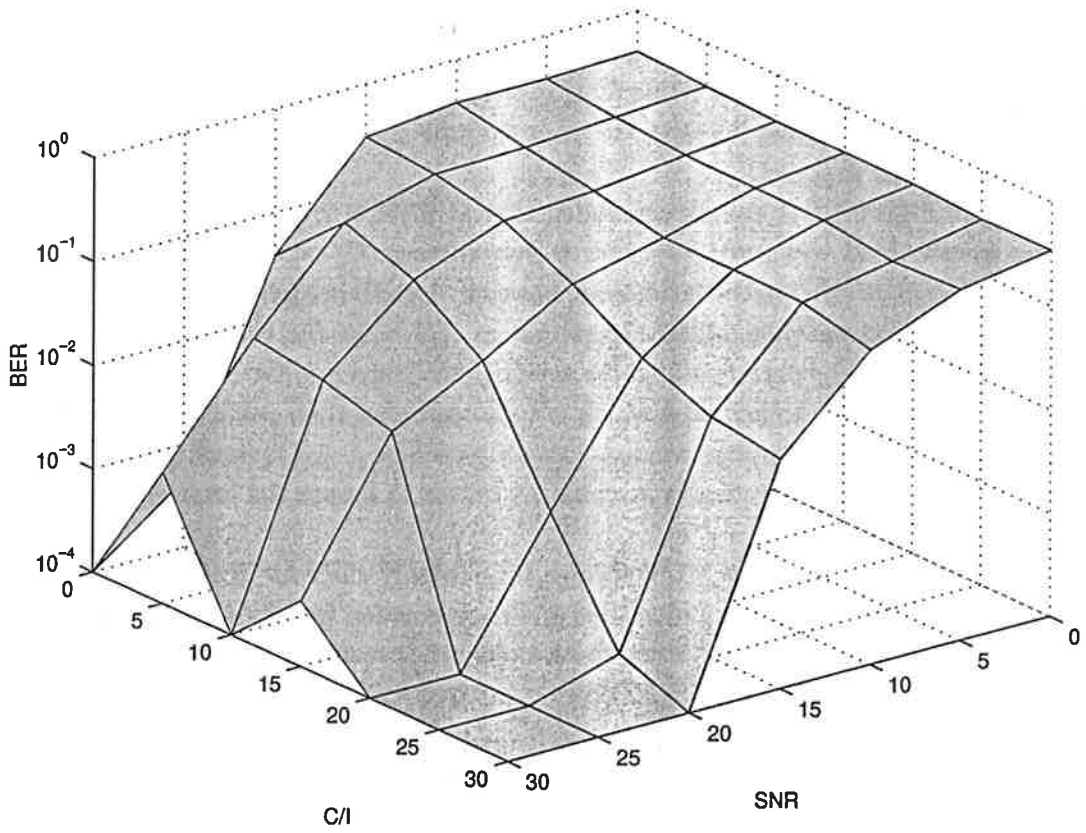


Figure 9: Performance of Joint MLSE. Channels are estimated from known training sequences. Performance compared to Conventional MLSE is significantly improved when C/I $\ll$ SNR.

Figures 10, 11 and 12 show the results in detail for SNR $= 20, 25, 30$ dB. For data transmission, a raw BER no higher than 5% is desirable. For lower values, channel encoding and interleaving will be able to correct practically all errors (see [1]).

Consistently, the Joint MLSE and the MLSE/DF show their worst performance for Interferer to Noise Ratios of about 15 dB. At this level the interferer seems harder to separate from the noise, but still strong enough to confuse the decoder.

From Figure 10, it can be concluded that at SNR $= 20$, 5% BER or less can be obtained when the interferer is very strong (C/I < 1 dB) or quite weak (C/I > 13 dB). In the latter case, however, 1–5 dB are gained compared to using a Conventional MLSE. Note that at C/I $= 25$ dB, the MLSE/DF algorithm actually performs slightly worse than Conventional MLSE. This comes from overfit to a model which is more complex than reality, i.e. the model assumes an interferer whereas in reality there is not. Obviously, BERs are very low, but theoretically it would be better to ignore the interferer in this case.

At SNR ≥ 25 dB Joint MLSE proves to be highly efficient. Strong interferers are almost completely identified and compensated for. From Figure 11 can be seen that even at its "worst" performance between C/I $= 5$ and 10 dB, Joint MLSE reduces the BER 10 times compared to Conventional MLSE.

