NEW RECEIVERS FOR DIFFERENTIALLY ENCODED OFFSET-QPSK

-INVESTIGATIONS IN DIFFERENTIAL DEMODULATION AND PER-SURVIVOR-PROCESSING ALGORITHMS-

by

Sven Hischke

Thesis Submitted for the Degree of

DOCTOR OF PHILOSOPHY

CITY UNIVERSITY

Department of Electrical, Electronic

and Information Engineering

July 1999

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADC	Analog/Digital Converter
ARMA	Autoregressive-moving average
ASIC	Application Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CBDE	COSSAP Block Diagram Editor
COSSAP	Communication System Simulation and Analysis Package
CPFSK	Continuous Phase Frequency Shift Keying
CPM	Continuous Phase Modulation
DAC	Digital/Analog Converter
DAVIS	Data Visualisation Tool
DECT	Digital European Cordless Telephone
DFE	Decision Feedback Equaliser
DPSK	Differential Phase Shift Keying
DSP	Digital Signal Processing
ETSI	European Telecommunications Standards Institute
FPGA	Field Programmable Gate Array
GSM	Global System for Mobile Communications
GMSK	Gaussian Minimum Shift Keying
HDL	Hardware Description Language
ISI	Intersymbol Interference
LMS	Least Mean Square
MAP	Maximum a Posteriori Probability
MLSE	Maximum Likelihood Sequence Estimation
MSE	Mean Square Error
MSK	Minimum Shift Keying
OQPSK	Offset-QPSK
PDC	Pacific Digital Cellular Standard
PSP	Per-Survivor Processing
QAM	Quadrature Amplitude Modulation

QPSK	Quadrature Phase Shift Keying
SDS	Stream Driven Simulator
SNR	Signal-to-Noise Ratio
TETRA	Terrestrial Trunked Radio
USDC	United States Digital Cellular Standard
WSSUS	Wide Sense Stationary Uncorrelated Scattering

List of Relevant Symbols

$a(n)$	Binary Data
B	Bandwidth
δ_{ij}	Kronecker Delta
Δ	Step-Size
$D(x_1, x_2)$	Euclidean Distance
e_l	Estimation Error
E_b, E_s	Signal Energy per Bit, per Symbol
η	Modulation Index
f_c	Carrier Frequency
f_d	Doppler Frequency
$f_k(t)$	Orthonormal Basis Functions
F_{MSE}	Mean Square Error
$g_r(t), g_t(t)$	Impulse Response of Pulse Shaping Filter at the Receiver, Transmitter
$\bar{\gamma}_c, \bar{\gamma}_s$	Average Received SNR per Channel, per Symbol
$h(t), h_l$	Channel Impulse Response, Discrete Channel Coefficient
$\mathbf{I}$	Identity Matrix
$\Im()$	Imaginary Component
$J_{j,l}, J()$	Path Metric
k	Number of Bits in a Symbol
$k_l(i)$	Kalman Gain
L	Number of Echoes in a Channel
$\lambda_{nj,l}, \lambda()$	Branch Metric
M	Number of States in the M-algorithm
M	Number of States of a Symbol
$n(t), n_l$	Additive Noise Signal, Sample
$n'(t), n'_l$	Filtered Additive Noise Signal, Sample
N_0	Noise Power Spectral Density
φ_m	Phase Value
$p()$	Probability Density Function (pdf)
$P(x|y)$	Conditional Probability

P_b, P_s	Bit/Symbol Error Probability
$P_o ut$	Out-of-Band Power
$r(t), r_l$	Received Signal after Matched Filtering, Sample
r	Roll-off Factor
$\rho(0)$	Diagonal Elements of $\mathbf{R}_{xx}$
$\mathbf{R}_{xx}$	Correlation Matrix
$\Re()$	Real Component
$s(t), s_l$	Transmitted Signal, Sample
$\bar{s}(t)$	Complex Envelope of $s(t)$
σ^2, S	Signal Power
σ_τ	root mean squared (rms) delay spread
$\bar{\tau}$	Mean Excess Delay
$\tau_m(\text{X dB})$	Excess Delay Spread (X dB)
T, T_s	Symbol Period
T_b	Bit Period
$v(t), v_l$	Received Signal, Sample
x^*	Conjugate Complex of x
$\hat{x}$	Estimate of x
$\mathbf{x}^{\mathrm{T}}$	Transpose of $\mathbf{x}$
$\mathbf{X}^{\mathrm{H}}$	Hermitian of $\mathbf{X}$
$z(t), z_l$	Complex Modulating Signal $z(t) = x(t) + \mathrm{j}\,y(t)$
$\tilde{z}_l$	Filtered Complex Data Symbol
$\mathbf{z}$	Vector of $\tilde{z}$

Acknowledgement

Declaration

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

Before a new mobile radio system will be standardised, an intense process of decision making is required concerning the different aspects of the system. One aspect is the definition of the air interface and in particular the employed modulation method. During the standardisation of the TETRA (Terrestrial Trunked Radio) system, the expert group decided to chose the linear modulation format $\pi/4$-QPSK. One important point for decision making was the requirement to support power efficient, nonlinear amplifiers. Although the second candidate, Offset-QPSK (OQPSK) shows further reduced envelope fluctuations, the nonavailibilty of a conventional differential demodulator for OQPSK led to a clear preference for $\pi/4$-QPSK. Differential demodulation is an attractive combined carrier phase estimation and demodulation technique on fast time selective fading channels.

The work presented in the thesis is concerned with the synthesis and analysis of receiver algorithms for differentially encoded OQPSK signals, which fulfill the requirements of modern mobile radio systems.

The thesis begins with an overview of the theoretical principles of digital modulation and optimum receivers. A description of mobile radio channels and the theoretical performance of PSK signals complete the basic chapter.

A new receiver structure for DOQPSK using differential demodulation followed by a Viterbi decoder is then proposed. On this basis, an advanced algorithm suitable for general roll-off factors and with reduced complexity is derived and investigated. Following the differential demodulation, original algorithms for joint coherent demodulation and equalisation, which are based on the principle of per-survivor processing (PSP) are synthesised and discussed. These receiver structures are extended for the use in time and frequency selective mobile radio channels.

With the aid of a computer simulation system the derived receiver structures are investigated. Simulation results on different linear and nonlinear channels are presented and compared with results obtained with standard $\pi/4$-DQPSK receivers and the theoretical bounds.

The thesis concludes with a description of the simulation tool COSSAP and gives examples of created and implemented simulation models. The results of a fixed-point analysis of one model is presented and the possibilities to synthesise a hardware implementation is discussed.

Chapter 1

Introduction

For a number of mobile communication systems which will be introduced in the near future, investigations of modulation formats and algorithms for channel estimation and equalisation are required in order to define the air interface.

Two important requirements should be fulfilled by the modulation format. Firstly, it should be bandwidth and spectrum efficient. Secondly, since the power sources are limited in subscriber radio units, the generation of a modulated signal with small envelope fluctuations is recommended. An envelope with reduced fluctuations supports power efficient, nonlinear amplifiers (see Appendix A). The attempt to fulfill these two requirements with one modulation format has been and still is subject of many investigations.

In the new European mobile communication standard TETRA (Terrestrial Trunked Radio) the modulation format $\pi/4$-QPSK was specified. The same modulation format is also used by the Pacific Digital Cellular System (PDC) and for the second generation United States Digital Cellular Standard (USDC). The choice of $\pi/4$-QPSK is motivated by two observations: Firstly, the envelope of $\pi/4$-QPSK has no zeros. This reduces the generation of harmonics and the distortion of the signal by nonlinear amplifiers [2]. Secondly, a differential precoding of $\pi/4$-QPSK signals allows for simple differential demodulation. Differential demodulation is an attractive combined carrier phase estimation and demodulation technique on fast time selective fading channels. Differential demodulation, however, does not resolve intersymbol interference introduced by a frequency selective channel.

In addition to $\pi/4$-QPSK, Offset-QPSK (OQPSK) was also a candidate for the TETRA system. Although OQPSK shows further reduced envelope fluctuations, the nonavailability of a differential demodulator for OQPSK led to a clear preference for $\pi/4$-QPSK. The difficulty of differential demodulation is due to the time overlap of the I- and Q-signals, because in OQPSK the I- and Q-channels are staggered, i.e., time

23

shifted by half a symbol period. This time shift leads to a mutual interference of the I- and Q-components. Conventional differential demodulators, like those described in [26], can not be applied anymore. Motivated by the assumption that OQPSK modulated signals require coherent demodulation [10], some authors ([20] and [6]) have investigated the use of differential demodulation. In both analyses, however, only unfiltered signals are considered.

Applying a coherent demodulation, the I- and Q-components can be decoded separately without crosscomponent interference, if the constant time shift between the two component signals is taken into account. The difficulty of coherent demodulation in mobile radio applications, however, is to find receivers, in which decoding is nearly optimal, with additional requirements for fast tracking in time selective channels, short acquisition time in burst mode transmission, and resolution of the intersymbol interference in frequency selective channels.

The purpose of the research work presented in the thesis has been the development of receiver algorithms for differentially encoded OQPSK signals. The new receiver algorithms have to fulfill the requirements of mobile radio systems. The main results in the accomplishment of the purpose are given in Chapter 4 and 5. Chapter 4 describes the implementation of differential demodulation techniques. The basic structure of the receiver, which uses a trellis decoder to resolve the mutual interference of the I- and Q-components after differential preprocessing, is discussed and an algorithm for raised cosine filtered signals with a roll-off factor $r = 1.0$ is given. An advanced algorithm suitable for general roll-off factors and with reduced complexity is then derived. Both algorithms are investigated in linear and nonlinear time selective mobile radio channels with the aid of computer simulation. In Chapter 5 algorithms for coherent demodulation and equalisation, which are based on per-survivor processing (PSP) are synthesised and discussed. Per-survivor processing provides a general framework for the approximation of maximum likelihood sequence estimation (MLSE) algorithms whenever the presence of unknown quantities prevents the precise use of the classical Viterbi algorithm. PSP combines a group of algorithms which implement the data aided estimation of unknown parameters within the Viterbi algorithm itself. In the investigations, the least mean square (LMS) algorithm, a hybrid algorithm, and a Kalman estimator were considered and compared. The Kalman estimator has been further extended for the use in frequency selective channels. At the end of the chapter, simulation results on

24

different time and frequency selective, linear and nonlinear mobile channels are given.

The implementation of the algorithms as simulation models and an introduction in the applied simulation environment COSSAP are presented in Chapter 6. The chapter also describes the fixed-point optimisation of the differential demodulation algorithm and its conversion into hardware description language (HDL). At the end of the chapter the possibilities to realise a hardware implementation is discussed.

To make the thesis largely self contained, Chapter 2 gives an overview of the basics of digital modulation, receiver techniques, and the characteristics of mobile radio channels. In Chapter 3 the principle of differential encoded Offset-QPSK is presented and a generic nonlinear amplifier model is introduced. The model is used for all investigations on nonlinear channels.

Chapter 2

Basics

2.1 Digital Modulation

Modern wireless communication systems use digital modulation techniques. Through the rapid progress in IC design and in digital signal processing (DSP) technology digital modulation becomes more cost effective than analog transmission systems. Therefore, digital systems are being used more and more to replace analog systems. In addition to the cost factor, digital modulation offers many advantages over analog modulation. These advantages include greater noise immunity and robustness to channel impairments, easier multiplexing of various forms of information (e.g., voice, data and video), and greater security.

In digital wireless communication systems, the modulating signal (e.g., the information) may be represented as a time sequence of symbols or pulses, where each symbol has M finite states. Each symbol represents k bits of information, where $M = 2^k$. There are many different digital modulation schemes that are now used in modern wireless communication systems, and many more are sure to be introduced. Some of these techniques have only small differences between one another, and each technique belongs to a family of related modulation methods. The most applied classification is to divide the techniques into linear and nonlinear methods.

2.1.1 Performance Criteria of a Modulation Format

Several factors influence the choice of a particular digital modulation format. A desirable modulation format provides low bit error rate (BER) at low received signal-to-noise ratios (SNR), performs well in multipath conditions, occupies a minimum of bandwidth, and is easy and cost effective to implement. Existing modulation formats do not simultaneously satisfy all these requirements. Some modulation formats are better in terms

27

of the bit error rate performance, while others are better in terms of bandwidth occupancy. Depending on the demands of the particular application, trade-offs are made when selecting a digital modulation format.

Power efficiency and bandwidth efficiency are two significant indicators of the performance of a modulation format. Power efficiency describes the ability of a modulation techniques to support detection of the data of a digital message at low power levels. The detection of a digital message on a given signal power depends on the noise immunity of the certain modulation format and on the demodulation technique. The power efficiency of a digital modulation format is often expressed as the ratio of the signal energy per bit to noise power spectral density (E_b/N_0) required at the receiver input for a certain error probability (e.g., 10^{-3}).

Bandwidth efficiency describes the ability of a modulation format to transmit a specific data rate within a limited bandwidth. In general, increasing the symbol rate implies decreasing the pulse width of a digital symbol, which increases the bandwidth of the signal. Thus, there is an unavoidable trade-off between symbol rate and bandwidth occupancy. However, some modulation formats perform better than others in dealing with this trade-off. Bandwidth efficiency reflects how efficiently the allocated bandwidth is utilised and is defined as the ratio of the throughput data rate to a given bandwidth where k bits are mapped on a symbol. If R is the data rate in bits per second, and B is the bandwidth occupied by the modulated signal, then bandwidth efficiency is expressed as

$$\frac{R}{B} \; bps/Hz \tag{2.1}$$

There is a fundamental upper bound on the achievable bandwidth efficiency for Gaussian signals and channels. Shannon's channel coding theorem states that for an arbitrarily small probability of error, the maximum possible bandwidth efficiency is limited by the noise in the channel, and is given by the famous channel capacity formula

$$\frac{C}{B} = log_2(1 + \frac{S}{N}) \tag{2.2}$$

where C is the channel capacity in bits per second, B is the bandwidth and S/N is the signal-to-noise ratio.

While power and bandwidth efficiency considerations are very important, other factors also affect the choice of a particular digital modulation format. For example,

for all personal communication systems which serve a large user community, the cost and complexity of the receiver must be minimised and a low energy consumption of the handheld devices is demanded. Therefore, the sensitivity against interferences resulting from nonlinear power amplifiers is an important key. The influence of the various types of mobile radio channels on the system performance is a factor just like the implemented signal processing algorithms in the receiver. In general, the modulation technique, the influence of nonlinear amplifiers, the interference and time varying effects of the channel, and the performance of the specific demodulator are analysed as a complete system by computer simulation in order to determine relative performance.

2.1.2 Linear Modulation Formats

In linear modulation techniques the transmitted signal $s(t)$ can vary in terms of amplitude, phase, frequency or combinations of these parameters as a function of the modulating digital signal $z(t)$. The transmitted signal can be expressed as

$$s(t) = \Re \left\{ A\, z(t) \cdot \mathrm{e}^{(\mathrm{j}2\pi f_c t)} \right\} \tag{2.3}$$

where A is the amplitude, f_c is the carrier frequency, and $z(t) = x(t) + \mathrm{j}y(t)$ is a complex envelope representation of the modulated signal which in general is complex. For wireless communication systems the most popular linear modulation techniques include M-PSK, DPSK, OQPSK, $\pi/4$-QPSK, and QAM, which are described below. The main attraction for these modulation formats is their reduced implementation complexity and "good" error rate performance (see Section 2.4).

Phase Shift Keying (PSK)

Digital phase modulated signals are implemented by assigning the phase φ_m of the carrier signal to one transmitting symbol. The amplitude of the carrier signal is not modulated.

$$\varphi_m = \frac{2\pi(m-1)}{M} \quad \text{with} \quad 1 \le m \le M \tag{2.4}$$

The block diagram of the M-PSK modulator for $M \ge 4$ is given in Fig. 2.1.

A number of k bits of the binary data $a(n \cdot T_b) \equiv a(n)$ are mapped to one of $M = 2^k$ possible symbols. At time instant $i \cdot T$ a symbol is mapped to two impulses

29
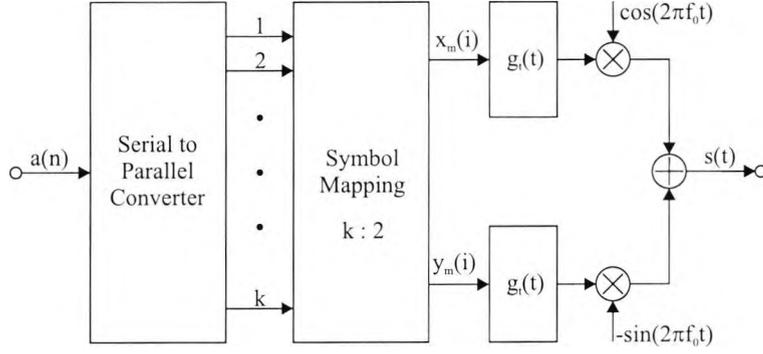
Figure 2.1: Block diagram of a M-PSK transmitter

with amplitudes $x_m(i)$ and $y_m(i)$ and with a pulse shape $g_t(t)$. The impulses are then multiplied with the two orthogonal carrier components. $g_t(t)$ can be interpreted as a normalised impulse response of a transmitter pulse shaping filter, i.e. an impulse response without dimension. Consequently, the amplitudes can also be defined as δ-impulses with weighting factors $x_m(t)$ and $y_m(t)$. The amplitudes $x_m(t)$ and $y_m(t)$ are chosen such that only the phase of the carrier signal is switched and the amplitude remains constant

$$z = |z_m| = \sqrt{x_m^2 + y_m^2} = \text{constant} \tag{2.5}$$

In spite of the expression, phase modulation which implies nonlinear processing, the format is linear, since only a limited number of phase values is selected by the data sequence[1]. The PSK signal can be described as

$$
\begin{aligned}
s(t) &= \Re \left\{ \left[\sum_{i=-\infty}^{+\infty} z \cdot e^{j \frac{2\pi(m(i)-1)}{M}} \cdot g_t(t - iT) \right] \cdot e^{j2\pi f_c t} \right\} \\
&= \Re \left\{ \bar{s}(t) \cdot e^{j2\pi f_c t} \right\}
\end{aligned}
\tag{2.6}
$$

where $\bar{s}(t)$ is the complex envelope of the modulated signal including all information of the signal with the exception of the carrier frequency f_c. Using trigonometric identities

---

[1]Linearity can be proved by formulating a PSK signal as an addition of two independent pulse amplitude modulated signals, which itself are linear

the above equation can be rewritten as

$$
\begin{aligned}
s(t) = \quad & \cos(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} z \cdot \cos(\frac{2\pi(m(i)-1)}{M}) \cdot g_t(t-iT) \\
& -\sin(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} z \cdot \sin(\frac{2\pi(m(i)-1)}{M}) \cdot g_t(t-iT)
\end{aligned}
\tag{2.7}
$$

$$
\begin{aligned}
s(t) = \quad & \cos(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} x_m(i) \cdot g_t(t-iT) \\
& -\sin(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} y_m(i) \cdot g_t(t-iT)
\end{aligned}
\tag{2.8}
$$

Furthermore, the signal may be represented as a linear combination of two orthonormal signal waveforms, $f_1(t)$ and $f_2(t)$, i.e.,

$$
\begin{aligned}
s(t) = \quad & \sqrt{\frac{E_g}{2}} \cdot \sum_{i=-\infty}^{+\infty} x_m(i) \cdot f_1(t-iT) \\
& +\sqrt{\frac{E_g}{2}} \cdot \sum_{i=-\infty}^{+\infty} y_m(i) \cdot f_2(t-iT)
\end{aligned}
\tag{2.9}
$$

where

$$
\begin{aligned}
f_1(t) &= \sqrt{\frac{2}{E_g}}\, g_t(t) \cdot \cos(2\pi f_c t) \\
f_2(t) &= -\sqrt{\frac{2}{E_g}}\, g_t(t) \cdot \sin(2\pi f_c t)
\end{aligned}
\tag{2.10}
$$

The orthonormal functions $f(t)$ will be used to derive the optimal receiver structure in the next section. E_g denotes the energy of the pulse $g_t(t)$.

One common used PSK format is Binary Phase Shift Keying (BPSK, 2-PSK). BPSK is a special case, since only one component is used. In the literature BPSK is also known as 2-PAM (Pulse Amplitude Modulation).

PSK signals can easily be investigated with the aid of a two dimensional signal space diagram. In this diagram the M possible combinations of the real amplitudes x_m and y_m of the orthogonal carrier components from (2.8) are shown as signal points of a complex amplitude. Fig. 2.2 gives the signal space diagrams of BPSK, QPSK, and 8-PSK.

Another common method to display PSK signals is the complex phase plot. Unlike the signal space diagram, in the phase plot the transitions between the signal points at

31

(a)   (b)   (c)
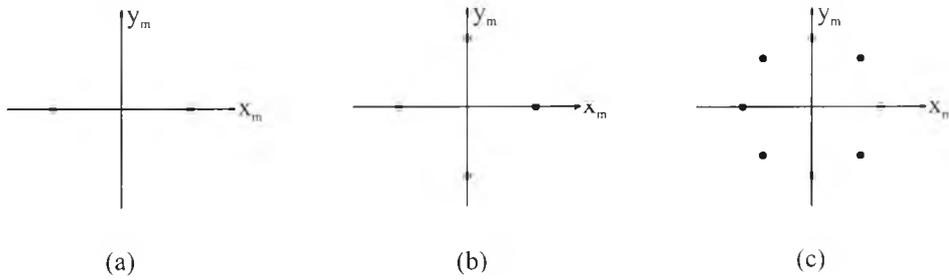
Figure 2.2: Signal space diagram of PSK signals (a) BPSK (b) QPSK (c) 8-PSK

the output of the pulse shaping filters are used. In this diagram the real and imaginary components of the complex envelope $\bar{s}(t)$ (2.11) at the output of the pulse shaping filters are drawn in mutual dependance in a system of rectangular coordinates.

$$
\begin{aligned}
s(t) &= \Re\left\{\bar{s}(t) \cdot e^{j2\pi f_c t}\right\} \\
&= \cos(2\pi f_0 t) \cdot \Re\left\{\bar{s}(t)\right\} - \sin(2\pi f_c t) \cdot \Im\left\{\bar{s}(t)\right\}
\end{aligned}
\tag{2.11}
$$

In Fig. 2.3 phase plots are given, obtained from a computer simulation. They show two filtered QPSK signals. One signal is obtained after filtering by a root raised cosine filter and the other by a raised cosine filter, the latter forms a Nyquist-1 filter with a roll-off factor $r = 0.35$. Though in PSK signals no information is carried in the envelope
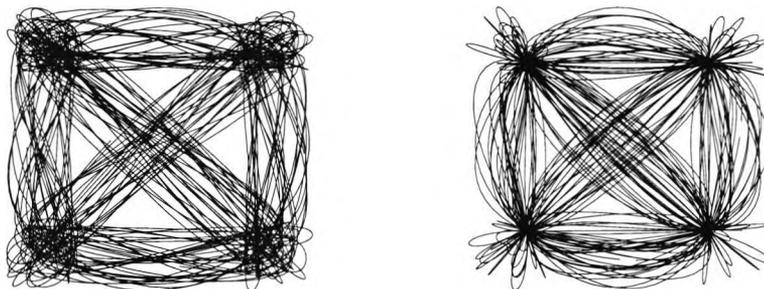
Figure 2.3: Complex phase plots of QPSK signals

(amplitude) the diagrams show that the envelope of the signals is not constant any more, since the signal has been filtered. This aspect, caused by the band limitation, is important in the development of wireless systems and is discussed in the following.

One feature of the PSK signal is that all amplitudes z_m have the same value, i.e., all waveforms have the same energy

$$
E_s = \int_{-\infty}^{\infty} s_m^2(t)dt = \frac{z^2}{2} \cdot E_g = \left[\frac{x_m^2}{2} + \frac{y_m^2}{2}\right] \cdot E_g
\tag{2.12}
$$

where $s_m(t)$ is the modulated signal of a single symbol. In order to calculate the

Euclidian distance, a two dimensional vector for each signal point is defined

$$\tilde{\mathbf{z}}_{\mathbf{m}} = [\tilde{x}_m, \tilde{y}_m] = \left[x_m \cdot \sqrt{\frac{E_g}{2}}, \quad y_m \cdot \sqrt{\frac{E_g}{2}} \right] \tag{2.13}$$

The Euclidean distance between two signal points $\tilde{\mathbf{z}}_{\mathbf{m}}$ and $\tilde{\mathbf{z}}_{\mathbf{n}}$ then is

$$D(\tilde{\mathbf{z}}_{\mathbf{m}}, \tilde{\mathbf{z}}_{\mathbf{n}}) = \mid \tilde{\mathbf{z}}_{\mathbf{m}} - \tilde{\mathbf{z}}_{\mathbf{n}} \mid = z \cdot \sqrt{2E_g} \cdot \left| \sin\left(\frac{\pi(m-n)}{M} \right) \right| \tag{2.14}$$

The minimum Euclidean distance between two adjacent signal phases is obtained when $\mid m - n \mid = 1$. In this case

$$D^{\min}(\tilde{\mathbf{z}}_{\mathbf{m}}, \tilde{\mathbf{z}}_{\mathbf{n}}) = z \cdot \sqrt{2E_g} \cdot \sin\left(\frac{\pi}{M} \right) \tag{2.15}$$

Differential Phase Shift Keying (DPSK)

A PSK signal contains the transmitted information in the discrete phase deviation compared to the unmodulated carrier. For an errorfree demodulation the absolute carrier phase at the receiver has to be known. Since, in many practical systems the carrier itself is not transmitted separately, a carrier derivation from the received modulated signal is mandatory. However, a phase ambiguity of $2\pi/M$ follows for M-ary PSK signals as a consequence of the nonlinear phase recovery in the receiver. To avoid this ambiguity differential PSK (DPSK) techniques are often employed.

Differential demodulation of PSK signals is a non-coherent technique which avoids the need for a coherent reference signal at the receiver. In differential PSK modulation the transmitted data information is contained in the difference in time consecutive absolute phase values

$$\varphi(i) = \varphi(i-1) + \Delta\varphi_m(i)$$

$$\text{with} \tag{2.16}$$

$$\Delta\varphi_m = \frac{2\pi(m-1)}{M} ; \quad 1 \le m \le M$$

While DPSK signaling has the advantage of reduced receiver complexity, it is important to know that the power efficiency of DPSK is lower than the power efficiency of PSK (see Section 2.4).

Offset-QPSK (OQPSK)

In the ideal unfiltered case the amplitude of a QPSK signal is constant. However, when QPSK signals are pulse shaped, they lose the constant envelope property. An

occasional phase shift of π results in a signal envelope, which is zero for the phase transition interval. A nonlinear amplification of the QPSK signal spreads the transmitted bandwidth and brings back the sidelobes. To prevent the generation of sidelobes, the linear modulation format QPSK should only be amplified with linear amplifiers, which, however, are less power efficient. A simple method to reduce the influences of nonlinear amplifiers is obtained through modified QPSK signals, which results in Offset-QPSK (OQPSK) or staggered QPSK.

OQPSK signaling is similar to QPSK signaling, except for the time alignment of the even and odd bit stream in the transmitter (Fig. 2.1). In a QPSK transmitter the bit transition of both bit streams occur at the same time instant, but in OQPSK signaling, the even and odd bit streams, $x(t)$ and $y(t)$, are offset in their relative alignment by half a symbol period. The transmitted signal is represented as

$$s(t) = \Re \left\{ \left[\sum_{i=-\infty}^{+\infty} x_m(i) \cdot g_t(t - iT) \right. \right.$$
$$\left. \left. + \mathrm{j}\, y_m(i) \cdot g_t(t - \frac{T}{2} - iT) \right] \cdot \mathrm{e}^{\mathrm{j}2\pi f_c t} \right\} \qquad (2.17)$$

The signal space diagram of OQPSK does not differ from the diagram of QPSK, which is shown in Fig. 2.2. The Euclidean distances are identical. It should be borne in mind, that as a result of the time shift of one bit stream, two transitions instead of one for QPSK occur during a symbol period. The transition $00 \rightarrow 11$, for example, is replaced by the two transitions $00 \rightarrow 10 \rightarrow 11$. Since the two components do not change simultaneously, a phase transition of π or $180°$ is avoided and a filtering of OQPSK signals does not force the signal envelope to zero (Fig. 2.4). Additionally, the envelope
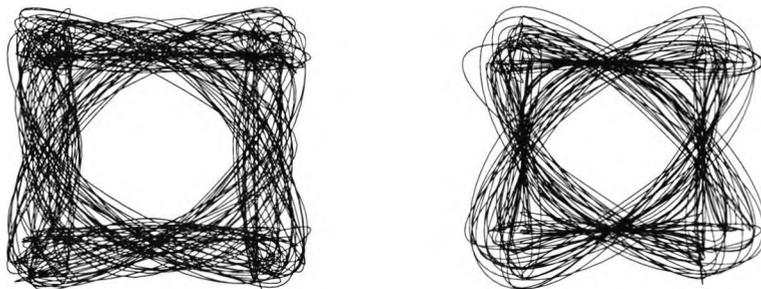
Figure 2.4: Complex phase plot of OQPSK signals (a) root raised cosine, (b) raised cosine with $r = 0.35$

has a reduced dynamic range compared to QPSK. Only in this reduced amplitude range the high demands on the linearity of the amplifier remain unchanged. Without amplification the spectrum of an OQPSK signal is identical to that of a QPSK signal

and both signals occupy the same bandwidth. The staggered alignment of the even and odd bit streams does not change the nature of the spectrum. OQPSK retains its bandlimited nature even after nonlinear amplification. Therefore it is very attractive for communication systems where bandwidth efficiency on the one side and power efficiency through the application of nonlinear amplifiers on the other side are important. The reason why OQPSK is not implemented in mobile radio systems is the nonavailability of non-coherent receivers, which meet the demands of such a system.

$\pi/4$-QPSK

$\pi/4$-QPSK is a quadrature phase shift keying technique which offers a compromise between OQPSK and QPSK in terms of the allowed maximum phase transition. It may be demodulated coherently or non-coherently. In $\pi/4$-QPSK systems, the maximum phase change is limited to $3/4\ \pi$ or $135°$, compared to π for QPSK and $\pi/2$ for OQPSK. The signal space diagram and the complex phase plot of $\pi/4$-QPSK is shown in Fig. 2.5. In $\pi/4$-QPSK, the signal points of the modulated signal are selected alternately from two QPSK constellations which are shifted by $\pi/4$ with respect to each other. Therefore a relative phase shift of $\pm\pi/4$ or $\pm3\pi/4$ can occur between two successive symbols. Switching between two signal point constellations every other symbol ensures that there is always a phase transition which enables a receiver to perform timing recovery and synchronisation. $\pi/4$-QPSK is extremely attractive due to the fact that it can be
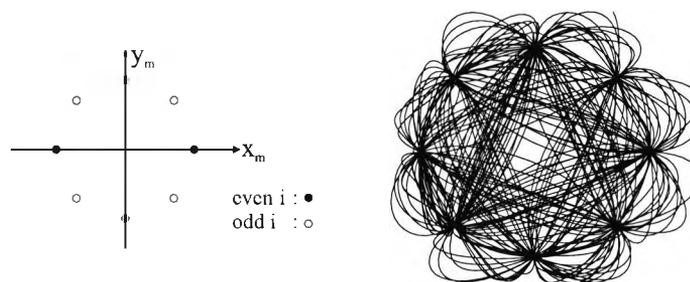
Figure 2.5: Signal space diagram and complex phase plot of a $\pi/4$-QPSK signal (raised cosine filtering with $r = 0.35$)

detected non-coherently, which clearly simplifies receiver design. When differentially encoded and demodulated $\pi/4$-QPSK is called $\pi/4$-DQPSK.

Without using nonlinear amplifiers the power and bandwidth efficiency of $\pi/4$-QPSK is identical to QPSK.

Quadrature Amplitude Modulation (QAM)

PSK techniques use two orthogonal components for modulation. All signal points (complex amplitudes) in the signal space diagram are located on a circle. If the number M of the signal points is increased, the minimum Euclidean distance is reduced, which decreases the probability of a correct data detection in the receiver. In a Quadrature Amplitude Modulation (QAM) signal the orthogonal carrier components are modulated independently, i.e., the signal points can be placed in the signal space, such that the minimum Euclidean distance is increased in comparison to PSK signals without changing the mean of the signal energy.

The QAM signal may be expressed as

$$
\begin{aligned}
s(t) = \quad & \Re\left\{\left[\sum_{i=-\infty}^{+\infty}(x_m(i) + \mathrm{j}\, y_m(i)) \cdot g_t(t - iT)\right] \cdot \mathrm{e}^{\mathrm{j}2\pi f_c t}\right\} \\
= \quad & \cos(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} x_m(i) \cdot g_t(t - iT) \\
& -\sin(2\pi f_c t) \cdot \sum_{i=-\infty}^{+\infty} y_m(i) \cdot g_t(t - iT)
\end{aligned}
\tag{2.18}
$$

In contrast to (2.6 - 2.8), the amplitudes x_m and y_m are independent

$$
|z_m| = |x_m + \mathrm{j}\, y_m| \neq \text{constant}
\tag{2.19}
$$

The principle of a QAM modulator is identical to the block diagram of a PSK modulator in Fig. 2.1. Three different signal space constellations of 16-QAM signals are given in Fig. 2.6. The square constellation has the largest minimum Euclidean distance when the mean of the signal energy is fixed. Therefore the square constellation is preferable for use on AWGN channels.
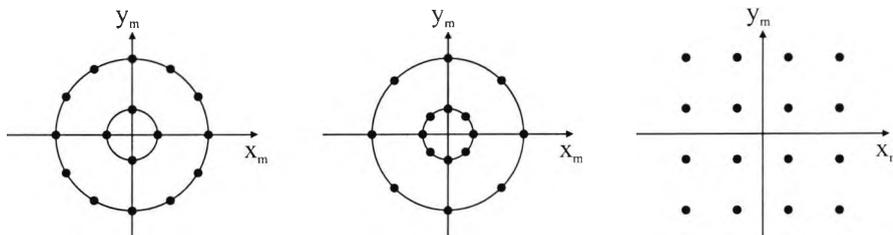
Figure 2.6: Signal space diagram of 16-QAM signals

By application of (2.13) the Euclidean distance between two QAM signal points

results in

$$D(\tilde{\mathbf{z}}_\mathbf{m}, \tilde{\mathbf{z}}_\mathbf{n}) = |\; \tilde{\mathbf{z}}_\mathbf{m} - \tilde{\mathbf{z}}_\mathbf{n} \;| = \sqrt{\frac{E_g}{2}} \cdot \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \qquad (2.20)$$

The specific Euclidean distances depend on the chosen signal space constellation. For $M \geq 8$ signal points M-QAM is more power efficient than M-PSK, i.e., the minimum Euclidean distance of an optimised M-QAM signal is larger than the distance of a M-PSK signal.

2.1.3 Nonlinear Modulation Formats

The amplitude of a nonlinear modulated signal can be constant, regardless of the variations in the modulating signal. Therefore, nonlinear modulated signals are ideal for transmitting over nonlinear quadripole networks like nonlinear amplifiers. An important class of modulation signals with constant envelopes are signals with continuous phase function. Such signals are generated by a Continuous Phase Modulation (CPM).

Continuous Phase Modulation (CPM)

In Continuous Phase Modulation (CPM), the frequency of a constant amplitude carrier signal is modulated by a data sequence. The frequency is changed in such a way that the phase function is continuous at all time instants. The principle of a CPM modulator is given by the block diagram in Fig. 2.7. The binary data $a(n)$ are mapped in groups of k bits to the 2^k possible symbols (amplitudes) z_m. Every symbol generates an impulse with amplitude z_m and with pulse shape $g_t(t)$. The impulses are then multiplied with
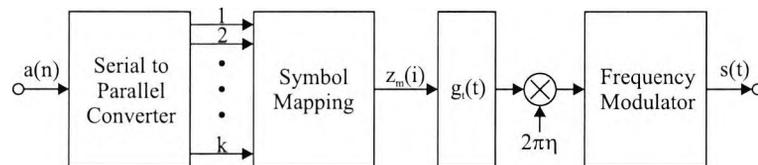
Figure 2.7: Block diagram of a CPM modulator

a constant, which is called the modulation index η. The resulting signal is input to a frequency modulator which maps a variation of the amplitude to a proportional variation of the carrier frequency. CPM signals are described analytically as

$$s(t) = \sqrt{\frac{2E_s}{T}} \cdot \cos(2\pi f_c t + \Phi(t, \mathbf{z})) \qquad (2.21)$$

37

$\Phi(t, \mathbf{z})$ represents the time varying phase of the carrier, which is defined as

$$\Phi(t, \mathbf{z}) = 2\pi\eta \cdot \sum_{i=-\infty}^{\infty} z_m(i) \cdot q(t - iT) \qquad (2.22)$$

where $\mathbf{z}$ is the sequence of M-ary information symbols selected from the alphabet $\pm 1, \pm 3, \ldots, \pm(M-1)$, η is the modulation index, and $q(t)$ is the phase waveform shape. The process of the continuous phase function is determined through the waveform $q(t)$, which result from the frequency impulse $g_t(t)$

$$q(t) = \int_0^t g_t(\tau)d\tau \qquad (2.23)$$

If $g_t(t) = 0$ for $t > T$, the CPM signal is called full response CPM. If $g_t(t) \neq 0$ for $t > T$, the modulated signal is called partial response CPM. Fig. 2.8 illustrates several pulse shapes for $g_t(t)$ and the corresponding phase function $q(t)$ where the pulses $g_t(t)$ have been normalised such that the area is equal to $\frac{1}{2}$. It is apparent that an infinite variety of CPM signals can be generated by choosing different pulse shapes $g_t(t)$ and by varying the modulation index η and the alphabet size M. A broader frequency impulse $g_t(t)$
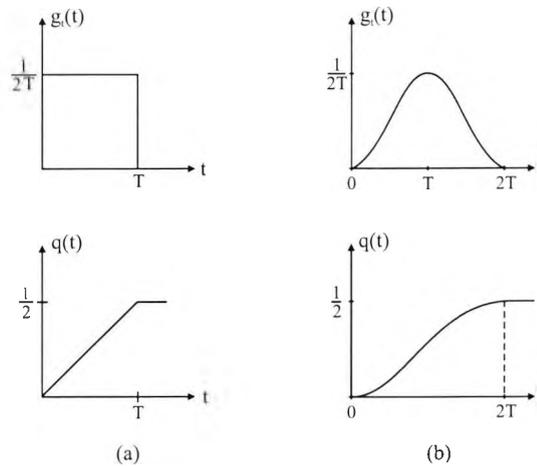
Figure 2.8: Pulse shapes for (a) full response CPM and (b) partial response CPM signals

leads to smoother phase trajectories and a more compact spectrum, but the complexity of the receiver increases. If the number of bits per symbol is increased, the transferable bit rate grows at the cost of more required bandwidth. A larger modulation index η normally improves the power efficiency, again resulting in an increase of the bandwidth requirement.

In contrast to a linear modulation signal, a CPM signal can only be defined as a power signal, that can not be described by the energy of a single pulse. Therefore,

the amplitude of the CPM signal in (2.21) is described by the energy E_s per symbol interval.

A CPM signal with a 1REC frequency impulse (Fig. 2.8a) is called CPFSK (Continuous Phase Frequency Shift Keying). If in addition binary symbols ($M = 2$) and a modulation index $\eta = 0.5$ are used, the modulation format is known as MSK (Minimum Shift Keying).

Minimum Shift Keying (MSK)

MSK is a special form of binary CPFSK (and therefore CPM) where the peak frequency deviation is equal to half the bit rate. In other words, MSK is a continuous phase FSK with a modulation index of 0.5. A modulation index of 0.5 corresponds to the minimum frequency spacing that is necessary to ensure the orthogonality of two FSK signals over a signaling interval of length T. Two FSK signals $s_H(t)$ (binary 1) and $s_L(t)$ (binary 0) are said to be orthogonal if

$$\int_0^T s_H(t)s_L(t)dt = 0 \tag{2.24}$$

A special feature of MSK is, that unlike the other CPM formats it can also be generated with an OQPSK modulator and in this way, it may be interpreted as a linear format. The equivalent lowpass formulation of the digitally modulated signal is

$$\bar{s}(t) = \sum_{i=-\infty}^{\infty} [z(2i)g_t(t - 2iT) - \mathrm{j}\, z(2i + 1)g_t(t - 2iT - T)] \tag{2.25}$$

where $g_t(t)$ is a sinusoidal pulse defined as

$$g_t(t) = \begin{cases} \sin \frac{\pi t}{2T} & : \quad 0 \le t \le 2T \\ 0 & : \quad \text{otherwise} \end{cases} \tag{2.26}$$

Equation (2.25) corresponds to the representation of an OQPSK signal where the amplitudes in the quadrature channels are not mutually independent.

In the GSM system (Global System for Mobile Communications) a modification of MSK is used leading to GMSK. In GMSK (Gaussian Minimum Shift Keying) the sidelobes of the spectrum are reduced by passing the modulating data waveform through a pre-modulation Gaussian pulse shaping filter with 3 dB bandwidth. Filtering smoothes the phase trajectories of the MSK signal and hence stabilises the instantaneous frequency variations. This has the effect of considerably reducing the sidelobes in the

39

signal spectrum. Figure 2.9 shows the computed power density spectrum of a GMSK signal for various values of the product (BT) of 3dB-bandwidth times bit duration. In the GSM system a BT product of 0.3 is used.
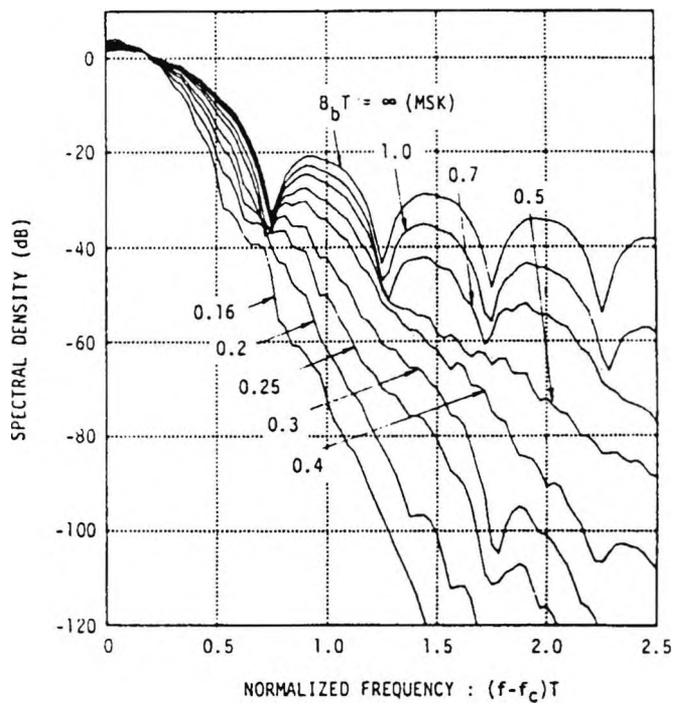
Figure 2.9: Power spectral density of a GMSK signal (from [34])

2.2 Optimum Receivers

In Section 2.1, various types of linear and nonlinear modulation formats were described that may be used to transmit digital information through a communication channel. This section deals with the design and performance characteristic of optimum receivers. In 2.2.1 the design for the AWGN channel is described whereas in 2.2.2 channels with intersymbol interference (ISI) and AWGN are treated.