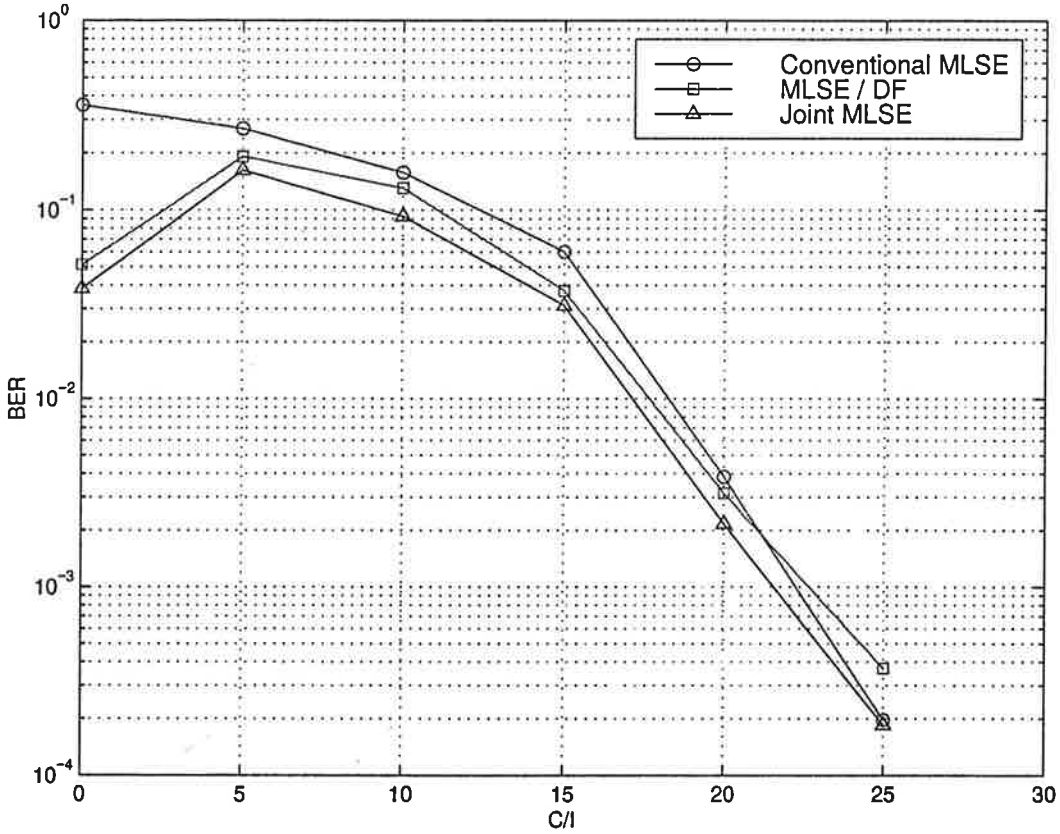


Figure 10: Performance at $SNR = 20$ dB of Conventional MLSE, MLSE/DF and Joint MLSE. Channels are estimated from known training sequences. Notice that at $C/I = 25$ dB the MLSE/DF performs worse than the Conventional MLSE, due to overfit.

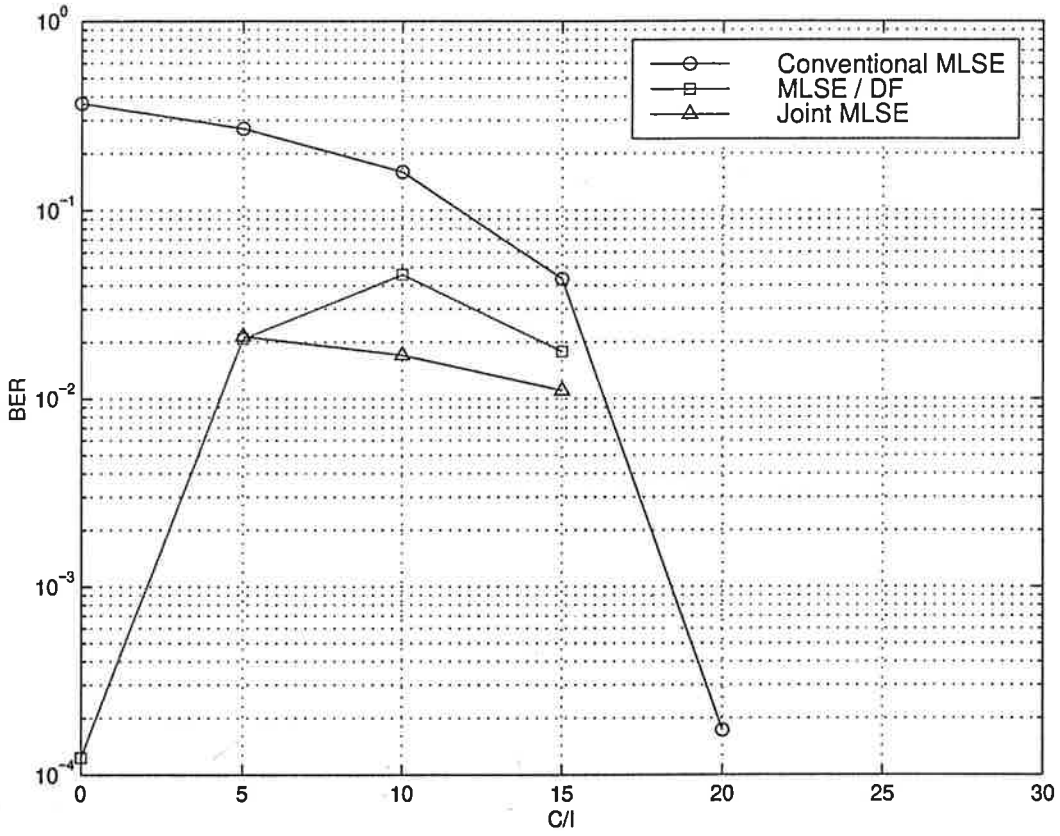


Figure 11: Performance at SNR = 25 dB of Conventional MLSE, MLSE/DF and Joint MLSE. Channels are estimated from known training sequences. Missing data points represent very small or zero Bit Error Rates (BER $< 10^{-4}$).

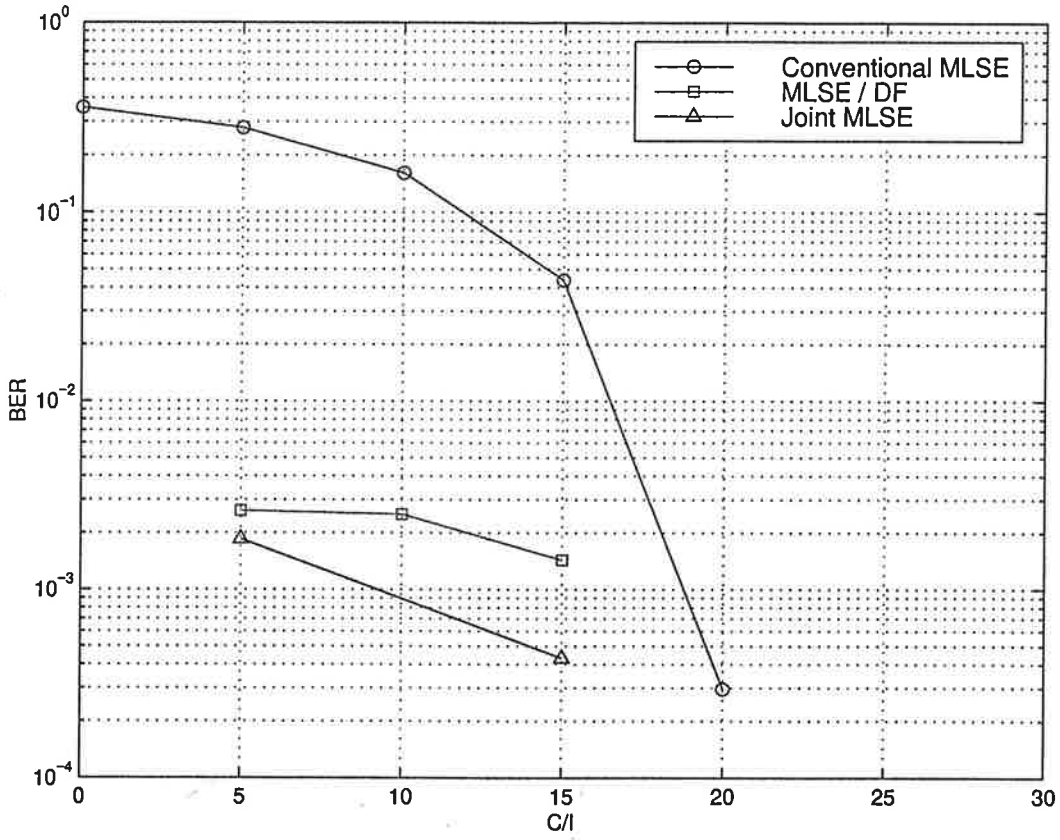


Figure 12: Performance at SNR = 30 dB of Conventional MLSE, MLSE/DF and Joint MLSE. Channels are estimated from known training sequences. Missing data points represent very small or zero Bit Error Rates (BER $< 10^{-4}$). Apparently, at very low noise levels the performance gain when using Joint Detection is huge.

4.2.2 Fading Channels

As discussed in Section 3.1.2, SNR and C/I for fading channels should be interpreted as *average values*. The individual bursts are sent through randomized channels, causing a great difference in quality. Performance of the decoders is reduced, since data transmitted in low-quality bursts will render a significant number of bit errors, which cannot be compensated for during high-quality bursts. In Figures 13, 14 and 15, results are shown in detail for SNR $= 20, 25, 30$ dB.

The plots show huge improvements in BER — in some cases up to 1000 times — for Joint Detection. All plots appear to reach a "noise floor", the level of which depends on the SNR. In Figure 15, the performance of Joint Detection at SNR = 30 dB appears to depend more on this noise from low-quality bursts than on the strength of the interferer.

Clearly, Joint Detection brings huge improvements to decoding for fading channels as well.

4.3 Choice of Decoder

From the results in 4.2, it is clear that the Joint MLSE decoder is the most efficient. This also verifies theory. However, the Joint MLSE algorithm increases complexity by $\mathcal{O}(M^{K+1})$ compared to the Conventional MLSE. As shown in the results, the MLSE/DF algorithm has a performance not far from the Joint MLSE, in most cases providing 95% of the Joint MLSE BER reduction. For the MLSE/DF, complexity is only increased by $\mathcal{O}(M)$ compared to the Conventional MLSE.

The tradeoff between complexity and performance deserves deeper analysis than is possible in this study. However, the less complex estimation of the interferer carried out by the MLSE/DF proves to be quite successful.

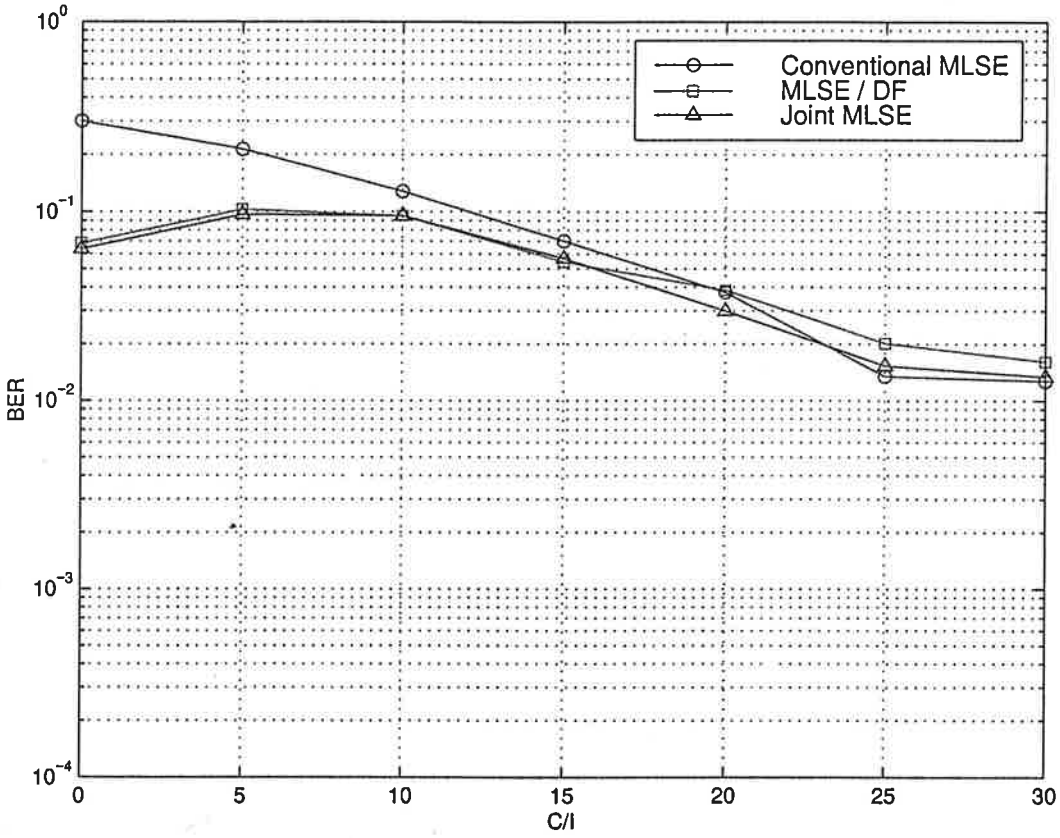


Figure 13: Performance at SNR = 20 dB of Conventional MLSE, MLSE/DF and Joint MLSE when decoding multipath fading carrier and interferer channels. Channels of each burst are estimated from known training sequences. The Joint Detection performance gain at BER = 5% is about 2 dB. Due to overfit, Joint Detection algorithms slightly degrade performance compared to Conventional MLSE when $C/I \geq 25$ dB.