2.2.1 Optimum Receiver for the AWGN Channel

In an AWGN channel the signal is corrupted by the addition of white Gaussian noise, as illustrated in Fig. 2.10. Thus, the received signal may be expressed as

$$v(t) = s_m(t) + n(t), \quad 0 \le t \le T \tag{2.27}$$

where $n(t)$ denotes a sample function of the additive white gaussian noise (AWGN) process and s_m is one of the M possible waveforms. Based on the observation of $v(t)$ over the signal interval, a receiver will be designed which is optimum in the sense that it minimises the error probability. It is assumed that no influences of preceding and following data pulses are effective in the observation period. A common procedure for
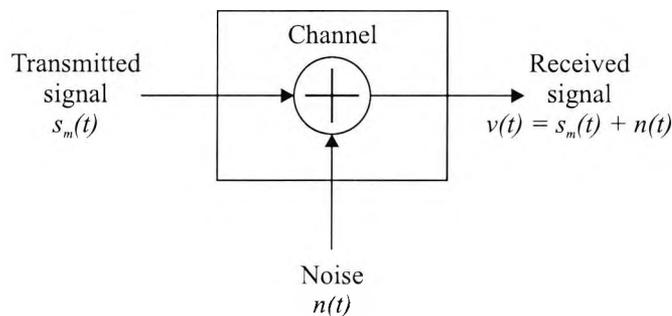
Figure 2.10: Model of an AWGN channel

the investigation of a receiver is the division of the receiver into two parts: the signal demodulator and the detector [26]. The task of the signal demodulator is to convert the received waveform $v(t)$ into an N-dimensional vector $\mathbf{v} = [v_1 v_2 \dots v_N]$, where N is the dimension of the transmitted signal waveforms (M-PSK, QAM: $N = 2$). The detector then has to decide which of the M possible signal waveforms s_m was transmitted based on the observation vector $\mathbf{v}$.

41

In the literature two realisations of the signal demodulator are described. One is based on the use of signal correlators and the second is based on the use of matched filters. Since both realisations are equivalent, in this thesis the description will be limited to the matched-filter demodulator.

Matched Filter Demodulator

Each of the transmitted signals of the set $s_m(t)$, $1 \leq m \leq M$ can be represented as a weighted linear combination of the N basis functions $f_k(t)$. For M-PSK and QAM ($N = 2$) $s_m(t)$ can be described as (compare (2.9))

$$
\begin{aligned}
s_m(t) = \quad & \sqrt{\frac{E_g}{2}}\, x_m(i) \cdot f_1(t) \\
+ & \sqrt{\frac{E_g}{2}}\, y_m(i) \cdot f_2(t)
\end{aligned}
\tag{2.28}
$$

The matched filter uses a bank of N linear filters, whose impulse responses are

$$
g_{r,k}(t) = f_k(T - t), \quad 0 \leq t \leq T
\tag{2.29}
$$

where $f_k(t)$ are the N basis functions and $g_{r,k} = 0$ outside the interval $0 \leq t \leq T$. The output signals of these filters are

$$
\begin{aligned}
r_k(t) &= \int_0^t v(\tau) g_{r,k}(t - \tau)\, d\tau \\
&= \int_0^t v(\tau) f_k(T - t + \tau)\, d\tau, \quad k = 1, 2, \ldots, N
\end{aligned}
\tag{2.30}
$$

If the output signals of the filters are sampled at $t = T$, then

$$
r_k(T) = \int_0^T v(\tau) f_k(\tau)\, d\tau = v_k, \quad k = 1, 2, \ldots, N
\tag{2.31}
$$

The sampled output signals of the filters at time $t = T$ are defined as the values v_k of the N-dimensional vector $\mathbf{v} = [v_1 v_2 \ldots v_N]$.

For the demodulator described above, the N filters are matched to the basis functions $f_k(t)$. In general, a filter is called a matched filter to the signal $s(t)$, when its impulse response is

$$
g_r(t) = s(T - t)
\tag{2.32}
$$

where $s(t)$ is the transmitted signal that is generated from the linear superposition of the basis functions $f_k(t)$. It is assumed that $s(t)$ is confined to the time interval

$0 \leq t \leq T$. A matched filter maximises the output signal-to-noise ratio (SNR) of a signal that is corrupted by AWGN.

To prove this property, it is assumed, that the received signal $v(t)$ is obtained from the transmitted signal $s(t)$ and the additive noise $n(t)$. The noise has zero mean and two-sided power spectral density $S_{nn}(f) = \frac{1}{2}N_0$ W/Hz. The signal $v(t)$ is passed through a filter with impulse response $g_r(t)$, $0 \leq t \leq T$, and its output is sampled at time $t = T$. At the sampling instant $t = T$, the filter response to the received signal $v(t)$ is

$$\begin{aligned} r(T) &= \int_0^T v(\tau) g_r(T-\tau)\, d\tau \\ &= \int_0^T s(\tau) g_r(T-\tau)\, d\tau + \int_0^T n(\tau) g_r(T-\tau)\, d\tau \\ &= r_s(T) + r_n(T) \end{aligned} \tag{2.33}$$

where $r_s(T)$ represents the signal component and $r_n(T)$ the noise component. In [26] it is shown that the maximum output signal-to-noise ratio (SNR$_0$), defined as

$$\mathrm{SNR}_0 = \frac{r_s^2(T)}{E[r_n^2(T)]}, \tag{2.34}$$

is achieved, when $g_r(t)$ is matched to the signal $s(t)$. The output (maximum) SNR obtained with the matched filter then is

$$\mathrm{SNR}_0 = \frac{2}{N_0} \int_0^T s^2(t)\, dt = \frac{2E_s}{N_0} \tag{2.35}$$

For a transmission without intersymbol interference, as it was presupposed up to now, additionally the first Nyquist criterium has to be fulfilled. In practical systems pulse shaping filters with raised cosine roll-off are often used. To satisfy the matched filtering theorem, the transmitter and receiver filter then have to be square root raised cosine filters.

Optimum Detector

A matched filter demodulator (or correlation demodulator) produces a vector $\mathbf{v} = [v_1 v_2 \ldots v_N]$ (2.31) which contains all the relevant information in the signal waveform. An optimum detector has to decide which decision rule is optimum based on the observation vector $\mathbf{v}$. Therefore, the probability of sending the signal $\mathbf{s}_m = [s_{1m} s_{2m} \ldots s_{Nm}]$

conditioned on the reception of the current signal vector $\mathbf{v}$ has to be maximised. This rule may be expressed as

$$P(\mathbf{s}_m \mid \mathbf{v}) = \max, \quad m = 1, 2, \ldots, M \tag{2.36}$$

The above decision criterion is called the maximum a posteriori probability (MAP) criterion. Using Bayes' rule, the posteriori probabilities are given by

$$P(\mathbf{s}_m \mid \mathbf{v}) = \frac{p(\mathbf{v} \mid \mathbf{s}_m) P(\mathbf{s}_m)}{p(\mathbf{v})} \tag{2.37}$$

where $p(\mathbf{v} \mid \mathbf{s}_m)$ is the conditional pdf of the observed vector given $\mathbf{s}_m$, and $P(\mathbf{s}_m)$ is the a priori probability of the mth signal being transmitted. The denominator in (2.37) is independent of which signal is transmitted. When the M signals are equally probable a priori, i.e., $P(\mathbf{s}_m) = 1/M$ for all M, then the decision rule based on finding the signal that maximises $P(\mathbf{s}_m \mid \mathbf{v})$ is equivalent to finding the signal that maximises $p(\mathbf{v} \mid \mathbf{s}_m)$.

The decision rule based on the maximum of $p(\mathbf{v} \mid \mathbf{s}_m)$ over the M signals is called the maximum-likelihood (ML) criterion. It is observed that a detector based on the MAP criterion and one that is based on the ML criterion make the same decision as long as the a priori probabilities $P(\mathbf{s}_m)$ are all equal, i.e., the signals $\mathbf{s}_m$ are equiprobable.

In the case of an AWGN channel, the likelihood function is given by

$$p(\mathbf{v} \mid \mathbf{s}_m) = \frac{1}{(\pi N_0)^{N/2}} \cdot e^{\left[-\sum_{k=1}^{N} \frac{(v_k - s_{mk})^2}{N_0} \right]}, \quad m = 1, 2, \ldots, M \tag{2.38}$$

To simplify computations, the natural logarithm of $p(\mathbf{v} \mid \mathbf{s}_m)$ is used, since it is a monotonic function.

$$\ln p(\mathbf{v} \mid \mathbf{s}_m) = -\frac{1}{2} N \ln(\pi N_0) - \frac{1}{N_0} \sum_{k=1}^{N} (v_k - s_{mk})^2 \tag{2.39}$$

This simplification leads to the result, that the maximum of $\ln p(\mathbf{v} \mid \mathbf{s}_m)$ over $\mathbf{s}_m$ is equivalent to finding the signal $\mathbf{s}_m$ that minimises the Euclidean distance

$$D(\mathbf{v}, \mathbf{s_m}) = \sum_{k=1}^{N} (v_k - s_{mk})^2 \tag{2.40}$$

$D(\mathbf{v}, \mathbf{s_m})$, $m = 1, 2, \ldots, M$ is called the distance metric. Therefore, for the AWGN channel, the decision rule based on the ML criterion is reduced to finding the signal $\mathbf{s}_m$ that is closest in distance to the received signal vector $\mathbf{v}$. This decision rule is known as minimum distance detection.

In summary, the optimum ML detector computes a set of M distances $D(\mathbf{v}, \mathbf{s_m})$ and selects the signal corresponding to the smallest distance metric. However, this detection rule is only valid in the case where all signals are equally probable and the MAP criterion is equivalent to the ML criterion.

2.2.2 Receiver for Channels with ISI and AWGN

In the last section, an optimum receiver was derived for ideal channels without intersymbol interference (ISI). Non-ideal channels with memory, such as the mobile radio channels, introduce signal distortion. A signal distortion results in intersymbol interference, which, if left uncompensated, causes high bit error rates. The solution to the ISI problem is to design a receiver that includes a technique for compensating or reducing the ISI in the received signal. The compensator for the ISI is called an equaliser.

In the following, three equalisation methods are presented. First a receiver structure is derived, that is optimal under ISI conditions , i.e., it satisfies the ML principle. Method two and three are suboptimal channel equalisation approaches. One method is based on the use of a linear filter (linear equaliser) whereas the other method uses a feedback structure to suppress the ISI (nonlinear equaliser).

Optimum Maximum-Likelihood Receiver

In the derivation of the optimum receiver structure equally probable data symbols as well as additive white gaussian noise are assumed. The ML criterion, derived in the last section, can still be used if the entire transmitted sequence $\mathbf{z}$ is considered instead of the individual symbols s_m. The received (equivalent lowpass) signal is then expressed as

$$\bar{v}(t) = \sum_i z_i h(t - iT) + n(t) \tag{2.41}$$

where $h(t)$ represents the response of the channel to the input signal pulse $g_t(t)$ and $n(t)$ the additive white gaussian noise. The received signal can be expanded into the series

$$\bar{v}(t) = \lim_{N \to \infty} \sum_{k=1}^{N} v_k f_k(t) \tag{2.42}$$

where $f_k(t)$ is a complete set of orthonormal functions and v_k are the observable random variables obtained by projecting $\bar{v}(t)$ onto the set $f_k(t)$. It can easily be shown that

$$v_k = \sum_i z_i h_{ki} + n_k, \quad k = 1, 2, \ldots \tag{2.43}$$

where h_{ki} and n_k are the values obtained from projecting $h(t - iT)$ and $n(t)$ onto $f_k(t)$. The sequence n_k is Gaussian with zero mean and covariance

$$\frac{1}{2} E(n_k^* n_m) = N_0 \delta_{km} \tag{2.44}$$

For an equivalent lowpass signal, which is considered here, the covariance of $n(t)$ is N_0 instead of $\frac{1}{2} N_0$ used in the last section.

The joint probability density function of the random variables $\mathbf{v}_N \equiv [v_1 v_2 \ldots v_N]$ conditioned on the transmitted sequence $\mathbf{z}_p \equiv [z_1 z_2 \ldots z_p]$, where $p \leq N$, is

$$p(\mathbf{v}_N \mid \mathbf{z}_p) = \left(\frac{1}{2\pi N_0}\right)^{N/2} e^{\left(-\frac{1}{2N_0} \sum_{k=1}^{N} \left| v_k - \sum_i z_i h_{ki} \right|^2\right)} \tag{2.45}$$

In the limit as the number N approaches infinity, the natural logarithm of $p(\mathbf{v}_N \mid \mathbf{z}_p)$ is proportional to the metrics $J'(\mathbf{z}_p)$, defined as

$$
\begin{aligned}
J'(\mathbf{z}_p) &= -\int_{-\infty}^{\infty} \left| \bar{v}(t) - \sum_i z_i h(t - iT) \right|^2 dt \\
&= -\int_{-\infty}^{\infty} |\bar{v}(t)|^2 dt + 2\Re \sum_i \left[z_i^* \int_{-\infty}^{\infty} \bar{v}(t) h^*(t - iT) dt \right] \\
&\quad - \sum_i \sum_m z_i^* z_m \int_{-\infty}^{\infty} h^*(t - iT) h(t - mT) dt
\end{aligned}
\tag{2.46}
$$

Since the integral of $|\bar{v}(t)|^2$ is common to all metrics, it may be discarded. Therefore, it is sufficient to use the simplified Metric J for the MLSE process

$$J(\mathbf{z}_p) = 2\Re \left(\sum_i z_i^* r_i \right) - \sum_i \sum_m z_i^* z_m u_{i-m} \tag{2.47}$$

r_i is the output signal of the matched filter

$$r_i \equiv r(iT) = \int_{-\infty}^{\infty} \bar{v}(t) h^*(t - iT) dt \tag{2.48}$$

and u_i represent the samples of the autocorrelation function of $h(t)$, taken periodically at $1/T$

$$u_i \equiv u(iT) = \int_{-\infty}^{\infty} h^*(t) h(t + iT) dt \tag{2.49}$$

46

r_i and u_i can now be used to substitute for $\bar{v}(t)$ in (2.48) using (2.41)

$$r_k = \sum_i z_i u_{k-i} + n'_k \tag{2.50}$$

with

$$n'_k = \int_{-\infty}^{\infty} n(t) h^*(t - kT) dt \tag{2.51}$$

where n'_k denotes the samples of the additive noise sequence at the output of the matched filter.

If ISI affects only a finite number of symbols, $u_i = 0$ for $|i| > L$, MLSE can be viewed as a problem of estimating the state of a discrete time finite state machine. This implies that the channel output with ISI may be represented by a trellis diagram with M^L states (M is the size of the symbol alphabet), and the maximum-likelihood estimate of the information sequence $(z_1, z_2, \ldots, z_p)$ is simply the most probable path through the trellis given the received demodulator output sequence $\mathbf{r}$. The Viterbi algorithm provides an efficient means for performing the trellis search [12]. Fig. 2.11 illustrates the block diagram of the optimum receiver for an AWGN channel with ISI.
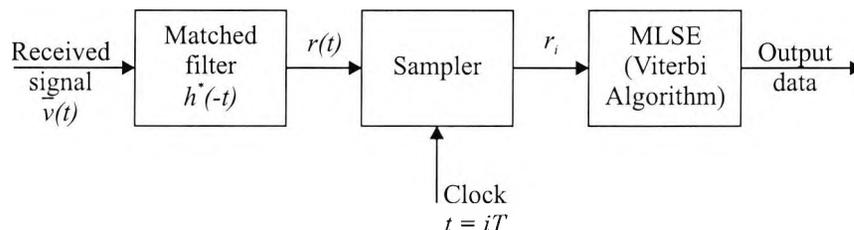
Figure 2.11: Optimum receiver for an AWGN channel with ISI

The noise sequence $\mathbf{n}' \equiv [n'_1 n'_2 \ldots n'_N]$ defined by the samples n'_k in (2.51) is no more uncorrelated and white. Though the set of noise variables n'_k is still a Gaussian distributed sequence with zero mean and autocorrelation function (compare (2.44))

$$\frac{1}{2} E(n'^*_k n'_m) = \begin{cases} N_0 u_{k-m} & (|k - m| \leq L) \\ 0 & (otherwise) \end{cases} \tag{2.52}$$

For calculating the error rate performance of the MLSE receiver and in order to compare the performance with other equalisation or estimation techniques it is common to whiten the noise sequence by further filtering of the sequence $\mathbf{r}$. Fig. 2.12 shows the block diagram of the optimum receiver with whitening filter. Since the filter succeeds
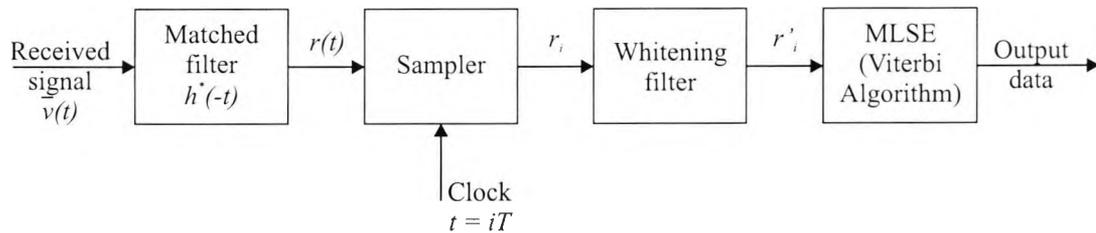
Figure 2.12: Optimum receiver with whitening filter

the sampling device, the effect of the matched filter is not affected. After the whitening filter the samples r'_k of the output sequence can be expressed as

$$r'_k = \sum_{i=0}^{L} u'_i z_{k-i} + \eta_k \tag{2.53}$$

where η_k are the samples of a white gaussian noise sequence and u' represents the cascade of the transmitter filter $g_t(t)$, the channel, the matched filter $h^*(-t)$, the sampler, and the discrete time noise whitening filter.

With the receiver structure based on the whitening approach it is possible to use the Euclidean distance in the Viterbi algorithm. The main advantage of this receiver can be seen in the simplified computation of the Euclidean distances between the current received sequence $\mathbf{r}'$ and all possible undisturbed sequences $\mathbf{u}'\mathbf{z}$. The Viterbi algorithm is briefly explained in the following.

Viterbi Algorithm

The Viterbi algorithm is a recursive optimal solution to the problem of estimating the state sequence of a discrete time finite state Markov process observed in memoryless noise. The algorithm was first proposed from Viterbi in 1967 as a method of decoding convolutional codes. Since that time, it has been recognised as an attractive solution to a variety of digital estimation problems, like the described MLSE. For MLSE the Viterbi algorithm will usually be employed for receiver structures which include the whitening filter. However, the whitening filter is not mandatory. Then the Euclidean distance in the Viterbi algorithm has to be replaced by a modified metric.

If the intersymbol interference affects $L+1$ symbols, the MLSE criterion is equivalent to the problem of estimating the state of a discrete time finite state machine. If the information symbols are M-ary, the channel filter has M^L states and can be described by a M^L-state trellis. The number of possible transitions between two subsequent time

48

instants is characterised by $N_{trans} = M \cdot M^L = M^{L+1}$. In the Viterbi algorithm, the metrics $J(\mathbf{z}_k)$ which are updated for each transition are calculated recursively by using the Euclidean distance

$$J(\mathbf{z}_k) = \min_{z_k} \left[J(\mathbf{z}_{k-1}) + \left| r'_k - \sum_{i=0}^{L} u'_i z_{k-i} \right|^2 \right]$$

$$= \min_{z_k} \left[J(\mathbf{z}_{k-1}) + \lambda(z_k) \right]$$

(2.54)

λ describes the branch metrics corresponding to the transitions that merge at a common state. Equation 2.54 shows that the Viterbi algorithm can be implemented with Add-Compare-Select units.

Starting with the M^L path metrics $J(\mathbf{z}_{k-1})$ M^{L+1} branch metrics $\lambda(z_k)$ are calculated and added in groups of M to the path metrics (Add). The M updated path metrics $J(\mathbf{z}_k)$ at every state are compared (Compare) and the smallest is selected (Select). The sequence $\mathbf{z}_k$ corresponding to the selected $J(\mathbf{z}_k)$ is saved while the $M - 1$ remaining sequences are discarded. Thus, M^L sequences with metrics $J(\mathbf{z}_k)$ are left and the procedure starts again.

The computation cost of the Viterbi algorithm is independent of the length of the transmitted data sequence. However, the length of the data vector, which includes the winner sequences, increases every iteration. Practical observations show that, if looking backwards, all survivor paths will merge after a finite number of time instants. Under this assumption, at time instant k the data z_{k-q} can be decoded. In practical systems five times the value of the channel memory is regarded as an experience value of the decision delay, i.e., $q \geq 5 \cdot L$.

Linear Equalisation

The computational complexity of the MLSE described in the last section grows exponentially with the length of the channel time dispersion. If L is the number of interfering symbols and M is the size of the symbol alphabet, the Viterbi algorithm computes M^{L+1} metrics for each new received symbol. Apart from the possibility to use reduced MLSE algorithms, like the M-algorithm, such a large computational complexity is expensive to implement. An alternative is the use of suboptimal channel equalisers to compensate for the ISI. The simplest approach employs a linear filter, which is described in this section.

A linear filter structure which is often used for equalisation is the transversal filter given in Fig. 2.13. The filter has a computational complexity that is a linear function of the channel dispersion length L. The output signal of the transversal filter at the
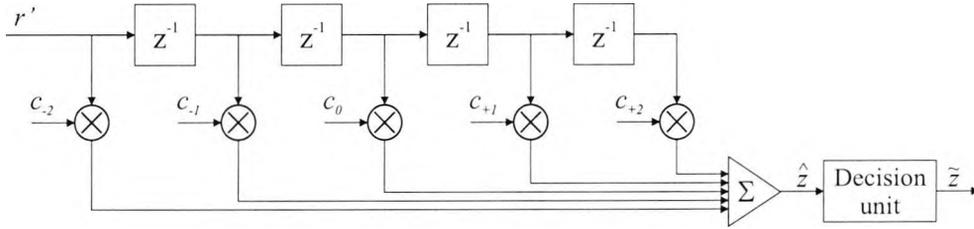
Figure 2.13: Structure of a linear transversal equaliser

input of the decision unit is

$$\hat{z}_k = \sum_{j=-K}^{K} c_j r'_{k-j} \tag{2.55}$$

where $\hat{z}_k$ is the estimate of the kth symbol, and c_j are the $2K + 1$ complex valued tap weight coefficients of the filter. The decision unit uses the distance criterion to form the decision $\tilde{z}_k$ from the estimate $\hat{z}_k$. If $\tilde{z}_k$ is not identical to the transmitted information symbol z_k, an error occurs. As mentioned earlier the average bit error probability is an often used measure of the performance of digital communications systems. Since the bit error probability is a highly nonlinear function of the filter coefficients c_j, it can not be used to optimise the tap weight coefficients of the equaliser. In practice two other criteria are implemented. One is the peak distortion criterion and the other is the mean square error criterion.

Peak Distortion Criterion: The peak distortion is simply defined as the worst-case intersymbol interference at the output of the equaliser. The cascade of the discrete time linear filter, which models the communications system, having coefficients u'_i and the equaliser with coefficients c_i can be represented by a single equivalent filter with impulse response

$$q_i = \sum_{j=-\infty}^{\infty} c_j u'_{i-j} \tag{2.56}$$

Here it is assumed that the equaliser has an infinite number of taps. The output at the kth sampling instant can then be expressed in the form

$$\hat{z}_k = q_0 z_k + \sum_{i \neq k} z_i q_{k-i} + \sum_{j=-\infty}^{\infty} c_j \eta_{k-j} \tag{2.57}$$

The first term in (2.57) gives a scaled version of the desired symbol. For convenience, q_0 is normalised to unity. The second term is the intersymbol interference. The peak value of this interference, which is called the peak distortion, is

$$\mathcal{D}(\mathbf{c}) = \sum_{\substack{i=-\infty \\ i \neq 0}}^{\infty} |q_i| = \sum_{\substack{i=-\infty \\ i \neq 0}}^{\infty} \left| \sum_{j=-\infty}^{\infty} c_j u'_{i-j} \right| \tag{2.58}$$

Thus, $\mathcal{D}(\mathbf{c})$ is a function of the equaliser tap weights.

With an equaliser having an infinite number of taps, it is possible to calculate the tap weights such that $\mathcal{D}(\mathbf{c}) = 0$. The intersymbol interference can then be eliminated completely. The values of the tap weights for $\mathcal{D}(\mathbf{c}) = 0$ are determined from the equation

$$q_i = \sum_{j=-\infty}^{\infty} c_j u'_{i-j} = \begin{cases} 1 & (i = 0) \\ 0 & (i \neq 0) \end{cases} \tag{2.59}$$

An implementable equaliser has only a finite number of filter taps. In the following $2K + 1$ coefficients are considered. Since $c_j = 0$ for $|j| > K$, the convolution of u'_i with c_i is zero outside the range $-K \leq i \leq K + L - 1$

$$q_i = \begin{cases} 0 & (i < -K) \\ 0 & (i > K + L - 1) \\ \left| \sum_j c_j u'_{i-j} \right| & (otherwise) \end{cases} \tag{2.60}$$

Although the equaliser has $2K + 1$ adjustable parameters, there are $2K + L$ nonzero values in the response q_i. Therefore, it is generally impossible to completely eliminate the intersymbol interference at the output of the equaliser. The problem then is to minimise $\mathcal{D}(\mathbf{c})$ with respect to the coefficients c_j

$$\mathcal{D}(\mathbf{c}) = \sum_{\substack{i=-K \\ i \neq 0}}^{K+l-1} |q_i| = \sum_{\substack{i=-K \\ i \neq 0}}^{K+L-1} \left| \sum_{j=-\infty}^{\infty} c_j u'_{i-j} \right| \tag{2.61}$$

Another problem of the linear equaliser is that large values of the calculated equaliser impulse response may result, especially for small values of the channel impulse response. The energy of the equaliser impulse response will then be very large and therefore may result in an unwanted large amplification of the noise power.

51

Mean Square Error (MSE) Criterion: In the MSE criterion, the tap weight coefficients c_j of the equaliser are adjusted to minimise the mean square value of the error

$$\varepsilon_k = z_k - \hat{z}_k \tag{2.62}$$

In [26] it is shown, that the MSE criterion results in an equaliser transfer function that differs only in the term which depends on the noise spectral density factor N_0 from the transfer function of the peak distortion criterion. That is, in the limit as $N_0 \to 0$, the two criteria yield the same solution in the tap weights. Consequently, when $N_0 = 0$, the minimisation of the MSE results in a complete elimination of the intersymbol interference. On the other hand, when $N_0 \neq 0$, there is both residual ISI and additive noise at the output of the equaliser. However, MSE has the advantage that the amplification of the noise power is less destructive.

Nonlinear Equalisation

Nonlinear equalisers are used in applications where the channel distortion is too severe to handle with a linear equaliser. Linear equalisers do not perform well on channels which have deep spectral nulls in their frequency response. In addition to the MLSE described before, the decision feedback equalisation (DFE) is the most known nonlinear equalisation method. The basic idea of a DFE is that once an information symbol has been detected and decided, the ISI that it causes on future symbols can be estimated and the estimated value can be subtracted before the detection of the next symbol is done. The decision feedback equaliser (DFE), illustrated in Fig. 2.14, consist of two filters, a feedforward filter (FFF) and a feedback filter (FBF). The input of the feedforward
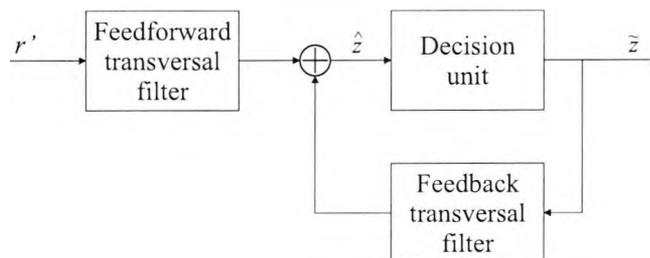
Figure 2.14: Structure of a decision feedback equaliser based on transversal filters

section is the received signal r'_k. The FFF is identical to the linear transversal equaliser described before. The FBF is driven by decisions on the output of the detector and

its coefficients can be adjusted to cancel the ISI on the current symbol from the past detected symbols. The equaliser output can be expressed as

$$\hat{z}_k = \sum_{j=-K_1}^{0} c_j r'_{k-j} + \sum_{j=1}^{K_2} c_j \tilde{z}_{k-j} \tag{2.63}$$

where $\hat{z}_k$ is an estimate of the kth information symbol, c_j are the tap coefficients of the filter, and $\tilde{z}_{k-j}$ are previously detected symbols. The equaliser is assumed to have $(K_1 + 1)$ taps in the FFF and K_2 taps in the FBF. It should be observed that this equaliser is nonlinear because the feedback filter contains previously detected symbols.

It can be shown that the minimum MSE for a decision feedback equaliser is almost always smaller than that of a linear equaliser [26]. The values of the feedback coefficients can be chosen in such a way, that the ISI from previously detected symbols can be eliminated completely, if the previous decisions are correct and if the memory of the FBF is greater or equal the channel memory, i.e., $K_2 \geq L$.

In time varying channels, such as mobile radio channels, the described equalisers require an algorithm to update the coefficients and track the channel variations. The adaptive equaliser will be discussed in Chapter 5.

2.3 Mobile Radio Channels

The characteristics of mobile radio channels affects the principles of mobile communications. Knowledge of the effects which occur during transmission over this channel is the basis for understanding the complex connections in modern digital mobile radio systems and its design principles.

The simplest model for a communication channel is the additive white Gaussian noise (AWGN) channel (Fig. 2.10). In this model, the transmitted signal $s(t)$ is corrupted by an additive random noise process $n(t)$. Physically, the additive noise process may arise from electronic components and amplifiers at the receiver of the communication system, or from interference encountered in transmission. The practical mobile channel is more complex than a simple AWGN channel. In a wireless mobile communication system, a signal can travel from transmitter to receiver over multiple reflective paths. The effect can cause fluctuations in the received signal's amplitude, phase, and angle of arrival, giving rise to the terminology multipath fading. Two types of fading effects characterise mobile communications: large-scale and small-scale fading. Large-scale fading represents the average signal power attenuation or path loss due to motion over large areas (depending on the propagation environment from several meters up to thousands of meters). Small-scale fading refers to the dramatic changes in signal amplitude and phase that can be experienced as a result of small changes (as small as half a wavelength) in the spatial separation between a receiver and transmitter. The received fading signal $r(t)$ can be partitioned in terms of the two fading effects

$$r(t) = m(t) \cdot r_0(t) \tag{2.64}$$

where $m(t)$ is called the large-scale fading component, and $r_0(t)$ is called the small-scale fading component. Figure 2.15 shows the relationship between large-scale and small-scale fading. In Fig. 2.15a, received signal power versus antenna displacement is plotted, whereas in Fig. 2.15b the large-scale fading component has been removed in order to view the small-scale fading at constant average power. The typical antenna displacement between the signal nulls of the small-scale fading component is approximately half a wavelength.

54
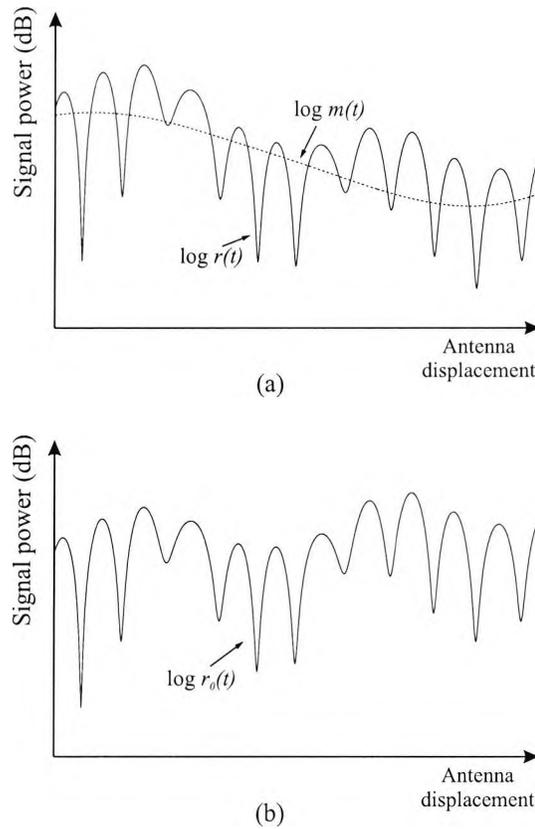
Figure 2.15: Large-scale fading and small-scale fading

2.3.1 Large-Scale Fading

Over time, some propagation models have emerged, which can be used to predict large-scale coverage for mobile communication systems design. A common model, the "Log-distance Path Loss" model is presented in the following.

Log-distance Path Loss Model

Both theoretical and measurement based propagation models show that the average received signal power decreases logarithmically with distance. The mean path loss, $\overline{L_p}(d)$, as a function of distance d between transmitter and receiver is proportional to the nth power of d relative to a reference distance d_0

$$\overline{L_p}(d) \propto \left(\frac{d}{d_0} \right)^n \tag{2.65}$$

or in decibels

$$\overline{L_p}(d)(\text{dB}) = L_p(d_0)(\text{dB}) + 10n \log \left(\frac{d}{d_0} \right) \tag{2.66}$$

55

Environment	Path Loss Exponent n
Free-space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Table 2.1: Path loss exponents for different environments

where $\overline{L_p}(d)(\text{dB}) = 10\log\overline{L_p}(d)$. The reference distance d_0 corresponds to a point located in the far field of the antenna. The reference path loss $L_p(d_0)$ is calculated using the free space path loss formula or through field measurements at distance d_0. $\overline{L_p}(d)$ is the average path loss for a given value of d. When plotted on a log-log scale, the modeled path loss is a straight line with a slope equal to $10n$ dB per decade. The value of n depends on the specific propagation environment. In free space $n = 2$, whereas n can be lower than 2 in the presence of a very strong guided wave phenomenon, and larger than 2 when obstructions are present (Tab. 2.1).

The path loss versus distance given in (2.66) is an average value, and therefore not adequate to describe the exact behaviour of a signal path. It is necessary to consider the fact that the surrounding environment clutter may be vastly different at two different locations having the same transmitter-receiver separation. Measurements have shown that for any value of d, the path loss $L_p(d)$ is a random variable which has a log-normal distribution (normal in dB) around the mean (distant depended) value $\overline{L_p}(d)$. That is

$$L_p(d)(\text{dB}) = L_p(d_0)(\text{dB}) + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma(\text{dB}) \qquad (2.67)$$

where X_σ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB).

The log-normal distribution describes the random shadowing effects which occur over a large number of measurement locations which have the same transmitter-receiver separation, but have different levels of clutter on the propagation path. This phenomenon is referred to as log-normal shadowing. A path loss exponent n and a standard deviation σ will therefore statistically describe the path loss for an arbitrary location having a specific transmitter-receiver separation. The log-normal shadowing model may be used in computer simulations to provide received power levels for random locations in communication system design and analysis. In practice, it is not unusual to use values of X_σ as high as 6-10 dB or greater.

2.3.2 Small-Scale Fading

Small-scale fading describes the rapid fluctuation of a radio signal over a short period of time or travel distance, so that large-scale path loss effects may be ignored. It is caused by wave interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. Different physical factors in the radio propagation channel influence small-scale fading. These include:

1. *Multipath propagation* - The presence of reflecting objects and scatterers in the channel creates a constantly changing environment that dissipates the signal energy in amplitude, phase, and time. The result is, that multiple versions of the transmitted signal arrive at the receiver and every version differs in time of arrival. The random phase and amplitudes of these different multipath components cause fluctuations in signal strength and/or signal distortion.

2. *Speed of the mobile* - The relative motion between the mobile and the base station results in random frequency modulation due to different Doppler shifts on each of the multipath components. Depending on the moving direction, the Doppler Shift may be positive or negative.

3. *Speed of surrounding objects* - If objects in the radio channel are in motion, they induce a time varying Doppler Shift. This effect must only be considered if the moving rate of the surrounding objects is in the order of the velocity of the mobile.

4. *The transmission bandwidth of the signal* - An estimate about the statistics of fading strength and the likelihood of time signal spreading can be gained from the ratio of the signal bandwidth to the coherence bandwidth of the channel.

The physical factors results in two mechanisms which are relevant for data transmission:

- Time spreading of the underlying digital pulses within the signal

- A time-variant behaviour of the channel due to motion

Table 2.2 shows, that each of the two mechanisms can be described in two domains (time or time-delay and frequency or Doppler shift), and gives an overview over the degradation categories and their corresponding mechanisms. The mechanisms and the degradation categories will be described in detail in the following.

Time-spreading mechanism due to multipath		Time-variant mechanism due to motion	
Time delay domain	**Frequency selective fading** multipath delay spread > symbol time	**Fast fading** channel fading rate > symbol rate	Doppler shift domain
	Flat fading multipath delay spread < symbol time	**Slow fading** channel fading rate < symbol rate	
Frequency domain	**Frequency selective fading** channel coherence BW < symbol rate	**Fast fading** channel coherence time < symbol time	Time domain
	Flat fading channel coherence BW > symbol rate	**Slow fading** channel coherence time > symbol time	

Table 2.2: Small-scale fading: mechanisms, and effects

For the remainder of this section, it will be assumed that the mobile channel is WSSUS (wide sense stationary uncorrelated scattering). WSS means that the fading statistics of a physical channel is approximately stationary for a sufficiently long time interval. Therefore, in this time interval the channel correlation functions are time invariant, i.e., the correlation functions do not depend on the specific time instants t_1 and t_2, but, instead, depend only on the time difference $\tau = t_1 - t_2$. The second assumption (US) states that the signals arriving with different delays are uncorrelated. The WSSUS model allows the description of the mobile radio channel through correlation functions. Figure 2.16 contains four functions that define the WSSUS model.

Signal time-spreading

As mentioned above, the time-spreading mechanism can be described in the time-delay domain and in the frequency domain. Figure 2.16a and 2.16b show the relationship. In Fig. 2.16a a multipath power delay profile versus time delay τ is plotted. For a typical wireless radio channel, the received signal usually consists of several discrete multipath components, sometimes referred to as taps. In order to compare different multipath channels, parameters are needed which quantify the channel. Three parameters are commonly used: mean excess delay, root mean squared (rms) delay spread, and excess delay spread (X dB). The mean excess delay is the first moment of the power delay profile and is defined as

$$\bar{\tau} = \frac{\int \tau |h(\tau)|^2 d\tau}{\int |h(\tau)|^2 d\tau} \tag{2.68}$$

58
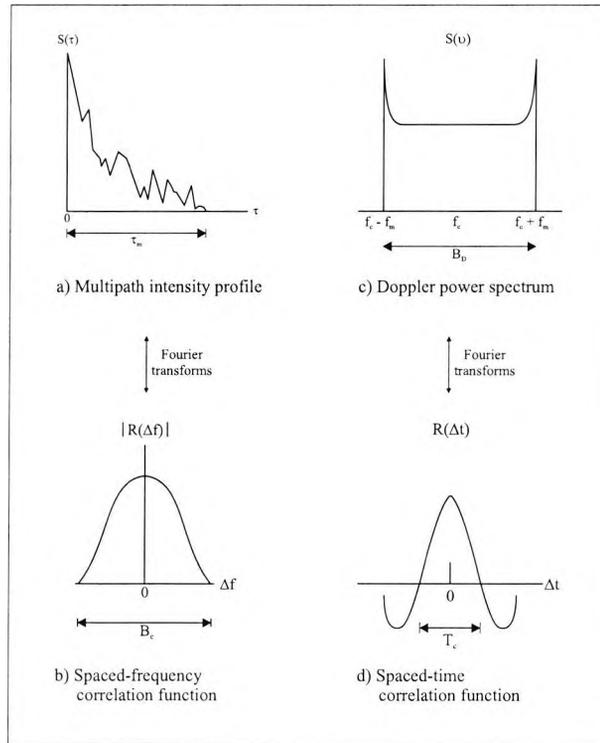
Figure 2.16: Relationships among the channel correlation functions and power density functions

The rms delay spread is the square root of the second central moment of the power delay profile

$$\sigma_\tau = \sqrt{\frac{\int (\tau - \bar{\tau})^2 |h(\tau)|^2 d\tau}{\int |h(\tau)|^2 d\tau}} \tag{2.69}$$

These two parameters are often used to describe the characterisation of the power delay profile. For the development of a practical receiver the third parameter is more important. The excess delay spread (X dB) or maximum excess delay (X dB) represents the time τ_m between the first and last received component where the signal exceeds some threshold level, defined by the strongest component. The threshold level might be chosen at 10 or 20 dB below the level of the strongest component. In a receiver design, this delay value is relevant to define the number of taps of the equaliser. Figure 2.17 shows an example of an indoor power delay profile and the different delay values. The noise threshold level is also given in the figure. The noise threshold is used to distinguish between received multipath components and thermal noise. If the noise threshold is set too low, then the noise will be detected as a multipath signal and the delay estimates would be wrong.

In a fading channel, the relationship between maximum excess delay time τ_m and

59
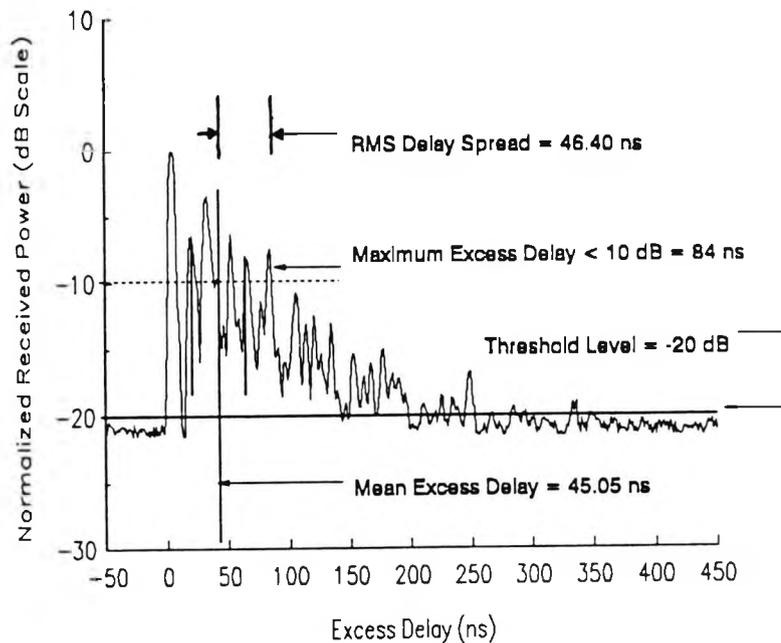
Figure 2.17: Example of an indoor power delay profile; rms delay spread, mean excess delay, maximum excess delay (10 dB), and threshold level (from[28])

symbol time T_s can be viewed in terms of two different degradations categories: frequency selective fading or flat fading (see Table 2.2). If $\tau_m > T_s$, then the channel appears frequency selective. This conditions occurs whenever a multipath component of a symbol is received beyond the symbol's time duration. The effect is the same if a transversal filter is used. Another name for the fading degradation caused by frequency selective fading is channel induced ISI. In the case of frequency selective fading, mitigating the distortion is possible if the multipath components are resolvable by the receiver. In [32] several such mitigation techniques are described.

In the frequency domain the time-spreading mechanism can be described by the spaced-frequency correlation function $|R(\Delta f)|$ (Fig. 2.16), that is the Fourier transform of $S(\tau)$. $|R(\Delta f)|$ represents the correlation between the channel's response to two signals as a function of the frequency difference of the two signals. It can also be considered as the channel's frequency transfer function. The relevant parameter to classify the channel in frequency selectivity or non-selectivity is the coherence bandwidth B_c. Coherence bandwidth is a statistical measure of the range of frequencies over which the channel can be considered "flat", i.e., all spectral components inside the range pass through the channel with approximately equal gain and linear phase. B_c and τ_m are

60

related reciprocally (within a multiplicative constant). As an approximation it is

$$B_c \equiv \frac{1}{\tau_m} \tag{2.70}$$

Furthermore an approximate relationship between coherence bandwidth and rms delay spread σ_τ can be found. The exact relation depends on the definition of B_c, i.e., the cutoff frequency of $|R(\Delta f)|$.

$$
\begin{aligned}
B_c &\approx \frac{1}{50\sigma_\tau} \quad |R(\Delta f)| \text{ above 0.9 of maximum value} \\
B_c &\approx \frac{1}{5\sigma_\tau} \quad |R(\Delta f)| \text{ above 0.5 of maximum value}
\end{aligned} \tag{2.71}
$$

In the frequency domain a mobile radio channel is referred to as frequency selective, if $B_c < B_s$, where B_s is the signal bandwidth. Frequency selective fading distortions occur whenever a signal's spectral components are not equally affected by the channel. Some of the signal's spectral components, which are falling outside the coherence bandwidth, will be affected differently by the channel compared to those components within B_c. The other case, $B_c < B_s$, results in a frequency non-selective or flat fading degradation. Flat fading does not introduce channel induced ISI distortion, but a performance degradation will be caused due to reduced SNR, whenever the signal is in a fading event. Typical flat fading channels show deep fades of 20 or 30 dB. To design a wireless communication system, the knowledge of the distribution of the amplitudes of the flat fading channel is important. The most commonly used amplitude distribution and the worst case fading channel is the Rayleigh channel. The Rayleigh flat fading channel model assumes that the received signal results from multiple reflective rays without a line-of-sight component. The amplitude which is induced by the channel varies according to the Rayleigh distribution. The probability density function (pdf) of the Rayleigh distribution is given by

$$
p(r) = \begin{cases} \frac{r}{\sigma^2} e^{\left[-\frac{r^2}{2\sigma^2} \right]} & (0 \leq r \leq \infty) \\ 0 & (r < 0) \end{cases} \tag{2.72}
$$

where r is the envelope amplitude of the received signal, and σ^2 is the time average power of the received signal.