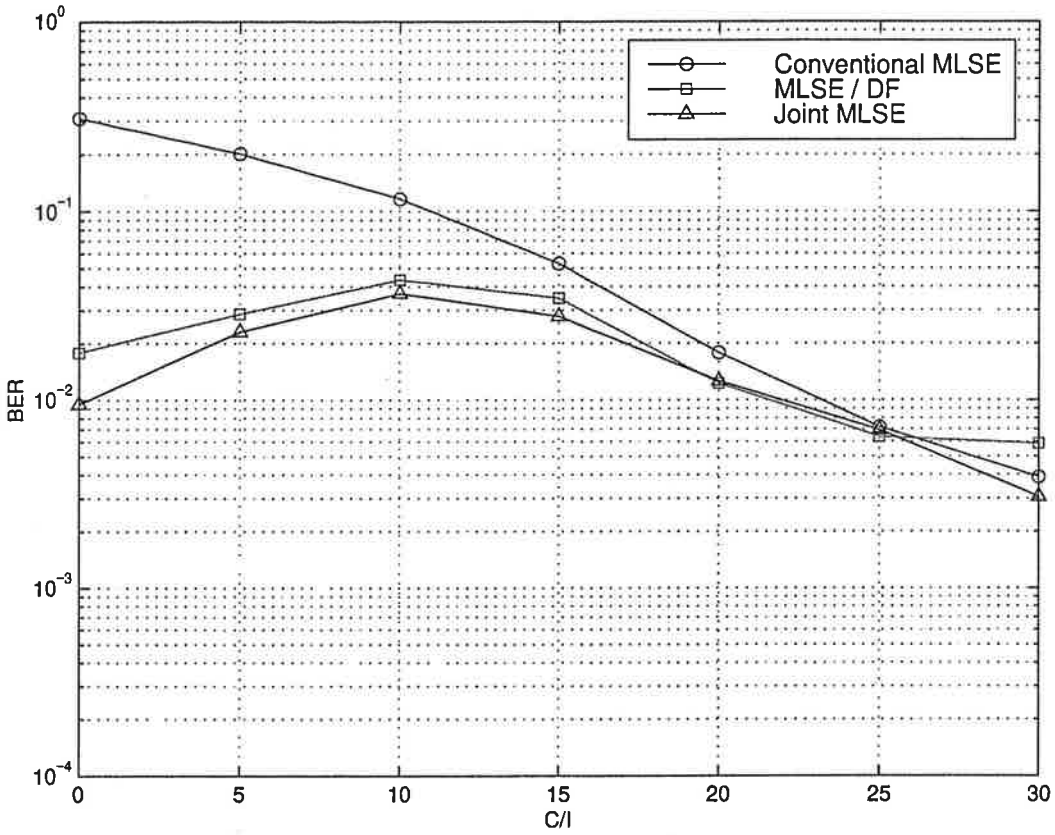


Figure 14: Performance at SNR = 25 dB of Conventional MLSE, MLSE/DF and Joint MLSE when decoding multipath fading carrier and interferer channels. Channels of each burst are estimated from known training sequences. For Joint Detection, a BER of 5% or less is achieved at any strength of the interferer, whereas the Conventional MLSE requires C/I > 15 dB for BER to fall below 5%. Due to overfit, Joint Detection algorithms slightly degrade performance compared to Conventional MLSE when C/I ≥ 25 dB.

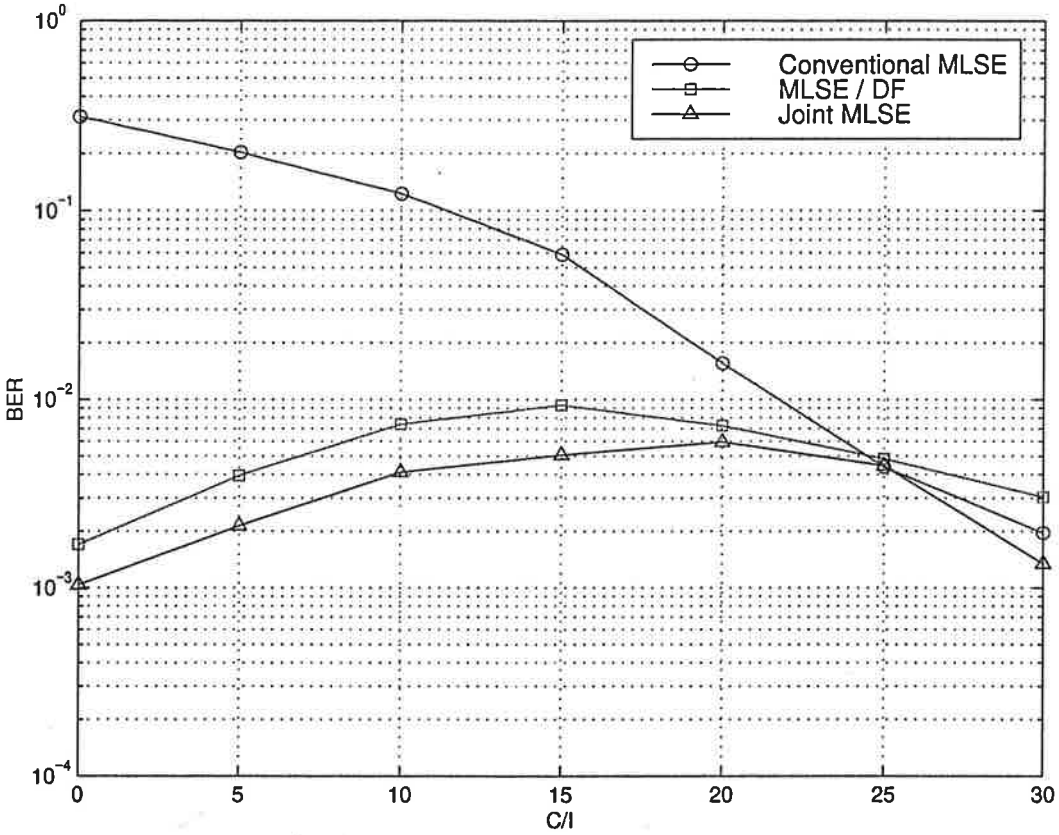


Figure 15: Performance at SNR = 30 dB of Conventional MLSE, MLSE/DF and Joint MLSE when decoding multipath fading carrier and interferer channels. Channels of each burst are estimated from known training sequences. At the low noise level of SNR = 30 dB the Joint Detection algorithms will efficiently decode both the carrier and interferer signals, irrespective of their relative strength.

4.4 Complexity Reduction by Channel Memory Truncation

One way to reduce the number of calculations is to assume a channel H in the decoder that is truncated in length compared to the real channel. If the tap or taps left out are small this approximation may seem reasonable. However, according to [2], this causes severe degradation in performance. This is illustrated by simulations presented in Figure 16. In this example a signal through a channel $H_{real} = \begin{pmatrix} 1 & 1 & \alpha \end{pmatrix}$ is decoded assuming the channel $H_{decoder} = \begin{pmatrix} 1 & 1 \end{pmatrix}$, i.e. the energy sent in the third tap α is ignored by the decoder. No interferer is present in the example.

Notice that the tap $\alpha = 0.1$ only represents -23 dB of the signal energy. Still, ignoring it clearly reduces performance. I.e. $\alpha = 0.3$ which represents about -14 dB of the signal energy, causes severe degradation results.

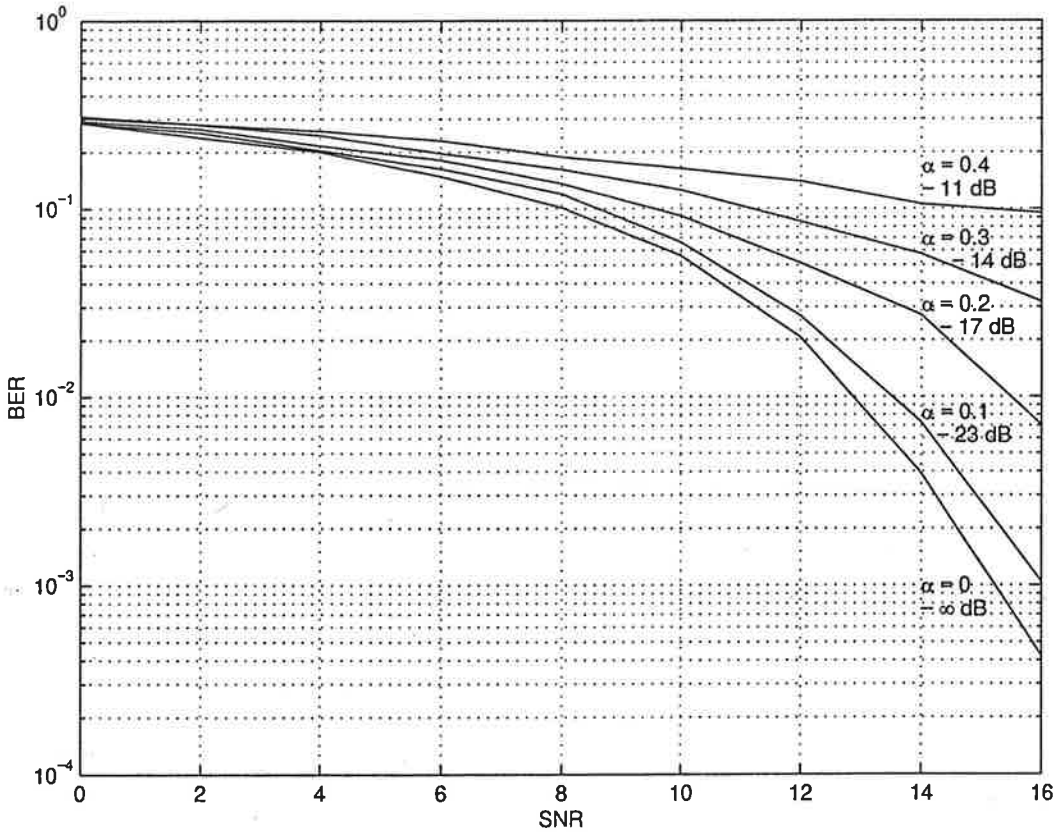


Figure 16: Performance using channel memory truncation. A signal through a channel $H_{real} = (\ 1\ 1\ \alpha\ )$ is decoded assuming the channel $H_{decoder} = (\ 1\ 1\ )$, $\alpha = 0, 0.1, \ldots, 0.4$.

5 Future and Associated Research

This study indicates that before Joint Detection interference cancellation techniques can be implemented, additional methods for identifying CCI environments need to be developed. The performance of the MLSE/DF algorithm in Figure 13 shows that, for high C/I values, performance is degraded when trying to decode the interferer.