If a dominant stationary (nonfading) signal component is present in the radio channel, such as a line-of-sight propagation path, the small-scale fading envelope distribution becomes Ricean instead of Rayleigh. In Ricean channels, random multipath components arriving at different angles are superimposed on a stationary dominant signal.

61

In general the probability to find a dominant component increases with decreasing distance between transmitter and receiver. It can be stated, that the signal distortions through fading events are weaker in the case of a strong dominating component. The Ricean distribution is given by

$$p(r) = \begin{cases} \frac{r}{\sigma^2} e^{\left[-\frac{r^2+A^2}{2\sigma^2}\right]} I_0\left(\frac{Ar}{\sigma^2}\right) & (A \geq 0,\ r \geq 0) \\ 0 & (r < 0) \end{cases} \qquad (2.73)$$

where A denotes the amplitude of the dominant signal. $I_0(\bullet)$ is the modified Bessel function of the first kind and zero order. Usually the Ricean distribution is described through the so called Ricean Factor K. The Ricean factor K is defined as the ratio between the deterministic signal power and the variance of the multipath components

$$K = \frac{A^2}{2\sigma^2} \quad \text{or in terms of dB:} \quad K(\text{dB}) = 10\log\frac{A^2}{2\sigma^2}\text{dB} \qquad (2.74)$$

The parameter K completely specifies the Ricean distribution. In the case of $A \to 0$, $K \to -\infty$, the Ricean distribution degenerates to a Rayleigh distribution. For $K \gg 1$ the Ricean distribution is approximately Gaussian around the mean. In Fig. 2.18 the pdf of the Ricean distribution for several values of K are illustrated.
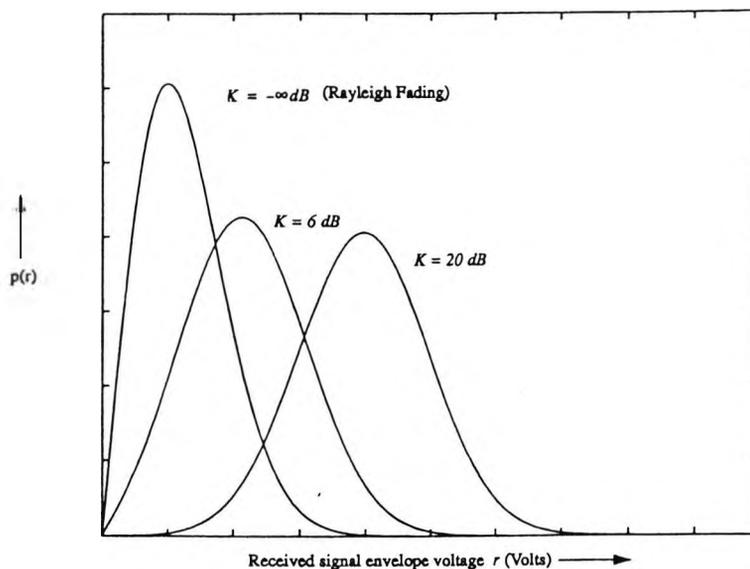
Figure 2.18: Probability density function of a Ricean distribution for different values of K (from [34])

62

Time variance of the channel

Delay spread and coherence bandwidth are parameters which describe the time dispersive nature of the channel in a local area. They do not offer information about the time varying nature of the mobile radio channel caused by either relative motion between the mobile and the base station, or by movements of objects in the channel. The parameters Doppler spread and coherence time describe the time varying nature of a radio channel in a small-scale region. Figure 2.16c shows a Doppler power spectral density $S(\nu)$ plotted as a function of the Doppler frequency shift ν. The Doppler spread B_D is a measure of the spectral broadening caused by the time rate of change of the mobile radio channel. It is defined as the range of frequencies over which the received Doppler spectrum is essentially non zero. The received signal spectrum of a pure sinusoidal tone, with a frequency f_c, includes spectral components in the range $f_c - f_d$ to $f_c + f_d$. The Doppler shift f_d is a function of the relative velocity of the mobile, and the angle α between the direction of motion of the mobile and direction of arrival of the scattered waves

$$f_d = \frac{v}{\lambda} \cos \alpha = f_m \cos \alpha \qquad (2.75)$$

where v is the velocity of the mobile and f_m is the maximum Doppler shift. The two extreme values of the Doppler frequency $(f_d = +f_m,\ f_d = -f_m)$ are obtained if a mobile drives directly to the base station $(\alpha = 0)$ or if it drives in the opposite direction $(\alpha = \pi)$. The width of the Doppler power spectrum is referred to as the spectral broadening or Doppler spread, denoted by f_m or B_D. Sometimes it is also called the fading bandwidth of the channel.

With the aid of the Doppler spread it is possible to decide whether a channel shows fast or slow fading. If the bandwidth B_s of the transmitted signal is less than the Doppler spread B_D, the mobile radio channel is called a fast fading channel. Conversely, a mobile radio channel is referred to as slow fading if $B_s > B_D$.

The time variant nature of a radio channel can also be characterised in the time domain. The time domain dual of the Doppler spread is the coherence time T_c. Figure 2.16d shows the spaced time correlation function $R(\Delta t)$. $R(\Delta t)$ is the autocorrelation function of the channel's response to a sinusoid. This function specifies the extend to which there is correlation between the channel's response to a sinusoid sent at time t_1 and the response to a similar sinusoid sent at time t_2, where $\Delta t = t_2 - t_1$. The coherence

time T_c is a statistical measure of the time duration over which the channel impulse response is essentially invariant, and quantifies the similarity of the channel response at different times. The Doppler spread and coherence time are inversely proportional to each other. The approximate relationship between the two parameters is

$$T_c \approx \frac{1}{B_D} \tag{2.76}$$

When T_c is defined more precisely as the time duration over which the channel's response to a sinusoid has a correlation greater than 0.5, the relationship between T_c and B_D is approximately

$$T_c \approx \frac{9}{16\pi B_D} \tag{2.77}$$

A popular rule of thumb for modern digital communications is to define the coherence time as the geometric mean of equations 2.76 and 2.77

$$T_c = \sqrt{\frac{9}{16\pi B_D^2}} = \frac{0.423}{B_D} \tag{2.78}$$

Similar to the Doppler spread, the coherence time T_c can be used to classify the mobile channel into fast fading and slow fading channels

$$
\begin{aligned}
T_s > T_c &\implies \quad \text{fast fading} \\
T_s < T_c &\implies \quad \text{slow fading}
\end{aligned}
\tag{2.79}
$$

where T_s is the time duration of a transmission symbol.

2.3.3 Examples of Mobile Radio Channels

Up to now the mobile channels are characterised by the technical and physical parameters. For the design of a digital communication system it is important to have models which describe the typical propagation conditions in urban or rural areas. Channel models are usually applied which divide the propagation conditions in four environment classes

- Typical Urban

- Bad Urban

- Rural Area

- Hilly Terrain

Typical Urban

This model is typical for built-up areas where there is no line-of-sight path but some reflections from large, distant buildings occur. In Fig. 2.19 the model impulse response of this channel type is shown. The following four figures which illustrate the environment classes in this section are taken from [7]. The Multipath power delay is drawn over
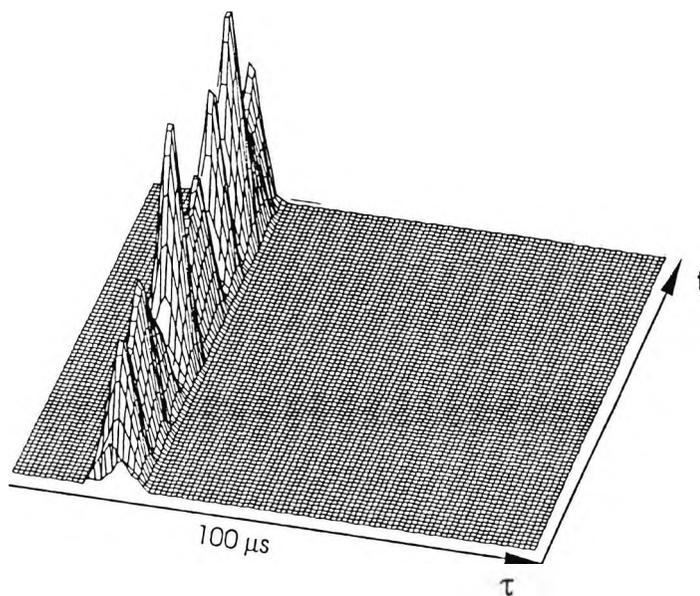
Figure 2.19: Model impulse response of typical urban channel versus time delay τ and time t (respectively location x)

a delay time of $100\mu s$ with a resolution of $5\mu s$. The time (location) axis corresponds to some wavelength. In an urban environment the multipath components are concentrated in a small delay interval ($< 5\mu s$). The bandwidth of the communication system defines, whether the echo paths can be resolved or accumulated to a single peak. The given impulse response shows one echo with very strong fading events. Such a channel with only a time variant behaviour and no delay spreading is called a time selective channel.

Bad Urban

In less structured environments the direct component is received with higher probability. Then the direct path shows only weak fading in the channel impulse response (Fig. 2.20). Additionally, significant reflection paths emerge, which are delayed relative to the direct path. Their number may vary between 2 and 10 and the strength of these
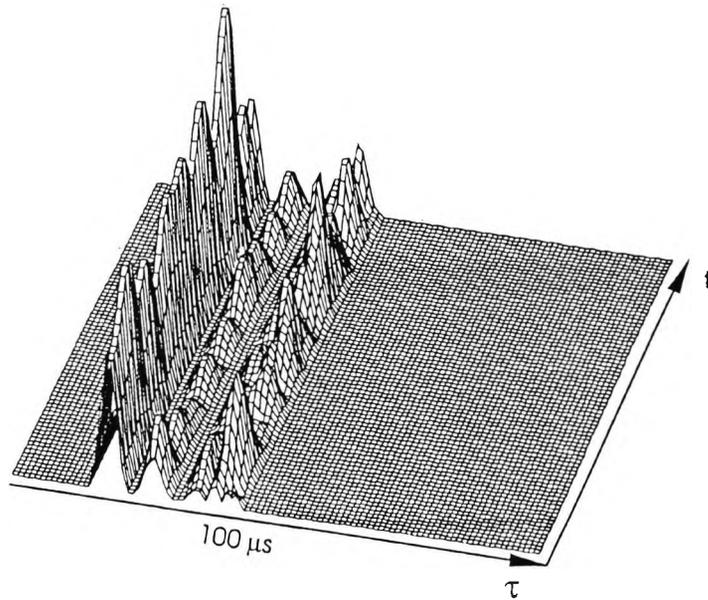
Figure 2.20: Model impulse response of bad urban channel versus time delay τ and time t (respectively location x)

paths typically decreases on the average with τ^{-4}. Echoes may usually be visible up to delays of $10\mu s$ depending on the transmitter power. For larger τ they may be hidden in the thermal noise, because the decrease of the received power through large-scale fading is large.

In contrast to the pure time selective channel, this model also includes delay components. Models which describe a time variant behaviour and include visible delay components are also known as time and frequency selective channels.

Rural Area

This model is typical for flat rural areas, where there is often a line-of-sight path (dominant component), no distance reflectors can be found, but scattering in the vicinity of the mobile is observed. The model defines one strong tap introducing Ricean fading and the scattering components with larger delays are only very weak (Fig. 2.21).

Hilly Terrain

This model is typical for hilly areas where reflections at distant hills are likely to occur. The delay time of the reflection components can be tens of microseconds, i.e., they
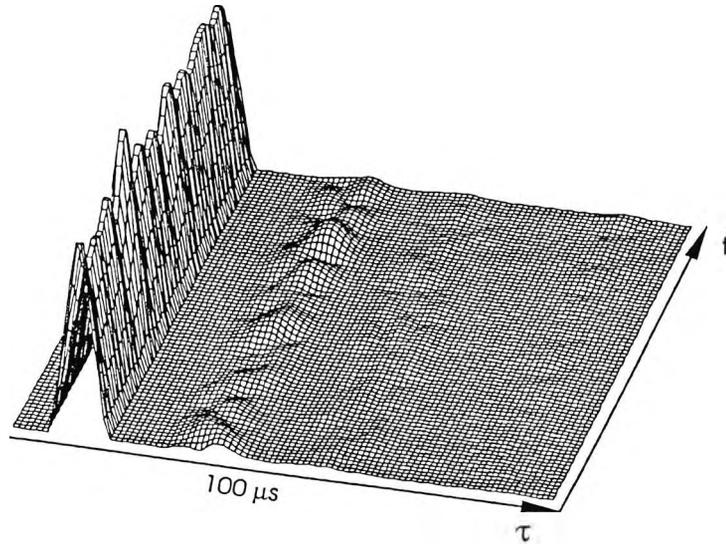
Figure 2.21: Model impulse response of rural area channel versus time delay τ and time t (respectively location x)

still supply visible contributions at such large distances. Figure 2.22 shows the impulse response when mountains build reflectors in the background. The reflections which can be resolved clearly have a delay relative to the direct path of up to approx. $80\mu s$.
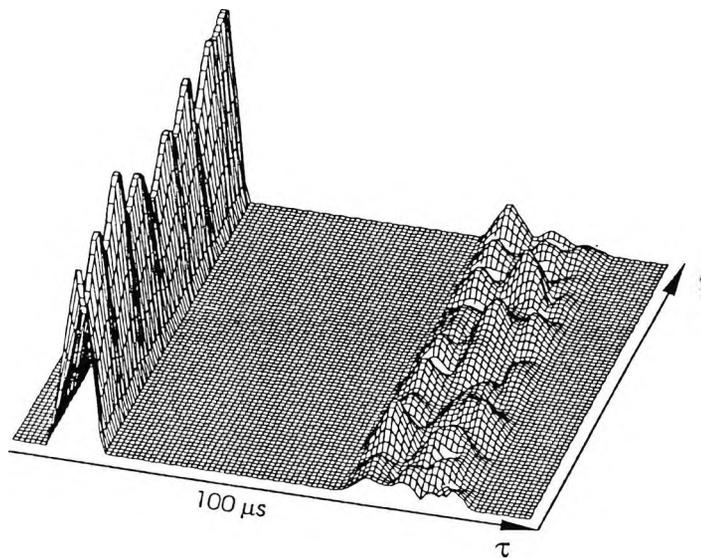
Figure 2.22: Model impulse response of hilly terrain channel versus time delay τ and time t (respectively location x)

2.4 Theoretical Performance of PSK Signals

In this section the formulas to calculate the bit error rate of PSK signals in different channels are listed. The precise derivation can be found in [26].

2.4.1 AWGN Channel

BPSK: For binary phase modulation with the signal space constellation from Fig. 2.2a (the two signals are antipodal), the bit error probability is

$$P_b = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_b}{N_0}} = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_s}{N_0}} \tag{2.80}$$

where $E_b = E_s$.

QPSK: A QPSK signal is made up of two binary phase modulation signals in phase quadrature. Since there is no crosstalk or interference between the signals on the two quadrature carriers, the bit error probability versus bit energy is identical to that of BPSK. However, it has to be considered, that in the case of QPSK $E_s = 2E_b$.

$$P_b = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_b}{N_0}} = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_s}{2N_0}} \tag{2.81}$$

The symbol error probability of QPSK is given by

$$P_s = \mathrm{erfc}\sqrt{\frac{E_s}{2N_0}} - \frac{1}{4}\left[\mathrm{erfc}\sqrt{\frac{E_s}{2N_0}}\right]^2 \tag{2.82}$$

For $E_s/N_0 \gg 1$ the term $[\mathrm{erfc}(\bullet)]^2$ can be ignored, and therefore, the symbol error probability is well approximated as

$$P_s = 2P_b \tag{2.83}$$

The exact calculation of the symbol error probability for signals with $M > 4$ is not possible in a closed manner. An approximation to the symbol error probability for large values of M is given by

$$P_s \approx \mathrm{erfc}\left(\sqrt{\frac{E_s}{N_0}}\sin\frac{\pi}{M}\right) \tag{2.84}$$

When a Gray code is used in the bit to symbol mapping, the equivalent bit error probability is approximated as

$$P_b \approx \frac{1}{\log_2(M)}\mathrm{erfc}\left(\sqrt{\frac{E_s}{N_0}}\sin\frac{\pi}{M}\right) \tag{2.85}$$

68

DPSK: A differentially encoded phase modulated (DPSK) signal allows two types of demodulation. In the first case, the demodulation is coherent followed by a differential detection. The symbol error probability for differentially encoded M-ary PSK is approximately twice the probability for M-ary PSK with absolute phase encoding

$$P_{s|DPSK} \approx 2P_{s|PSK} \tag{2.86}$$

With Gray coding the bit error probability is twice the BER of PSK

$$P_{b|DPSK} \approx 2P_{b|PSK} \approx \frac{2}{\log_2(M)} P_{s|PSK} \tag{2.87}$$

In the second case the demodulation of DPSK is incoherent. For DBPSK the bit error probability can then be calculated as

$$P_b = \frac{1}{2} e^{-E_b/N_0} \tag{2.88}$$

For $M > 2$ the formulas to calculate the BER of M-DPSK become quite difficult. In the publications of the European Telecommunications Standards Institute (ETSI, TETRA) the following rather simple but relatively exact equation for DQPSK is given by

$$P_b \approx \frac{1}{2} \text{erfc}\left(\sqrt{\frac{2E_s}{N_0}} \sin\left(\frac{\pi}{8}\right)\right) \tag{2.89}$$

In general, for M-DPSK with $M > 4$ the performance of DPSK is approximately 3dB poorer than that of PSK.

2.4.2   Multipath Rayleigh Fading Channel

In [26], the bit error probability for QPSK and DQPSK on a multipath Rayleigh fading channel is derived, assuming that a pair of information bits is mapped into the four phases according to a Gray coding rule. The expression for the bit error probability for an L-path diversity channel (with independent paths) is

$$P_b = \frac{1}{2}\left[1 - \frac{\mu}{\sqrt{2-\mu^2}} \sum_{k=0}^{L-1} \binom{2k}{k} \left(\frac{1-\mu^2}{4-2\mu^2}\right)^2\right] \tag{2.90}$$

where

$$\mu = \sqrt{\frac{\bar{\gamma}_c}{1+\bar{\gamma}_c}} \tag{2.91}$$

for coherent PSK and

$$\mu = \frac{\bar{\gamma}_c}{1 + \bar{\gamma}_c} \tag{2.92}$$

for DPSK. $\bar{\gamma}_c$ is the average received SNR per channel. Under the assumption that $\bar{\gamma}_c$ is identical for all L channels it is defined as

$$\bar{\gamma}_c = \frac{E_s}{N_0} E(\alpha_k^2) \tag{2.93}$$

where α_k is the attenuation of the kth channel. The average SNR per symbol then is $\bar{\gamma}_s = L\bar{\gamma}_c$.

In a one path Rayleigh fading channel (time selective channel), the bit error rate simplifies to

$$P_b = \frac{1}{2}\left[1 - \frac{\mu}{\sqrt{2 - \mu^2}}\right] \tag{2.94}$$

Chapter 3

Differential Encoded Offset-QPSK

3.1 Transmitter Concept

Offset-QPSK or staggered QPSK is a well known linear modulation format. It is used instead of QPSK in systems in which small variations of the bandlimited signal envelope and, therefore, less sensitivity to nonlinear distortions introduced by the power amplifier are important requirements. In the american digital cellular standard (IS95), for example, two different modulation formats for uplink and downlink are specified. The downlink modulation is QPSK. On the other hand, for the uplink an OQPSK modulation is employed in the mobile devices so that power efficient amplifiers can be applied. OQPSK is also used in different satellite systems. However, in all systems the OQPSK signal is demodulated coherently.
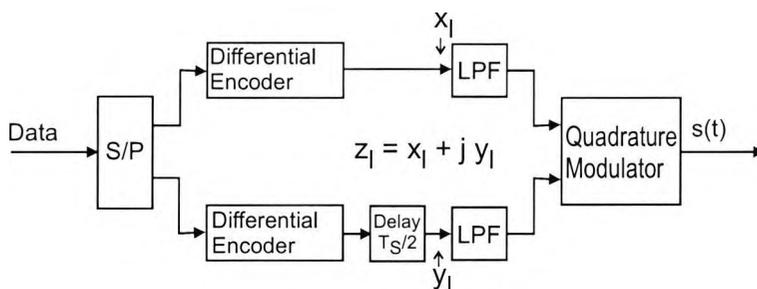
Figure 3.1: DOQPSK transmitter model

Figure 3.1 shows the block diagram of the DOQPSK transmitter which is used in all investigations in this thesis. From the serial data stream two bits are mapped in parallel to the two branches of the quadrature modulator. The bits of each branch are differentially encoded and filtered with square root raised cosine Nyquist 1 filters. The delay of half a symbol period $T_s/2$ in one branch gives the desired offset of one component in the modulated signal $s(t)$. It should be pointed out that the DOQPSK transmitter in Fig. 3.1 differs from the standard OQPSK transmitter by merely including a differential encoder in each channel.

71

3.2 Behaviour on Nonlinear Channels

3.2.1 Amplitude Distribution

In Chapter 2 the two modulation schemes $\pi/4$-QPSK and OQPSK are described. By means of the complex phase plots, the fluctuations of the envelopes of the modulation schemes are compared. The comparison becomes more clear, if the amplitude distributions of both schemes are considered. In Fig. 3.2 the amplitude distributions of $\pi/4$-DQPSK and DOQPSK at the transmitter side are shown. Both modulation schemes use a square root raised cosine Nyquist filter. Figure 3.2a illustrates the result for filtering with a roll-off factor of 0.35 and Fig. 3.2b for $r = 0.5$. The amplitudes
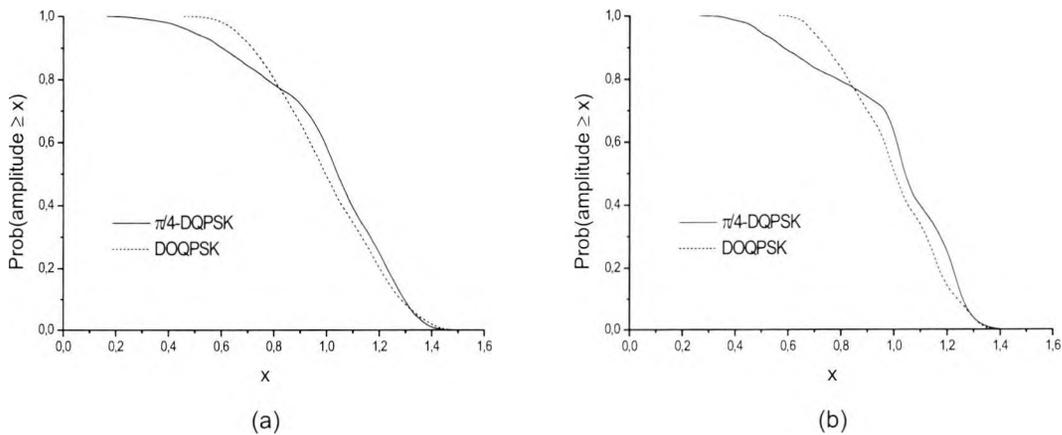
Figure 3.2: Inverse amplitude distributions (a) $r = 0.35$ and (b) $r = 0.5$, amplitudes are normalised with D

are normalised to the average amplitude D, which is the root of the average power of the modulated signal. If a filter with a roll-off factor of 0.35 is used, $\pi/4$-DQPSK has a minimum amplitude of $0.18D$, whereas the amplitude of DOQPSK is never less than $0.46D$. However, in this case the peak amplitude of DOQPSK is slightly larger. DOQPSK has a maximum amplitude of $1.51D$ instead of $1.48D$ for the peak amplitude of $\pi/4$-DQPSK. The resulting peak-to-average power ratios are 3.56 dB and 3.41 dB for DOQPSK and $\pi/4$-DQPSK, respectively. The performance changes clearly for the benefit of DOQPSK if filters are used with larger r. For a roll-off factor of 0.5 the values are $0.58D$, $1.41D$, and 3.01 dB for DOQPSK and $0.28D$, $1.44D$, and 3.15 dB for $\pi/4$-DQPSK. A further enlarging of the roll-off factor increases the performance difference. DOQPSK then becomes even more attractive for nonlinear amplifiers.

3.2.2 Generic Amplifier Model

The amplitude distributions give information about the fluctuations of the envelope. In order to be able to investigate and to compare the effects of the different amplitude distributions on the system performance when a nonlinear amplifier is used, a generic amplifier model is defined.

	in	out
p(0)	0	0
p(2a)	2a	a
p(1)	1	1-a
p(2)	2	2-a
p(i)	i	i-a

(a)

	in	out
p(0)	0	0
p(0.3)	0.3	0.3
p(0.6)	0.6	0.6
p(b)	b=c-e	b
p(d)	d=c+2e	c

(b)
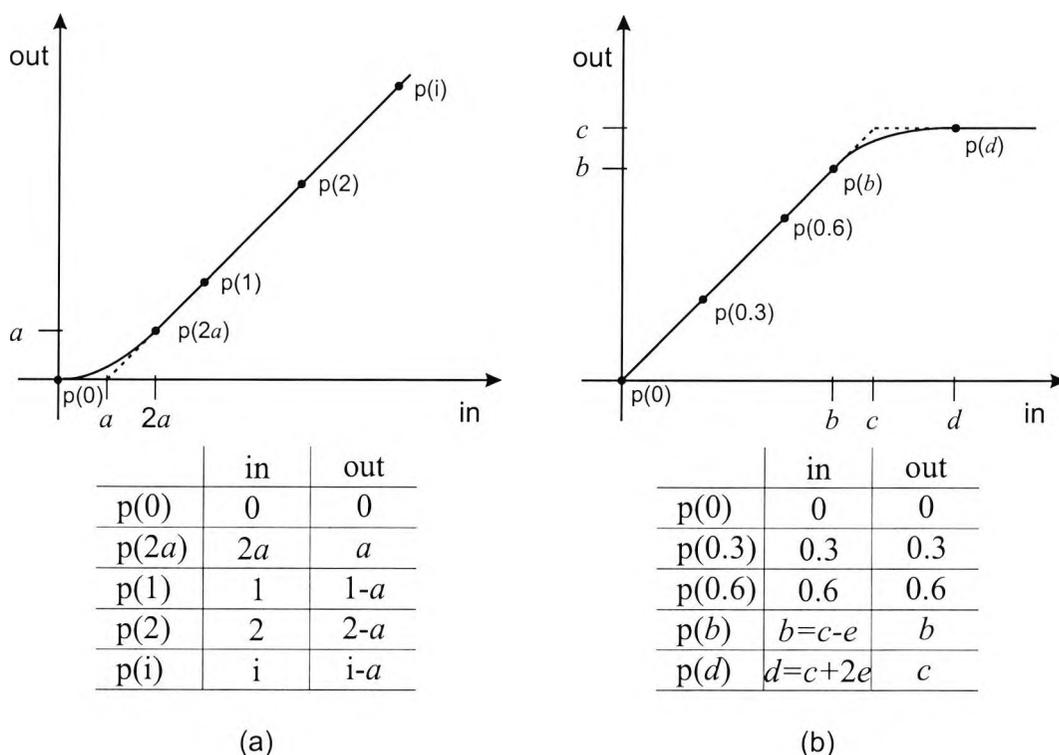
Figure 3.3: Amplitude transfer characteristic of a generic power amplifier (a) weak conduction, (b) saturation

In nonlinear amplifiers typically two effects due to the nonlinear characteristic occur, AM-AM conversion and AM-PM conversion. The generic amplifier model defined for the investigations in this thesis considers only the AM-AM conversion due to a nonlinear input output characteristic of the amplitude. The output phase versus input amplitude characteristic (AM-PM conversion) is not included in the model, since the purpose was to evaluate the dominant effect of the nonlinear amplitude factor alone. It will be shown in the next section with another amplifier model that the influence of the AM-PM effects on the system performance can be neglected. Two kinds of nonlinear amplitude distortions may appear at a power amplifier: saturation of the amplifier for high input voltage and blocking of the amplitude when the input signal is lower than the amplifier threshold (weak conduction region). To examine the different effects of nonlinearity,

two separate amplifier models are used (Fig. 3.3). The transfer characteristics are modelled by a third order spline function. The tables below the figures show the fixed points in the spline function. The weak conduction region in the left curve is mainly influenced through the threshold parameter a. In the right curve the parameter c adjusts the edge-of-saturation amplitude. The parameter e defines the shaping of the curve in the saturation region. For the following investigations e will be fixed at 0.3. The amplitudes in the model are also normalised to the average amplitude D.

3.2.3 ETSI Amplifier Model

In addition to the generic amplifier model defined in this thesis, a nonlinear amplifier model is also employed in the investigations according to a document of ETSI (European Telecommunications standards institution, ETSI/STC-RES 6.2 91 145, November 1991). The model is based on two nonlinear transfer functions. The nonlinear curves are given by

$$SAT(x) = \frac{1}{2} \left(\sqrt{(x+s)^2 + d^2} - \sqrt{(x-s)^2 + d^2} \right) \tag{3.1}$$

These curves have odd symmetry and tend asymptotically to $\pm s$. The shape factor is determined by the value of d. The values chosen for the simulation are $s = 2.4$ and $d = \{0.4, 0.7\}$. The different values of d result in two different curves. Both curves consider only amplitude distortions through the saturation effect (Fig. 3.4).
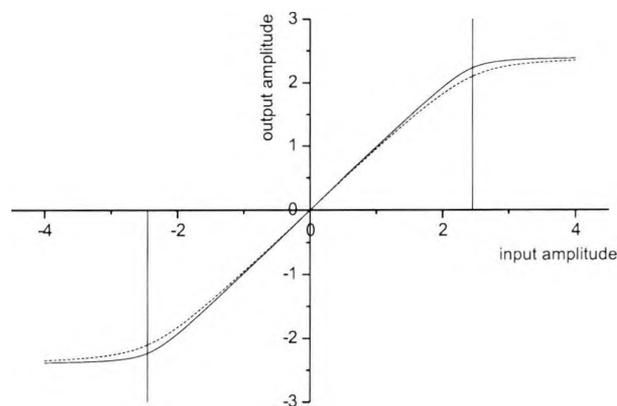
Figure 3.4: Nonlinear transfer functions of the ETSI amplifier model

In contrast to the description in the ETSI document, the amplifier model is implemented as a baseband simulation COSSAP module (Fig. 3.5). The baseband signal is split into two paths, one of which is subject to a delay of one sample and a phase

shift of $\frac{2}{3}\pi$. The sampling rate and the phase shift are chosen in such a way, that the result corresponds to $\frac{1}{3}$th of a cycle of a carrier defined in the ETSI document. The delayed and non-delayed signals are input to the two different nonlinear transfer functions. Finally, the output signals of the nonlinearities are summed. As a matter of the complex multiplication and the delay of one arm of the amplifier the AM-PM effects are generated.

AM/AM–AM/PM Conversion ETSI/STC–RES6.2(91)
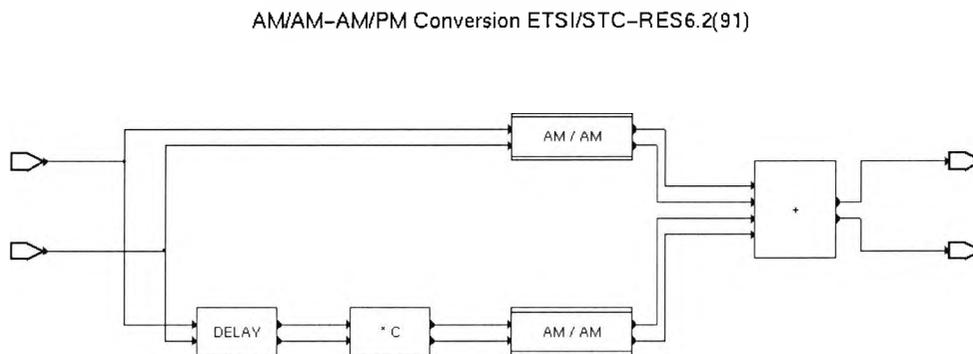
Figure 3.5: Block diagram COSSAP module of the ETSI nonlinear amplifier model

In the investigations on the generic amplifier model, the average power of the modulated signals is fixed and the transfer characteristic of the amplifier is varied by the given parameters. The ETSI model, however, describes a fixed transfer function. Therefore, the average power of the modulated signal has to be adjusted in such a way that the peak amplitude is in the region of the saturation. The BER results in the next chapters were achieved with an average signal power of 6, which corresponds to an amplitude of 2.45 in each arm (see the mark in Fig. 3.4).

3.2.4 Out-of-Band Power

One criterion to assess the behaviour of a modulation format on nonlinear channels is the out-of-band power. The out-of-band power ratio P_{out} is defined as:

$$P_{out} = 1 - \frac{\int\limits_{-B/2}^{B/2} W(f)\,df}{\int\limits_{-\infty}^{\infty} W(f)\,df} \qquad (3.2)$$

where $W(f)$ is the simulated power spectral density of the signal around the carrier frequency. P_{out} is the fraction of the total transmitted power that is found at frequencies

separated more than $B/2$ from the carrier frequency. Figures 3.6 and 3.7 show the computed (fractional) out-of-band power as a function of bandwidth B times symbol duration T_s for the amplifiers with amplitude characteristics from Fig. 3.3.

Figure 3.6 illustrates the effect of a nonlinearity due to the weak conduction region. In general and independent on the chosen roll-off factor, DOQPSK shows less spectral spreading than $\pi/4$-DQPSK, since the former has less energy in the deadband. An increasing of the roll-off factor even results in an increasing of the gain of DOQPSK. It is also interesting that even with small values of the threshold parameter a, e.g., 0.05 or 0.15, there is still some far-out spectrum radiation for both schemes, which have minimum amplitudes larger than these values. This is due to the fact that the presence of the threshold amplitude a in the transfer curve acts as a constant offset to the output amplitude and introduces an additional noise-like signal with constant amplitude a and random phase, superimposed on the amplified distortion-free signal. Thus the out-of-band radiation is directly connected with a and occurs even for input signals with minimum amplitude above a.
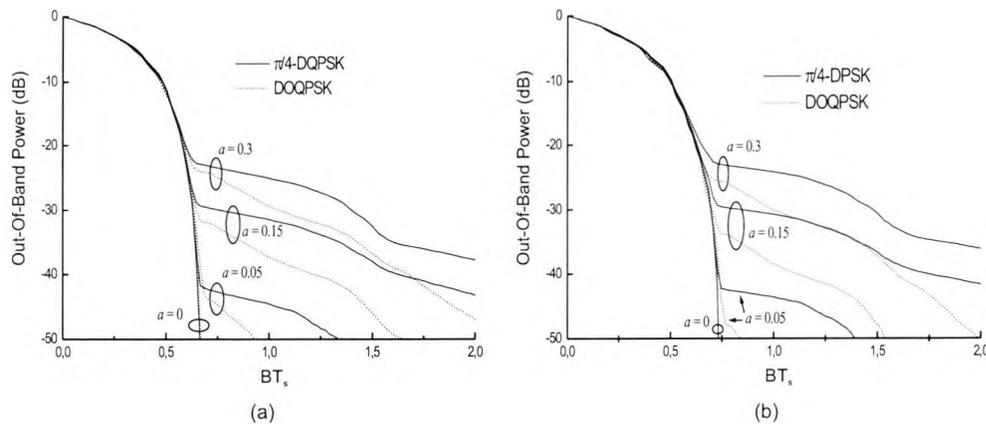
Figure 3.6: Out-of-band power, weak conduction (a) $r = 0.35$, (b) $r = 0.5$

The effect of a nonlinearity caused by saturation is shown in Fig. 3.7. Due to the similar amplitude distributions of both modulation schemes for higher amplitudes, the differences in the out-of-band power are smaller than in Fig. 3.6. In spite of the fact that the peak amplitude of DOQPSK is larger than that one of $\pi/4$-DQPSK, DOQPSK has less energy in the region around the center frequency of the adjacent channel. This effect can be explained, that not the peak amplitude alone but the amplitude distribution over the entire dynamic range influences the out-of-band power. In Fig. 3.2 it is illustrated, that DOQPSK shows only a worse performance in the

76

amplitude distribution in comparison to $\pi/4$-DPSK in a very small region.

The out-of-band power of DOQPSK signals compared to $\pi/4$-DQPSK signals is further decreased for pulse shaping filters with larger roll-off factors ($r > 0.5$). This result holds for both the saturation region (Fig. 3.7) and the weak conduction region (Fig. 3.6).
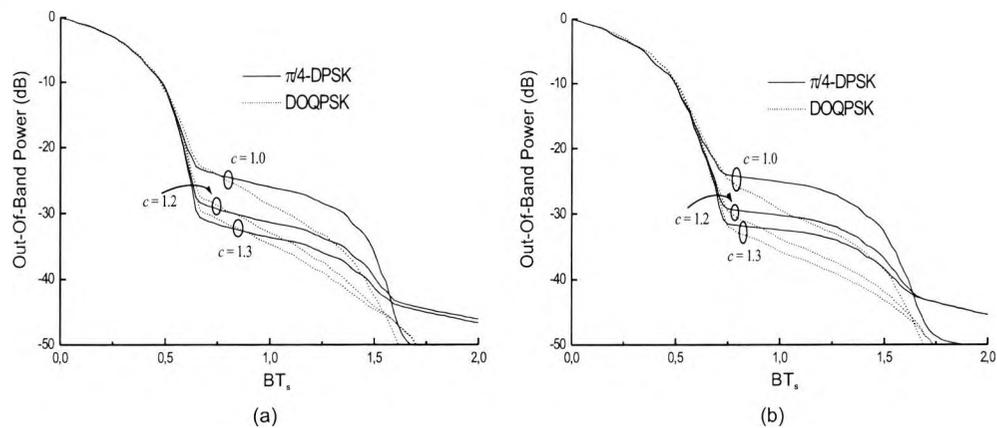
Figure 3.7: Out-of-band power, saturation (a) $r = 0.35$, (b) $r = 0.5$

3.2.5 Power Density Spectrum

The out-of-band power results, which have been obtained by simulations with the ETSI model are comparable to the curves in Fig. 3.7 and, therefore, they are not included in the thesis. However, the ETSI model is used to demonstrate the effect of the nonlinearities on the power spectral density (psd) of the modulation schemes. Different to the out-of-band power, the psd shows the actual spectral requirements of the modulation format and demonstrate the sidelobes. To better outline the difference
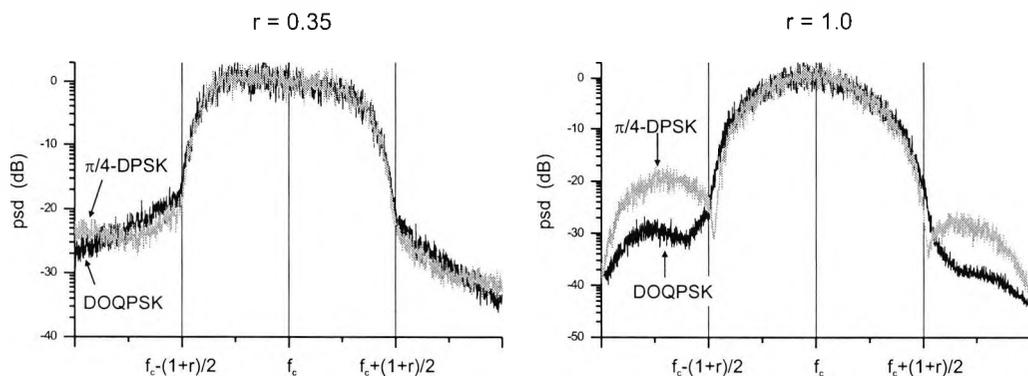
Figure 3.8: Power density spectra

of both modulation schemes, additionally to $r = 0.35$ the spectra for a roll-off factor of 1.0 are also given in Fig. 3.8. The spectra for $r = 1.0$ show a tiny bit smaller center lobe of $\pi/4$-DPSK, however, they show also the strong reduced sidelobes of DOQPSK. If the roll-off factor of the pulse shaping filters is reduced ($r = 0.35$), the difference in the power density spectra decreases.

The results of the psd for $r = 0.35$ and $r = 0.5$ and the nonlinear ETSI amplifier are in accordance with the results on the out-of-band power for the generic amplifier model. It can be summarised, that saturation has only a minor effect on the difference in the spectral shaping of both modulation formats, when the roll-off factor is less than $r = 0.5$. Therefore, gains in BER can only be expected for DOQPSK in comparison to $\pi/4$-DPSK when the nonlinearity effects of the weak conduction region are dominant.

Chapter 4

Differential Demodulation of DOQPSK-Signals

4.1 Receiver Algorithms

An efficient means for carrier phase estimation is differential demodulation. Then the estimation is based on the phase difference between the current and the last symbol. The difficulty of differential demodulation of differential encoded Offset-QPSK signals is due to the time overlap of the I- and Q-components. In this chapter receiver algorithms will be presented which can cope with this problem. First investigations on a receiver structure were presented in [14]. Figure 4.1 shows the block diagram of the receiver. It is based on a differential preprocessing (demodulation) algorithm and a trellis decoder to resolve the mutual interference of the I- and Q-components.
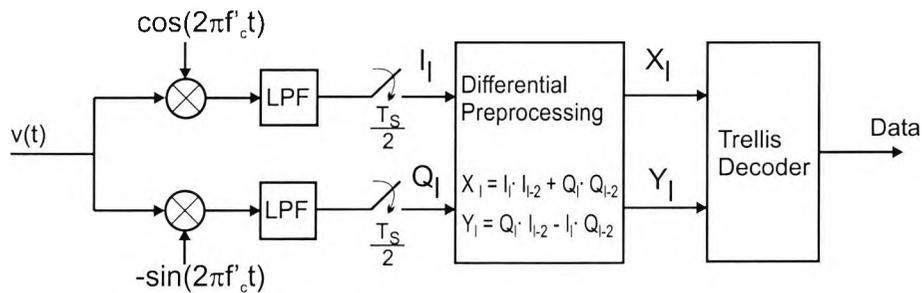
Figure 4.1: DOQPSK receiver model (differential demodulation)

The received signal $v(t)$ is down converted with a synthesiser, in which frequency f'_c is slightly mistuned to the carrier frequency f_c and low pass filtered with square root raised cosine filters resulting in the signal $r(t)$. The I- and Q-components of the complex baseband signal $r(t)$ are sampled twice per symbol interval T_s, i.e., once per bit interval. The discrete signal r_l at epoch l for an AWGN channel then becomes

$$r_l = I_l + j\,Q_l = \left[z_l + \sum_{i=1}^{\infty} g_i(z_{l-i} + z_{l+i})\right] e^{j\phi_l} + n'_l,$$

$$g_i = 0, \quad i = 2, 4, 6, \ldots,$$

(4.1)

79

where g_i designates the joint impulse response of the transmitter and receiver filter. Due to the Nyquist 1 criterion the impulse response has zeros at multiples of a symbol period. ϕ_l denotes the time varying phase, which rate of change is slow in comparison to a symbol or bit interval, and $z_l = x_l + \mathrm{j}\, y_l$ gives the differentially encoded, complex data at the transmitter (see Fig. 3.1). The filtered noise signal is represented by n_l'.