In Section 5.1, a detector designed by Bo Bernhardsson [5] is presented. Section 5.2 discusses blind detection, a method to estimate the channel without knowing the training sequence. Finally, in Section 5.3, different methods to reduce complexity of the Viterboi algorithm are discussed.

5.1 An Interference Detection Method

The goal of the detector is to determine whether there is channel interference from another source. It is assumed that the remaining noise after (perfect) channel estimation and Viterbi decoding of the main source has either of the forms

$$\begin{aligned} \varepsilon_k &= e_k && \text{no interference} \\ \varepsilon_k &= Gv_k + e_k && \text{interference} \end{aligned}$$

where $e_k \in N(0, \sigma^2)$ is Gaussian noise, G is the interference transfer function and v_k is the output resulting from the accidental decoding of an interfering signal.

If the interfering signal is 8-PSK modulated, v_k can be assumed to have the form

$$v_k = e^{i(n_k\pi/4+\phi_k)}$$

where $n_k \in (0, 1, \ldots, 7)$ is determined by the transmitted interference symbol k and ϕ_k is the unknown phase. It might be natural to model ϕ_k with a rectangular random variable in $[0, 2\pi]$, independent of e_k.

The Case of a One-tap $G = g_0 = m$

The complex random variable ε_k will have a probability distribution that is circular and radially has a Rice distribution

$$p_R(r, m, \sigma) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2 + m^2}{2\sigma^2}\right) I_0\left(\frac{mr}{\sigma^2}\right); \qquad r \geq 0$$

where I_0 is the modified Bessel function of the first kind. The case with no interference is the special case $m = 0$ where p_R reduces to a Rayleigh

distribution

$$p_R(r, 0, \sigma) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

For large m/σ the distribution is close to Gaussian with mean m

$$p_R(r, m, \sigma) \approx \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r-m)^2}{2\sigma^2}\right); \qquad m/\sigma \text{ large}$$

For more information, see [6].

The random variable is illustrated for $\phi_k = 0$ with 114 samples in Figure 17.

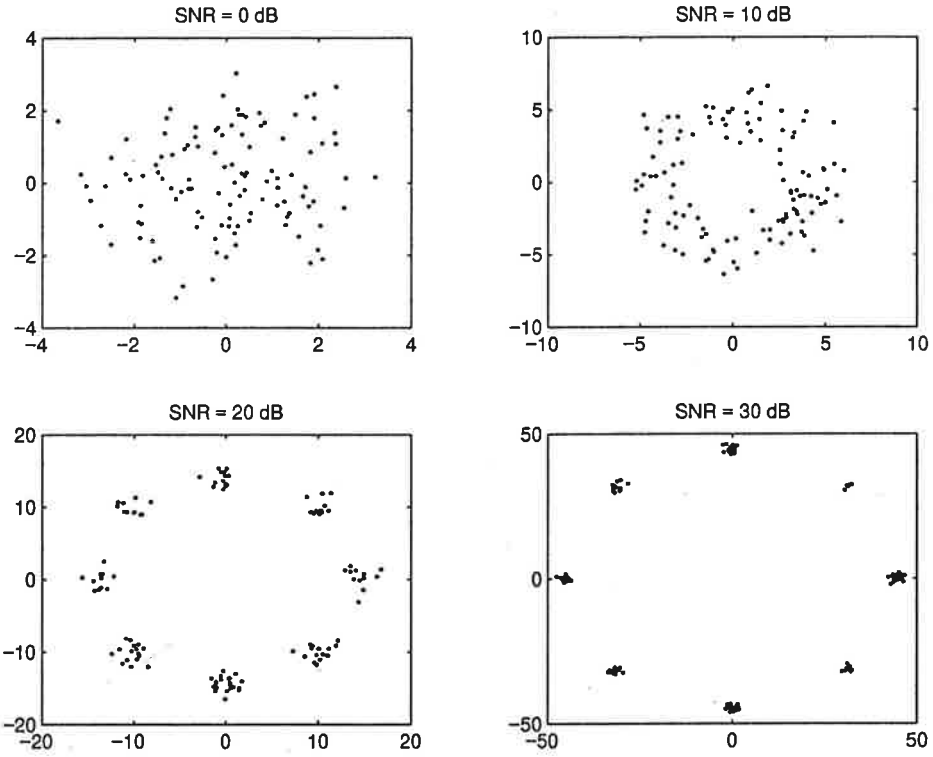


Figure 17: The random variable ε_k plotted in IQ signal space for $\sigma = 1$ and varying m. The SNR is defined as $10 \log m^2/2\sigma^2$.

Maximum Likelihood Estimation of m and σ

The m and σ can be estimated by solving

$$\max_{m,\sigma} \sum_i \log\left(\frac{r_i}{\sigma^2}\right) - \frac{r_i^2 + m^2}{2\sigma^2} + \log\left(I_0\left(\frac{m r_i}{\sigma^2}\right)\right).$$

Simulations show that this function may have local minima, especially when $m < \sigma$. In addition, the ML estimator is complex to implement.

Moment Estimator

A less complex, but still accurate, method is the moment estimator. The moments of r are given by

$$M_\nu = E(r^\nu) = (2\sigma^2)^{\nu/2}\Gamma(\frac{\nu+2}{2})_1F_1(-\frac{\nu}{2}, 1, -\frac{s^2}{2\sigma^2})$$

where ${}_1F_1(a,b,c)$ is the hyper-geometric function. From the moments

$$M_1 = E(r_i) = m$$

$$M_2 = E(r_i - M_1)^2 = m^2 + 2\sigma^2,$$

a test quantity F is constructed:

$$F(\gamma) = \frac{M_2}{M_1^2} = 1 + \frac{1}{\gamma}; \qquad \gamma = \frac{m^2}{2\sigma^2}$$

In Figures 18 and 19, plots of the test quantity values for 200 simulated bursts are presented. In these plots, the detector tries to separate the cases of an interferer with $I/N = 10$ dB from no interferer at all. It can be concluded that given a sample length of 148, which is about one GSM burst length, practically no wrong decisions are made. When given a sample length of 26, which is the length of the GSM training sequence, about 0.3% of the decisions are incorrect.