As a consequence of the time offset in the transmitter and the bit-by-bit sampling in the receiver, z_l contains only the data of the I-channel for even sampling time and the data of the Q-channel for odd sampling time. Thus,

$$z_l = \begin{cases} x_l & ; \quad l = 2k \\ \mathrm{j}\, y_l & ; \quad l = 2k + 1 \end{cases} \tag{4.2}$$

Depending on the binary and differentially encoded data x_l and y_l at the transmitter side, the following equations at even $(l = 2k)$ and odd $(l = 2k + 1)$ sampling times can be formulated as

$$\begin{aligned} r_{2k} &= I_{2k} + \mathrm{j}\, Q_{2k} \\ &= \left[x_{2k} + \mathrm{j} \cdot \sum_{i=1}^{\infty} g_i \cdot (y_{2k-i} + y_{2k+i}) \right] \cdot \mathrm{e}^{\mathrm{j}\,\phi_{2k}} + n_{2k}' \end{aligned} \tag{4.3}$$

$$\begin{aligned} r_{2k+1} &= I_{2k+1} + \mathrm{j}\, Q_{2k+1} \\ &= \left[\sum_{i=1}^{\infty} g_i \cdot (x_{2k+1-i} + x_{2k+1+i}) + \mathrm{j} \cdot y_{2k+1} \right] \cdot \mathrm{e}^{\mathrm{j}\,\phi_{2k+1}} + n_{2k+1}' \end{aligned} \tag{4.4}$$

The differential preprocessing after the receiver filters performs the mathematical operation $r_l r_{l-2}^*$, where r^* is the conjugate complex of r. It eliminates the unknown phase ϕ_l and provides the input for the metric computation of the trellis decoding algorithm. The real component X_l and the imaginary component Y_l of the signal at the output of the differential preprocessing unit are

$$X_l = \Re\left\{ r_l \cdot r_{l-2}^* \right\} = I_l \cdot I_{l-2} + Q_l \cdot Q_{l-2} \tag{4.5a}$$

$$Y_l = \Im\left\{ r_l \cdot r_{l-2}^* \right\} = Q_l \cdot I_{l-2} - I_l \cdot Q_{l-2} \tag{4.5b}$$

The trellis decoder in Fig. 4.1 compensates the intersymbol interference introduced through the differential demodulation.

The structure of the demodulator is the same for any type of Nyquist filtering. Depending on the chosen roll-off factor only the trellis decoder has to be changed, since a different number of symbols may interfere mutually due to the differential preprocessing.

As an example the I-channel is considered. Due to the Nyquist 1 criterion, there is no interference from other I-channel bits. However, there may be a number of Q-channel bits which interfere with the considered I-channel bit.

4.1.1 Algorithm for a Roll-Off Factor of 1.0

If pulse shaping filters with a roll-off factor of 1.0 are used, the signal also satisfies the Nyquist 2 criterion in addition to the Nyquist 1 criterion. In this case, only the preceding and succeeding Q-channel bit interfere on a given I-channel bit and vice versa. Figure 4.2 illustrates this subject.
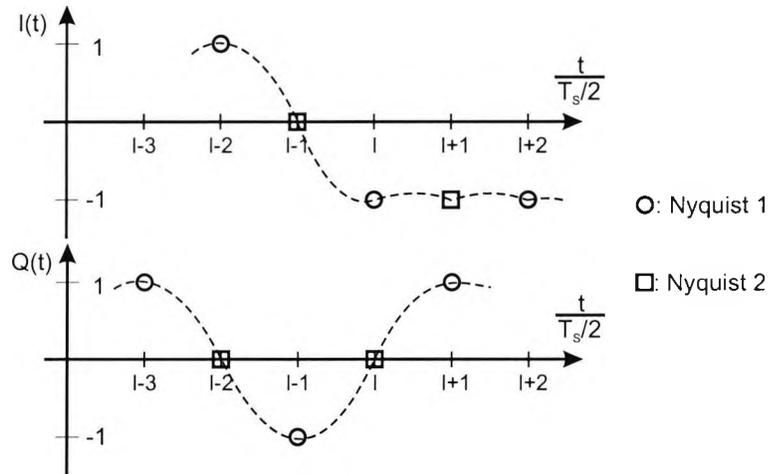
Figure 4.2: I- and Q-channel signals at the receiver filter output which satisfy Nyquist criteria 1 and 2

The impulse response g_i in 4.1 then is

$$g_i = \begin{cases} 1.0 & ; \quad i = 0 \\ 0.5 & ; \quad i = 1 \\ 0 & ; \quad other \end{cases} \tag{4.6}$$

and hence, 4.3 and 4.4 are simplified to

$$r_{2k} = [x_{2k} + 0.5\,\mathrm{j}\,(y_{2k-1} + y_{2k+1})] \cdot \mathrm{e}^{\mathrm{j}\,\phi_{2k}} + n'_{2k} \tag{4.7}$$

$$r_{2k+1} = [0.5(x_{2k} + x_{2k+2}) + \mathrm{j}\,y_{2k+1}] \cdot \mathrm{e}^{\mathrm{j}\,\phi_{2k+1}} + n'_{2k+1} \tag{4.8}$$

In order to compute the nominal values $\overline{X}_l$ and $\overline{Y}_l$ of the trellis decoder, no noise is considered in r_l. Consequently, at sampling intervals $l = 2k$ the nominal values are

81

defined as

$$\overline{X}_{2k} = \Re\left\{r_{2k} \cdot r_{2k-2}^*\right\}$$
$$= x_{2k}x_{2k-2} + \frac{1}{4}(y_{2k-1} + y_{2k+1})(y_{2k-3} + y_{2k-1}) \qquad (4.9a)$$

$$\overline{Y}_{2k} = \Im\left\{r_{2k} \cdot r_{2k-2}^*\right\}$$
$$= \frac{1}{2}x_{2k-2}(y_{2k-1} + y_{2k+1}) - \frac{1}{2}x_{2k}(y_{2k-3} + y_{2k-1}) \qquad (4.9b)$$

and at sampling intervals $l = 2k + 1$

$$\overline{X}_{2k+1} = \Re\left\{r_{2k+1} \cdot r_{2k-1}^*\right\}$$
$$= y_{2k+1}y_{2k-1} + \frac{1}{4}(x_{2k} + x_{2k+2})(x_{2k-2} + x_{2k}) \qquad (4.10a)$$

$$\overline{Y}_{2k+1} = \Im\left\{r_{2k+1} \cdot r_{2k-1}^*\right\}$$
$$= -\frac{1}{2}y_{2k-1}(x_{2k} + x_{2k+2}) + \frac{1}{2}y_{2k+1}(x_{2k-2} + x_{2k}) \qquad (4.10b)$$

The equations lead to a simple trellis. In the case of $l = 2k$, both $\overline{X}_{2k}$ and $\overline{Y}_{2k}$ are functions of y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k}, and y_{2k+1}. Therefore, transitions from an initial state $(y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k})$ to a final state $(x_{2k-2}, y_{2k-1}, x_{2k}, y_{2k+1})$ have to be considered. To these transitions, values of $\overline{X}_{2k}$ and $\overline{Y}_{2k}$ as specified in (4.9) are associated. Similarly, transitions from an initial state $(x_{2k-2}, y_{2k-1}, x_{2k}, y_{2k+1})$ to the final state $(y_{2k-1}, x_{2k}, y_{2k+1}, x_{2k+2})$ have to be taken into account at odd sampling intervals. From the decoded bits x_l and y_l an estimate of the input data sequence of the transmitter is obtained with standard differential decoding.

Note that $\overline{X}_{2k+1} = \overline{X}_{2k}$ and $\overline{Y}_{2k+1} = -\overline{Y}_{2k}$ when the initial and final states are considered at even and odd sampling intervals. This means that the resulting 16 state trellis with two transitions leaving and entering each node is essentially time invariant. A further simplification can be carried out, since both $\overline{X}_l$ and $\overline{Y}_l$ are invariant when the initial and final state are complemented. Applying this symmetry to the trellis decoder and bearing in mind that the bits are decoded differentially an 8 state trellis results. In Table 4.1 the transitions and their associated values of $\overline{X}_l$ and $\overline{Y}_l$ are given.

Data decoding in the trellis decoder is implemented using the Viterbi algorithm. To simplify computational complexity, the metric in the Viterbi algorithm is obtained from the squared Euclidean distance between the input vector and the nominal complex value of each state, i.e., it is assumed that the additive noise is white and Gaussian. However, the differential preprocessing together with the two samples per symbol in

82

initial state	final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
1 1 1 1	1 1 1 1	2	0
1 1 1 1	1 1 1-1	1	-1
1 1 1-1	1 1-1 1	0	2
1 1 1-1	1 1-1-1	-1	1
1 1-1 1	1-1 1 1	1	0
1 1-1 1	1-1 1-1	1	-1
1 1-1-1	1-1-1 1	-1	0
1 1-1-1	1-1-1-1	-1	-1
1-1 1 1	-1 1 1 1	0	-2
1-1 1 1	-1 1 1-1	-1	-1
1-1 1-1	-1 1-1 1	2	0
1-1 1-1	-1 1-1-1	1	1
1-1-1 1	-1-1 1 1	-1	0
1-1-1 1	-1-1 1-1	-1	1
1-1-1-1	-1-1-1 1	1	0
1-1-1-1	-1-1-1-1	1	1

Table 4.1: Trellis for DOQPSK with $r = 1.0$

principle destroy this assumption. The accumulated metric $J(j_l)$ of state j at epoch l and the branch metric λ are calculated from the well-known equations

$$J(j_l) = \min\{J(n_{l-1}) + \lambda_{nj,l-1}, J(m_{l-1}) + \lambda_{mj,l-1}\} \qquad (4.11a)$$

$$\lambda_{uj,l} = \left(X_l - \overline{X}_{uj,l}\right)^2 + \left(Y_l - \overline{Y}_{uj,l}\right)^2 \quad ; u = n, m \qquad (4.11b)$$

$\overline{X}_{uj,l}$ and $\overline{Y}_{uj,l}$ indicate the nominal values from the trellis table. n and m represent the two states in the trellis leading to the common state j and $\min\{y_1, y_2\}$ is the minimum function between y_1 and y_2. Due to the differential preprocessing the Euclidean metrics are not more optimal as indicated above. However, at higher signal-to-noise ratios the resulting performance degradation can be neglected, as will be shown by computer simulations.

4.1.2 Algorithm for Roll-Off Factors less than 1.0

If pulse shaping filters with a roll-off factor less than 1.0 are used, the second Nyquist criterion is not fulfilled any more. The impulse response of one channel has nonzero values at sampling times of all preceding and succeeding bits of the other channel

$$g_i \neq 0 \quad ; \quad i = 1, 3, 5, 7, \ldots \qquad (4.12)$$

Therefore, an infinite number of Q-channel bits interfere with a certain I-channel bit and vice versa. Consequently an infinite number of states in the trellis decoder would be needed to resolve the intersymbol interference from one channel into the other channel. To overcome this problem, the impulse response considered in the trellis decoder to calculate the nominal values $\overline{X}_l$ and $\overline{Y}_l$ in the trellis table, is truncated after i bit intervals at the cost of a performance degradation. Table 4.2 illustrates the required number of states in the trellis decoder, when the receiver considers an impulse response of length $2i + 1$ bit intervals. The values of the impulse responses are calculated for the

Samples i	$g_i(r = 0.5)$	$g_i(r = 0.35)$	Number of states in the trellis decoder $2^{2 \cdot i + 1}$
1	0.60021	0.61858	8
3	-0.12004	-0.16244	128
5	0.01715	0.05703	2048
7	0.00572	-0.01382	32768
9	-0.00260	-0.00185	524288

Table 4.2: Values g_i of the impulse response and number of required states in the trellis decoder

roll-off factors $r = 0.5$ and $r = 0.35$ which are used in several communication systems. The table shows that the complexity of the trellis decoder increases dramatically for $i > 3$. Therefore, the considered impulse response is truncated at $i = 3$ in the following investigations. Under this condition, (4.1) simplifies to

$$r_l = [z_l + g_1(z_{l-1} + z_{l+1}) + g_3(z_{l-3} + z_{l+3})] e^{j\phi_l} + n_l' \qquad (4.13)$$

Note that the sampling value g_2 of the impulse response is omitted in (4.13), since due to the Nyquist 1 criterion and the twofold sampling of a symbol, for all even sampling values the impulse response is zero. A roll-off factor $r < 1$ and a truncation of the impulse response after 3 samples ($i = 3$) result in a trellis decoder with 128 states. The input signals X_l and Y_l of the trellis decoder are functions of the transmitted bits y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1} and y_{l+3}

$$X_l = \Re \left\{ r_l \cdot r_{l-2}^* \right\}$$
$$= F \left\{ y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n_l' \right\} \qquad (4.14a)$$

$$Y_l = \Im \left\{ r_l \cdot r_{l-2}^* \right\}$$
$$= F \left\{ y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n_l' \right\} \qquad (4.14b)$$

84

The detailed yet lengthy equations relating the bits x_l and y_l to the nominal signals $\overline{X}_l$ and $\overline{Y}_l$ can be found similarly to (4.9) and (4.10) when the additive noise is omitted. The equations are given in Appendix B. Here transitions from an initial state $(y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1})$ into a final state $(y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3})$ occur. It is observed that the bits x_{l-4} and x_{l+2} do not have any influence on X_l and Y_l since, due to the Nyquist 1 criterion, the impulse response is equal to zero at these specific sampling intervals. Consequently, four transitions always have the same pairs of nominal values in the trellis. However, each of the four transitions represents a part of different trellis paths. Therefore, the bits x_{l-4} and x_{l+2} are important to define the initial and final state of the transition in the trellis diagram. Without these bits the history information of the different paths would be lost and Viterbi decoding is not possible. Both, initial and final state depend on eight bits resulting in a trellis with 256 states. Since lower and upper part of the trellis are symmetric, as has been shown for $r = 1.0$, the number of states can be reduced to 128 (see trellis table in Appendix B).

Data decoding can always be done by using the Viterbi algorithm. Then 256 branch metrics have to be calculated in each step through the trellis. A complexity reduction compared to the Viterbi algorithm can be achieved in the case of trellis decoding if not all paths of the trellis are considered. The number of transitions which have to be computed can then be reduced dramatically. Different algorithms with a reduced number of states are described in the literature. A good solution for the application in a DOQPSK receiver is the selection algorithm, which is commonly known as the M-algorithm [18]. In [3] the algorithm is also referred to as SA(B) (Search Algorithm with B active states) .

The M-algorithm stores merely the M most probable paths through the trellis. Therefore, the computation complexity for the branch metrics in each step is reduced to 2M. The complete operations needed in each time interval are:

- Access the 2M branch metrics associated with the branches emerging from the present M states. Add these branch metrics to the previous path metrics to yield the new path metrics.

- If two of the 2M paths end in the same state, select the path with the smaller accumulated metric as the survivor path, similar to the Viterbi algorithm. Delete the other path by the path list.

- Select the M best paths from the remaining $\leq$ 2M paths.

- Store the M new states and path metrics.

- Decode the data symbol associated with the path with smallest metric and again begin with the first step. Decoding is typically done with a delay of five times the channel memory.

Figure 4.3 illustrates the path build up process. Here a part of a trellis with $S = 10$ states is shown. The number of states used in the M-algorithm is chosen as three.
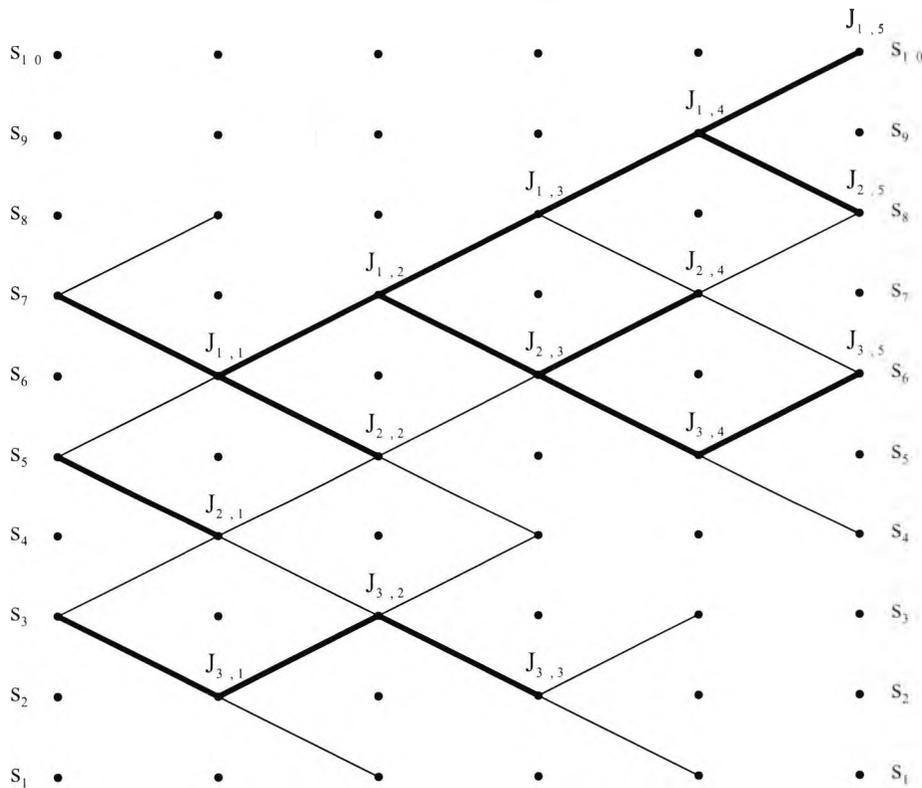
Figure 4.3: Illustration of the M-algorithm with M=3 for a trellis with 10 states and a binary transmission

It is essential to chose the best value for M, since the performance of the decoding process depends on M. In the literature no universal calculation formula for M is given. Generally, the value of M depends on the respective application and has to be determined by simulation. Table 4.3 gives a comparison between the required simulation times, indicated by the time factor, and the achieved bit error rates for an AWGN channel at a signal-to-noise ratio of 10 dB. For all investigations the M-algorithm has been obtained from the original trellis with 128 states, i.e., the filter impulse response in the trellis decoder has been truncated after 3 samples ($i = 3$).

Algorithm	Time factor	BER
128 state Viterbi	1.00	$1.2142 * 10^{-2}$
32 state M-algorithm	0.45	$1.2168 * 10^{-2}$
16 state M-algorithm	0.26	$1.3826 * 10^{-2}$
14 state M-algorithm	0.24	$1.5481 * 10^{-2}$
8 state Viterbi	0.19	$1.6981 * 10^{-2}$

Table 4.3: Comparison between simulation time and bit error rate; AWGN channel with $E_s/N_0 = 10$ dB

The Viterbi algorithm with 8 states, which is adapted to a roll-off factor of 1.0, leads to the shortest simulation time. Assuming $r = 0.35$, the trellis decoder is mismatched and, therefore, results in the highest bit error rate. The M-algorithm with 32 states requires less than half of the simulation time of the Viterbi algorithm, at almost identical bit error rate. Sorting of the M most probable paths from 2M possible paths at any time interval is implemented with the Quicksort algorithm. If the M-algorithm with 16 states is used, only a small degradation compared to the Viterbi algorithm with 128 states is observed. However, a gain in computer simulation time by a factor of 4 is achieved. Therefore, for all further investigations the Viterbi algorithm with 8 (for $r = 1.0$) and 128 (for $r = 0.35$) states and the M-algorithm with 32 and 16 states (for $r = 0.35$) will be used.

4.2 Results on Linear Channels

The different algorithms described above are firstly compared in linear channels. In general, the receiver algorithms for differential encoded Offset-QPSK signals, which include differential demodulation, are not optimal. This is due to the fact, that firstly the distances in the Viterbi algorithm are assumed Gaussian, and secondly for roll-off factors less than 1.0 only a part of the impulse response is considered in the trellis. Therefore, over the whole SNR area the DOQPSK receivers can not achieve the same performance as a $\pi/4$-DQPSK receiver, which also includes differential demodulation. The differential demodulation in a $\pi/4$-DQPSK receiver does not lead to an interference between the I- and the Q-channel. In this case, a symbol-by-symbol detection independent of the roll-off factor is sufficient. Figures 4.4 and 4.5 compare the bit error rates (BER) of $\pi/4$-DQPSK and different DOQPSK receiver algorithms as a function of the ratio of symbol energy-to-noise power density E_s/N_0 for an AWGN channel. The results have been obtained by Monte Carlo simulation with the aid of the simulation tool COSSAP. The curves in Fig. 4.4 show that the receiver algorithms for DOQPSK
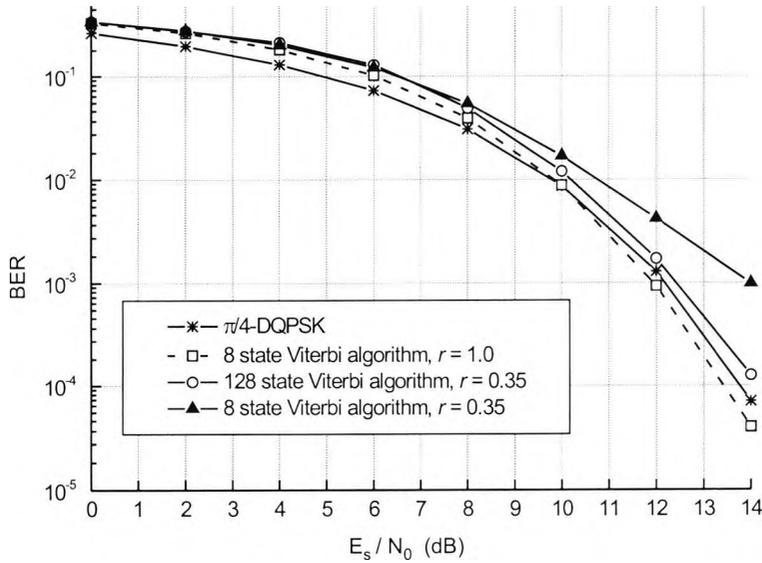
Figure 4.4: $\pi/4$-DPSK and DOQPSK on an AWGN channel

lead to a degradation compared to $\pi/4$-DQPSK for error rates higher than 10^{-2}. The losses for smaller values of E_s/N_0 are a result of the implemented non-optimal Euclidean metrics. However, in the range of interest the curves for the Viterbi algorithm with 8 states for a roll-off factor of 1.0 and the algorithm with 128 states for $r = 0.35$ are comparable to that of $\pi/4$-DQPSK. The Viterbi algorithm with 8 states, which

is matched to $r = 1.0$, shows a noticeable performance degradation when filters with $r = 0.35$ are used. It can be concluded, that the above simplifications in the metric calculation and in the reduced impulse response do not have a major effect on the BER performance. However, to completely neglect the influence of the roll-off factor is still not possible.

In the following the performance of the M-algorithm will be investigated. In Fig. 4.5 the results for different M-algorithms are shown. The lower and upper bounds are given by the curves of $\pi/4$-DQPSK and by the 8 state Viterbi for $r = 0.35$. The M-algorithm with 32 states achieves the same performance as the Viterbi algorithm with 128 states although only the half simulation time is needed. The performance loss due to a further reduction to 16 states in the M-algorithm is negligible. On the other hand, with 16 states a gain in simulation time of a factor of four is achieved (see Table 4.3) compared to the Viterbi algorithm with 128 states.
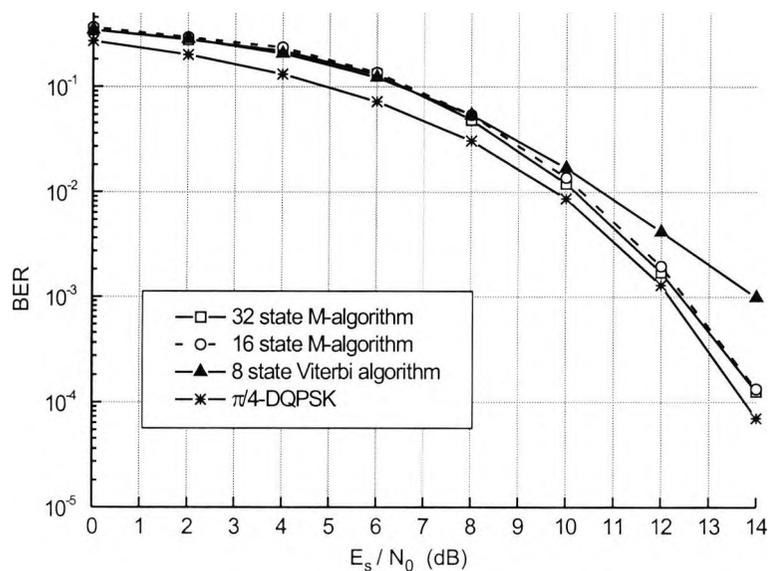
Figure 4.5: M-algorithm on an AWGN channel ($r = 0.35$)

As mentioned in the basics chapter, the AWGN channel represents the best achievable transmission channel, while the Rayleigh fading channel can be regarded as the worst case channel. In a narrowband mobile radio communication system, Rayleigh fading is often a good approximation for real channels due to the small product of symbol duration times coherence bandwidth of the channel. Results for such a type of channel are illustrated in Fig. 4.6. The product of maximum Doppler frequency f_d and symbol duration T_s is chosen to be 0.0046 and 0.0093. At 450 MHz and a data rate of 36 kbit/s (18 ksym/s), which are the parameters of the TETRA system, the

two values corresponds to a vehicle speed of 200 and 400 km/h. As expected, the good results of DOQPSK in the AWGN channel are also confirmed in the Rayleigh fading channel. For a roll-off factor of 1.0 and $f_d * T_s = 0.0093$ the degradation compared to $\pi/4$-DQPSK is negligible in the BER range of interest. The M-algorithm with 32 states and $r = 0.35$ shows only a small degradation over the full range of E_s/N_0. Note that
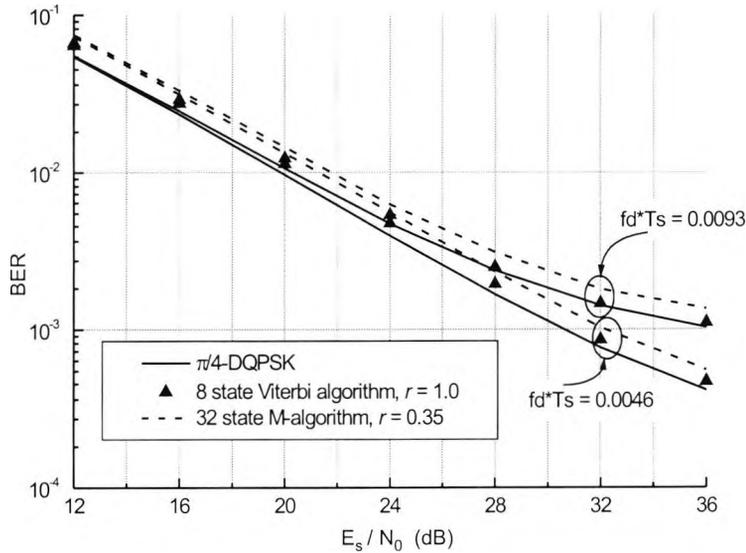
Figure 4.6: $\pi/4$-DPSK and DOQPSK on a Rayleigh fading channel

in contrast to the AWGN channel, on the Rayleigh fading channel the results obtained with the 8 state Viterbi and $r = 0.35$ are nearly the same as those of the M-algorithm. This is due to the fact that the high error rates during the fading events (small values of E_s/N_0) dominate the bit error results and, therefore, the smaller error rate of the M-algorithm in the area of higher E_s/N_0 values (Fig. 4.5) is of no significance.

4.3 Results on Nonlinear Channels

A major motivation for the implementation of linear modulation formats with reduced envelope fluctuations is the cost reduction for linearisation of nonlinear amplifier amplitude transfer curves. In Chapter 3 the behaviour of the two modulation schemes $\pi/4$-DQPSK and DOQPSK concerning the out-of-band power and the spectra of the transmitted signals are considered for two types of nonlinear amplifiers. In this section the bit error performance of the DOQPSK receiver using differential demodulation on nonlinear channels is investigated.

Weak Conduction Region

The investigations on the behaviour in the weak conduction region are carried out with the amplifier defined in Fig. 3.3a. Figure 4.7 compares both modulation schemes DOQPSK and $\pi/4$-DQPSK at a bit error rate of 10^{-3} under relative adjacent channel power P_{AC} and without noise for a threshold parameter a of 0.3. P_{AC} defines the relative difference of the average powers of the adjacent channel to the information bearing channel, i.e., a P_{AC} of 10 dB means that the average power of the adjacent channel is 10 dB above the average power of the information bearing channel. The two adjacent channels are of the same type as the original signal. Channel spacing is $f_s(1 + r)$, i.e., there is no overlap of the different spectra without the nonlinear amplifier. The simulation results in Fig. 4.7a are obtained for adjacent channels which
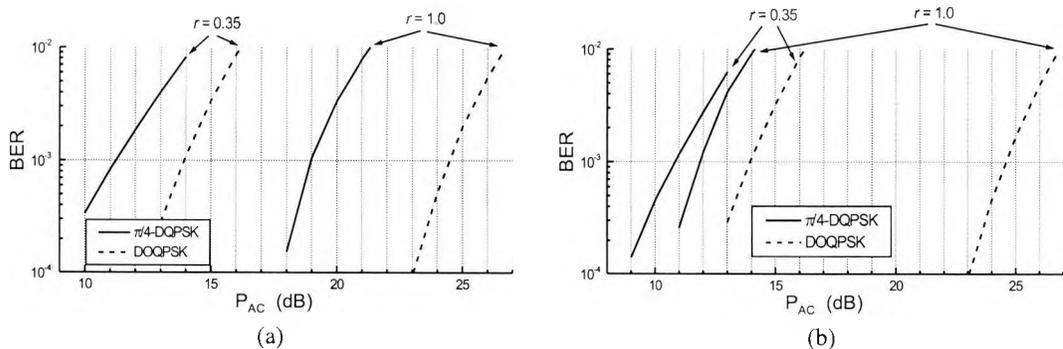
Figure 4.7: BER performance under ACI without noise; parameter $a = 0.3$ (a) synchronised, (b) shift by half a symbol period

are synchronised to the measured channel. In this case the use of DOQPSK results in a gain against $\pi/4$-DQPSK of more than 5.5 dB for a roll-off factor of 1.0 and about 3.0 dB for $r = 0.35$. If the assumption of synchronisation is not fulfilled, e.g., if the

adjacent channels are shifted by half a symbol period in the worst case (Fig. 4.7b), $\pi/4$-DQPSK shows a distinct degradation, especially for a higher roll-off factor. The values of DOQPSK remain almost stable.

Figure 4.8 illustrates the BER performance for two smaller values of the threshold parameter a. If the value of the threshold parameter is reduced, the absolute value of
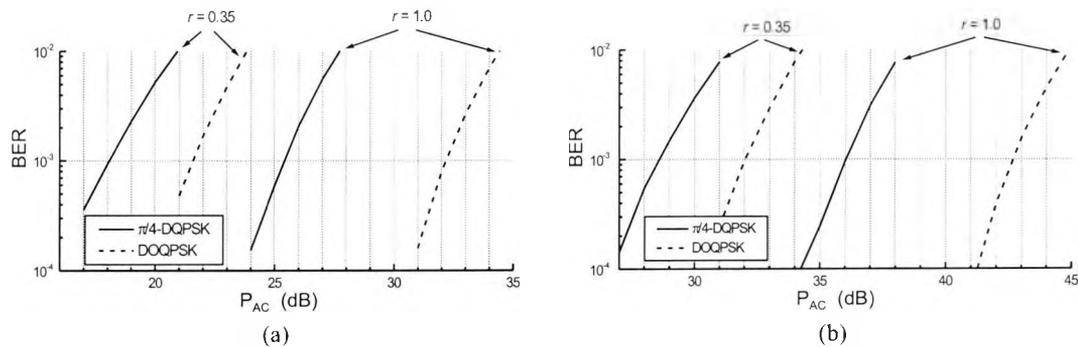
Figure 4.8: BER performance under ACI without noise (a) parameter $a = 0.15$, (b) $a = 0.05$

P_{AC} at the same bit error rate increases. However, the gain of DOQPSK is independent of the chosen threshold value a. From another point of view DOQPSK can operate at a higher threshold value a to obtain the same error rate as $\pi/4$-DQPSK. Thus, the effort to linearise the amplifier characteristic is reduced, when DOQPSK is implemented.

Figure 4.9 gives the bit error performance under a fixed relative adjacent channel power of 10 dB in an AWGN environment. In the simulations the nonlinear amplifier characteristic with a parameter a of 0.3 is used. A P_{AC} of 10 dB has been chosen in
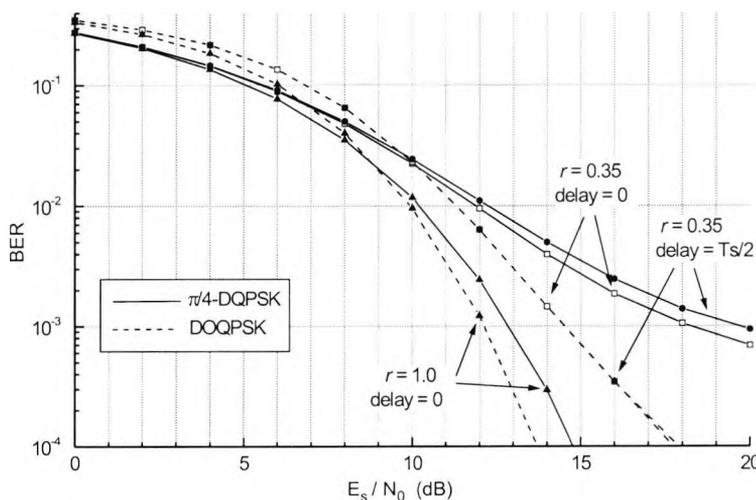
Figure 4.9: BER performance under ACI with $P_{AC} = 10$ dB and parameter $a = 0.3$ on an AWGN channel

92

Fig. 4.9, since it follows from Fig. 4.7, that a BER of less than 10^{-3} can be obtained for both modulation formats in the noise free case. Consequently, higher values of P_{AC} would shift the BER floor in Fig. 4.9 to higher bit error rate values. As expected, the gain of DOQPSK under nonlinear amplification is confirmed on an AWGN channel. If an error rate of 10^{-3} is considered, DOQPSK achieves a gain of 3.7 dB in the case of $r = 0.35$ and synchronised adjacent channels, and a gain of 5.1 dB, if the adjacent channels are shifted by $T_s/2$. The simulation results for a roll-off factor of 1.0 show in the synchronised case only a small gain of 1.0 dB. If a delay of $T_s/2$ is used (not included in the figure), DOQPSK has nearly the same performance as for zero delay, while the BER results of $\pi/4$-DQPSK decrease to the values of $r = 0.35$.

Figure 4.10 compares the algorithms in the time selective Rayleigh fading environment. The two bottom curves represent the results from Fig. 4.6. In the linear case DOQPSK shows a small degradation compared to $\pi/4$-DQPSK. If the nonlinear ampli-
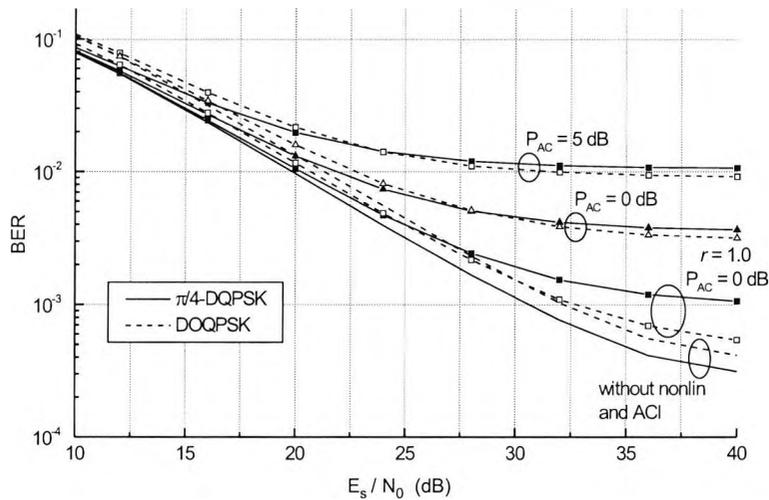
Figure 4.10: BER performance under ACI, $r = 0.35$, $a = 0.3$, and $f_d T_s = 0.0046$ on a Rayleigh fading channel

fier is included in the simulation, the behaviour is changed. Depending on the chosen parameters r and the relative power of the adjacent channels, a performance gain of DOQPSK is obtained. In a Rayleigh fading environment and a roll-off factor less than 1.0 the time shift of the adjacent channels has no significant influence on the BER .

Saturation Region

The investigations on amplitude distribution and out-of-band power in Chapter 3 indicate that in the case of saturation, differences in the error rate performance of both

modulation schemes are hardly to be expected, when a roll-off factor of 0.35 is chosen. In Fig. 4.11 this assumption is confirmed by the BER results. In contrast to an improvement of nearly 10 dB, when the edge-of-saturation parameter of the generic amplifier model is $c = 1.0$ and the roll-off factor is 1.0, only a gain of 1 dB is obtained, when $r = 0.35$. For $c = 1.3$ (Fig. 4.11b) the influence of the amplitude nonlineartiy on
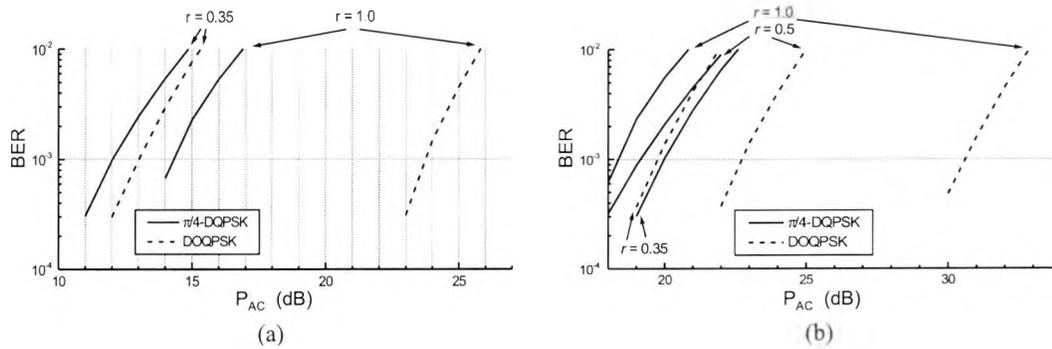
Figure 4.11: BER performance under ACI without noise using amplifier model described in Fig. 3.3b (a) parameter $c = 1.0$, (b) $c = 1.3$

the error performance is even in favour for $\pi/4$-DQPSK when r is small. As mentioned earlier, if a roll-off factor of 0.35 is used, DOQPSK has a higher peak value of the amplitude compared to $\pi/4$-DQPSK. The higher peak amplitude of DOQPSK may result (e.g. for $c = 1.3$) in a performance loss against $\pi/4$-DQPSK. However, the curves in Fig. 4.11b also shows that for larger roll-off factors the use of DOQPSK results in a gain even if the edge-of-saturation parameter c is increased. In contrast to the results in the weak conduction region a time shift of the adjacent channels has merely a negligible influence on the performance.

In Fig. 4.12 the results on an AWGN channel with $P_{AC} = 10$ dB and a parameter
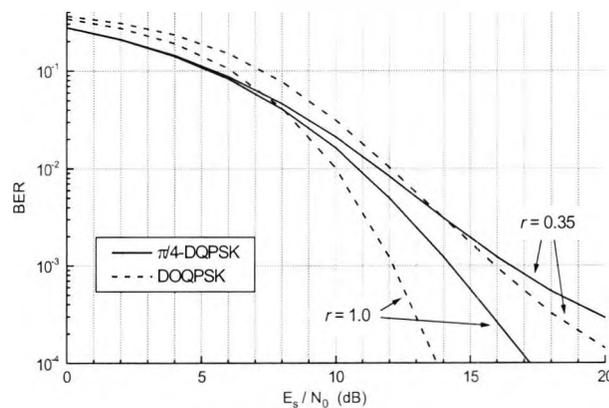
Figure 4.12: BER performance under ACI with $P_{AC} = 10$ dB and parameter $c = 1.0$ on an AWGN channel

94

$c = 1.0$ are given. The upper two curves illustrate that for $r = 0.35$, a common roll-off factor in communication systems, both modulation schemes have nearly the same error rate performance. Simulations in Rayleigh fading environments also yield no significant performance differences.

In addition to the investigation with the generic amplifier model, simulations with the ETSI model have been carried out. Table 4.4 list the E_s/N_0 values which are needed to achieve a given bit error rate. The results were obtained both on a linear AWGN

linear channel				
		E_s/N_0 for a bit error rate of		
algorithm	roll-off	$1 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
$\pi/4$-DQPSK	0.5	9.9	10.7	12.2
DOQPSK	0.5	10.0	10.8	12.2
$\pi/4$-DQPSK	0.35	9.9	10.7	12.2
DOQPSK	0.35	10.2	11.0	12.4
nonlinear channel $P_{AC} = 10$ dB, signal power $= 6$				
		E_s/N_0 for a bit error rate of		
algorithm	roll-off	$1 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
$\pi/4$-DQPSK	0.5	12.4	14.3	21.5
DOQPSK	0.5	11.6	12.6	14.8
$\pi/4$-DQPSK	0.35	12.4	14.2	19.2
DOQPSK	0.35	13.0	14.6	19.1
loss in bit error performance				
		E_s/N_0 for a bit error rate of		
algorithm	roll-off	$1 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
$\pi/4$-DQPSK	0.5	2.5	3.6	9.3
DOQPSK	0.5	1.5	1.8	2.6
$\pi/4$-DQPSK	0.35	2.5	3.5	7.1
DOQPSK	0.35	2.8	3.6	6.7

Table 4.4: BER performance under ACI using the ETSI amplifier model

channel and on a nonlinear AWGN channel with $P_{AC} = 10$ dB. The signal power is chosen in such a way, that the average amplitude corresponds to the edge-of-saturation value of the amplifier transfer curves (see Fig. 3.4). The third part of the table provides the loss in bit error performance, i.e., it gives the figures by which value E_s/N_0 has to be increased, such that the same bit error rate can be achieved as on a linear channel. As it was to be expected no significant differences between both modulation schemes can be observed, if filters with a roll-off factor of 0.35 are used. However, for $r = 0.5$ DOQPSK yields a gain of 6.7 dB at a bit error rate of $1 \cdot 10^{-3}$. This is due to the fact that DOQPSK shows less amplitude fluctuations and thus only loses 2.6 dB in the

nonlinear channel compared to the linear channel, while $\pi/4$-DQPSK loses more than 9 dB.

It can be summarised that both modulation schemes show no great differences in the BER on linear channels, when differential demodulation in the receiver is applied. In nonlinear channels, and depending on the roll-off factor, DOQPSK has advantages due to the smaller fluctuations of the envelope and, therefore, may achieve gains of up to 6 dB compared to $\pi/4$-DQPSK. A further advantage of DOQPSK is that a time shift of the adjacent channels can be neglected, while the same time shift has a strong influence on the performance of $\pi/4$-DQPSK.

Chapter 5

Coherent Demodulation and Equalisation of DOQPSK-Signals with Per-Survivor-Processing

Without modification, the receiver type described in the last chapter is only suitable for purely time selective channels, since the differential processing does not consider any intersymbol interference introduced through a frequency selective channel. However, mobile radio channels are often both time and frequency selective (see Section 2.3). If the frequency selectivity is already considered in the synthesis of the receiver for differentially encoded OQPSK, a new structure using coherent demodulation and equalisation is obtained. In Section 2.2, both optimum and suboptimum receivers were introduced that compensate for ISI in the transmission of digital information through bandlimited, non-ideal channels. The optimum receiver employed maximum likelihood sequence estimation for detecting the information sequence from the samples of the matched filter, while the suboptimum receivers employed either a linear or a nonlinear equaliser. In the description of the three equalisation methods, it was assumed that the channel impulse responses were known at the receiver. However, in most communication systems that use equalisers, the channel characteristic are unknown a priori and, in many cases, the channel response is time-variant. In such a case, data transmission results both in a data sequence and channel estimation problem at the receiving side.

It is sufficiently known from the literature that the common maximum likelihood estimation of the data z and of the channel impulse response h can not be carried out recursively. An optimum decoding requires a decoding of the entire message. However, the algorithm for data decoding and channel estimation can be split up into two steps

$$[z_o, h_o] = \arg \left[\min_h \left[\min_z \sum_{l=0}^{N} \left| r_l - \sum_{i=0}^{L} h_i \cdot z_{l-i} \right|^2 \right] \right] \tag{5.1}$$

where z_o and h_o are the optimal choices of z and h. The algorithm is optimal in

97

slowly fading channels with respect to the maximum likelihood criterion. However, it is not recursive and has a high implementation complexity. Therefore, quite often equalisation and decoding are carried out suboptimally in two separate steps leading to a considerable performance degradation of the overall system.

Figure 5.1 shows a sequential realisation of a receiver. In this case, linear transversal filters are often used for channel equalisation or a decision feedback equaliser is implemented. Both algorithms require a training sequence for acquisition, where the adaption is done with conventional algorithms, e.g. LMS (least mean square) algorithm. The performance of the receiver can be increased through a transfer of 'soft'
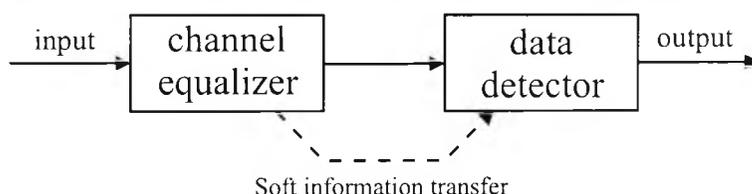
Figure 5.1: Sequential channel estimation and data decoding

information between both algorithms (see dashed line in Fig. 5.1). In contrast to a 'hard' information, which is only binary (good, bad channel), a 'soft' information gives a more detailed measure about the quality of the channel. However, a considerable loss compared to optimal ML decoding may still occur depending on the channel impulse response.

A further method is parallel channel estimation and decoding with data aided feedback from the Viterbi decoder to the channel estimator, as shown in Fig. 5.2. The channel estimator is identical in structure to the adaptive linear transversal equaliser in the sequential realisation of the receiver. Though the given receiver can be implemented recursively it is, however, not optimal. A further problem is the error propaga-
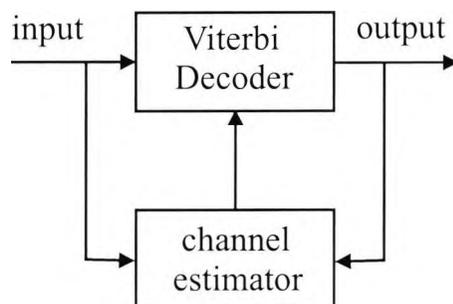
Figure 5.2: Data aided channel estimation

tion which is still increased by the decoding delay in the Viterbi decoder. For startup

operation, the algorithm requires a training sequence to perform the initial adjustment of the filter tap coefficients. In an adaptive mode of operation, the receiver simply uses its own decisions of the Viterbi decoder to form an error signal.

In contrast to the briefly mentioned algorithms, the investigation here emphasise a new coherent receiver structure for DOQPSK algorithms, which do not require a training sequence but nevertheless lead to a near optimum BER performance.

5.1   Receiver Algorithms

The optimum decoding and channel estimation algorithm (5.1) could not be implemented in a mobile radio receiver for complexity reasons. In the following, realisable, suboptimum structures will be considered which are derived from the optimum algorithm.

The classical, conventional, suboptimum MLSE receiver initially performs a matched filtering that considers the channel impulse response and the impulse response of the transmission filter, followed by a sampling device with decorrelation and a Viterbi decoder. Figure 5.3 shows the overall communications system employing a conventional MLSE receiver. The unknown parameters of the different blocks of the receiver can
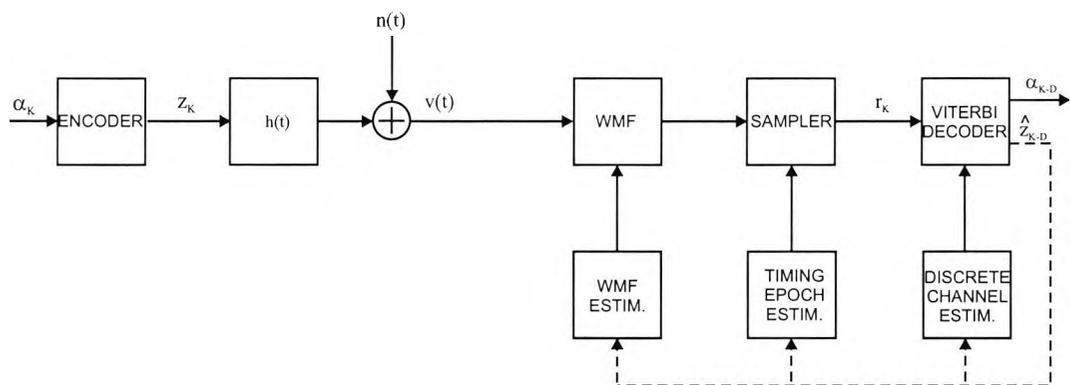
Figure 5.3: Conventional MLSE with unknown channel parameters

be estimated from the decoded, partly erroneous, and delayed data which results in a suboptimum implementation.

An alternative to the classical approach is per-survivor processing (PSP). PSP [27] combines a group of algorithms which implement the data aided structure within the Viterbi algorithm itself. Per-survivor processing uses the code sequence, that belongs

to the survivor path to estimate the unknown parameters without delay. To apply PSP in a DOQPSK receiver, a specific implementation of PSP for joint data and channel estimation using a blind trellis search technique [29] is developed.

The block diagram of the receiver, which combines a Viterbi algorithm and discrete channel estimation within the trellis decoder is given in Fig. 5.4. For reasons of a simplified receiver implementation, the standard channel matched filters with subsequent decorrelation device, which have to be adapted to the time varying channel, are replaced by a suboptimum Nyquist 1 root raised cosine roll-off filters. In comparison
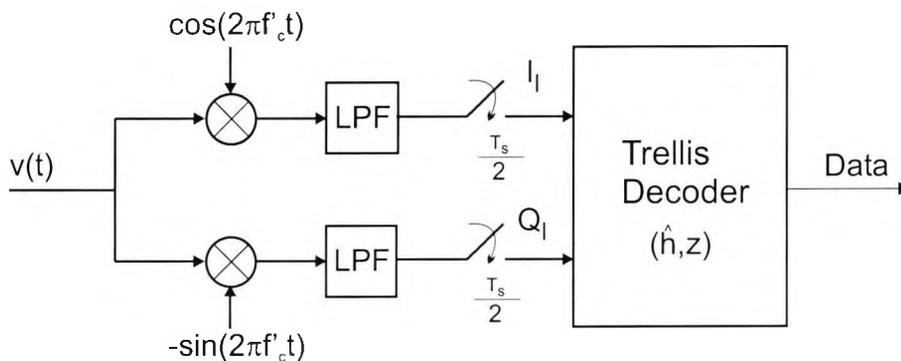
Figure 5.4: DOQPSK receiver model (coherent demodulation)

to the non-coherent receiver of Chapter 4, in the coherent receiver of Fig. 5.4 no differential preprocessing is employed before trellis decoding. The twofold sampling per symbol considers the special feature of Offset-QPSK signals.

For the subsequent investigations it is useful to define a vector $\mathbf{z}$ that contains the filtered complex data symbols z. A Nyquist filtered OQPSK signal with a roll-off factor of 1.0 can then be described as

$$\mathbf{z}_l^{\mathrm{T}} = [\tilde{z}_l, \ldots, \tilde{z}_{l-N}] \tag{5.2}$$

$\mathbf{z}^{\mathrm{T}}$ is the transpose of $\mathbf{z}$, and

$$\tilde{z}_l = [0.5 \cdot z_{l-1} + z_l + 0.5 \cdot z_{l+1}] \tag{5.3}$$

where $z_l = x_l + \mathrm{j}\, y_l$ is the complex data symbol.

100

5.1.1 Blind Equalisation Based on the Maximum-Likelihood Criterion

It is convenient to use the equivalent, discrete-time channel model described in the basics chapter. Recall that the received and filtered signal r at time instant l is

$$r_l = I_l + \mathrm{j}\, Q_l = \sum_{i=0}^{L} h_i \tilde{z}_{l-i} + n'_l \tag{5.4}$$

where h_i are the samples of the equivalent discrete-time channel coefficients, $\tilde{z}$ represents the filtered samples of the information sequence, and n'_l is the filtered noise signal. In vector notation the received signal is determined as

$$r_l = \mathbf{h}_l^{\mathrm{T}} \cdot \mathbf{z}_l + n'_l \tag{5.5}$$

The vector $\mathbf{h}$ represents the $L+1$ taps of the impulse response of the time variant and frequency selective channel

$$\mathbf{h}_l^{\mathrm{T}} = [h_l(0), \dots, h_l(L)] \tag{5.6}$$

For a block of N received data samples, the joint probability density function of the received data vector $\mathbf{r}_l^{\mathrm{T}} = [r_l, r_{l-1}, \dots, r_{l-N-1}]$ conditioned on the impulse response vector $\mathbf{h}_l^{\mathrm{T}} = [h_l(0), \dots, h_l(L)]$ and the data vector $\mathbf{z}_l^{\mathrm{T}} = [\tilde{z}_l, \dots, \tilde{z}_{l-N-1}]$ is

$$p(\mathbf{r}|\mathbf{h}, \mathbf{z}) = \frac{1}{(2\pi\sigma^2)^N} \, \mathrm{e}^{\left(-\frac{1}{2\sigma^2} \sum_{n=l}^{(l-N-1)} \left| r_n - \sum_{i=0}^{L} h_i \tilde{z}_{n-i} \right|^2 \right)} \tag{5.7}$$

The joint maximum-likelihood estimates of $\mathbf{h}$ and $\mathbf{z}$ are the values of these vectors that maximise the joint probability density function $p(\mathbf{r}|\mathbf{h}, \mathbf{z})$ or, equivalently, the values of $\mathbf{h}$ and $\mathbf{z}$ that minimise the term in the exponent. Hence, the ML solution is simply the minimum over $\mathbf{h}$ and $\mathbf{z}$ of the metric

$$J'(\mathbf{z}, \mathbf{h}) = \sum_{n=l}^{(l-N-1)} \left| r_n - \sum_{i=0}^{L} h_i \tilde{z}_{n-i} \right|^2 \tag{5.8}$$

$$= ||\mathbf{r} - \mathbf{Z}\mathbf{h}||^2$$

where the matrix $\mathbf{Z}$ is called the data matrix and is defined as

$$\mathbf{Z} = \begin{bmatrix} \tilde{z}_l & \tilde{z}_{l-1} & \tilde{z}_{l-2} & \cdots & \tilde{z}_{l-L} \\ \tilde{z}_{l-1} & \tilde{z}_{l-2} & \tilde{z}_{l-3} & \cdots & \tilde{z}_{l-1-L} \\ \tilde{z}_{l-2} & \tilde{z}_{l-3} & \tilde{z}_{l-4} & \cdots & \tilde{z}_{l-2-L} \\ \vdots & \vdots & \vdots & & \vdots \\ \tilde{z}_{l-(N-1)+L} & \tilde{z}_{l-(N-1)+L-1} & \tilde{z}_{l-(N-1)+L-2} & \cdots & \tilde{z}_{l-(N-1)+L-L} \end{bmatrix} \tag{5.9}$$

Two observations can be made. First of all, when the data vector $\mathbf{z}$ (or the data matrix $\mathbf{Z}$) is known, as, it is the case when a training sequence is available at the receiver, the ML channel impulse response estimate is obtained by minimising (5.8) over $\mathbf{h}$. On the other hand, when the channel impulse response $\mathbf{h}$ is known, the optimum ML detector for the data sequence $\mathbf{z}$ performs a trellis search by utilising the Viterbi algorithm for the ISI channel.

When neither $\mathbf{z}$ nor $\mathbf{h}$ are known, the minimisation of $J'(\mathbf{z}, \mathbf{h})$ may be performed jointly over $\mathbf{z}$ and $\mathbf{h}$. The next sections deal with different adaptive MLSE algorithms based on PSP for differential encoded OQPSK signals which do not require a training sequence for joint channel and data estimation.

5.1.2 Matrix Inversion

In [4] an algorithm is presented which avoids the time delay of the channel estimate in the data aided structure. Instead of producing the channel estimate based on the output bits of the Viterbi decoder, as in Fig. 5.2, the estimate is based on the data symbols corresponding to the most probable transmitted data sequence at the current time nT. Thus, the estimation is based on tentative symbols that have not been decided by the Viterbi algorithm but, on the other hand, no time delay results. Figure 5.5 shows a part of a trellis illustrating the principle. At time nT the most probable path is the
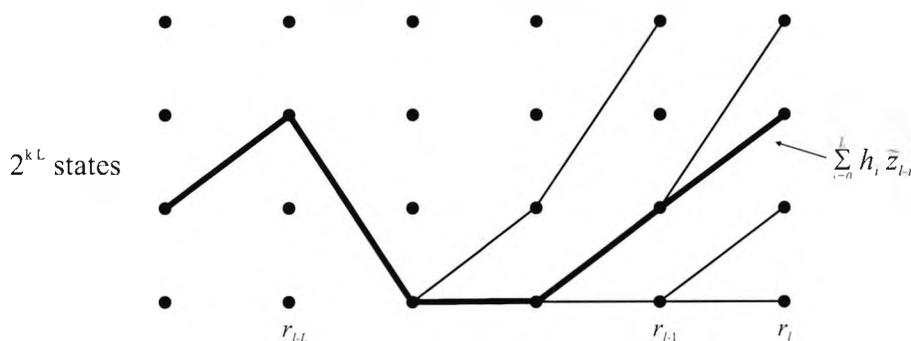
Figure 5.5: Illustration of a Viterbi algorithm detector

one with least accumulated metric. This path is represented by the bold line in the figure. Note, that the last two symbols may not yet be decoded optimally. However, the algorithm is based on the assumption, that the most probable path corresponds to the transmitted sequence and the channel may be estimated based on the symbols corresponding to that specific path.

102

In the last section the problem of joint maximum-likelihood estimation of the channel and the data was addressed when neither $\mathbf{z}$ nor $\mathbf{h}$ are known. The matrix inversion algorithm uses the most probable path in the trellis to define a data matrix $\mathbf{Z}$. In contrast to the matrix given in (5.9) the employed matrix is a square matrix of order $L + 1$. Based on the L last received samples of the most probable path, the following system of equations may be set up

$$r_l = \sum_{i=0}^{L} h_i \tilde{z}_{l-i} + n'_l$$

$$r_{l-1} = \sum_{i=0}^{L} h_i \tilde{z}_{l-1-i} + n'_{l-1}$$

$$\vdots$$

$$r_{l-L} = \sum_{i=0}^{L} h_i \tilde{z}_{l-L-i} + n'_{l-L}$$

(5.10)

In matrix form the received signal may be written as

$$\mathbf{r} = \mathbf{Z}\mathbf{h} + \mathbf{n}'$$

(5.11)

With the aid of the above equation, the vector $\hat{\mathbf{h}}$ of the channel estimates can be calculated from the most probable path. Given the vector of the received signal

$$\mathbf{r}_l^{\mathrm{T}} = [r_l, \cdots, r_{l-L}]$$

(5.12)

which contains $L + 1$ complex samples, and the data matrix estimate

$$\hat{\mathbf{Z}} = \begin{bmatrix} \hat{z}_l & \hat{z}_{l-1} & \hat{z}_{l-2} & \cdots & \hat{z}_{l-L} \\ \hat{z}_{l-1} & \hat{z}_{l-2} & \hat{z}_{l-3} & \cdots & \hat{z}_{l-1-L} \\ \hat{z}_{l-2} & \hat{z}_{l-3} & \hat{z}_{l-4} & \cdots & \hat{z}_{l-2-L} \\ \vdots & \vdots & \vdots & & \vdots \\ \hat{z}_{l-L} & \hat{z}_{l-L-1} & \hat{z}_{l-L-2} & \cdots & \hat{z}_{l-2L} \end{bmatrix}$$

(5.13)

obtained from the corresponding data of the most probable path, an inversion of $\hat{\mathbf{Z}}$ is used to result in an estimate $\hat{\mathbf{h}}$ of the channel

$$\hat{\mathbf{h}} = \hat{\mathbf{Z}}^{-1} \cdot \mathbf{r}$$

(5.14)

In order to suppress noise and interference, $\hat{\mathbf{h}}$ can be averaged over several estimation intervals.

In conclusion the algorithm can be described as follows: Using the Viterbi algorithm as usual the metrics for the survival paths are found based on the previous channel

estimate. The minimum metric state determines the most probable path and from this path $\hat{\mathbf{Z}}$ is given. The inversion of $\hat{\mathbf{Z}}$ is used to calculate $\hat{\mathbf{h}} = \hat{\mathbf{Z}}^{-1} \cdot \mathbf{r}$. The result is saved so that the mean value over several intervals could be determined.

The approach leads to several problems and has one major drawback. The computation based on only one (most probable) path relies on a tentative data decision which may be wrong and, therefore, results in a BER degradation. This can be avoided by computing more than one channel impulse response, e.g., one per survivor path. However, this leads to another problem. The computation of the inverse of a matrix will result in a high receiver complexity and with every additional estimate of a channel impulse response the complexity will increase strongly. The third problem of this approach is the major drawback and also the reason why this approach is not further pursued. The matrix $\hat{\mathbf{Z}}$ may become singular, i.e., the inverse $\hat{\mathbf{Z}}^{-1}$ does not exist. Therefore, the algorithm will not be applied in its original form, but will be modified to become the hybrid algorithm. The hybrid algorithm is partly developed from the LMS algorithm, which is investigated next.

5.1.3 LMS Algorithm

In the next three sections adaptive algorithms will be presented which are based on the principle of joint data and channel estimation using per-survivor processing. That means, that the estimates $\hat{\mathbf{h}}$ and the data values $\mathbf{z}$ are obtained with PSP directly from the survivor path to a state in the trellis decoder. This is indicated in Fig. 5.6. In
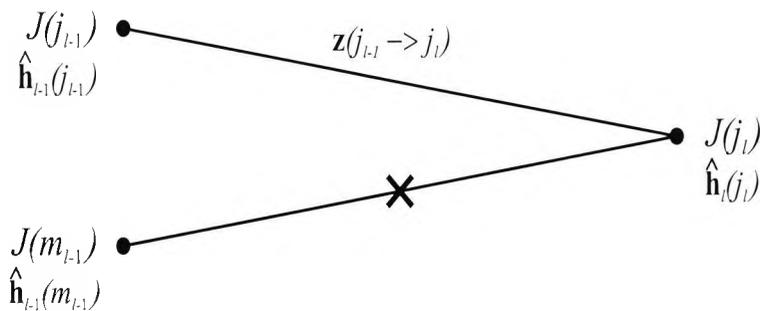
Figure 5.6: Segment of a trellis with per-survivor processing and metrics $J()$

order to avoid double indexing of the time instant l both for the channel variable and the state (see $\hat{\mathbf{h}}$ in the figure), the following assumptions are made which are valid for the investigations in this chapter:

104

- Generally the time is indicated as an index of the variable.

- If the variable is combined with a specific state in the trellis decoder and state and variable have the same time index, the time index is only indicated with the state.

- Transitions are generally characterised by state transitions.

The first step of data decoding is common to the three algorithms. Data decoding in the algorithms will be done by using the Viterbi algorithm. The accumulated metric J of the state j at a time instant l is generally calculated from the quadratic norm. Selection of the minimum metric gives

$$J(j_l) = \min \left[J(j_{l-1}) + \left| r_l - \hat{\mathbf{h}}^{\mathrm{T}}(j_{l-1}) \cdot \mathbf{z}(j_{l-1} \to j_l) \right|^2 ; \right.$$
$$\left. J(m_{l-1}) + \left| r_l - \hat{\mathbf{h}}^{\mathrm{T}}(j_{m-1}) \cdot \mathbf{z}(m_{l-1} \to j_l) \right|^2 \right] \tag{5.15}$$

Having calculated the path with the lowest accumulated metric into each state, channel updating can be done. If the trellis has N states, N channel estimates have to be computed. At this point the three algorithms diverge, since different adaptive channel tracking is employed.

The first algorithm which will be considered is the well known LMS (least mean square) algorithm. The LMS algorithm is an important member of the family of stochastic gradient-based algorithms. A significant feature of the LMS algorithm is its simplicity. The algorithm does not require measurements of the pertinent correlation functions, nor does it use matrix inversion. Indeed, it is the simplicity of the LMS algorithm that has made it the standard against other adaptive algorithms. The following description of the LMS algorithm will be limited to the important facts. The precise derivation can be taken from [17].

The updating of the channel estimate is done by a recursive relation

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} - \Delta_{LMS} \cdot \frac{\delta \hat{F}_{MSE}}{\delta \hat{\mathbf{h}}_{l-1}^{\mathrm{H}}} \tag{5.16}$$

Instead of the ideal objective function F_{MSE} which includes the expected values, the LMS algorithm employs an instantaneous estimate $\hat{F}_{MSE}$ that is based on sample values of the received signal and the data values $\mathbf{z}$ obtained directly from the survivor path in the trellis

$$\hat{F}_{MSE} = |e_l|^2 = \left| r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l \right|^2 \tag{5.17}$$

105

In contrast to the description in the literature and the Chapter 2, here the LMS algorithm is part of the PSP where the tap inputs are represented by the decoded data $\mathbf{z}$ and the desired signal in the algorithm is a sample of the received signal r.

Inserting of (5.17) into (5.16) yields the basic equation of the LMS algorithm

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} + \Delta_{LMS} \cdot e_l \cdot \mathbf{z}_l^* \tag{5.18}$$

with the estimation error

$$e_l = r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l \tag{5.19}$$

The important parameter which influences the performance of the adaptation process is the step-size parameter Δ_{LMS}. Assuming a stationary environment the step-size has to guarantee two forms of convergence:

- Convergence in the mean, which means that
$$E[\hat{\mathbf{h}}_l] \to \mathbf{h}_o \text{ as } l \to \infty$$
where $\mathbf{h}_o$ is the optimum (Wiener) solution.

- Convergence in the mean square, which means that
$$F_{MSE,l} \to F_{MSE,\infty} \text{ as } l \to \infty$$
where $F_{MSE,\infty}$ is always greater than the minimum mean-squared error $F_{MSE,min}$ that corresponds to the Wiener solution, because $F_{MSE,\infty}$ is obtained through a gradient algorithm.

For this two forms of convergence to hold, the step-size parameter Δ_{LMS} has to satisfy different conditions related to the eigenvalues of the correlation matrix of the tap inputs, i.e., of the data.

The first bound and the condition of stability is given by

$$0 < \Delta_{LMS} < \frac{2}{\lambda_{\max}} \tag{5.20}$$

where $\lambda_{\max}$ is the largest eigenvalue of the correlation matrix $\mathbf{R}_{zz}$. If the step-size Δ_{LMS} is set within the bounds defined by (5.20), then the mean of the channel estimate $\hat{\mathbf{h}}_l$ computed by using the LMS algorithm converges to the optimum solution $\mathbf{h}_o$ as the number of iterations approaches infinity. Under this condition, the LMS algorithm is stable and convergent in the mean.

If the step-size Δ_{LMS} is small compared to $2/\lambda_{\max}$, i.e., convergence in the mean is well satisfied, a second criterion has to be fulfilled. The recursive algorithm in (5.18) for adjusting the estimate of the channel coefficients employs unbiased noisy estimates of the received signal. The noise in the estimates causes random fluctuations in the coefficients about their optimal values and, thus, leads to an increase in MSE at the output. That is, the final MSE is $F_{MSE} = F_{MSE,min} + F_{MSE,\Delta}$, where the term $F_{MSE,\Delta}$ due to the estimation noise is named excess mean-squared error. It is desirable to have $F_{MSE,\Delta} < F_{MSE,min}$. This goal can be achieved if Δ_{LMS} will be selected such that

$$\Delta_{LMS} < \frac{2}{\sum_{i=1}^{L+1} \lambda_i} \tag{5.21}$$

If the relation from matrix algebra is used that the sum of the eigenvalues of a positive definite matrix equals the sum of the elements on its main diagonal, then

$$\Delta_{LMS} < \frac{2}{(L+1) \cdot \rho(0)} \tag{5.22}$$

where the main diagonal elements are equal to $\rho(0)$, which itself is equal to the mean-square value of the input at each of the $L+1$ taps.

The analysis of the LMS algorithm presented previously has been limited to a stationary environment. Now, the ability of the LMS algorithms to operate in a non-stationary environment will be briefly considered. In a slowly time-variant environment the minimum MSE and the optimum coefficient vector are also time-variant. The LMS algorithm attempts to follow the moving minimum $F_{MSE,min}$, but it is always lagging behind due to its use of estimated gradient vectors. As a consequence, another form of error is obtained, called the lag error. The total MSE error can now be expressed as

$$\hat{F}_{MSE} = F_{MSE,min} + F_{MSE,\Delta} + F_{MSE,lag} \tag{5.23}$$

where $F_{MSE,lag}$ denotes the mean square error due to the lag. In Fig. 5.7 the errors $F_{MSE,\Delta}$ and $F_{MSE,lag}$ are plotted as a function of Δ_{LMS}. It can be observed that $F_{MSE,\Delta}$ increases with an increase in Δ_{LMS} while $F_{MSE,lag}$ decreases with an increase in Δ_{LMS}. When the statistical time variations of the signal occur rapidly, the lag error will dominate the performance of the adaptive equaliser. In such a case, $F_{MSE,lag} \gg F_{MSE,min} + F_{MSE,\Delta}$, even when the largest possible value of Δ_{LMS} is used. When this condition occurs, the LMS algorithm is inappropriate for the specific application and other adaptive algorithms have to be employed.
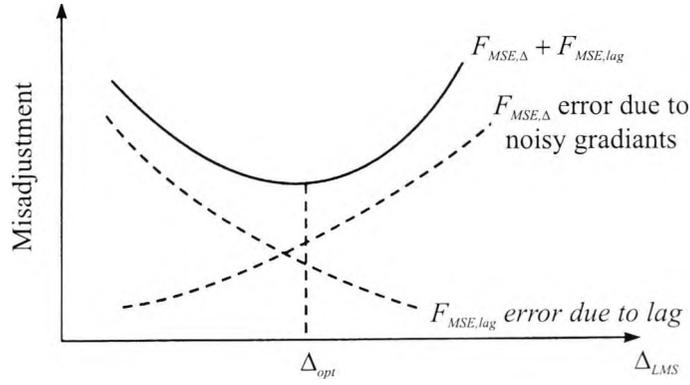
Figure 5.7: Total misadjustment versus step-size Δ_{LMS}

5.1.4 Hybrid Algorithm

The algorithm described in this section is named a hybrid algorithm since it employs estimation approaches both from the LMS algorithm and from the matrix inversion algorithm. The adaption procedure of the channel coefficients is based on the stochastic gradient algorithm, similar to the LMS algorithm.

$$\hat{h}_l(i) = \hat{h}_{l-1}(i) - \Delta_{hyb} \cdot \frac{\delta \left| e_l(i) \right|^2}{\delta \hat{h}_{l-1}^*(i)} \tag{5.24}$$

The difference to (5.16) now is, that the error e is computed separately for every individual coefficient h of the channel impulse response on every state. Note that the objective function now is the channel impulse response and not the received signal. The algorithm will thus give not an optimal bit error performance but result in a more exact channel estimate which might be used in subsequent algorithms which rely on a channel side information. The error e is given by

$$e_l(i) = e(i, j_{l-1} \rightarrow j_l) = \tilde{h}(i, j_l) - \hat{h}(i, j_{l-1}) \tag{5.25}$$

where $\tilde{h}(i, j_l)$ is the current value obtained from the received signal

$$r_l = \mathbf{h}^{\mathsf{T}}(j_l) \cdot \mathbf{z}(j_{l-1} \rightarrow j_l) + n_l' \tag{5.26}$$

by neglecting the noise term, solving the equation with respect to each coefficient $h(i, j_l)$, and by substituting the coefficients $h(k, j_l)$ for $k \neq i$ through the estimated values $\hat{h}(k, j_{l-1})$ of the last sampling instant. The current, possibly erroneous value of the channel coefficient i, which is obtained from the survivor path to state j_l can be

108

expressed as

$$\bar{h}(i, j_l) = F\left\{ r_l, \hat{h}(k, j_{l-1}), \mathbf{z}^{-1}(j_{l-1} \to j_l) \right\} , \quad k \neq i \tag{5.27}$$

Instead of the matrix inversion the hybrid algorithm only divides by the data vector. The application of the stochastic gradient algorithm results in a smoothing of the coefficients.

$$\hat{\mathbf{h}}(j_l) = \hat{\mathbf{h}}(j_{l-1}) + \Delta_{hyb} \cdot \mathbf{e}(j_{l-1} \to j_l) \tag{5.28}$$

Note that the step-size parameters Δ_{LMS} and Δ_{hyb} in (5.18) and (5.28) are different. In order to compare the performance of the LMS and the hybrid algorithm a relation between both step-size parameters has to be defined. If the equations (5.18) and (5.28) are compared, one yields

$$\Delta_{LMS} \cdot (r_l - \hat{h}_{l-1} \cdot \tilde{z}_l) \cdot \tilde{z}_l^* = \Delta_{hyb} \cdot (\bar{h} - \hat{h}_{l-1}) \tag{5.29}$$

which results in

$$\Delta_{hyb} = E\left[|\tilde{z}_l|^2\right] \cdot \Delta_{LMS} \tag{5.30}$$

In a OQPSK system using filters with a roll-off factor of 1.0 $\tilde{z}$ is defined as

$$\tilde{z}_l = [0.5 \cdot z_{l-1} + z_l + 0.5 \cdot z_{l+1}] \tag{5.31}$$

In this case the expected value $E\left[|\tilde{z}_l|^2\right]$ is 1.5, and therefore

$$\Delta_{hyb} = 1.5 \cdot \Delta_{LMS} \tag{5.32}$$

5.1.5 Kalman Estimator

The major advantage of the described LMS and hybrid algorithm is their computational simplicity. However, the price paid for the simplicity is slow convergence, especially when the autocorrelation matrix of the input of the estimator has a large spread of the eigenvalues. The slow convergence is due to the fact, that the gradient algorithms have only a single adjustable parameter for controlling the convergence rate, namely the step-size parameter Δ. In order to obtain faster convergence, it is necessary to apply algorithms involving additional parameters. Therefore, based on the iterative channel estimation with the LMS algorithm, an optimisation of the estimate with the aid of

a simplified Kalman estimator is developed. The derivation of the algorithm follows the steps in [22] and [17]. The Kalman estimator requires an ARMA process model of the frequency selective fading channel. Figure 5.8 shows the model of the time and frequency selective fading channel defined by the vector $\mathbf{h}$. The AR part of the vector
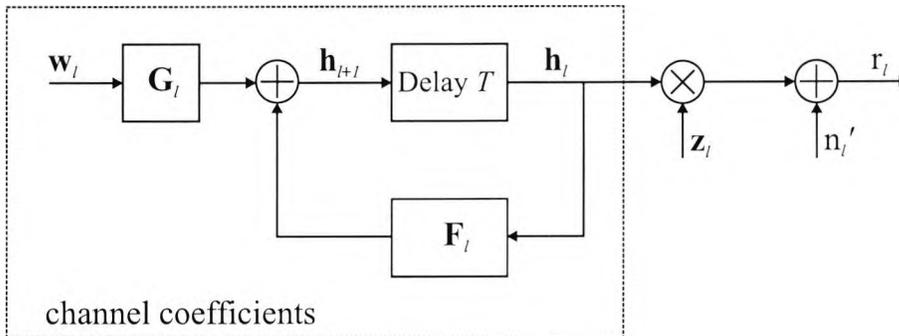
Figure 5.8: ARMA model of a Rayleigh fading channel

$\mathbf{h}_l$ at time epoch l in general considers also past samples of the channel coefficients, i.e., it is

$$\mathbf{h}_l^\mathrm{T} = [h_l(0), h_{l-1}(0) \ldots, h_{l-k}(0), \ldots, h_l(L), h_{l-1}(L), \ldots, h_{l-k}(L)] \tag{5.33}$$

If several AR coefficients are used, a good approximation of the power spectral density function of a uniform direction Rayleigh fading process can be achieved. Since, however, this specific power density function seldom occurs, as has been shown by measurements (see e.g. [15]), a first order approximation is sufficient. In a first order approximation $h_{l-1}(i) \ldots h_{l-k}(i) = 0$. In this case the vector $\mathbf{h}$ is $(L+1)$-dimensional, $\mathbf{F}_l$ and $\mathbf{G}_l$ then are $(L+1) \times (L+1)$ matrices. If furthermore, the input process

$$\mathbf{w}_l^\mathrm{T} = [w_l(0), \ldots, w_l(i), \ldots, w_l(L)] \tag{5.34}$$

with powers $\sigma_w^2(i)$ of its components $w(i)$ is not averaged, $\mathbf{G}_l$ becomes a unit matrix. Assuming that the channel coefficients are uncorrelated and unbiased (see Section 2.3), i.e.,

$$\mathrm{E}\left[h(i) \cdot h^*(j)\right] = \mathrm{E}\left[|h(i)|^2\right] \cdot \delta_{ij} , \quad (\delta_{ij} \hat{=} Kronecker\, Delta) \tag{5.35}$$

the matrix $\mathbf{F}_l$ is diagonal. The filtered data can then be expressed

$$r_l = \mathbf{z}_l^\mathrm{T} \cdot \mathbf{h}_l + n_l' = \mathbf{z}_l^\mathrm{T} \cdot (\mathbf{F}_{l-1} \cdot \mathbf{h}_{l-1} + \mathbf{w}_{l-1}) + n_l' \tag{5.36}$$

If, furthermore, the Rayleigh process is stationary (WSSUS), $\mathbf{F}_l = \mathbf{F}$. For a Rayleigh fading process the diagonal elements of the matrix $\mathbf{F}$ are equal to the Bessel function

$J_0(2\pi f_d \cdot T_s)$ and may be further simplified to $\beta = \mathrm{e}^{-2\pi f_d T_s}$. At this point, it should be mentioned that for reasons of a simplified formulation $\hat{\mathbf{h}}_l$ and $\mathbf{z}_l$ are used instead of $\hat{\mathbf{h}}(j_l)$ and $\mathbf{z}(j_{l-1} \to j_l)$ in the following.

The vector of the estimated channel coefficients, which is obtained with the aid of a Kalman filter adapted to the channel model can be expressed [17] as

$$\hat{\mathbf{h}}_l = \mathbf{F} \cdot \hat{\mathbf{h}}_{l-1} + \mathbf{k}_l \cdot \left(r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l \right) = \mathbf{F} \cdot \hat{\mathbf{h}}_{l-1} + \mathbf{k}_l \cdot e_l \tag{5.37}$$

where $\mathbf{k}$ is the Kalman gain factor

$$\mathbf{k}_l = \mathbf{F} \cdot \mathbf{R}_{l-1} \cdot \mathbf{z}_l^* \cdot \left[\mathbf{z}_l^{\mathrm{T}} \cdot \mathbf{R}_{l-1} \cdot \mathbf{z}_l^* + \sigma_{N'}^2 \right]^{-1} \tag{5.38}$$

with $\mathrm{E}\left[n_l' \cdot n_k'^* \right] = \sigma_{N'}^2 \cdot \delta_{lk}$. The equation can be simplified by replacing the diagonal matrix $\mathbf{F}$ with the element β

$$\mathbf{k}_l = \beta \cdot \mathbf{R}_{l-1} \cdot \mathbf{z}_l^* \cdot \left[\mathbf{z}_l^{\mathrm{T}} \cdot \mathbf{R}_{l-1} \cdot \mathbf{z}_l^* + \sigma_{N'}^2 \right]^{-1} \tag{5.39}$$

since $\mathbf{F} = \beta \cdot \mathbf{I}$. To compute the Kalman Gain factor $\mathbf{k}_l$ the knowledge of the error covariance matrix $\mathbf{R}_{l-1}$ is required. The error covariance matrix is defined as

$$\mathbf{R}_{l-1} = \mathrm{E}_{n',w}\left[\left(\mathbf{h}_{l-1} - \hat{\mathbf{h}}_{l-1} \right) \cdot \left(\mathbf{h}_{l-1} - \hat{\mathbf{h}}_{l-1} \right)^{\mathrm{H}} \right] \tag{5.40}$$

It can be shown [17] that the error covariance matrix can be computed recursively with the aid of the Riccati equation

$$\mathbf{R}_l = \mathbf{F} \cdot \mathbf{R}_{l-1} \cdot \mathbf{F}^{\mathrm{H}} - \mathbf{k}_l \cdot \mathbf{z}_l^{\mathrm{T}} \cdot \mathbf{R}_{l-1} \cdot \mathbf{F}^{\mathrm{H}} + \mathbf{Q}_l \tag{5.41}$$

The matrix $\mathbf{Q}_l$ is diagonal and contains the signal powers $\sigma_w^2(i)$ of the input processes $w(i)$ of the vector from (5.34), if the process is stationary. Since the feedback matrix $\mathbf{F}$ of the channel process is diagonal with elements β, the following relationship between the power of a channel path and the power of the input process of the ARMA model holds

$$\mathrm{E}\left[|h(i)|^2 \right] = \frac{\sigma_w^2(i)}{1 - \beta^2} \tag{5.42}$$

The equation of the Kalman gain can further be simplified if it is considered, that the values of the main diagonal will dominate in the error covariance matrix $\mathbf{R}_l$. An averaging of the diagonal elements of $\mathbf{R}_l$ over the data results in the diagonal matrix $\tilde{\mathbf{R}}_l$ with elements $\rho_l(i)$. The Kalman gain is then obtained as

$$\mathbf{k}_l = \beta \cdot \tilde{\mathbf{R}}_{l-1} \cdot \mathbf{z}_l^* \cdot \left[\sum_{i=0}^{L} \rho_{l-1}(i) \cdot |\tilde{z}_{l-i}|^2 + \sigma_{N'}^2 \right]^{-1} \tag{5.43}$$

A further averaging of the denominator of $\mathbf{k}_l$ over the data gives

$$\mathbf{k}_l = \beta \cdot \tilde{\mathbf{R}}_{l-1} \cdot \mathbf{z}_l^* \cdot \left[c \cdot \sum_{i=0}^{L} \rho_{l-1}(i) + \sigma_{N'}^2 \right]^{-1} \tag{5.44}$$

where c is a constant which depends on the roll-off factor of the Nyquist filter. In the last section it has been shown, that c is 1.5 for $r = 1.0$.

Unlike the LMS algorithm, every channel coefficient is now multiplied by an individual gain factor

$$k_l(i) = \beta \cdot \rho_{l-1}(i) \cdot \tilde{z}_{l-i}^* \cdot \left[c \cdot \sum_{i=0}^{L} \rho_{l-1}(i) + \sigma_{N'}^2 \right]^{-1} \tag{5.45}$$

By means of the Riccati equation it can be shown that for higher signal-to-noise ratios E_s/N_0 an approximation for the diagonal elements of the error covariance matrix can be found

$$\rho_{l \to \infty}(i) \approx \sigma_w^2(i) = \mathrm{E}\left[|h(i)|^2 \right] \cdot \left(1 - \beta^2 \right) \tag{5.46}$$

The Kalman gain factor $k(i)$, thus, is proportional to the average power $\mathrm{E}\left[|h(i)|^2 \right] = \sigma_i^2$. In the steady state, consequently, the step size $k(i)$ for the adaption of the channel coefficients is proportional to the average relative signal power of that specific path. This result can also be applied to the hybrid channel estimation algorithm and even to other adaptation algorithms, if the step size is chosen proportional to the signal power of a path.

5.1.6 Application of a Simplified Kalman Estimator to the LMS Algorithm

The understanding of the simplified Kalman estimator is now used to optimise the step-size parameter Δ_{LMS} of the LMS algorithm and to show that the simplified Kalman algorithm can be interpreted as a LMS algorithm with power dependent coefficients. For this purpose, the equations for the estimated channel coefficients (5.18) and (5.37) are given again.

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} - \Delta_{LMS} \cdot \left(r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l \right) \cdot \mathbf{z}_l^* \tag{5.47}$$

$$\hat{\mathbf{h}}_l = \mathbf{F} \cdot \hat{\mathbf{h}}_{l-1} + \mathbf{k}_l \cdot \left(r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l \right) \tag{5.48}$$

112

The comparison of both equations leads to the result that

$$\mathbf{k}_l = \Delta_{LMS} \cdot \mathbf{z}_l^*$$ (5.49)

Thus, the parameter Δ_{LMS}, which is identical for all channel paths, is changed to a set of parameters $\Delta_{LMS}(i)$, which have different values for the different channel paths.

$$\Delta_{LMS}(i) = k_l(i) \cdot (\tilde{z}_{l-i}^*)^{-1}$$ (5.50)

Inserting equation (5.45) into (5.50) and applying the approximation (5.46) yields

$$\Delta_{LMS}(i) = \beta \cdot \sigma_i^2 \cdot (1 - \beta^2) \cdot \left[c \cdot \sum_{i=0}^{L} \sigma_i^2 \cdot (1 - \beta^2) + \sigma_{N'}^2 \right]^{-1}$$ (5.51)

Equation (5.51) can further be simplified, if $c = 1.5$ (roll-off factor of 1.0) and the signal power is normalised to $\sigma^2 = \sum_{i=0}^{L} \sigma_i^2 = 1$

$$\Delta_{LMS}(i) = \frac{\beta \cdot \sigma_i^2 \cdot (1 - \beta^2)}{1.5 \cdot (1 - \beta^2) + \sigma_{N'}^2}$$ (5.52)

with $\sigma_{N'}^2 = 10^{-\frac{E_s/N_0}{10}}$.

The computation of the step-size parameters by means of the simplified Kalman algorithm involves some restrictions. Equation (5.52) only considers the power of the individual paths and the signal-to-noise ratio. The Doppler frequency, which is included in the parameter β, has only a minor influence. In contrast to the results obtained with the Ricatti equation, the simplified equation is only valid in a specific region of the signal-to-noise ratio ($E_s/N_0 > 10$ dB). In general the calculated values are smaller than the exact values from the Ricatti equation. However, both computation methods can not be used for higher E_s/N_0 values (> 20 dB), since the Rayleigh fading process model, which is a first order Markov Process, is not suitable for this region.

5.2 Results

The performance of the DOQPSK receiver, which implements per-survivor processing, depends mainly on the correct choice of the step-size parameter Δ. Equation (5.52) gives the calculation method for the step-size parameter, if the LMS algorithm is used to estimate the channel impulse response. The calculated values of the parameter for different Doppler frequencies and E_s/N_0 values are listed in Table 5.1. As mentioned

$f_d \cdot T_s$	E_s/N_0				
	10 dB	15 dB	20 dB	30 dB	40 dB
0.0001	0.012	0.037	0.106	0.435	0.633
0.0003	0.036	0.1	0.24	0.565	0.654
0.0015	0.145	0.31	0.487	0.638	0.658
0.0046	0.296	0.471	0.579	0.64	0.647
0.0093	0.392	0.528	0.593	0.625	0.628

Table 5.1: Values of the step-size parameter Δ_{LMS}

before, for E_s/N_0 values higher than 20 dB the calculation method shows a saturation effect. Although the Doppler frequency increases, the value of the step-size parameter decreases as a matter of a simplified process model.

In the following sections the influence of the step-size parameter on the acquisition and BER performance is investigated in detail.