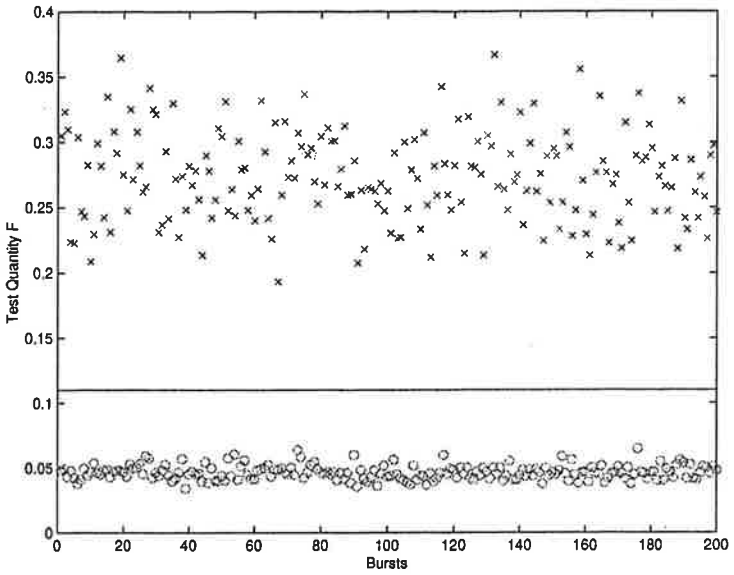


Figure 18: The moment detector Test Quantity F using a sample length of 148 symbols. 200 bursts are simulated. The interferer strength is $I/N = 10$ dB ($\times$), compared to no interferer present ($\circ$). Practically no wrong decisions are made.

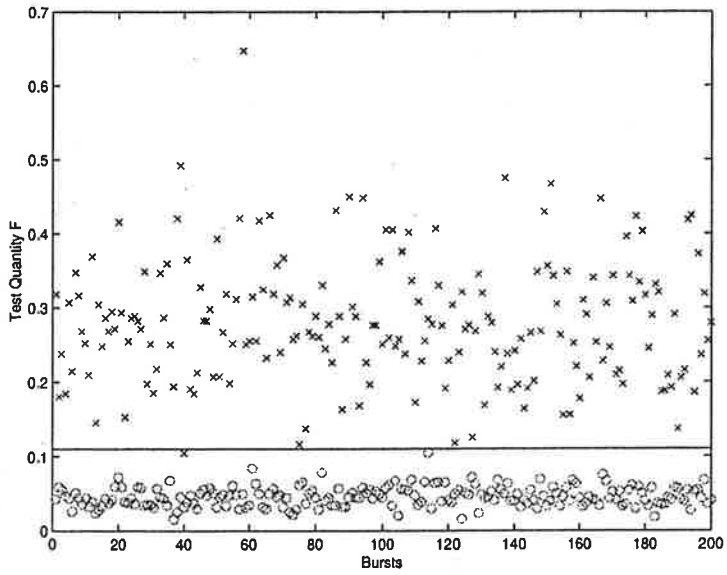


Figure 19: The moment detector Test Quantity F using a sample length of 26 symbols. 200 bursts are simulated. The interferer strength is $I/N = 10$ dB ($\times$), compared to no interferer present ($\circ$). About 0.3% of the decisions will be incorrect.

5.2 Blind Detection

In this study, the assumption has been made that the training sequence of the interferer is known to the decoder. In reality there is a need for identification and estimation of the interfering channel without this knowledge. Thus, the problem is to identify the interferer, only knowing that it is transmitting symbols from a known alphabet (i.e. the 8-PSK waveforms) through an unknown ISI channel. This is a *Blind Detection* problem, which needs to be further studied in order to make a successful implementation of the Joint Detection decoders.

Blind Detection could be applied to the residuals of the training sequence left after estimating the carrier channel H. In GSM, this would give a 26-symbol residual sequence to analyze for identifying an interferer, requiring a fast convergence of the estimation algorithm. Other possible solutions would be to implement various kinds of continuous channel trackers, i.e. algorithms built-in to the decoders, aiming to improve the channel estimates simultaneously while decoding the signal.

One algorithm that can be used for recursive estimation of a channel is the *Sato* algorithm, which is described in [7]. The description refers to a slightly different field of usage, but with some adaptions it may be useful. Basically, an equalization filter is recursively updated to minimize a loss function. A suitable loss function for PSK Modulation could be

$$V(\hat{c}) = \left[|\hat{u}|^2 - 1\right]^2,$$

where $\hat{c}$ is the equalization filter and $\hat{u}$ is a decision on a sent symbol. This function uses the information that the amplitude of all waveforms are equal for PSK Modulation. However, this algorithm is very sensitive to the initial parameters of $\hat{c}$. In addition, it requires several hundreds of symbols to converge.

Fredrik Gustafsson's work [7] suggests a more complex, but fast converging, method of estimating a channel (see [7]). It consists of one Kalman filter for each possible symbol sequence sent. Each Kalman filter produces an estimation of the channel, given its specific sent sequence. This requires a huge number of Kalman filters, especially for long sequences. However, the number of Kalman filters, and thus complexity, can be reduced using the same ideas as the Viterbi algorithm (see Section 2.3.3). According to Gustafsson, such a method will converge in 4–5 samples, which is a very good performance.

5.3 Complexity reduction of the Viterbi algorithm

The Viterbi algorithm is an implementation of the optimal MLSE. As discussed in Section 2.3.3 its complexity is $\mathcal{O}(NM^{L+1})$. This complexity becomes too high for practical implementations when the channel has a long memory, i.e. L is large. Since the Joint Detection methods developed in this thesis have even higher complexity, the issue of complexity reduction is of even more importance. Below, two methods of reducing the number of states, and consequently the complexity, will be discussed.