5.2.1 Acquisition Performance and Channel Tracking

A short acquisition time and a fast tracking performance is essential for a blind trellis search algorithm in fast time selective channels. In order to investigate the acquisition performance the next two figures give the squared error of the channel coefficients of the LMS and the hybrid algorithm on a single path Rayleigh fading channel. The mean squared error has been averaged over 30 independent simulations. In Fig. 5.9 the product of Doppler frequency and symbol duration $f_d T_s$ is fixed to 0.0003 and results are given for an E_s/N_0 of 10 and 80 dB. In general, both algorithms show a similar behaviour. An increase of the signal-to-noise ratio E_s/N_0 leads to a strong decrease of the squared error. However, while the convergence of the LMS algorithm has been observed within 50 symbols in the case of 10 dB E_s/N_0, the rate of convergence of the hybrid algorithm is slower. BER investigations show, that for the LMS algorithm no further bit errors occur after 10 bits. The slower rate of convergence of the hybrid
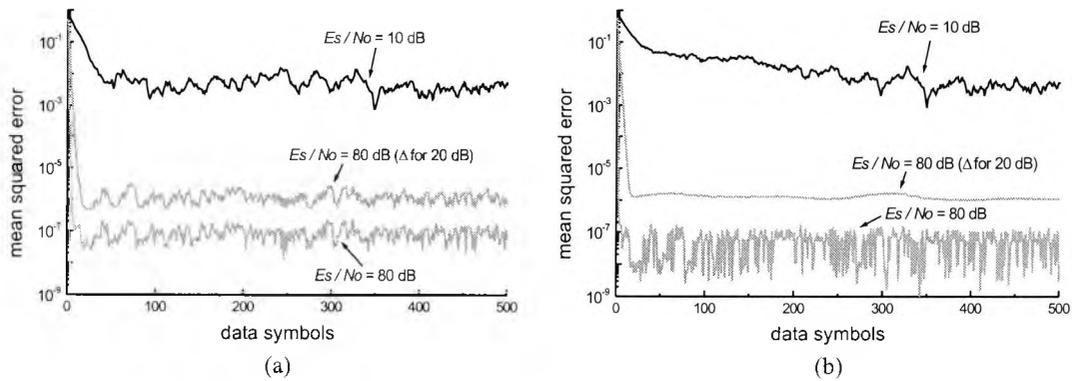
Figure 5.9: Acquisition performance on a single path Rayleigh fading channel with $f_d T_s = 0.0003$ (a) LMS algorithm, (b) hybrid algorithm

algorithm results in bit errors within the first 30 bits. On the other hand, the hybrid algorithm shows a smaller steady state squared error for higher E_s/N_0 values. In the illustrated case of $E_s/N_0 = 80$ dB and a non-optimal step-size parameter calculated for $E_s/N_0 = 20$ dB the simulation with the hybrid algorithm results in a smoother curve. Figure 5.10 illustrates the acquisition performance for a higher Doppler frequency ($f_d T_s = 0.0046$). In this case, the hybrid algorithm yields the same rate of convergence as the LMS algorithm The number of bit errors during the acquisition
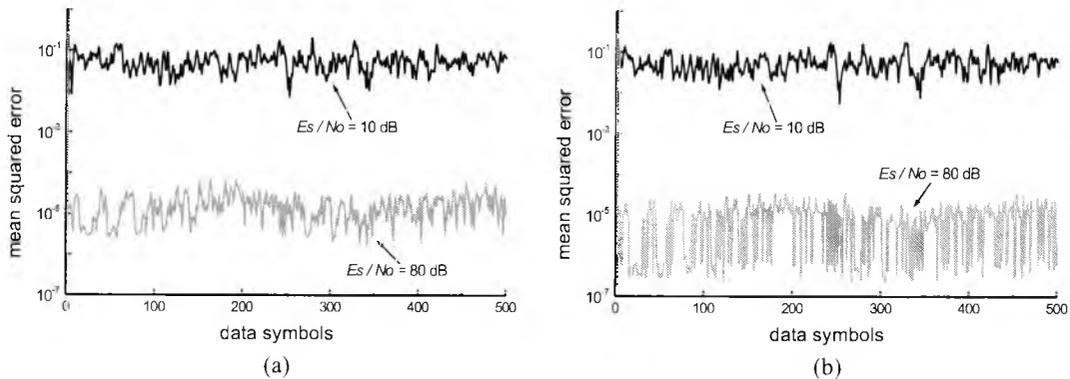
Figure 5.10: Acquisition performance on a single path Rayleigh fading channel with $f_d T_s = 0.0046$ (a) LMS algorithm, (b) hybrid algorithm

process are also similar. However, again the hybrid algorithm shows a less steady state squared error.

Before the acquisition performance on a two path Rayleigh fading channel will be considered, the next two figures illustrate the effect of an incorrectly chosen step-size. The figures show the estimates of the real and imaginary part of the channel coefficients on a two path Rayleigh fading channel. The nominal values of the coefficients are not visible since they are hidden by the curves of the estimates. In Fig. 5.11 the step-size

115

parameters for both channel paths are chosen equal. This is the correct choice for the simulation illustrated in the left picture, where both channel paths have the same power. The right picture shows the case that the power of the echo path is reduced by 60 dB compared to the direct path. In this case, the mismatched step-size parameter
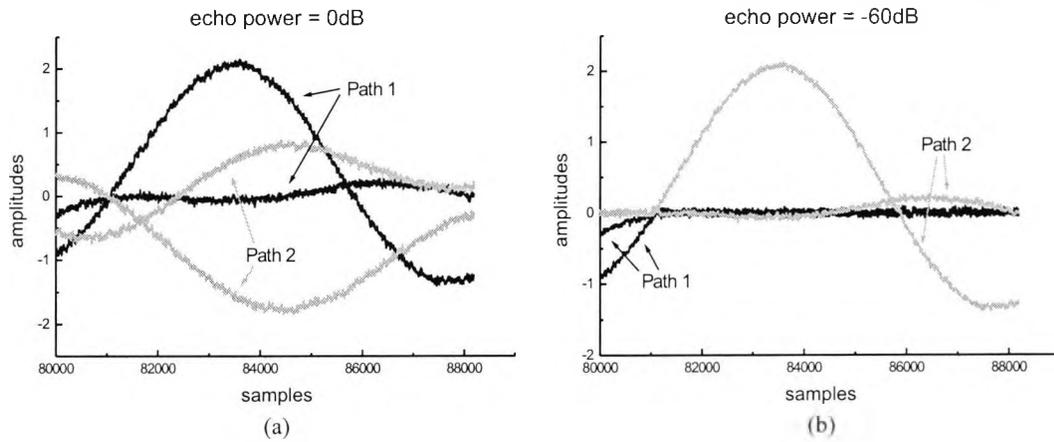
Figure 5.11: Channel tracking on a two path Rayleigh fading channel with $f_d T_s = 0.0003$ and $\Delta(1) = \Delta(2)$

$\Delta(2)$ results in a wrong channel estimate. At the sampling time, where all channel coefficients cross the zero amplitude, the algorithm exchanges the path assignment and, therefore, a bitslip occurs leading to a high BER.

In Fig. 5.12 the step-size parameter of the direct path is twice the parameter of the echo path. If the channel paths have the same power no significant difference to Fig. 5.11a is obtained. However, the wrong assignment of the channel paths is prevented
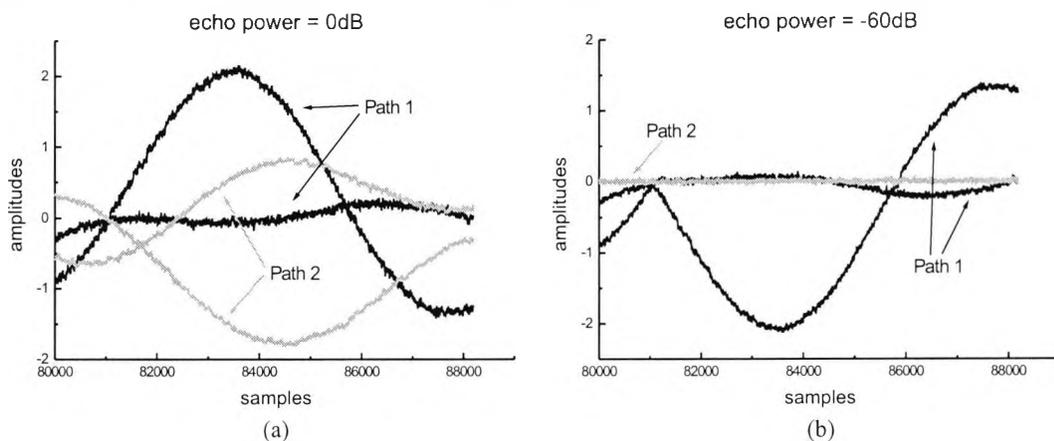
Figure 5.12: Channel tracking on a two path Rayleigh fading channel with $f_d T_s = 0.0003$ and $\Delta(1) = 2 \cdot \Delta(2)$

in the case of the reduced echo power. Nevertheless, an error event occurs if all channel coefficients cross the zero amplitude at the same time. In this case, it is possible that

116

the estimation algorithm inverts the channel coefficients. Due to the fact, that the receiver uses differential decoding this short error event in the channel estimation leads only to limited number of bit errors.

In Fig. 5.13 and 5.14 the squared errors of the channel coefficients of the LMS and the hybrid algorithm on a two path Rayleigh fading channel are presented. The signal-to-noise ratio in these simulations is fixed to 10 dB and the power of the echo path is reduced by 10 dB compared to the direct path. The step-size parameters $\Delta(1)$ and $\Delta(2)$ are computed from (5.52) in accordance with the ratio of the paths power. In contrast to the results on the one path channel, the hybrid algorithm shows the
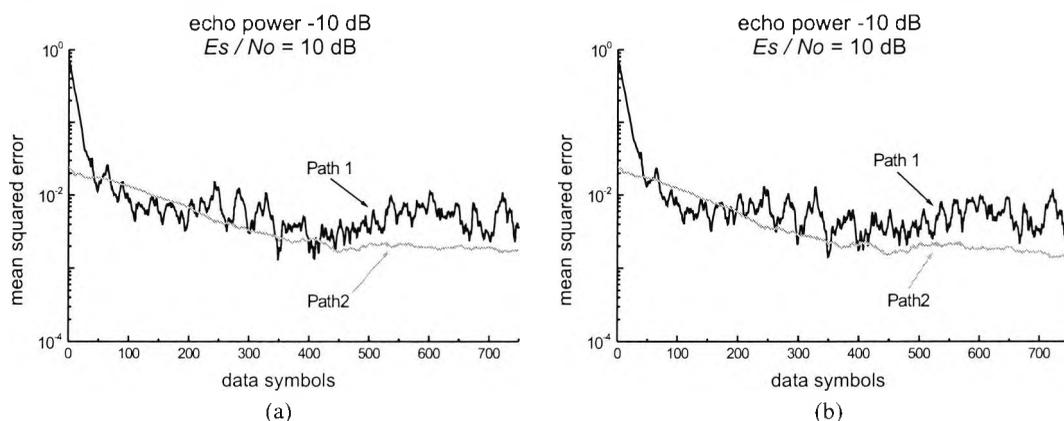
Figure 5.13: Acquisition performance on a two path Rayleigh fading channel with $f_d T_s = 0.0003$ (a) LMS algorithm, (b) hybrid algorithm

same acquisition performance as the LMS algorithm even in the case of a small value of $f_d T_s$. The rate of convergence of the echo path is, depending on the Doppler frequency, slower than the rate of the direct path. However, the steady state squared error is also smaller, which is in accordance to the reduced power. Due to the averaging of the squared error over 30 simulations some effects are not visible in the figures. It can be observed that the channel estimates of the LMS algorithm show significant peaks of the squared errors for individual configurations of the simulations. This is especially true for two simulations with $f_d T_s = 0.0046$ which result in a squared error of more than 1.5 in the tracking mode. The simulations of the hybrid algorithm with the same configuration do not show this effect. This effect results from the structure of the LMS algorithm:

- The objective function of the LMS algorithm is the received signal and not the channel impulse response.
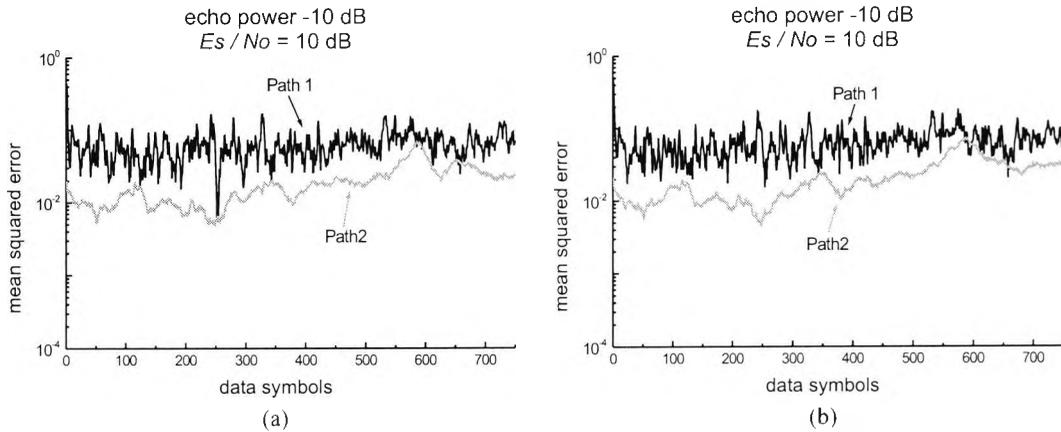
Figure 5.14: Acquisition performance on a two path Rayleigh fading channel with $f_d T_s = 0.0046$ (a) LMS algorithm, (b) hybrid algorithm

- The simulation model for the trellis decoder provides the estimates of the channel impulse response for data visualisation. These estimates are the best estimates of the actual sampling epoch, and hence are tentative. The data, however, are decoded with delay and therefore may belong to another channel response.

- A squared error which is larger than 1.0 indicates that the estimate of the channel impulse response has been inverted.

5.2.2 BER on Linear Channels

Similar to the differential demodulator and where no whitening matched filter is used, the coherent receiver with per-survivor processing is also not optimal. Though the coherent receiver uses no differential preprocessing, and therefore the calculation of the Euclidean distances is correct, the twofold sampling per symbol again results in a trellis with an infinite number of states for a roll-off factor less than one. The following results are obtained with a trellis adapted to $r = 1.0$. Then the number of states in the trellis only depends on the number of channel paths and the path delays. The developed receiver for a single path channel results in a 4 state trellis and the receiver for a two path channel in 8 states for a path delay of $T_s/2$ and 16 states when a path delay of T_s is assumed.

In Fig. 5.15 the simulation results on an AWGN channel for coherent $\pi/4$-DQPSK and DOQPSK are compared. In the $\pi/4$-DQPSK receiver no channel estimation algorithm is necessary on an AWGN channel. Therefore, $\pi/4$-DQPSK shows better results than the DOQPSK receiver which uses the LMS algorithm for channel estimation. The

118
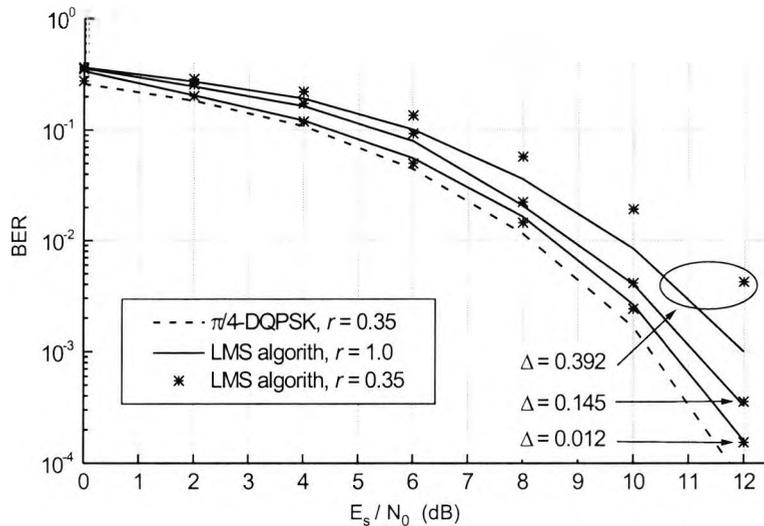
Figure 5.15: DOQPSK employing the LMS algorithm with different values of Δ on an AWGN channel

results for DOQPSK are obtained with values of the step-size parameter according to $f_d T_s = 0.0001$, 0.0015, and 0.0093. An increase of the Doppler frequency leads to a degradation of the BER. The receiver with a 4 state trellis, which is matched to $r = 1.0$, shows a performance degradation only for higher values of the step-size parameter when filters with $r = 0.35$ are used. On a static AWGN channel and for small values of Δ no difference between $r = 1.0$ and $r = 0.35$ are observed. The hybrid algorithm shows the same BER performance if the step-size is calculated from (5.32). Thus, only the BER curves obtained with the LMS algorithm are presented in the following.

Time Selective Channel

The coherent $\pi/4 - DQPSK$ receiver, which has been used as a reference on the AWGN channel, is only suitable for static channels, because no adaptive equaliser is included. Therefore, on time selective channels the obtained BER performance of DOQPSK is compared with the theoretical bounds, which are given in Section 2.4.2.

The BER of DOQPSK on a pure time selective Rayleigh fading channel is given in Fig. 5.16. In Fig. 5.16a the simulation results of DOQPSK for different Doppler frequencies are compared with the theoretical bound, where the receiver velocity is assumed to be zero. In case of a small Doppler frequency and, therefore, a small step-size parameter, the theoretical bound can almost be achieved with the PSP receiver. If the Doppler frequency is increased, the bit error rate increases as expected. In
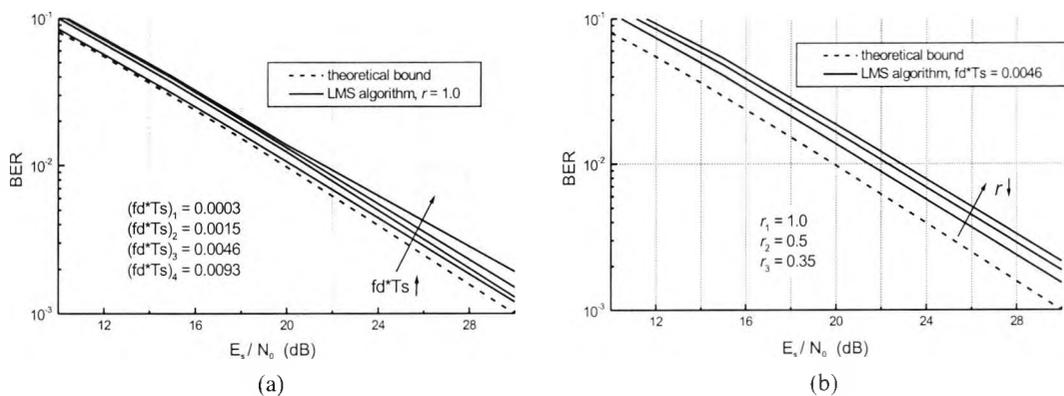
Figure 5.16: DOQPSK employing the LMS algorithm on a Rayleigh fading channel (a) different Doppler frequencies, (b) different roll-off factors

Fig. 5.16b the influence of the roll-off factor is shown. As mentioned above, the trellis decoder in the receiver is adapted to a roll-off factor of 1.0. The curves in the figure illustrate that the use of filters with a smaller roll-off factor leads to an increase in the bit error rate. To demonstrate this effect, the value of $f_d T_s$ has to be chosen large enough ($f_d T_s = 0.0046$), since otherwise the degradation is not observable and can be neglected.

Time and Frequency Selective Channel

According to the Kalman estimator (5.52), the step sizes depend on the signal-to-noise ratio. In a practical receiver it is, however, not appropriate to compute the step size for every single value of E_s/N_0. The previous results were also achieved with one fixed step-size parameter. The curves in Fig. 5.17 give the results for three fixed values of the step-size parameters on a two path Rayleigh fading channel. The step sizes are
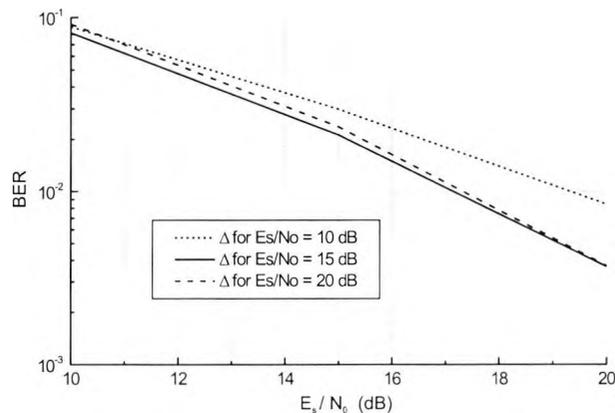
Figure 5.17: Comparison of different step sizes, relative echo power path is -10 dB, $r = 1.0$

120

calculated from (5.52) for an E_s/N_0 of 10, 15 and 20 dB and the curves are obtained by simulations in a range of E_s/N_0 from 10 to 20 dB. The curve according to a step size calculated for an E_s/N_0 of 15 dB shows, that for a wide range of E_s/N_0 the step size which has been calculated for the value in the middle of the E_s/N_0-interval is sufficient. This rule is kept during all following investigations.

Figure 5.18 shows the simulation results of the DOQPSK receiver on a two path Rayleigh fading channel. In Fig. 5.18a, the delay of the echo path is fixed to one symbol period. The lower curve gives the theoretical bound of DQPSK with two fold diversity in a maximal ratio receiver. In case of equal signal powers and a small Doppler frequency the gain of a diversity receiver can almost be achieved. The residual bit error rate is due to the nonzero receiver velocity in the simulations. An increase of the Doppler frequency leads to a clear degradation in the BER. If the power of the echo path is reduced compared to the direct path and the step sizes are adapted to this new environment, the BER also increases. However, depending on the remaining echo power a gain compared to the theoretical bound without diversity can be achieved. This effect is illustrated by the curve for a relative echo power of -10 dB. Another interesting effect, which is not shown in the figures, has been found by further simulations. If the power of the echo path is equal to the direct path, but the step size of the echo path in the receiver is adapted to a smaller power value (e.g. -20 dB), the result is comparable to the theoretical bound without diversity. This result can be used to derive a strategy
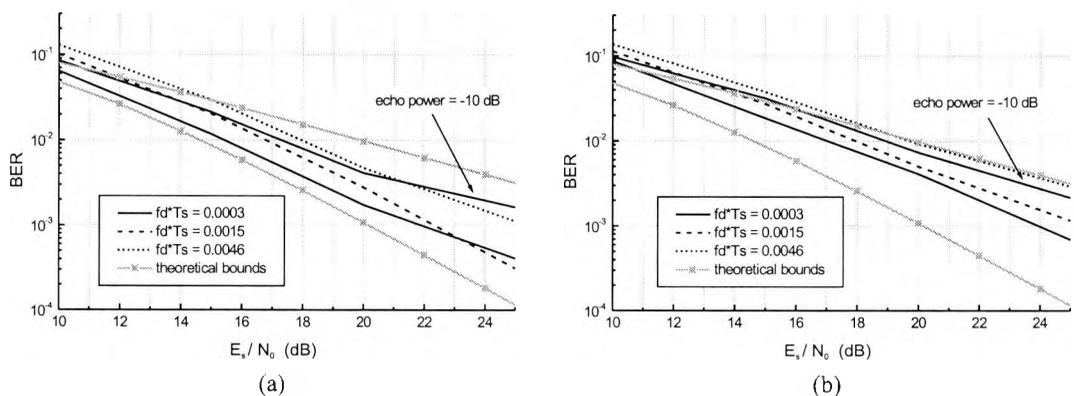
Figure 5.18: DOQPSK on a two path Rayleigh fading channel (a) echo delay $= T_s$, (b) echo delay $= T_s/2$

for the power estimation of the different paths. If a continuous data transmission is assumed, the step-size parameters can be adjusted adaptively, starting with very small values for the echo paths.

The results, which are obtained with a path delay of $T_s/2 = T_b$ (Fig. 5.18b) are generally worse than the results for T_s. This is due to the fact, that the paths are not uncorrelated anymore. Thus, the achievable gain is reduced. However, due to the offset of one component in the transmitter and the twofold sampling per symbol in the receiver, only the DOQPSK algorithm has the possibility to resolve an echo with a delay of $T_s/2$ and to use this information to achieve a diversity gain.

The derived algorithm and the developed simulation model are also suitable for channels with more than one echo path. The problem which then emerges is the length of the simulation time which is required in order to get a reliable result. For a single path Rayleigh fading channel the required number of bits in the simulation can be derived analytically from the Rayleigh distribution and the level crossing rate. The simulation length for multipath channels, however, has to be determined empirically from the statistics of the achieved BER results. For a three path Rayleigh fading cannel, in which a reliable BER of 10^{-6} is expected, the simulation time increases up to several days. Therefore, such results can not be presented here.

Additionally to the investigations of the PSP algorithm for channels with more than one echo path, the receiver performance has to be evaluated when the receiver is not matched to the number of channel paths. The curves in Fig. 5.19a are obtained using
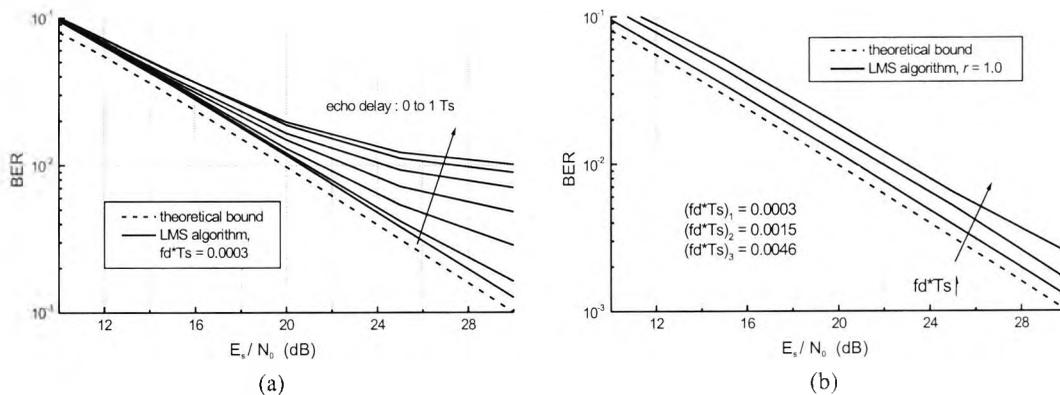
Figure 5.19: (a) Two path Channel, no echo equaliser (b) single path Channel, echo equaliser

a receiver matched to a single path channel but actually the channel is a two path Rayleigh fading channel with -20 dB echo power. It is shown that the bit error rate may increase dramatically depending on the delay of the echo path. Only a small delay of $1/8T_s$ leads to a bit error rate similar to the result obtained with no echo delay. In Fig. 5.19a the BER is parameterised with a delay of the echo path with multiplies of

$1/8T_s$. A delay higher than $6/8T_s$ saturates in the upper curve. In Fig. 5.19b curves are presented for the reverse case when the number of resolvable echoes in the receiver is greater than the number of echoes in the channel. The results are obtained from simulations on a single path Rayleigh fading channel. The receiver algorithm, however, is adapted to a two path channel with a channel delay of T_s and an echo power of -20 dB. In this case, a small degradation compared to the results in Fig. 5.16a can be observed. If the receiver considers a higher echo power, the bit error rate increases. An echo power of 0 dB in the receiver can lead to a bit error rate of 0.5 if the receiver interchanges the paths within a synchronous decoding process.

5.2.3 BER on Nonlinear Channels

On linear channels the coherent DOQPSK algorithm using per-survivor processing has proved, that this type of receiver is suitable for applications in mobile radio systems. In this section the DOQPSK receiver with PSP will be investigated on nonlinear channels. Since the design of the transmitted signal mainly determines the performance on nonlinear channels it can be expected that the coherent receiver achieves similar results as the receiver using differential demodulation. Therefore, this section is limited to the presentation of some figures which do confirm this statement. The simulations, however, were carried out as complete as those for differential demodulation with the amplifiers described in Chapter 3.

Figure 5.20 compares the coherent receivers of DOQPSK and $\pi/4$-DQPSK at a bit error rate of 10^{-3} under relative adjacent channel power P_{AC} and without noise on a static channel. The simulation results in Fig. 5.20a are obtained using the generic
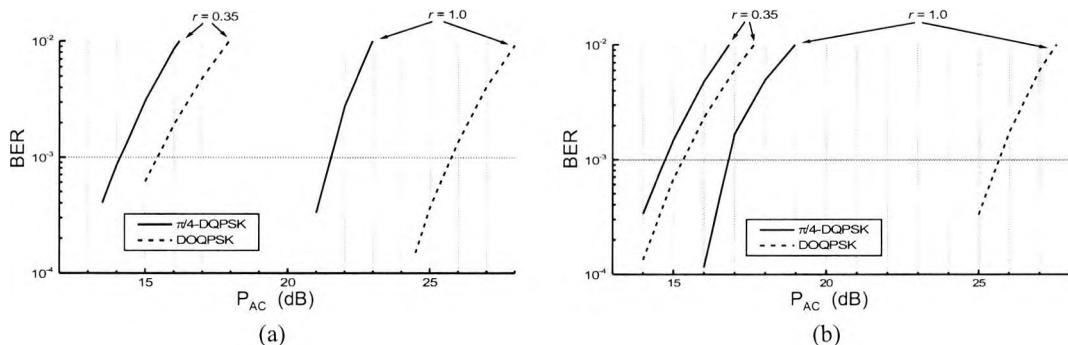
Figure 5.20: BER performance under ACI without noise (a) parameter $a = 0.3$, (b) parameter $c = 1.0$

amplifier model for the weak conduction region and a value of the parameter a of 0.3.

In this case the use of DOQPSK with no channel adaptation ($\Delta = 0$) results in a gain against $\pi/4$-DQPSK of more than 4 dB for a roll-off factor of 1.0 and about 1.5 dB for $r = 0.35$. The implementation of the saturation model and a value of the parameter c of 1.0 yields the curves in Fig. 5.20b. Comparable to the results presented in Chapter 4, a roll-off factor of 1.0 results in an improvement of 9 dB, whereas, when filters with $r = 0.35$ are used, only a gain of 1 dB is obtained. If a channel adaption is used in the DOQPSK receiver, the gain will decrease depending on the chosen step size.

The next two figures show that the time shift of the adjacent channels leads to the same behaviour as described for the receiver with differential demodulation. Figure 5.21 illustrates the simulation results under a fixed relative adjacent channel power of 13 dB in an AWGN environment. In the simulations the nonlinear amplifier characteristic with a parameter a of 0.3 is used. The results for $r = 1.0$ and synchronised channels show no significant differences between DOQPSK and $\pi/4$-DQPSK. This is due to the fact, that in this case the adjacent channels with $P_{AC} = 13$dB have no influence on the performance in the considered BER region (see Fig. 5.20a). However, if the adjacent channels are shifted by half a symbol period, $\pi/4$-DQPSK shows a distinct degradation, whereas the values of DOQPSK remain almost stable. For a roll-off factor of 0.35 a
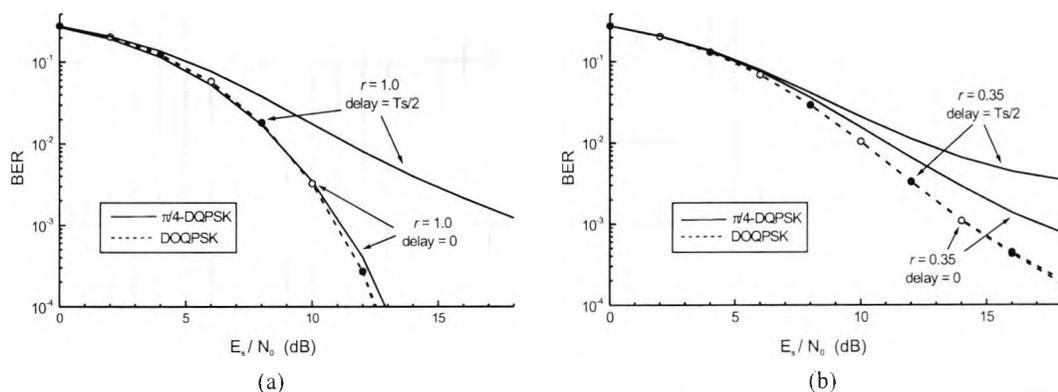
Figure 5.21: BER performance under ACI with $P_{AC} = 13$ dB and parameter $a = 0.3$ on an AWGN channel (a) $r = 1.0$, (b) $r = 0.35$

difference in the bit error rate is already visible when the channels are synchronised. The time shift of $T_s/2$ has also no influence on the bit error rate of DOQPSK for this small roll-off factor. The curves of $\pi/4$-DQPSK, on the other hand, show a degradation which is, however, less significant as for $r = 1.0$

5.3 Comparison of the Receiver using Differential Demodulation with the Coherent Receiver using Per-Survivor Processing

In the first sections of this chapter, the results of coherent demodulation and equalisation of DOQPSK signals with per-survivor processing are compared with the results of a coherent $\pi/4$-DQPSK receiver and the theoretical bounds. In this section the coherent DOQPSK receiver will be compared with the non-coherent receiver described in Chapter 4.

5.3.1 Bit Error Rate Performance

In the Basic Chapter the formulas of the bit error probability of differentially encoded phase shift keying (DPSK) signals are given. It has been shown, that on an AWGN channel a difference exist in the error probability between the coherent demodulation with differential detection and the differential demodulation. The results for DOQPSK obtained by simulation are illustrated in Fig. 5.22a to confirm this statement. The coherent receiver shows a distinct gain, independent of the chosen roll-off factor. For a roll-off factor of 0.35 a gain of 2 dB can be achieved in the E_s/N_0 region of interest. On a fast time selective channel the conditions may change (Fig. 5.22b). On fast time se-
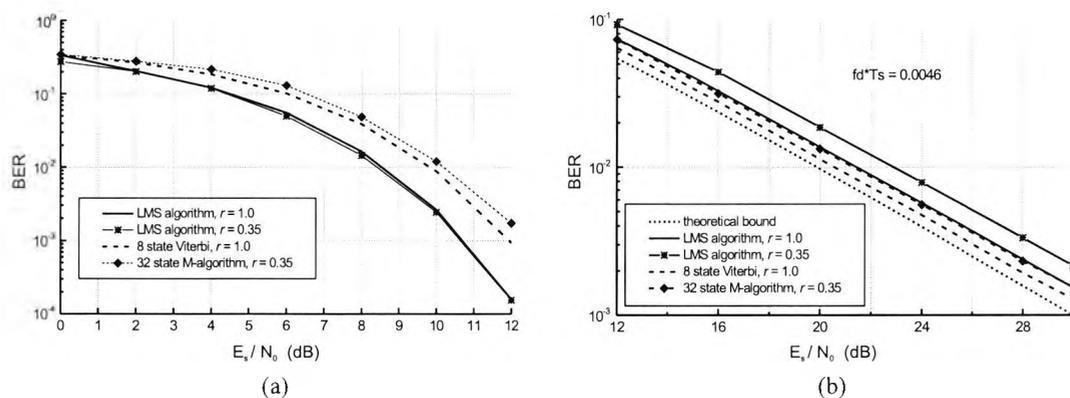
Figure 5.22: Comparison of the coherent (straight lines) and non-coherent (dashed and dotted lines) DOQPSK receivers (a) AWGN channel, (b) Rayleigh fading channel

lective channels the phase estimation in the differential demodulator, which is based on the phase difference between the current and the last symbol, leads to a better performance than can be obtained with the slower adaptive channel estimator in the coherent

receiver. A major drawback of the non-coherent receiver is the fact, that the better tracking capability of differential demodulation can only be used in non-frequency selective channels. Differential demodulation does not resolve intersymbol interference introduced by a frequency selective channel. The coherent receiver, however, does not only resolve the intersymbol interference, it even achieves a diversity gain through the power of the echo paths.

5.3.2 Receiver Complexity

In addition to the results with regards to the error rate, implementation complexity is also a decisive point when a receiver algorithm for a new system has to be selected. The comparison of the complexity will be done firstly by the required number of states in the trellis and secondly by the number of multiplications and additions for an additional processing prior to the trellis decoding. The simulation time will not be used to make a statement about the complexity, because the simulation time depends on several other effects like the load and the performance of the computer and the quality of the written code. The influence on the performance through the quality of the C code will be described in an example in the next chapter.

Both the coherent and the non-coherent receiver use a trellis decoder for data decoding. The non-coherent receiver uses a differential preprocessing device introducing additional computational load which needs a number of multiplications and additions at every sampling interval. On the other hand, the coherent receiver includes channel estimation which needs some operations in every state of the trellis. Table 5.2 gives an overview of the required complexity of the different receiver algorithms for different roll-off factors. For a roll-off factor of 1.0 the non-coherent receiver uses a trellis with

	Viterbi	M-alg.	LMS alg.	hybrid alg.
number states in the trellis r = 1.0 r = 0.35	8 128	32/16	4 4	4 4
operations in every state of the trellis			8m/6a[1]	8m/6a
additional operations at every sampling interval	4m/2a	4m/2a[2]		

[1] m = multiplications, a = additions

[2] plus sorting and selection processing within the trellis

Table 5.2: Comparison of the receiver complexity for a one path channel

twice the number of states as the coherent receiver. Since the differential demodulation needs no additional channel estimation in the trellis decoding process, the non-coherent receiver has an entire smaller complexity than the coherent receiver. For roll-off factors less than 1.0 the number of states in the trellis of the non-coherent receiver increases dramatically, whereas the coherent receiver still uses the 4 state trellis without a significant loss in BER. In this case the use of the coherent receiver is more efficient. The non-coherent receiver, which includes the M-algorithm uses a reduced number of states for data decoding. The algorithms require, however, additional sorting and selection processing, since it is based on the 128 state trellis. Therefore, the grade of complexity of the M-algorithm can be classified between the Viterbi algorithm and the coherent receiver.

The table illustrates that there is no difference in complexity between the LMS algorithm and the hybrid algorithm. This statement holds only for a single path channel. In a two path channel with an echo delay of T_s, for example, the hybrid algorithm needs 13 multiplications and 11 additions, while the LMS algorithm needs 12 multiplications and 13 additions. Since the multiplications are more complex than the additions, the complexity of the hybrid algorithm is higher.

The comparisons of the receiver structures in performance and complexity have shown that both structures have advantages and disadvantages. The receiver with differential demodulation is a good choice for fast time selective channels, since it includes a simple channel estimation. The coherent receiver with joint data decoding and channel estimation is especially suitable for time and frequency selective channels.

Chapter 6

Simulation Environment and Design of Receiver Models

6.1 Simulation Tool COSSAP™

COSSAP (Communication System Simulation and Analysis Package) is a complete digital signal processing design tool suite that is used by system designers to create, explore, and test algorithms, architectures, and implementations for a variety of digital signal processing applications such as wireless transmission, speech coding, data- and voice-band modem design, image processing, and data storage. Figure 6.1 illustrates a complete design environment for digital signal processing systems. In addition to
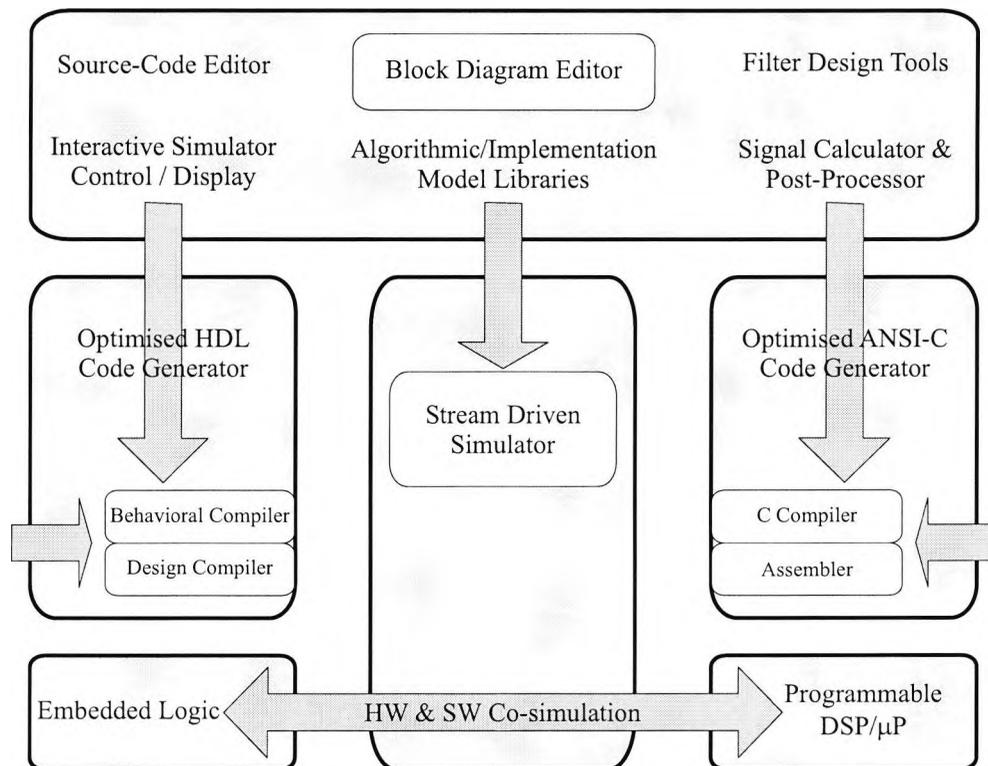
Figure 6.1: The COSSAP System Design Environment

the COSSAP packages, other Synopsys® products, which are necessary for hardware

implementation, are given in the figure. The principle of the entire implementation and verification flow is now described briefly. Systems of algorithms are comprehensively specified by system developers using the COSSAP block diagram editor (CBDE, see Fig. 6.2). In COSSAP a system will be defined as a hierarchical block diagram in which each block represents a function, and where interconnections represent the data flow. There are more than 1,200 models in COSSAP's extensive libraries. Communications system engineers, for example, will find virtually all models necessary for modeling complete transceivers for a number of wireless standards, including GSM, IS-95, DECT, and so on. Many models appear in floating-point, fixed-point, and HDL versions. In addition, it is easy to extend the existing libraries by writing new models in C or C++. The block diagrams are used as input specifications for simulation and implementation.

Figure 6.2: COSSAP Block Diagram Editor

Extensive parameterised simulations can then be performed using the COSSAP stream driven simulator (SDS). In contrast to the most conventional simulators for digital signal processing, which use time driven simulation, the COSSAP SDS is based on a full dynamic data flow approach. Each block of the lowest level in the block diagram (primitive model) consists of a software function. The blocks in an upper level are hierarchical models, which can include primitive models or other hierarchical models. Traditional simulators require a system clock and handle the simulation by advancing

this internal clock. This makes the simulation slow and requires the developer to invest a lot of time in configuring a particular design, especially when the design consists of multiple data rates and asynchronous parts. The SDS does not use an internal clock. The decision on when a block has to be activated is based on the availability of data on the input ports of this block (Fig. 6.3). Each block defines a minimum number of signal elements in order to start the processing. This method supports applications that process data at more than a single rate naturaly.
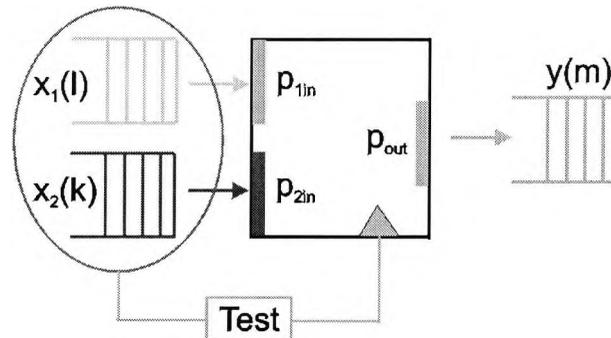
Figure 6.3: Stream Driven Process

The third main part of COSSAP is the data visualisation tool DAVIS (Fig. 6.4). DAVIS provides graphical depiction, visual analysis, and post-processing of the data. DAVIS also includes a calculator to interact with graphs. The data can originate from simulation results, dataset information, or other data that can be specified. Because DAVIS can be used to depict data generated independently of COSSAP, to add text and labels, and to print and save graphs, it can also be used as a general graphical tool for purposes that are not related to COSSAP.

After the complex system of algorithms has been described and the performance and behaviour have been verified with COSSAP the implementation in hardware is possible. Implementation on programmable DSPs is done through COSSAP DSP code generation, where the C code is generated for a target processor. COSSAP supports optimised C code generation for DSP target processors from several manufacturers. Implementation onto hardwired DSP architectures, like FPGAs and ASICs, is done firstly by generating HDL code through the COSSAP HDL code generator, and secondly by compiling the architecture to logical gates through Synopsys' Design Compiler.

In the following sections, the procedure of creation, simulation, and analysis of a system is explained on selected examples from the research work.
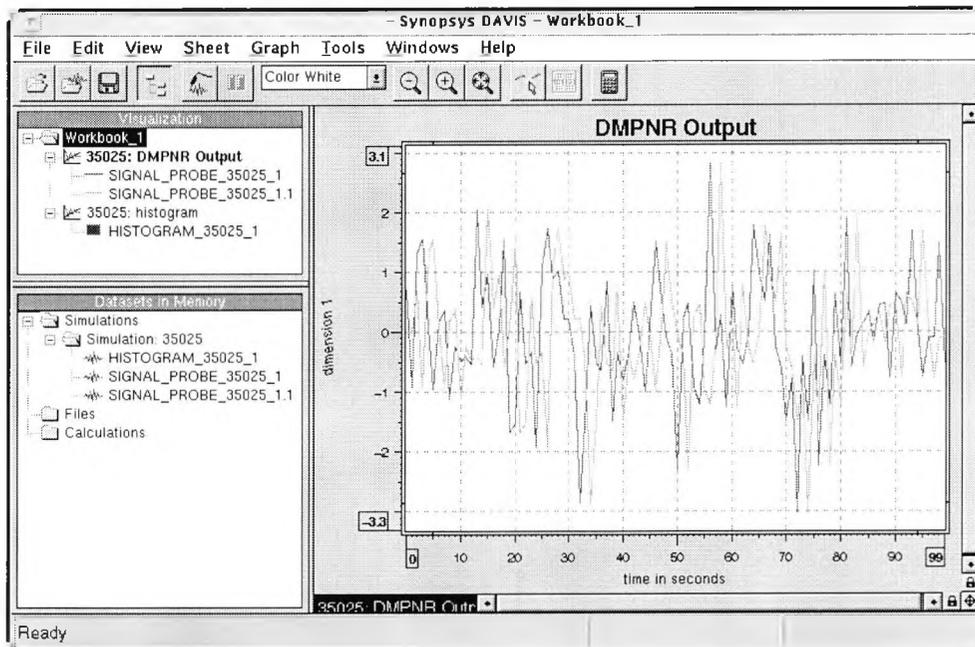
Figure 6.4: Display Window of the DAVIS tool

6.2 Floating-Point Modelling

6.2.1 Creation of the Models

Differential Demodulation

The block diagram given in Fig. 6.5 represents a complete communication system, which includes a pseudo random bit source, the OQPSK transmitter with a root raised cosine roll-off filter, an AWGN channel, a receiver filter, two further receiver blocks and a bit comparator with an error counter. All blocks, except for the two receiver blocks, are standard blocks and have been selected from the COSSAP model library. The new receiver algorithm, which is derived in Chapter 4, is implemented in accordance with Fig. 4.1 through two COSSAP models. The first block includes the differential
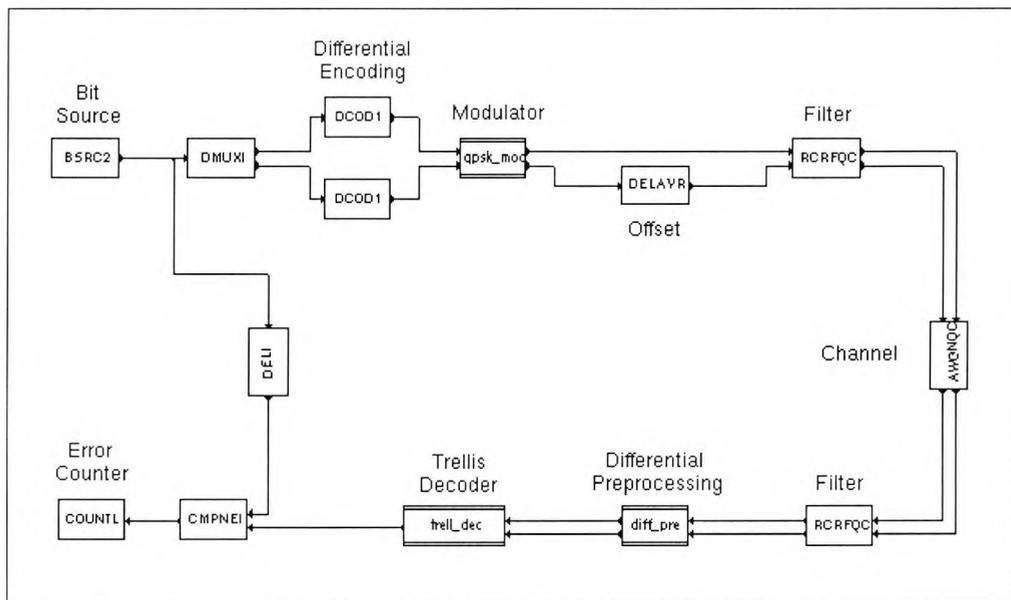
Figure 6.5: Block diagram of a DOQPSK system

preprocessing, which is independent from the subsequent trellis decoder algorithm. In the second block the different trellis decoding algorithms (8 and 128 state Viterbi and M-algorithm) are implemented.

The straightforward differential preprocessing is implemented as a hierarchical model. In Fig. 6.6 the analytical expression and the blocks, which define the hierarchical model diff_pre are given. In addition to the computation of the differential preprocessing two further functions are implemented in the model. The block SRDR performs a rate reduction from the sampling rate to the bit rate and the block DELR deletes the first

three bits. A deletion of the first three blocks is necessary, since the hierarchical model needs three bits to achieve the steady state, because of the history from the delay blocks.

$$X(I) = I(I)^*I(I-2) + Q(I)^*Q(I-2)$$
$$Y(I) = Q(I)^*I(I-2) - I(I)^*Q(I-2)$$
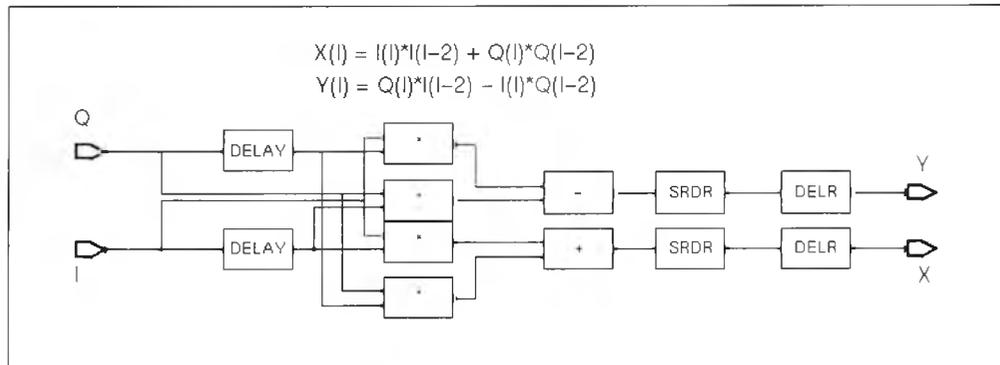
Figure 6.6: Hierarchical Model of the Differential Preprocessing

The block of the trellis decoder in Fig. 6.5 indicates by the two double lines, that it is also a hierarchical model. Unlike differential preprocessing, the actual decoder algorithm, which is the core signal processing element, is implemented as a primitive model. The hierarchical model trell_dec is merely composed of this primitive model and an additional block that deletes a number of bits which are necessary for the transients of the Viterbi decoder. The 8 state and 128 state Viterbi decoder are realised by a single model. In Appendix C the model definition file and the Generic C file are given. Generic C is a special COSSAP C dialect, which simplifies the creation of primitive models. Depending on the parameter value STATES and its input datasets, the model performs either the 8 state Viterbi for a roll-off factor of 1.0 or the 128 state Viterbi for roll-off factors less than 1.0. The trellis decoder using the M-algorithm is given in a second primitive model.

As mentioned in the last chapter, the quality of the C code has influence on the simulation time. The model m_alg is a good example to describe this influence. The first model m_alg, which has been created to investigate the M-algorithm, includes the standard sorting algorithm within the model's C code. An advanced model (m_alg_ss) uses function calls for the sorting operation and the current model m_alg_qs includes function calls for the quicksort algorithm. The required simulation times of the different models for different numbers of states with a simulation length of 10^6 bits are given in Tab. 6.1. The 128 state Viterbi decoder represents the reference in time and BER. The simulation is carried out with an E_s/N_0 of 10 dB and a roll-off factor of 0.35.

The table shows, that the M-algorithm with 32 states including the standard sorting algorithm within the C code obviously requires more time than the Viterbi algorithm. The longer computing time results from the additional complexity due to the sorting algorithm. The simple use of function calls for the sorting algorithm reduces the simulation time to 60 percent of the model including the sorting algorithm. A further gain is

	128 state Viterbi algorithm BER: $1.2142 \cdot 10^{-2}$		
time factor	32m 27s 1.0		
	32 state M-algorithm BER: $1.2168 \cdot 10^{-2}$		
	standard within the code	standard function calls	quicksort function calls
time factor	37m 54s 1.168	22m 28s 0.692	14m 32s 0.448
	16 state M-algorithm BER: $1.3826 \cdot 10^{-2}$		
	standard within the code	standard function calls	quicksort function calls
time factor	13m 52s 0.427	9m 49s 0.303	8m 17s 0.255
	14 state M-algorithm BER: $1.5481 \cdot 10^{-2}$		
	standard within the code	standard function calls	quicksort function calls
time factor	12m 06s 0.373	8m 58s 0.276	7m 52s 0.242

Table 6.1: Comparison of the required simulation time and the bit error rate of different implementation methods

obtained if the quicksort algorithm is implemented instead of the standard algorithm. The use of the 16 state M-algorithm using quicksort leads to an almost four times faster simulation time than the Viterbi algorithm, while the bit error rate shows only a small degradation.

This example should illustrate that a comparison in complexity between different algorithms solely on the basis of the simulation time is hardly useful. Additional information, which is independent from the implementation method, is required to make reliable statements about the complexity. Therefore, in Section 5.3, such information is used.

Coherent Demodulation and Equalisation

The floating-point investigations on the coherent receiver which includes channel equal-
isation on the principle of per-survivor processing were started with a primitive model
for a single path channel. The model was included in the same simulation environment
as the blocks of the differential demodulator. Several receiver models were then devel-
oped, which are adapted to specific environments like two or three path channels or
an echo delay of a symbol or bit period (T_s or T_b). The channel update in the models
is based either on the hybrid or on the LMS algorithm. During further development,
different models were merged into a general coherent adaptive Viterbi decoder model,
which is suitable for multipath channels. The code of one model, which is based on the
LMS algorithm, is given in Appendix C. Due to the use of vectors and pointers, the
parameter (NumOfStates) and the datasets can be used to vary the number of channel
paths and the echo delays.

6.2.2 Simulation

One important aspect, in order to get correct simulation results, is the choice of the
simulation length. On a pure AWGN channel a confidence interval can be used to de-
termine the number of bits, which are required to make a statement about the bit error
rate. In Fig. 6.7, confidence intervals for the Monte Carlo simulation technique based
on the Normal approximation are given. It should be noted that the figure shows the
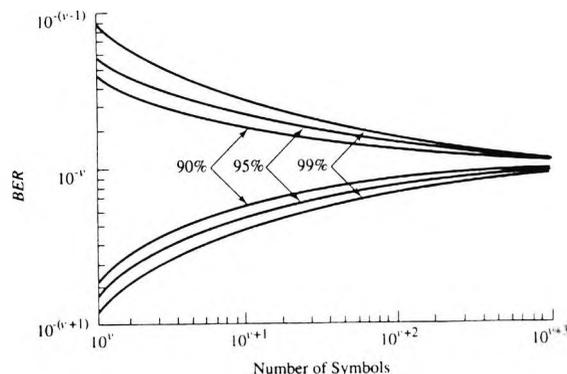
Figure 6.7: Confidence bands on BER when observed value is $10^{-\nu}$ for the Monte Carlo
technique based on the normal approximation (from [9])

confidence intervals for a single BER point. Typically, in a simulation BER estimates
are made at several values of E_s/N_0, and the values in between are interpolated. In

this case, for every value of E_s/N_0 a different number of symbols is required to achieve a value in the desired confidence interval. Therefore, the simulation results presented in this thesis are obtained through an error counter, which does not count predefined number of items, but stops the simulation when 1000 errors are obtained. A simulation with 1000 errors corresponds to $10^{\nu+3}$ symbols in Fig. 6.7, and therefore corresponds to a confidence interval $\pm 10\%$ around the measured BER.

This rule of thumb is valid only on an AWGN channel. The definition of the simulation length for a Rayleigh channel or even for a multipath channel is more complicated. The simulation length for a pure Rayleigh fading channel can be determined by using the error probability for a given threshold amplitude and the level crossing rate of the complex signal amplitude. The simulation length further increases if multipath channels are used. Here, an empirical estimate by increasing the simulation length and comparison of the obtained BER is used to define the simulation length in a two path Rayleigh fading channel. For multipath channels with more than two paths the simulation length increases in such a way, that a Monte Carlo simulation is not practicable anymore.

The floating-point investigations of the derived algorithms conclude the research work funded by the German Research Council which is the basis of this thesis. However, a subsequent research project funded by the German Ministry of Education and Research deals with the hardware implementation aspects of the developed algorithms. Since some results are already available from this project, they are mentioned at this point. In addition, the following sections should give an overview of future investigations in this area.

6.3 Fixed-Point Optimisation

6.3.1 General Procedure

A system on a chip implementation follows several iterative steps: algorithm design and redesign, fixed-point analysis and optimisation, VHDL implementation, Register Transfer Logic (RTL) design and further hardware oriented design steps. Since a fixed-point implementation may have influence on the overall system performance, this chapter deals with investigations on the receiver performance under finite word-length.

After the system functionality has been verified in a floating-point implementation and an optimum design solution has been obtained, quantisation is introduced and word-length optimisation is carried out. In this design phase each of the model blocks in the floating-point design, which have to be investigated, are replaced by a corresponding bittrue model. The bittrue models are either taken from the COSSAP bittrue library as predefined models, generated through hierarchical modeling of existing models, or developed from Generic C code. The fixed-point optimisation procedure allows to choose word-lengths such that the performance of the system stays within an acceptable margin. Following are the general steps for a fixed-point optimisation:

1. The statistical variances (σ^2) of the input and output signals of all models are measured.

2. The signal processing of the model to be optimised in word-length is replaced by a bittrue equivalent, while the rest of the models remain in floating point. Analog/digital converters (ADC) and digital/analog converters (DAC) are inserted at the input and output of the model to enable interaction between fixed- and floating-point models.

3. A simple and efficient procedure to define the dynamic range of the ADC and DAC, is to use three times of the standard deviation 3σ of the signal as maximum and minimum ranges. For a Gaussian distributed signal, the amplitude will be in this range for more than 99%. Clipping of the signal in the ADC will therefore only occur for one out of 100 samples.

4. Two's complement arithmetic is used during all phases of word-length optimisation by introducing an offset value of $-2^{(N-1)}$ for ADC and DAC, where N is the

word-length.

5. To represent fractions in fixed point, radix_fxp library models are used. For these models, where the number of bits to the left of the decimal point (NumIntWidth) has to be specified for the input and output signal, the NumIntWidth parameter is chosen such that the dynamic range is covered.

Maximum word-lengths are chosen in a first step to match the floating-point performance. A typical starting value is 16 bits for interface word-lengths between modules.

6.3.2 Input Word-Length Optimisation

The criteria used for input word-length optimisation are the measured SNR observed after quantisation at the output of the DAC and the bit error rate of the overall system.

Starting with the floating-point implementation of the design (Fig. 6.5), ADC and DAC blocks are inserted and put back-to-back at the input of the differential pre-processing. The maximum and minimum ranges are set to $+3\sigma$ and -3σ. Since the dynamic range of $\pm 3\sigma$ depends on both the signal energy and the noise power density of the AWGN channel, two values of 3σ are simulated and compared.

Figures 6.8 and 6.9 show the results for different input word-lengths and a fixed E_s/N_0 of 10 dB.
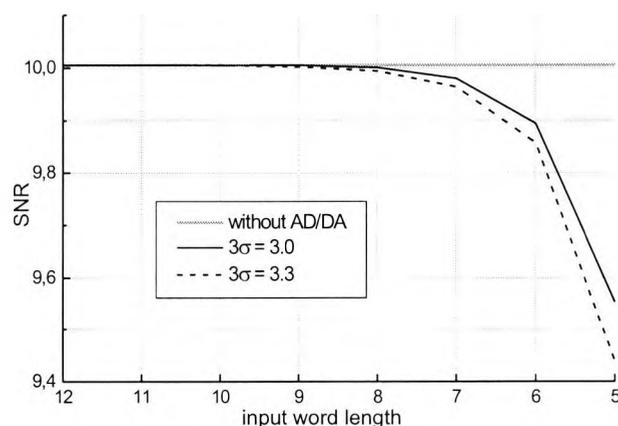
Figure 6.8: Measured SNR as a function of the word-length and of the chosen max/min range

A word-length of 8 bits and a range of ± 3 is chosen for the following investigations, because the SNR and the BER remain almost constant if more than 8 bits are used.

Figure 6.9: Bit error rate as a function of the word-length and of the chosen max/min range

However, for a smaller word-length than 8 bits the resolution of the signal and thus the SNR decreases and the bit error rate increases.

6.3.3 Word-Length Optimisation of the Model "Differential Preprocessing"

After the input word-length has been determined, the word-length of every block in the differential preprocessing model has to be optimised. Blocks with similar functions, which are arranged in parallel, can be considered together. This means, that the four MULT_FXP (multiplication) blocks in Fig. 6.10 can be optimised in one step, just like the blocks SUB_FXP (subtraction) and ADD_FXP (addition). The blocks DELAYB

Figure 6.10: Block diagram of the fixed-point implementation of the differential preprocessing

(delay), SRDR (sampling reduction) and DELR (delete) offer no possibility of a word-length modification. The word-length of these blocks depends only on the output

140

word-length of the preceding blocks.

The optimisation of fixed-point models mainly considers two aspects: the implementation of the truncation for large values and the type of rounding. The following modes are available in the COSSAP fixed-point models:

- Round Mode (RM)

    0 : Truncation of least significant bits (LSB).

    1 : Output values are rounded by adding 1/2 LSB (to the truncated output) and subsequent truncation of least significant bits (mathematical rounding). If necessary, clipping to the maximum positive value is done.

    2 : Positive values are rounded as for Round Mode = 1, while negative values are rounded by subtracting 1/2 LSB (of the truncated output) and subsequent truncation of least significant bits (symmetrical rounding). If necessary, clipping to maximum positive (minimum negative) value is done.

    3 : Output values are truncated by truncating the least significant bits of the absolute value and keeping the sign bit (truncation to zero).

- Saturation Mode (SM)

    0 : Truncation of most significant bits. No saturation occurs.

    1 : Output values greater than the maximum positive (less than minimum negative) value are replaced with the maximum positive value (minimum negative value).

    2 : Output values greater than the maximum positive (less than minimum negative) value are replaced with the maximum positive value (minimum negative value +1 LSB). This corresponds to a symmetrical clipping operation. This clipping does not introduce offset (if the input has a symmetrical amplitude distribution).

Note that the saturation parameter has no effect, if an overflow does not occur. This is especially the case when the output number of bits to the left of the decimal point is larger than the the required processing word-length.

Word-Length of the Models MULT_FXP

In the preceding subsection a range of $\pm 3 = 6$ and a word-length of 8 bits (256 steps) for the ADC have been determined, which results in a step size of 2.344^{-2}. A multiplication without rounding and saturation would lead to an output word-length which is the sum of the two input word-lengths. In this case an output word-length of 16 bits (65536 steps) would result, leading to a range of $6 \cdot 6 = 36$, and a step size of 5.493^{-4}. It can be shown with the quality measures BER and SNR that both the dynamics of ± 18 and the resolution are not required at this point. Therefore, a significant reduction of the output word-length can be achieved. In order to avoid complicated denotation the abbreviations given in Tab. 6.2 are employed for all further investigations.

InpWidth	:	word-length of the input bit vector
InpIntBits	:	number of integer bits of the input vector
OutWidth	:	word-length of the output bit vector
OutIntBits	:	number of integer bits of the output vector
RM	:	round mode
SM	:	saturation mode

Table 6.2: Abbreviations for the word-length optimisation

In the first fixed-point block following the ADC, the position of the decimal point can be chosen arbitrarily. This determination, however, has then to be considered in all following blocks. Here two of the eight bits of the input word are defined as bits to the left of the decimal point. Therefore, the optimisation of the block starts with the input declaration of:

InpWidth	:	8
InpIntBits	:	2

A rule of thumb for word-length optimisation including both rounding and saturation is, that rounding should always be done first followed by saturation. Table 6.3 gives the simulation results for different output word-lengths and round modes. Since the input word-length is 8, the investigation starts with the maximum output word-length of $8 + 8 = 16$. The table shows that a reduction of OutWidth to 9 bits results in a negligible degradation. A further reduction leads to an increase of the error rate. On the other hand, if a round mode of 0 or 3 is used, a noticeable increase in BER occurs even for OutWidth of 9. The results of mathematical rounding (RM=1) and

symmetrical rounding (RM=2) are comparable. However, since a symmetrical signal is used, the symmetrical round mode is chosen.

OutWidth	OutIntBits	RM	BER	SNR
16	4	0	8.6291^{-3}	6.7668
12	4	2	8.5946^{-3}	6.7713
11	4	2	8.5517^{-3}	6.7715
10	4	2	8.5517^{-3}	6.7764
9	**4**	**2**	$\mathbf{8.5350^{-3}}$	**6.7606**
8	4	2	8.8846^{-3}	6.7497
9	4	0	10.2502^{-3}	6.7430
9	4	1	8.5233^{-3}	6.7509
9	4	3	10.6406^{-3}	6.4579

Table 6.3: Rounding, model MULT_FXP

The reduction of OutIntBits influences the dynamic range of the model. The input signal of the differential preprocessing is symmetrical with a mean amplitude of 1.0. A multiplication of two input samples results in a sample, which again has a mean amplitude of 1.0. Therefore, the dynamic range of the output signals of the multiplication models can keep the same range as the input. Table 6.4 confirms this statement. The results of BER and SNR remain stable as long as OutIntBits $\geq$ InpIntBits $= 2$. A further reduction of OutIntBits leads to a distinct degradation. The actual saturation mode has no influence on the performance, since no overflow occurs. In conclusion the

OutWidth	OutIntBits	SM	BER	SNR
9	4	0	8.5350^{-3}	6.7606
8	3	0	8.5350^{-3}	6.7606
7	**2**	**0**	$\mathbf{8.5350^{-3}}$	**6.7606**
6	1	0	2.6419^{-3}	7.0726
7	2	1	8.5350^{-3}	6.7606
7	2	2	8.5350^{-3}	6.7606

Table 6.4: Saturation, model MULT_FXP

results of the word-length optimisation of the MULT_FXP models are given in Tab. 6.5.

Word-Length of the Models ADD_FXP and SUB_FXP

Since the steps to optimise word-length, round mode, and saturation mode are identical to those in the model MULT_FXP, the documentation of the word-length optimisation of the models ADD_FXP and SUB_FXP is limited to the presentation of the final

InpWidth	:	8
InpIntBits	:	2
OutWidth	:	7
OutIntBits	:	2
RM	:	2
SM	:	0
Range	:	$\pm 4.5 = 9$
Step size	:	7.031^{-2}

Table 6.5: Results of the word-length optimisation of the MULT_FXP models

results. These are given in Tab. 6.6. The word-length and the range of the DAC follow directly from these results.

InpWidth	:	7
InpIntBits	:	2
OutWidth	:	6
OutIntBits	:	2
RM	:	2
SM	:	1
Range	:	$\pm 4.5 = 9$
Step size	:	1.406^{-1}

Table 6.6: Results of the word-length optimisation of the ADD_FXP and SUB_FXP models

The above word-length optimisation of the differential demodulation could simply be carried out since all models were available as fixed-point blocks from a COSSAP library. The trellis decoder model, however, exists only as a floating-point block. Therefore, before a word-length optimisation of this model can be done, a new fixed-point block has to be created. This task is one part of the next, more practical, project following this thesis. The goal of the next project is to implement the algorithms, which have been derived in this thesis, into hardware, especially into an ASIC. For this job the algorithms have firstly to be implemented as fixed-point COSSAP models and investigations on the required word-length has to be carried out. The next step will be the implementation of the models in a hardware description language (VHDL). At this point additional Synopsys products have to be used for comparing the performance of the VHDL models with the COSSAP floating-point models. Starting with the VHDL models it is possible to synthesise the code and to create a gate map for a final ASIC design. The description of the tasks sounds simple, however, the realisation requires a large knowledge about the different aspects (fixed-point, VHDL, ASIC design) and therefore will be the task of a new project.

Chapter 7

Summary and Conclusion

This thesis contains the synthesis and analysis of new receivers for differentially encoded Offset-QPSK (OQPSK). The modulation format Offset-QPSK is characterised by two advantages compared to established mobile radio modulation formats. Firstly, the linear modulation format OQPSK has a better bandwidth and spectral efficiency than a nonlinear format such as, e.g., GMSK. Secondly, the inevitable fluctuations of the envelope are considerably smaller for OQPSK than in the case of other linear formats. A signal with reduced envelope fluctuations can be transmitted over nonlinear power efficient amplifiers without deterioration of its quality. Nonlinear amplifiers are attractive when mobiles with battery units are used. In order to exploit the bandwidth efficiency of linear formats in conjunction with reduced envelope fluctuations, the linear format $\pi/4$-QPSK was designed. $\pi/4$-QPSK is used in several modern communication systems. OQPSK has even less fluctuations of the envelope than $\pi/4$-QPSK. However, OQPSK is not used in modern digital radio systems, since no receivers were derived up to now, which fulfill the requirements for mobile radio communications.

To illustrate the problems of receiver design for digital signals and to give a survey to readers, who are not familiar with this specific topic, the first part of the thesis dealt with an introduction to digital modulation and optimum receiver structures. Mobile communication faces the problem of a communication medium which corrupts the signal in several ways. The mobile channel does not only introduce additive noise, but also produces a multipath propagation of the signal. Multipath reception may lead to the well known time selective fading, however, it may also lead to additional smearing of an impulse, which is called frequency selectivity. Furthermore, due to the use of nonlinear amplifiers the spectrum of the signal may be broadened, thus adjacent channel interference may occur. Therefore, in the basics chapter, mobile radio systems were defined with the emphasis on the description of mobile radio channels for different environments. In order to get a reference for the simulation results, which were obtained with the new receiver structures, the theoretical performance of PSK signals were given

145

to conclude the basics chapter.

The main part of the thesis started with the description of a DOQPSK transmitter. Offset-QPSK (or staggered QPSK) is a QPSK modulation format, where one quadrature arm is delayed by half a symbol period relative to the other quadrature arm, avoiding phase shifts of 180°. The new DOQPSK transmitter differs from the conventional OQPSK transmitter by differential precoding. Two new receiver structures for DOQPSK signals were synthesised and analysed. The first receiver, based on differential demodulation, is intended to be used for purely time selective channels, thus it is suited for mobile systems where the ratio of maximum multipath delay to symbol period is small. The second receiver can be used both for time and frequency selective mobile radio channels and jointly applies trellis decoding and channel estimation.

The first receiver initially performs a differential demodulation in the complex baseband. The cross channel interference between the I- and Q-channel, which results from the offset of one channel, was then reduced through a succeeding trellis decoder. The number of states and the metrics in the trellis decoder depend on the filters in the transmitter and the receiver. If Nyquist filters with a roll-off factor of 1.0 were used, then an 8 state trellis decoder was sufficient. For a roll-off factor less than 1.0, trellis decoders with 128 states were examined. In this case, the trellis decoder with the Viterbi algorithm based on Euclidean metrics considers only a part of the impulse response. Therefore, and since the nonlinear processing within the differential demodulation destroys the assumption of the Euclidean metrics, the receivers are not optimal. However, the investigations of the DOQPSK receivers showed only a small degradation of the BER in time selective linear channels compared to $\pi/4$-DQPSK. A complexity reduction of the Viterbi algorithm with 128 states was achieved with the aid of the M-algorithm. The investigations have shown that a 16 state M-algorithm resulted in a gain of four in simulation time, while only a further small degradation in the bit error rate was obtained. Due to the fact that the receiver uses differential demodulation, the structure is only suitable for purely time selective channels. Therefore, for time and frequency selective channels a second receiver structure was derived.

The second receiver structure does not apply differential preprocessing and is based on algorithms for adaptive channel estimation and data decoding of DOQPSK signals. Decoding and channel estimation are carried out jointly and blind in a trellis decoder. Blind means, that no training sequence is necessary. In this receiver a channel impulse

response is computed adaptively for every state in the trellis according to the principle of per-survivor processing (PSP). Different algorithms were considered to estimate the channel impulse response within the Viterbi algorithm. The well known LMS algorithm was adapted to the requirements of PSP and a new hybrid algorithm was derived. Both algorithms were initially investigated in time selective channels. A trellis decoder with 4 states was synthesised for Nyquist filters with $r = 1.0$. In contrast to the differential receiver structure, the use of mismatched filters with roll-off factors less than 1.0 produced only minor degradations in the BER results. Thus, a trellis decoder with more states, which considers a longer impulse response, was not necessary.

For both time and frequency selective channels a simplified Kalman algorithm was derived. The new algorithm considers multipath Rayleigh fading signals. The Rayleigh fading process of each multipath component was modelled by a first order Markov process. Averaging procedures in the receiver derivation of the Kalman estimator led to the simplified Kalman algorithm. The most important result was, that a simple extension of the conventional LMS and hybrid algorithm could be found. In contrast to the standard LMS algorithm, each component of the multipath channel is calculated with a specific step size, which is adapted to the normalised power of this component. Applying the simplified Kalman algorithm resulted in a diversity gain in frequency selective channels compared to the standard LMS and hybrid algorithm. The diversity gain depended on the power and the delay of the echo path. It could further be shown, that the performance of the algorithm was close to the theoretical bounds of multipath Rayleigh fading channels. Due to the offset of one component in the OQPSK signal, it was furthermore possible to resolve echo paths with a delay of multiples of half a symbol period with the new receiver structure. This fact is an additional benefit compared to a coherent $\pi/4$-DQPSK receiver which can merely resolve echo delays of multiples of the symbol period.

An important feature besides the performance of a new receiver structure is the realisation complexity. A comparison between the two receivers was carried out. The relevant parameter which defines the complexity is the number of states in the receiver. The receiver with joint channel estimation and data detection needs 4 states in the minimum. The number of states is then increased by the power of 2 with every additional echo delay of $T_s/2$. On the other hand the receiver with differential demodulation requires 8 states, when the roll-off factor of the Nyquist filters is one. For smaller roll-off

factors the number of states increases as mentioned above.

The BER investigations with the two receiver structures have shown that the algorithm with differential demodulation allows higher velocities of the mobile than the PSP based algorithm. This is because in differential demodulation the channel phase does not influence the decoding process as long as the channel phase is almost constant for two succeeding symbols. Therefore, in time selective channels the tracking performance of the receiver with differential demodulation was observed to outperform the other receiver. Another important feature besides tracking is the acquisition performance. Acquisition is a major topic in burst mode transmission. Acquisition of the differential demodulation algorithm is dominated merely by the intersymbol interference through the differential demodulation process. Since the major interference is found within three bits, acquisition is a minor problem for the differential demodulator. A similar acquisition performance could be observed for the receiver with PSP. The simplified Kalman algorithm which was adapted to the LMS and the hybrid algorithm, was investigated in several environments and the results for both algorithms confirmed the statement, that no training sequence was required. The acquisition time can be bridged with some tail bits.

The simulation results of the two new receiver structures in time selective and in time and frequency selective channels showed, that only a small degradation in linear channels was obtained compared to the $\pi/4$-DQPSK receivers and the theoretical bounds. The advantage of the smaller fluctuations of the envelope of DOQPSK is relevant, when nonlinear amplifiers are used. Therefore, a generic amplifier model was defined to investigate the performance of the receivers in the weak conduction region and when saturation occurs. As was expected from the amplitude distribution of the transmitted signal, in the weak conduction region the DOQPSK receivers achieved a gain of several dB compared to the corresponding $\pi/4$-DQPSK receivers. The measured gain depended on the roll-off factors of the Nyquist filters. On the other hand, the amplitude distributions of both modulation formats are comparable for higher amplitudes. Therefore, the expected gain was reduced when saturation occurred. A distinct benefit could only be obtained for higher roll-off factors.

It can be concluded, that two new receiver structures for differentially encoded OQPSK signals were synthesised and analysed. The new receiver structures showed respectable results in different mobile radio environments and proved their superiority

148

in nonlinear channels over other linear modulation formats. At the end of the thesis some aspects for further investigations on the hardware implementation of the receiver algorithms were discussed. In order to give a statement about the hardware complexity, further work on the implementation is required. Only with these results will it be possible to estimate whether the algorithms have a chance for implementation in future mobile radio systems.

Appendix A

Nonlinear Distortions

In mobile radio transmitters power efficient amplifiers should be used to amplify the signal. For such a situation it is desirable that the output is an amplified true reproduction of the input signal. Therefore, the amplifier must operate as a linear device. However, the inherent nonlinearity of real amplifiers results in an output which is a distorted version of the input. In Fig. A.1 a blockdiagram of a nonlinear system is illustrated. For an real input signal X the output signal is $Y = F(X)$, where F denotes the nonlinear transfer characteristics of the system.
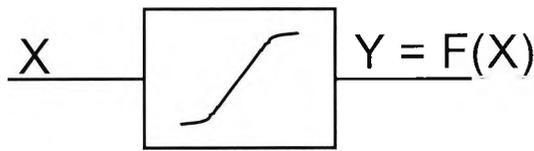
Figure A.1: Nonlinear System

When a sinusoidal signal of a single frequency is applied at the input of a nonlinear system, the resulting output contains frequency components that are integer multiples of the input signal. The harmonics are generated by the nonlinearity of the system and therefore the distortion is called harmonic distortion. The harmonic distortion is measured by comparing the magnitudes of the harmonics with the fundamental component (input frequency) of the output.

Consider the input signal to be the form

$$x(t) = X_1 \cdot \cos(\omega_1 t) \tag{A.1}$$

where $f_1 = \omega_1/2\pi$ is the frequency and X_1 is the amplitude of the input signal. Let the output of the nonlinear system be

$$y(t) = Y_0 + Y_1 \cdot \cos(\omega_1 t) + Y_2 \cdot \cos(2\omega_1 t) + Y_3 \cdot \cos(3\omega_1 t) + \ldots \tag{A.2}$$

where Y_0 is the dc (direct current) component of the output, Y_1 is the amplitude of the fundamental component, and Y_2, Y_3 are the amplitude of the second and third

harmonic components. The harmonic distortion of the nth harmonic is defined as [19]

$$HD_n = \frac{|Y_n|}{|Y_1|} \tag{A.3}$$

The total harmonic distortion (THD) is defined as to be the ratio of the rms (root-mean-square) value of the harmonics to the amplitude of the fundamental component.

$$THD = \frac{\sqrt{Y_2^2 + Y_3^2 + \cdots + Y_n^2}}{|Y_1|} = \sqrt{HD_2^2 + HD_3^2 + \cdots + HD_n^2} \tag{A.4}$$

In a real system more then a single frequency sinusoidal signal is applied at the input. In such a case the nonlinearity results not only on the fundamental and harmonics but also additional frequencies called beat frequencies at the output. The distortion due to the components at the beat frequencies is defined as intermodulation distortion. To characterise this type of distortion consider the input signal of the nonlinear system to be

$$x(t) = X_1 \cdot \cos(\omega_1 t) + X_2 \cdot \cos(\omega_2 t) \tag{A.5}$$

where $f_1 = \omega_1/2\pi$ and $f_2 = \omega_2/2\pi$ are the two input frequencies. The spectrum of the output signal shows in addition to the harmonics of the two frequencies ($n \cdot f_1$ and $n \cdot f_2$) also terms at multiples of the sum and difference frequencies $n \cdot (f_1 + f_2)$ and $n \cdot (f_1 - f_2)$.

Appendix B

Trellis for a roll-off Factor of 0.35

B.1 Equations of the nominal values $\overline{X}_l$ and $\overline{Y}_l$

Sampling intervals $l = 2k$:

$$
\begin{aligned}
\overline{X}_{2k} &= \Re\left\{r_{2k} \cdot r_{2k-2}^*\right\} \\
&= x_{2k}x_{2k-2} + g_1^2(y_{2k-1} + y2k + 1)(y_{2k-3} + y_{2k-1}) \\
&\quad + g_1 g_3\left[(y_{2k-1} + y_{2k+1})(y_{2k-5} + y_{2k+1}) + (y_{2k-3} + y_{2k+3})(y_{2k-3} + y_{2k-1})\right] \\
&\quad + g_3^2(y_{2k-3} + y_{2k+3})(y_{2k-5} + y_{2k+1})
\end{aligned}
$$

(B.1)

$$
\begin{aligned}
\overline{Y}_{2k} &= \Im\left\{r_{2k} \cdot r_{2k-2}^*\right\} \\
&= g_1 x_{2k-2}(y_{2k-1} + y_{2k+1}) + g_3 x_{2k_2}(y_{2k-3} + y_{2k+3}) \\
&\quad - g_1 x_{2k}(y_{2k-3} + y_{2k-1}) - g_3 x_{2k}(y_{2k-5} + y_{2k+1})
\end{aligned}
$$

(B.2)

Sampling intervals $l = 2k + 1$:

$$
\begin{aligned}
\overline{X}_{2k+1} &= \Re\left\{r_{2k+1} \cdot r_{2k-1}^*\right\} \\
&= y_{2k+1}y_{2k-1} + g_1^2(x_{2k} + x_{2k+2})(x_{2k-2} + x_{2k}) \\
&\quad + g_1 g_3([(x_{2k} + x_{2k+2})(x_{2k-4} + x_{2k+2}) + (x_{2k-2} + x_{2k+4})(x_{2k-2} + x_{2k})] \\
&\quad + g_3^2(x_{2k-2} + x_{2k+4})(x_{2k-4} + x_{2k+2})
\end{aligned}
$$

(B.3)

$$
\begin{aligned}
\overline{Y}_{2k+1} &= \Im\left\{r_{2k+1} \cdot r_{2k-1}^*\right\} \\
&= -g_1 y_{2k-1}(x_{2k} + x2k + 2) - g_3 y_{2k-1}(x_{2k-2} + x_{2k+4}) \\
&\quad + g_1 y_{2k+1}(x_{2k-2} + x_{2k}) + g_3 y_{2k+1}(x_{2k-4} + x_{2k+2})
\end{aligned}
$$

(B.4)

where $g_1 = 0.61858$ and $g_3 = -0.16244$ for a roll-off factor of $r = 0.35$.