The most common way to reduce complexity is to reduce the number of states in the Viterbi algorithm by decision feedback, see [14]. For example, a five tap channel, $L = 4$, of an 8-PSK signal has 4096 states with corresponding metrics. The ideas behind MLSE and decision feedback can be combined by making a decision after, say, three taps. The two remaining taps are still allowed to affect the signal through DF. The number of states in the Viterbi trellis is now reduced to 64 which is easier to handle. Clearly, all combinations of symbols are not tested and the method is no longer optimal. However, assuming that most of the energy is in the first taps, the reduced performance is redeemed by the reduction of complexity.

Another way to reduce the number of states is to analyze the metrics of the paths and only let some of the paths survive. In [8] two ways of limiting the number of paths that survive are analyzed. They are called the M-algorithm and the T-algorithm. The basic principle of the M-algorithm is the retention of only the q best states at each stage of the trellis. A search must be made among the metrics of the states in order to select the best ones. These states survive whereas the others are discarded. The T-algorithm also discards the survivors with the worst metrics. The decision is now based on a threshold of the difference between a certain metric and the best metric. Consequently, survivors whose metric is too far off the currently best metric are discarded. Simulation results claim that the T-algorithm follows the performance of the full Viterbi algorithm very well whereas the M-algorithm is worse for high SNR. It should be noted that while the M-algorithm has a constant complexity the T-algorithm has not. In certain cases its complexity will increase, but will still be significantly lower than the full Viterbi algorithm.

The discussion above shows that there exists methods to reduce the complexity of the Viterbi algorithm. Similar methods could be constructed and applied to the Joint Detection equalizers used in the thesis.

6 Conclusion

This study shows that Joint Detection is a powerful method for eliminating co-channel interference in 8-PSK modulation, provided a carrier Signal to Noise Ratio of more than 15 dB, and an interferer stronger than background noise. A full-state Joint MLSE algorithm was implemented in Matlab and showed good performance though at the cost of great numerical complexity. As a less complex alternative, a Conventional MLSE for the carrier including a built-in Decision Feedback algorithm for the interferer was implemented and evaluated. In most cases, this MLSE/DF algorithm provided about 95% of the performance gain from Joint MLSE. Thus, for practical implementation, MLSE/DF may be an effective solution.

The results are based on the assumption that carrier and interferer bursts are synchronized, and that the training sequences of both carrier and interferer are known to the channel estimator. In order to detect and estimate an interferer using an unknown training sequence, which is the most realistic scenario, further studies need to be carried out in the field of Blind Detection. Some ideas for rurther research are presented in this study. As for detection, a moment estimator assuming a one-tap interferer is presented.

Implementing the decoders and simulation environments has demanded much work. Still, execution time remains a bottleneck when carrying out simulations, especially for the full-state Joint MLSE. For future simulations, implementing the decoders in a faster, compiling language may be considered.

A Table of Acronyms

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CCI	Co-Channel Interference
C/I	Carrier to Interferer ratio
CPM	Continous Phase Modulation
EDGE	Enhanced Data rates for GSM Evolution
GMSK	Gaussian Minimum Shift Keying
GSM	Global System for Mobile communication
I/N	Interferer to Noise ratio
ISI	Inter-Symbol Interference
LS	Least Squares
MLSE	Maximum Likelihood Sequence Estimator
MLSE/DF	MLSE with a built-in Decision Feedback estimator for the interferer
PSK	Phase Shift Keying
SNR	Signal to Noise Ratio (= Carrier to Noise ratio)
TDMA	Time Division Multiple Access
WCDMA	Wideband Code Division Multiple Access

References

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

[2] N. Edvardsson. *Studies of Joint Detection MLSE in the GSM system.* Master's thesis, Ericsson Radio Systems, Kista, 1996.

[3] L. Olbjer, U. Holst and J. Holst. *Tidsserieanalys.* Dept. of Mathematical Statistics, Lund Institute of Technology, 1996.

[4] European Telecommunications Standards Institute. *EDGE - Evaluation of 8-PSK.* ETSI, 1998. *Private communication.*

[5] Bo Bernhardsson. *Private communication.* Dept. of Automatic Control, Lund Institute of Technology.

[6] T. Öberg. *Modulation, detektion och kodning.* Studentlitteratur, 1998.

[7] F. Gustafsson and B. Wahlberg. "Blind equalization by direct examination of the input sequences," *IEEE Transactions on Communication*, Vol. 43, No. 7, pp. 2213–2222, 1995.

[8] H. Murata, S. Yoshida. "Maximum-Likelihood Sequence Estimation for Coded Modulation in the Presence of Co-Channel Interference and Intersymbol Interference," *1996 IEEE 46th Vehicular Technology Conference*, IEEE, New York, USA, 1996, 3 vol. xxxix+1887 pp. p. 701-5 vol. 2.

[9] M. Mouly and M.-B. Pautet. *The GSM System for Mobile Communication.* Cell & Sys, 1992.

[10] European Telecommunications Standards Institute. *Radio transmission and reception, GSM 05.05.* ETSI, 1997.

[11] E. Dahlman, B. Gudmundson, M. Nilsson and J. Sköld. "UMTS/IMT-2000 Based on Wideband CDMA," *IEEE Commun.*, vol. 36, no. 9, pp. 70–79, Sept. 1998.

[12] G. Blom. *Sannolikhetsteori med tillämpningar.* Studentlitteratur, Lund, 1984.

[13] L. Yiin and G. Stuber. "MLSE and Soft-Output Equalization for Trellis Code Continous Phase Modulation", *IEEE Transactions on Communication*, Vol. 45, No. 6, pp. 651–652, June 1997.

[14] A. Duel-Hallen and C. Heegard. "Delayed Decision-Feedback Sequence estimation", *IEEE Transactions on Communication*, Vol. 37, No. 5, pp. 428–436, May 1989.