B.2 Trellis table

initial state		final state		$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
0	1 1 1 1 1 1 1 1	0	1 1 1 1 1 1 1 1	1.832	0.000
	1 1 1 1 1 1 1 1	1	1 1 1 1 1 1 1 1-1	2.129	0.325
1	1 1 1 1 1 1 1-1	2	1 1 1 1 1 1 1-1 1	1.832	0.000
	1 1 1 1 1 1 1-1	3	1 1 1 1 1 1 1-1-1	2.129	0.325
2	1 1 1 1 1 1-1 1	4	1 1 1 1 1-1 1 1	0.598	-1.562
	1 1 1 1 1 1-1 1	5	1 1 1 1 1-1 1 1-1	1.000	-1.237
3	1 1 1 1 1 1-1-1	6	1 1 1 1 1-1-1 1	0.598	-1.562
	1 1 1 1 1 1-1-1	7	1 1 1 1 1-1-1-1	1.000	-1.237
4	1 1 1 1 1-1 1 1	8	1 1 1 1-1 1 1 1	-0.168	1.825
	1 1 1 1 1-1 1 1	9	1 1 1 1-1 1 1 1-1	0.129	2.149
5	1 1 1 1 1-1 1-1	10	1 1 1 1-1 1 1-1 1	-0.168	1.825
	1 1 1 1 1-1 1-1	11	1 1 1 1-1 1 1-1-1	0.129	2.149
6	1 1 1 1 1-1-1 1	12	1 1 1 1-1-1 1 1	-1.402	0.912
	1 1 1 1 1-1-1 1	13	1 1 1 1-1-1 1 1-1	-1.000	1.237
7	1 1 1 1 1-1-1-1	14	1 1 1 1-1-1-1 1	-1.402	0.912
	1 1 1 1 1-1-1-1	15	1 1 1 1-1-1-1-1	-1.000	1.237
8	1 1 1 1-1 1 1 1	16	1 1 1-1 1 1 1 1	1.106	0.000
	1 1 1 1-1 1 1 1	17	1 1 1-1 1 1 1 1-1	1.000	0.325
9	1 1 1 1-1 1 1-1	18	1 1 1-1 1 1 1-1 1	1.106	0.000
	1 1 1 1-1 1 1-1	19	1 1 1-1 1 1 1-1-1	1.000	0.325
10	1 1 1 1-1 1-1 1	20	1 1 1-1 1-1 1 1	1.000	-1.562
	1 1 1 1-1 1-1 1	21	1 1 1-1 1-1 1 1-1	1.000	-1.237
11	1 1 1 1-1 1-1-1	22	1 1 1-1 1-1-1 1	1.000	-1.562
	1 1 1 1-1 1-1-1	23	1 1 1-1 1-1-1-1	1.000	-1.237
12	1 1 1 1-1-1 1 1	24	1 1 1-1-1 1 1 1	-0.894	-0.650
	1 1 1 1-1-1 1 1	25	1 1 1-1-1 1 1 1-1	-1.000	-0.325
13	1 1 1 1-1-1 1-1	26	1 1 1-1-1 1 1-1 1	-0.894	-0.650
	1 1 1 1-1-1 1-1	27	1 1 1-1-1 1 1-1-1	-1.000	-0.325
14	1 1 1 1-1-1-1 1	28	1 1 1-1-1-1 1 1	-1.000	-1.562
	1 1 1 1-1-1-1 1	29	1 1 1-1-1-1 1 1-1	-1.000	-1.237
15	1 1 1 1-1-1-1-1	30	1 1 1-1-1-1-1 1	-1.000	-1.562
	1 1 1 1-1-1-1-1	31	1 1 1-1-1-1-1-1	-1.000	-1.237
16	1 1 1-1 1 1 1 1	32	1 1-1 1 1 1 1 1	-0.168	-1.825
	1 1 1-1 1 1 1 1	33	1 1-1 1 1 1 1 1-1	0.129	-2.149
17	1 1 1-1 1 1 1-1	34	1 1-1 1 1 1 1-1 1	-0.168	-1.825
	1 1 1-1 1 1 1-1	35	1 1-1 1 1 1 1-1-1	0.129	-2.149
18	1 1 1-1 1 1-1 1	36	1 1-1 1 1-1 1 1	-1.402	-0.912
	1 1 1-1 1 1-1 1	37	1 1-1 1 1-1 1 1-1	-1.000	-1.237
19	1 1 1-1 1 1-1-1	38	1 1-1 1 1-1-1 1	-1.402	-0.912
	1 1 1-1 1 1-1-1	39	1 1-1 1 1-1-1-1	-1.000	-1.237

	initial state		final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
41	1 1-1 1-1 1 1-1	82	1-1 1-1 1 1-1 1	1.000	1.562
	1 1-1 1-1 1 1-1	83	1-1 1-1 1 1-1-1	0.493	1.887
42	1 1-1 1-1 1-1 1	84	1-1 1-1 1-1 1 1	2.531	0.000
	1 1-1 1-1 1-1 1	85	1-1 1-1 1-1 1-1	2.129	0.325
43	1 1-1 1-1 1-1-1	86	1-1 1-1 1-1-1 1	2.531	0.000
	1 1-1 1-1 1-1-1	87	1-1 1-1 1-1-1-1	2.129	0.325
44	1 1-1 1-1-1 1 1	88	1-1 1-1-1 1 1 1	-1.000	-1.562
	1 1-1 1-1-1 1 1	89	1-1 1-1-1 1 1-1	-1.507	-1.237
45	1 1-1 1-1-1 1-1	90	1-1 1-1-1 1-1 1	-1.000	-1.562
	1 1-1 1-1-1 1-1	91	1-1 1-1-1 1-1-1	-1.507	-1.237
46	1 1-1 1-1-1-1 1	92	1-1 1-1-1-1 1 1	0.531	-2.474
	1 1-1 1-1-1-1 1	93	1-1 1-1-1-1 1-1	0.129	-2.149
47	1 1-1 1-1-1-1-1	94	1-1 1-1-1-1-1 1	0.531	-2.474
	1 1-1 1-1-1-1-1	95	1-1 1-1-1-1-1-1	0.129	-2.149
48	1 1-1-1 1 1 1 1	96	1-1-1 1 1 1 1 1	-1.402	-0.912
	1 1-1-1 1 1 1 1	97	1-1-1 1 1 1 1-1	-1.507	-1.237
49	1 1-1-1 1 1 1-1	98	1-1-1 1 1 1-1 1	-1.402	-0.912
	1 1-1-1 1 1 1-1	99	1-1-1 1 1 1-1-1	-1.507	-1.237
50	1 1-1-1 1 1-1 1	100	1-1-1 1 1-1 1 1	-1.000	0.000
	1 1-1-1 1 1-1 1	101	1-1-1 1 1-1 1-1	-1.000	-0.325
51	1 1-1-1 1 1-1-1	102	1-1-1 1 1-1-1 1	-1.000	0.000
	1 1-1-1 1 1-1-1	103	1-1-1 1 1-1-1-1	-1.000	-0.325
52	1 1-1-1 1-1 1 1	104	1-1-1 1-1 1 1 1	0.598	-1.562
	1 1-1-1 1-1 1 1	105	1-1-1 1-1 1 1-1	0.493	-1.887
53	1 1-1-1 1-1 1-1	106	1-1-1 1-1 1-1 1	0.598	-1.562
	1 1-1-1 1-1 1-1	107	1-1-1 1-1 1-1-1	0.493	-1.887
54	1 1-1-1 1-1-1 1	108	1-1-1 1-1-1 1 1	1.000	0.000
	1 1-1-1 1-1-1 1	109	1-1-1 1-1-1 1-1	1.000	-0.325
55	1 1-1-1 1-1-1-1	110	1-1-1 1-1-1-1 1	1.000	0.000
	1 1-1-1 1-1-1-1	111	1-1-1 1-1-1-1-1	1.000	-0.325
56	1 1-1-1-1 1 1 1	112	1-1-1-1 1 1 1 1	-1.000	1.562
	1 1-1-1-1 1 1 1	113	1-1-1-1 1 1 1-1	-1.507	1.237
57	1 1-1-1-1 1 1-1	114	1-1-1-1 1 1-1 1	-1.000	1.562
	1 1-1-1-1 1 1-1	115	1-1-1-1 1 1-1-1	-1.507	1.237
58	1 1-1-1-1 1-1 1	116	1-1-1-1 1-1 1 1	0.531	2.474
	1 1-1-1-1 1-1 1	117	1-1-1-1 1-1 1-1	0.129	2.149
59	1 1-1-1-1 1-1-1	118	1-1-1-1 1-1-1 1	0.531	2.474
	1 1-1-1-1 1-1-1	119	1-1-1-1 1-1-1-1	0.129	2.149
60	1 1-1-1-1-1 1 1	120	1-1-1-1-1 1 1 1	1.000	-1.562
	1 1-1-1-1-1 1 1	121	1-1-1-1-1 1 1-1	0.493	-1.887
61	1 1-1-1-1-1 1-1	122	1-1-1-1-1 1-1 1	1.000	-1.562
	1 1-1-1-1-1 1-1	123	1-1-1-1-1 1-1-1	0.493	-1.887

	initial state		final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
62	1 1-1-1-1-1-1 1	124	1-1-1-1-1-1 1 1	2.531	0.000
	1 1-1-1-1-1-1 1	125	1-1-1-1-1-1 1-1	2.129	-0.325
63	1 1-1-1-1-1-1-1	126	1-1-1-1-1-1-1 1	2.531	0.000
	1 1-1-1-1-1-1-1	127	1-1-1-1-1-1-1-1	2.129	-0.325
64	1-1 1 1 1 1 1 1	127	-1 1 1 1 1 1 1 1	1.832	0.000
	1-1 1 1 1 1 1 1	126	-1 1 1 1 1 1 1-1	2.129	0.325
65	1-1 1 1 1 1 1-1	125	-1 1 1 1 1 1 1-1 1	1.832	0.000
	1-1 1 1 1 1 1-1	124	-1 1 1 1 1 1-1-1	2.129	0.325
66	1-1 1 1 1 1-1 1	123	-1 1 1 1 1-1 1 1	0.598	-1.562
	1-1 1 1 1 1-1 1	122	-1 1 1 1 1-1 1-1	1.000	-1.237
67	1-1 1 1 1 1-1-1	121	-1 1 1 1 1-1-1 1	0.598	-1.562
	1-1 1 1 1 1-1-1	120	-1 1 1 1 1-1-1-1	1.000	-1.237
68	1-1 1 1 1-1 1 1	119	-1 1 1 1-1 1 1 1	-0.168	1.825
	1-1 1 1 1-1 1 1	118	-1 1 1 1-1 1 1-1	0.129	2.149
69	1-1 1 1 1-1 1-1	117	-1 1 1 1-1 1-1 1	-0.168	1.825
	1-1 1 1 1-1 1-1	116	-1 1 1 1-1 1-1-1	0.129	2.149
70	1-1 1 1 1-1-1 1	115	-1 1 1 1-1-1 1 1	-1.402	0.912
	1-1 1 1 1-1-1 1	114	-1 1 1 1-1-1 1-1	-1.000	1.237
71	1-1 1 1 1-1-1-1	113	-1 1 1 1-1-1-1 1	-1.402	0.912
	1-1 1 1 1-1-1-1	112	-1 1 1 1-1-1-1-1	-1.000	1.237
72	1-1 1 1-1 1 1 1	111	-1 1 1-1 1 1 1 1	1.106	0.000
	1-1 1 1-1 1 1 1	110	-1 1 1-1 1 1 1-1	1.000	0.325
73	1-1 1 1-1 1 1-1	109	-1 1 1-1 1 1-1 1	1.106	0.000
	1-1 1 1-1 1 1-1	108	-1 1 1-1 1 1-1-1	1.000	0.325
74	1-1 1 1-1 1-1 1	107	-1 1 1-1 1-1 1 1	1.000	-1.562
	1-1 1 1-1 1-1 1	106	-1 1 1-1 1-1 1-1	1.000	-1.237
75	1-1 1 1-1 1-1-1	105	-1 1 1-1 1-1-1 1	1.000	-1.562
	1-1 1 1-1 1-1-1	104	-1 1 1-1 1-1-1-1	1.000	-1.237
76	1-1 1 1-1-1 1 1	103	-1 1 1-1-1 1 1 1	-0.894	-0.650
	1-1 1 1-1-1 1 1	102	-1 1 1-1-1 1 1-1	-1.000	-0.325
77	1-1 1 1-1-1 1-1	101	-1 1 1-1-1 1-1 1	-0.894	-0.650
	1-1 1 1-1-1 1-1	100	-1 1 1-1-1 1-1-1	-1.000	-0.325
78	1-1 1 1-1-1-1 1	99	-1 1 1-1-1-1 1 1	-1.000	-1.562
	1-1 1 1-1-1-1 1	98	-1 1 1-1-1-1 1-1	-1.000	-1.237
79	1-1 1 1-1-1-1-1	97	-1 1 1-1-1-1-1 1	-1.000	-1.562
	1-1 1 1-1-1-1-1	96	-1 1 1-1-1-1-1-1	-1.000	-1.237
80	1-1 1-1 1 1 1 1	95	-1 1-1 1 1 1 1 1	-0.168	-1.825
	1-1 1-1 1 1 1 1	94	-1 1-1 1 1 1 1-1	0.129	-2.149
81	1-1 1-1 1 1 1-1	93	-1 1-1 1 1 1 1-1	-0.168	-1.825
	1-1 1-1 1 1 1-1	92	-1 1-1 1 1 1 1-1-1	0.129	-2.149
82	1-1 1-1 1 1-1 1	91	-1 1-1 1 1-1 1 1	-1.402	-0.912
	1-1 1-1 1 1-1 1	90	-1 1-1 1 1-1 1-1	-1.000	-1.237

	initial state		final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
83	1-1 1-1 1 1 1-1-1	89	-1 1-1 1 1 1-1-1 1	-1.402	-0.912
	1-1 1-1 1 1 1-1-1	88	-1 1-1 1 1 1-1-1-1	-1.000	-1.237
84	1-1 1-1 1-1 1 1	87	-1 1-1 1-1 1 1 1	1.832	0.000
	1-1 1-1 1-1 1 1	86	-1 1-1 1-1 1 1 1-1	2.129	-0.325
85	1-1 1-1 1-1 1-1	85	-1 1-1 1-1 1-1 1 1	1.832	0.000
	1-1 1-1 1-1 1-1	84	-1 1-1 1-1 1-1-1	2.129	-0.325
86	1-1 1-1 1-1-1 1	83	-1 1-1 1-1-1 1 1	0.598	1.562
	1-1 1-1 1-1-1 1	82	-1 1-1 1-1-1 1-1	1.000	1.237
87	1-1 1-1 1-1-1-1	81	-1 1-1 1-1-1-1 1	0.598	1.562
	1-1 1-1 1-1-1-1	80	-1 1-1 1-1-1-1-1	1.000	1.237
88	1-1 1-1-1 1 1 1	79	-1 1-1-1 1 1 1 1	-0.894	0.650
	1-1 1-1-1 1 1 1	78	-1 1-1-1 1 1 1 1-1	-1.000	0.325
89	1-1 1-1-1 1 1-1	77	-1 1-1-1 1 1 1-1 1	-0.894	0.650
	1-1 1-1-1 1 1-1	76	-1 1-1-1 1 1 1-1-1	-1.000	0.325
90	1-1 1-1-1 1-1 1	75	-1 1-1-1 1-1 1 1	-1.000	1.562
	1-1 1-1-1 1-1 1	74	-1 1-1-1 1-1 1-1	-1.000	1.237
91	1-1 1-1-1 1-1-1	73	-1 1-1-1 1-1-1 1	-1.000	1.562
	1-1 1-1-1 1-1-1	72	-1 1-1-1 1-1-1-1	-1.000	1.237
92	1-1 1-1-1-1 1 1	71	-1 1-1-1-1 1 1 1	1.106	0.000
	1-1 1-1-1-1 1 1	70	-1 1-1-1-1 1 1-1	1.000	-0.325
93	1-1 1-1-1-1 1-1	69	-1 1-1-1-1 1-1 1	1.106	0.000
	1-1 1-1-1-1 1-1	68	-1 1-1-1-1 1-1-1	1.000	-0.325
94	1-1 1-1-1-1-1 1	67	-1 1-1-1-1-1 1 1	1.000	1.562
	1-1 1-1-1-1-1 1	66	-1 1-1-1-1-1 1-1	1.000	1.237
95	1-1 1-1-1-1-1-1	65	-1 1-1-1-1-1-1 1	1.000	1.562
	1-1 1-1-1-1-1-1	64	-1 1-1-1-1-1-1-1	1.000	1.237
96	1-1-1 1 1 1 1 1	63	-1-1 1 1 1 1 1 1	0.598	1.562
	1-1-1 1 1 1 1 1	62	-1-1 1 1 1 1 1 1-1	0.493	1.887
97	1-1-1 1 1 1 1-1	61	-1-1 1 1 1 1 1-1 1	0.598	1.562
	1-1-1 1 1 1 1-1	60	-1-1 1 1 1 1 1-1-1	0.493	1.887
98	1-1-1 1 1 1-1 1	59	-1-1 1 1 1 1-1 1 1	1.000	0.000
	1-1-1 1 1 1-1 1	58	-1-1 1 1 1 1-1 1-1	1.000	0.325
99	1-1-1 1 1 1-1-1	57	-1-1 1 1 1 1-1-1 1	1.000	0.000
	1-1-1 1 1 1-1-1	56	-1-1 1 1 1 1-1-1-1	1.000	0.325
100	1-1-1 1 1-1 1 1	55	-1-1 1 1-1 1 1 1	-1.402	0.912
	1-1-1 1 1-1 1 1	54	-1-1 1 1-1 1 1 1-1	-1.507	1.237
101	1-1-1 1 1-1 1-1	53	-1-1 1 1-1 1 1-1 1	-1.402	0.912
	1-1-1 1 1-1 1-1	52	-1-1 1 1-1 1 1-1-1	-1.507	1.237
102	1-1-1 1 1-1-1 1	51	-1-1 1 1-1 1-1 1 1	-1.000	0.000
	1-1-1 1 1-1-1 1	50	-1-1 1 1-1 1-1 1-1	-1.000	0.325
103	1-1-1 1 1-1-1-1	49	-1-1 1 1-1-1-1 1	-1.000	0.000
	1-1-1 1 1-1-1-1	48	-1-1 1 1-1-1-1-1	-1.000	0.325

Continuation on the next page

	initial state		final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
104	1-1-1 1-1 1 1 1	47	-1-1 1-1 1 1 1 1	1.000	1.562
	1-1-1 1-1 1 1 1	46	-1-1 1-1 1 1 1-1	0.493	1.887
105	1-1-1 1-1 1 1-1	45	-1-1 1-1 1 1-1 1	1.000	1.562
	1-1-1 1-1 1 1-1	44	-1-1 1-1 1 1-1-1	0.493	1.887
106	1-1-1 1-1 1-1 1	43	-1-1 1-1 1-1 1 1	2.531	0.000
	1-1-1 1-1 1-1 1	42	-1-1 1-1 1-1 1-1	2.129	0.325
107	1-1-1 1-1 1-1-1	41	-1-1 1-1 1-1-1 1	2.531	0.000
	1-1-1 1-1 1-1-1	40	-1-1 1-1 1-1-1-1	2.129	0.325
108	1-1-1 1-1-1 1 1	39	-1-1 1-1-1 1 1 1	-1.000	-1.562
	1-1-1 1-1-1 1 1	38	-1-1 1-1-1 1 1-1	-1.507	-1.237
109	1-1-1 1-1-1 1-1	37	-1-1 1-1-1 1-1 1	-1.000	-1.562
	1-1-1 1-1-1 1-1	36	-1-1 1-1-1 1-1-1	-1.507	-1.237
110	1-1-1 1-1-1-1 1	35	-1-1 1-1-1-1 1 1	0.531	-2.474
	1-1-1 1-1-1-1 1	34	-1-1 1-1-1-1 1-1	0.129	-2.149
111	1-1-1 1-1-1-1-1	33	-1-1 1-1-1-1-1 1	0.531	-2.474
	1-1-1 1-1-1-1-1	32	-1-1 1-1-1-1-1-1	0.129	-2.149
112	1-1-1-1 1 1 1 1	31	-1-1-1 1 1 1 1 1	-1.402	-0.912
	1-1-1-1 1 1 1 1	30	-1-1-1 1 1 1 1-1	-1.507	-1.237
113	1-1-1-1 1 1 1-1	29	-1-1-1 1 1 1-1 1	-1.402	-0.912
	1-1-1-1 1 1 1-1	28	-1-1-1 1 1 1-1-1	-1.507	-1.237
114	1-1-1-1 1 1-1 1	27	-1-1-1 1 1-1 1 1	-1.000	0.000
	1-1-1-1 1 1-1 1	26	-1-1-1 1 1-1 1-1	-1.000	-0.325
115	1-1-1-1 1 1-1-1	25	-1-1-1 1 1-1-1 1	-1.000	0.000
	1-1-1-1 1 1-1-1	24	-1-1-1 1 1-1-1-1	-1.000	-0.325
116	1-1-1-1 1-1 1 1	23	-1-1-1 1-1 1 1 1	0.598	-1.562
	1-1-1-1 1-1 1 1	22	-1-1-1 1-1 1 1-1	0.493	-1.887
117	1-1-1-1 1-1 1-1	21	-1-1-1 1-1 1-1 1	0.598	-1.562
	1-1-1-1 1-1 1-1	20	-1-1-1 1-1 1-1-1	0.493	-1.887
118	1-1-1-1 1-1-1 1	19	-1-1-1 1-1-1 1 1	1.000	0.000
	1-1-1-1 1-1-1 1	18	-1-1-1 1-1-1 1-1	1.000	-0.325
119	1-1-1-1 1-1-1-1	17	-1-1-1 1-1-1-1 1	1.000	0.000
	1-1-1-1 1-1-1-1	16	-1-1-1 1-1-1-1-1	1.000	-0.325
120	1-1-1-1-1 1 1 1	15	-1-1-1-1 1 1 1 1	-1.000	1.562
	1-1-1-1-1 1 1 1	14	-1-1-1-1 1 1 1-1	-1.507	1.237
121	1-1-1-1-1 1 1-1	13	-1-1-1-1 1 1-1 1	-1.000	1.562
	1-1-1-1-1 1 1-1	12	-1-1-1-1 1 1-1-1	-1.507	1.237
122	1-1-1-1-1 1-1 1	11	-1-1-1-1 1-1 1 1	0.531	2.474
	1-1-1-1-1 1-1 1	10	-1-1-1-1 1-1 1-1	0.129	2.149
123	1-1-1-1-1 1-1-1	9	-1-1-1-1 1-1-1 1	0.531	2.474
	1-1-1-1-1 1-1-1	8	-1-1-1-1 1-1-1-1	0.129	2.149
124	1-1-1-1-1-1 1 1	7	-1-1-1-1-1 1 1 1	1.000	-1.562
	1-1-1-1-1-1 1 1	6	-1-1-1-1-1 1 1-1	0.493	-1.887

initial state	final state	$\overline{X}_{2k} = \overline{X}_{2k+1}$	$\overline{Y}_{2k} = -\overline{Y}_{2k+1}$
125 1-1-1-1-1-1 1-1	5 -1-1-1-1-1 1-1 1	1.000	-1.562
1-1-1-1-1-1 1-1	4 -1-1-1-1-1 1-1-1	0.493	-1.887
126 1-1-1-1-1-1-1 1	3 -1-1-1-1-1-1 1 1	2.531	0.000
1-1-1-1-1-1-1 1	2 -1-1-1-1-1-1 1-1	2.129	-0.325
127 1-1-1-1-1-1-1-1	1 -1-1-1-1-1-1-1 1	2.531	0.000
1-1-1-1-1-1-1-1	0 -1-1-1-1-1-1-1-1	2.129	-0.325

Table B.1: Trellis Table for a roll-off Factor of 0.35

Appendix C

Software Implementation of the Algorithms

C.1   Receiver I : Differential Demodulation

C.1.1   Viterbi Decoder

Definition File (mdef-file)

```
.NAME
 viterb

.SHORT_DESCRIPTION
 Viterbi decoder with variable number of states

.DESCRIPTION
 This models performs viterbi decoding for a DOQPSK-System.
 With the parameter STATES it can be selected a trellis decoder with
 8 states for a roll-off factor r = 1.0 or a decoder with 128 states
 for a roll-off factor r < 1.0.
 The input datasets contain the data table, which is used in the
 algorithm.

.IMPLEMENTATIONS
 1 SOURCE_CODE  C  viterb
   Simulation source code derived from the DSP generic C-code
 2 DSP_CODE_GENERIC GC viterb
   DSP generic C-code

.CLASSES
 1 DOQPSK

.PARAMETERS
 1 STATES (I)

.INPUT_DATASETS
 1  X_REF
 1.1.1  X_REF (R)
 1.2 NONE
 2  Y_REF
 2.1.1  Y_REF (R)
 2.2 NONE

.INPUT_PORTS
 1  X_IN (R) [1]
 2  Y_IN (R) [1]

.OUTPUT_PORTS
 1  DATA_OUT (I) [1]
```

```
.ERROR_CODES
 1 NONE

.HISTORY
  07.03.1996 S.Hischke

.END
```

Generic C File (gc-file)

```c
/***************************************************************/
/*                                                           */
/* Filename: viterb.gc                             */
/*                                                           */
/* Author:  Sven Hischke                           */
/* Company: Fachhochschule Giessen/Friedberg           */
/*                                                           */
/***************************************************************/
/*                                                           */
/*      Created: Mar-07-1996                            */
/*      Revisions: 1.2                                  */
/*      Name        Date        Reason                  */
/*      S.Hischke   Apr-29-1996                         */
/*      S.Hischke   Feb-24-1997  change of the computation of */
/*                               the detected data       */
/***************************************************************/
/*      IMPORTED LIBRARIES                              */
/***************************************************************/
#include <math.h>


/***************************************************************/
/*      COSSAP DECLARATIONS FOR COMPILATION WITH COINS     */
/***************************************************************/

PARAMETER(1) long no_states;

INPUT_DATASET(1.1) STATE float ref_x[no_states * 2];
INPUT_DATASET(2.1) STATE float ref_y[no_states * 2];

INPUT_PORT(1) float *X_in;
INPUT_PORT(2) float *Y_in;

OUTPUT_PORT(1) long *data_out;

BLOCKFACTOR int BlockFactor;

STATE float gamma[no_states];
STATE float gamma1[no_states];
STATE float gamma2[no_states];
STATE long info_reg_0[no_states];
STATE long info_reg_1[no_states];
STATE long transition_0[no_states];
STATE long transition_1[no_states];
STATE int sign;

/***************************************************************/
/*              Initialization Phase   init = 1           */
/***************************************************************/
void init_viterb()
{
  int i;

  for(i = 0;i < no_states; i++)
    {
    gamma[i] = 0;
```

163

```c
        info_reg_0[i] = 0;
        info_reg_1[i] = 0;
      }
    sign = 1;

/**********************************************************************/
/*                Calculation of the transition table               */

    for(i = 0;i < no_states; i++)
      {
        if((i % 2) == 0)
          {
            transition_0[i] = i/2;
            transition_1[i] = (no_states - 1) - (i / 2);
          }
        else
          {
            transition_0[i] = (i - 1) / 2;
            transition_1[i] = (no_states - 1) - ((i - 1) / 2);
          }
      }
}


/**********************************************************************/
/*                Signal Processing Phase   init = 2                */
/**********************************************************************/

void viterb()
{
  int    i, state, info_diff;
  float delta_x, delta_y, gamma_low;
  long  help_reg;
  SCRATCH float lambda[no_states * 2];

  LOOP(BlockFactor)

/**********************************************************************/
/*                BMU branch metric unit                            */
/**********************************************************************/

/**********************************************************************/
/*                Calculation of lambda                             */

    for(i = 0;i < (no_states * 2);i++)
      {
        delta_x = ref_x[i] - *X_in;
        delta_y = sign * ref_y[i] - *Y_in;
        lambda[i] = pow(delta_x,2) + pow(delta_y,2);
      }

/**********************************************************************/
/*                ACSU add compare select unit                      */
/**********************************************************************/

/**********************************************************************/
/*                Calculation of gamma                              */
```

```c
    for(i = 0;i < no_states;i++)
      {
       gamma1[i] = lambda[i*2] + gamma[transition_0[i]];
       gamma2[i] = lambda[i*2+1] + gamma[transition_1[i]];
      }

/**********************************************************************/
/*          Compare which gamma is the lower one             */
/*   The info_reg of the state with the lower gamma is saved      */

    for(i =0;i < no_states;i++)
      {
       if(gamma1[i] <= gamma2[i])
         {
          gamma[i] = gamma1[i];
          info_reg_1[i] = info_reg_0[transition_0[i]];
         }
       else
         {
          gamma[i] = gamma2[i];
          info_reg_1[i] = info_reg_0[transition_1[i]];
         }

/**********************************************************************/
/*    info_reg is shifted one to the left and the result of      */
/*       info_difF is placed as the LSB in the info_reg         */

       info_reg_1[i] = info_reg_1[i] << 1;
       info_diff = ((i & (no_states/4)) && 1) ^ ((i &
                  (no_states/16)) && 1);
       info_reg_1[i] = info_reg_1[i] + info_diff;
      }

/**********************************************************************/
/*              SMU survivor memory unit                   */
/**********************************************************************/

/**********************************************************************/
/*The contents of the new info_reg is saved into the old info_reg*/

    for(i = 0;i < no_states;i++)
      {
       info_reg_0[i] = info_reg_1[i];
      }

/**********************************************************************/
/*       Sorting of the state number and the gamma value       */
/*            of the state with the least gamma value          */

    state = 0;
    for(i = 1;i < no_states;i++)
      {
       if(gamma[state] > gamma[i])
         {
          state = i;
         }
      }
```

```
        gamma_low = gamma[state];

/***********************************************************************/
/*        Reduction of all gamma values by the least value           */

        for(i = 0; i < no_states;i++)
          {
           gamma[i] = gamma[i] - gamma_low;
          }

/***********************************************************************/
/*      The info_reg is shifted 31 to the right and the LSB is       */
/*                        given to the output                        */
/*      The buffers are incremented and the sign is changed          */

        help_reg = info_reg_0[state] >>31;
        *data_out = help_reg & 1;
        data_out++;
        X_in++;
        Y_in++;
        sign = sign * -1;

   ENDLOOP /* BlockFactor */
}

/***********************************************************************/
/* END OF FILE                                                       */
/***********************************************************************/
```

C.1.2 M-algorithm

Definition File (mdef-file)

```
.NAME
m_alg_qs

.SHORT_DESCRIPTION
M-Algorithm

.DESCRIPTION
This model includes the M-algorithm for decoding a DOQPSK system.
The data table, which is used in the algorithm, will be read from
the input datasets. The parameter STATES represent the reduced
number of states in the trellis and is dependent on the data table.
Sorting algorithm: quicksort

.IMPLEMENTATIONS
1 SOURCE_CODE   C  m_alg_qs
  Simulation source code derived from the DSP generic C-code
2 DSP_CODE_GENERIC GC m_alg_qs
  DSP generic C-code

.CLASSES
1 DOQPSK

.PARAMETERS
1 STATES (I)

.INPUT_DATASETS
1  X_REF_0
1.1.1  X_REF_0 (R)
1.2 NONE
2  X_REF_1
2.1.1  X_REF_1 (R)
2.2 NONE
3  Y_REF_0
3.1.1  Y_REF_0 (R)
3.2 NONE
4  Y_REF_1
4.1.1  Y_REF_1 (R)
4.2 NONE

.INPUT_PORTS
1  X_IN (R) [1]
2  Y_IN (R) [1]

.OUTPUT_PORTS
1  DATA_OUT (I) [1]

.ERROR_CODES
1 NONE

.HISTORY
  29.01.1998 S.Hischke

.END
```

Generic C File (gc-file)

```c
/********************************************************************/
/*                                                                */
/* Filename: m_alg_qs.gc                                  */
/*                                                                */
/* Author:  Sven Hischke                                  */
/* Company: Fachhochschule Giessen/Friedberg              */
/*                                                                */
/********************************************************************/
/*                                                                */
/*      Created: Jan-29-1998                              */
/*      Revisions: 1.0                                    */
/*      Name         Date         Reason               */
/*                                Quicksort             */
/********************************************************************/
/*      IMPORTED LIBRARIES                               */
/********************************************************************/
#include <math.h>
#include "my_h.h"
/********************************************************************/
/*      DEFINITIONS                                      */
/********************************************************************/
#define no_states 128


/********************************************************************/
/*      COSSAP DECLARATIONS FOR COMPILATION WITH COINS   */
/********************************************************************/

PARAMETER(1)  long red_states;

INPUT_DATASET(1.1) STATE float ref_x_0[no_states];
INPUT_DATASET(2.1) STATE float ref_x_1[no_states];
INPUT_DATASET(3.1) STATE float ref_y_0[no_states];
INPUT_DATASET(4.1) STATE float ref_y_1[no_states];

INPUT_PORT(1) float *X_in;
INPUT_PORT(2) float *Y_in;

OUTPUT_PORT(1) long *data_out;

BLOCKFACTOR int BlockFactor;

STATE float gamma_old[2*red_states];
STATE long info_reg[2*red_states];
STATE long transition_0[no_states];
STATE long transition_1[no_states];
STATE int sign;
STATE int state_old[red_states];

/********************************************************************/
/*                Initialization Phase   init = 1         */
/********************************************************************/

void init_m_alg_qs()
{
  int i;
```

```c
/*              Cration of the starting states              */

  for(i=0;i<red_states;i++)
   {
    state_old[i] = i;
   }

/*        Initialization of the gamma values with 0        */

  for(i = 0;i < 2*red_states; i++)
    {
     gamma_old[i] = 0;
     info_reg[i] = 0;
    }

  sign = 1;

/*********************************************************************/
/*              Calculation of the transition table              */

  for(i = 0;i < no_states/2; i++)
    {
     transition_0[i] = 2*i;
     transition_0[no_states-i-1] = 2*i + 1;
     transition_1[i] = 2*i + 1;
     transition_1[no_states-i-1] = 2*i;
    }
}

/*********************************************************************/
/*            Signal Processing Phase   init = 2              */
/*********************************************************************/

void m_alg_qs()
{
  int    i, j, k, m, info_diff, help_state;
  float delta_x,delta_y,help_gamma;
  long  help_reg;
  SCRATCH float lambda_0[red_states];
  SCRATCH float lambda_1[red_states];
  SCRATCH int state_new[2*red_states];
  SCRATCH float gamma_0[red_states];
  SCRATCH float gamma_1[red_states];

  LOOP(BlockFactor)

/*********************************************************************/
/*                Calculation of lambda                  */

    for(i = 0;i < red_states;i++)
      {
        delta_x = ref_x_0[state_old[i]] - *X_in;
        delta_y = sign * ref_y_0[state_old[i]] - *Y_in;
        lambda_0[i] = pow(delta_x,2) + pow(delta_y,2);

        delta_x = ref_x_1[state_old[i]] - *X_in;
```

```
        delta_y = sign * ref_y_1[state_old[i]] - *Y_in;
        lambda_1[i] = pow(delta_x,2) + pow(delta_y,2);
      }

/*****************************************************************/
/*                   Calculation of gamma                      */

    for(i = 0;i < red_states;i++)
      {
       gamma_0[i] = lambda_0[i] + gamma_old[i];
       gamma_1[i] = lambda_1[i] + gamma_old[i];
      }

    for(i = 0;i < red_states;i++)
      {
       state_new[i] = transition_0[state_old[i]];
       state_new[i+red_states] = transition_1[state_old[i]];
       gamma_old[i] = gamma_0[i];
       gamma_old[i+red_states] = gamma_1[i];
       info_reg[i+red_states] = info_reg[i];
      }

/*****************************************************************/
/*  Sorting the gamma values according to ascending precedence */

    sort_qs(0,2*red_states-1, gamma_old, state_new, info_reg);

/*****************************************************************/
/*        Checking, if two of the chosen transitions           */
/*                  end in the same state                      */

    i = 0;
    m = 0;
    while(i < red_states-1)
      {
       for(j = i+1;j < red_states;j++)
         if(state_new[i] == state_new[j])
           {
            for(k = j;k < 2*red_states-1;k++)
              {
               state_new[k] = state_new[k+1];
               gamma_old[k] = gamma_old[k+1];
               info_reg[k] = info_reg[k+1];
               m = 1;
              }
           }
       i++;
       if(m == 1)
         {
          i = 0;
          m = 0;
         }
      }

/*****************************************************************/
/*        Reduction of all gamma values by the least value     */
/*    info_reg is shifted one to the left and the result of    */
```

```
/*        info_diff is placed in the info_reg as the LSB         */

      help_gamma = gamma_old[0];
      for(i = 0; i< red_states;i++)
        {
        gamma_old[i] = gamma_old[i] - help_gamma;
        state_old[i] = state_new[i];
        info_reg[i] = info_reg[i]<<1;
        info_diff = ((state_old[i] & 32) && 1) ^
                    ((state_old[i] & 8) && 1);
        info_reg[i] = info_reg[i] + info_diff;
        }

/*****************************************************************/
/*      info_reg is shifted 31 to the right and the LSB is       */
/*                     to the output                             */
/*      The buffers are incremented and the sign is changed      */

      help_reg = info_reg[0] >>31;
      *data_out = help_reg & 1;
      data_out++;
      X_in++;
      Y_in++;
      sign = sign * -1;

  ENDLOOP /* BlockFactor */
}

/*****************************************************************/
/*      END OF FILE                                              */
/*****************************************************************/
```

171

C.2 Receiver II : Per-Survivor Processing

Definition File (mdef-file)

```
.NAME
DOQPSK_PSP_BIT

.SHORT_DESCRIPTION
general, coherent adaptive viterbi decoder for DOQPSK (LMS algorithm)

.DESCRIPTION
This model performs coherent viterbi decoding for a OQPSK-System
with a complex time invariant channel
The coefficient update is performed using a least mean square
(LMS) algorithm.
This algorithm uses the following equation:

    H_(k+1) = H_(k) + step * error * data*

where H_(k) is the following complex filter coefficients vector
of the k-th iteration:

    H_(k) = (h_(0,k), h(1,k), ..., h(NumOfFilterCoeff-1,k))^T

The filter tap delay corresponds to the bit duration (T_bit).

The parameter NumOfStates depends on the number of filter taps:

  NumOfStates = 2 * 2^NumOfFilterCoeff

The initial coefficients of the equalizer and the step sizes of the
LMS algorithm for every filter coefficient are specified using
input datasets.


.IMPLEMENTATIONS
1   SOURCE_CODE    C  DOQPSK_PSP_BIT  (DOQPSK_PSP_BIT)
2   DSP_CODE_GENERIC   GC  DOQPSK_PSP_BIT  (DOQPSK_PSP_BIT)

.CLASSES
1   DOQPSK

.INPUT_PORTS
1   InReal  (R)    [1]
    Real part of the input signal
2   InImag  (R)    [1]
    Imaginary part of the input signal

.OUTPUT_PORTS
1   OutData  (I)    [1]
    Data output
2   OutCoeffReal  (R)    [NumOfCoeff/2]
    Real part of the estimated channel coefficients
3   OutCoeffImag  (R)    [NumOfCoeff/2]
    Imaginary part of the estimated channels coefficients
```

```
.PARAMETERS
 1 NumOfStates (I)
   Number of states in the trellis

.INPUT_DATASETS
 1          Coefficients
 1.1        NONE
 1.1.1      RECORD_1  (R)
 1.1.1.*    Coefficients*
            The (complex) coefficients are stored from h(0) to h(k)
            in the order
            h_Real(0), h_Imaginary(0)
            h_Real(1), h_Imaginary(1)
                 ...
            h_Real(k), h_Imaginary(k)

 2          Step
 2.1        NONE
 2.1.1      RECORD_1  (R)
 2.1.1.*    ELEMENT*
            The step sizes are stored in the order
            step(0)
            step(1)
             ...
            step(k)

.HISTORY
  created by svenh on Fri Jul 31 11:15:10 1998
  generated by COINS, COSSAP Release 1998.02, Synopsys Inc., 1998

.END
```

Generic C File (gc-file)

```
/*                                                    */
/*   Filename : doqpsk_psp_bit.gc                     */
/*   Author   : Sven Hischke                          */
/*   Company  : Fachhochschule Giessen-Friedberg      */
/*                                                    */
/*   Revision : 1.0                                   */
/*   Date     : Aug-14-1998                           */
/*   Changes  :                                       */
/*                                                    */
/*   Imported Libraries                               */
#include <math.h>

/*   Declarations                                     */

INPUT_PORT(1) register SIGNAL *InReal;
INPUT_PORT(2) register SIGNAL *InImag;

OUTPUT_PORT(1) register long *OutData;
OUTPUT_PORT(2) register SIGNAL *OutCoeffReal;
OUTPUT_PORT(3) register SIGNAL *OutCoeffImag;

PARAMETER(1) long NumOfStates;

LENGTH(INPUT_DATASET(1.1)) long NumOfCoeff;
INPUT_DATASET(1.1) CONST SIGNAL Coefficients[NumOfCoeff];
INPUT_DATASET(2.1) CONST SIGNAL Step[NumOfCoeff/2];

RATE(OUTPUT_PORT(2)) = NumOfCoeff/2;
RATE(OUTPUT_PORT(3)) = NumOfCoeff/2;

STATE SIGNAL CoeffVecR[NumOfCoeff/2 * NumOfStates];
STATE SIGNAL CoeffVecI[NumOfCoeff/2 * NumOfStates];
STATE SIGNAL DataR[NumOfCoeff * NumOfStates];
STATE SIGNAL DataI[NumOfCoeff * NumOfStates];
STATE SIGNAL Metric[NumOfStates];
STATE long DecData[NumOfStates];
STATE long SampTime;

BLOCKFACTOR long BlockFactor;

/*   Initialization Phase (init = 1)                  */

void init_doqpsk_psp_bit()
{
  register SIGNAL *pmetric, *pinbuffer, *pcoeffr, *pcoeffi;
  register long *pdec;
  register long NumOfCoeff_h = NumOfCoeff >> 1;
  int      i,j,int0,int1,int2,int3,int4,int5;

/* initialize transition data vector */

  for(i = 0; i < 2*NumOfStates; i++)
    for(j = 0; j < NumOfCoeff_h; j++)
      {
        int0 = j/2;
```

```
        int1 = pow(4,int0);
        int2 = pow(-1,(i/int1));
        int3 = pow(-1,(i/(4*int1)));
        int4 = pow(-1,(i/(2*int1)));
        int5 = pow(-1,(i/(8*int1)));
        if( (j % 2) == 0)
         {
          DataR[(i*NumOfCoeff_h)+j] = int4;
          DataI[(i*NumOfCoeff_h)+j] = (int2 + int3) /2;
         }
        else
         {
          DataR[(i*NumOfCoeff_h)+j] = (int4 + int5) /2;
          DataI[(i*NumOfCoeff_h)+j] = int3;
         }
      }

/* initialize filter coefficient vector and metric vector*/

  pcoeffr = CoeffVecR;
  pcoeffi = CoeffVecI;
  pmetric = Metric;
  pdec = DecData;
  LOOP(NumOfStates)
    pinbuffer = Coefficients;
    *pmetric++ = 0.0;
    *pdec++ = 0;
    LOOP(NumOfCoeff_h)
      *pcoeffr++ = *pinbuffer++;
      *pcoeffi++ = *pinbuffer++;
    ENDLOOP
  ENDLOOP

/* initialize sampling time */

  SampTime = 1;
 }

/*  Processing Phase (init = 2)                        */

void doqpsk_psp_bit()
  {
  register SIGNAL *pinbuffer, *pcoeff0r, *pcoeff1r, *pcoeff0i, *pcoeff1i;
  register SIGNAL *perr01r, *perr02r, *perr01i, *perr02i, *perr11i;
  register SIGNAL *perr11r;
  register SIGNAL *perr12r, *perr12i, *plam0u, *plam1u, *plam0l, *plam1l;
  register SIGNAL *pstep, *pmet1, *pmet2, *pgam1, *pgam2, *pvecr, *pveci;
  register SIGNAL gam_min, *pdatar, *pdata1r, *pdatai, *pdata1i;
  register long *pdec, decdata,  hilf_reg, *pdec1, *pdec2;
  register long NumOfCoeff_h = NumOfCoeff >> 1;
  int      i,j, n, data_diff;

  SCRATCH SIGNAL ErrorR[2 * NumOfStates];
  SCRATCH SIGNAL ErrorI[2 * NumOfStates];
  SCRATCH SIGNAL Lambda[NumOfStates*2];
  SCRATCH SIGNAL Gamma1[NumOfStates];
  SCRATCH SIGNAL Gamma2[NumOfStates];
```

```
    SCRATCH SIGNAL VectorR[NumOfCoeff_h *NumOfStates];
    SCRATCH SIGNAL VectorI[NumOfCoeff_h * NumOfStates];
    SCRATCH long DecData_h[NumOfStates];

    LOOP(BlockFactor)

/* initialize pointers */

    pcoeff0r = CoeffVecR;
    pcoeff0i = CoeffVecI;
    pcoeff1r = CoeffVecR;
    pcoeff1i = CoeffVecI;
    perr01r = ErrorR;
    perr01i = ErrorI;

    LOOP(2*NumOfStates)
      *perr01r++ = 0.0;
      *perr01i++ = 0.0;
    ENDLOOP

    perr01r = ErrorR;
    perr01i = ErrorI;
    perr11r = ErrorR + NumOfStates;
    perr11i = ErrorI + NumOfStates;

    if(SampTime == 1)
     {
      pdatar = DataR;
      pdatai = DataI;
     }
    else
     {
      pdatar = DataI;
      pdatai = DataR;
     }

/* compute the error : e = (r-h*z) */

    LOOP(NumOfStates)
      LOOP(NumOfCoeff_h)
        *perr01r += *pcoeff0r   * *pdatar   - *pcoeff0i   * *pdatai;
        *perr01i += *pcoeff0r++ * *pdatai++ + *pcoeff0i++ * *pdatar++;
      ENDLOOP
      LOOP(NumOfCoeff_h)
        *perr11r += *pcoeff1r   * *pdatar   - *pcoeff1i   * *pdatai;
        *perr11i += *pcoeff1r++ * *pdatai++ + *pcoeff1i++ * *pdatar++;
      ENDLOOP
      *perr01r = *InReal - *perr01r++;
      *perr01i = *InImag - *perr01i++;
      *perr11r = *InReal - *perr11r++;
      *perr11i = *InImag - *perr11i++;
    ENDLOOP

/* compute lambda : l = |e^2| */

    perr01r = ErrorR;
    perr01i = ErrorI;
```

```
    perr11r = ErrorR + NumOfStates;
    perr11i = ErrorI + NumOfStates;
    plam0u = Lambda;
    plam1u = Lambda + NumOfStates;

    LOOP(NumOfStates)
      *plam0u = pow(*perr01r++,2) + pow(*perr01i++,2);
      plam0u++;
      *plam1u = pow(*perr11r++,2) + pow(*perr11i++,2);
      plam1u++;
    ENDLOOP

/* compute metric : g = g + l */

    plam0u = Lambda;
    plam0l = Lambda + (NumOfStates/2);
    plam1u = Lambda + NumOfStates;
    plam1l = Lambda + NumOfStates + (NumOfStates/2);
    pmet1 = Metric;
    pmet2 = Metric + (NumOfStates/2);
    pgam1 = Gamma1;
    pgam2 = Gamma2;

    LOOP(NumOfStates/2)
      *pgam1++ = *pmet1    + *plam0u++;
      *pgam2++ = *pmet2    + *plam0l++;
      *pgam1++ = *pmet1++ + *plam1u++;
      *pgam2++ = *pmet2++ + *plam1l++;
    ENDLOOP

/* select the metric and compute filter coefficient */

    pmet1 = Metric;
    pgam1 = Gamma1;
    pgam2 = Gamma2;
    pvecr = VectorR;
    pveci = VectorI;
    perr01r = ErrorR;
    perr02r = ErrorR + (NumOfStates/2);
    perr01i = ErrorI;
    perr02i = ErrorI + (NumOfStates/2);
    perr11r = ErrorR + NumOfStates;
    perr12r = ErrorR + NumOfStates + (NumOfStates/2);
    perr11i = ErrorI + NumOfStates;
    perr12i = ErrorI + NumOfStates + (NumOfStates/2);
    pcoeff0r = CoeffVecR;
    pcoeff1r = CoeffVecR + (NumOfCoeff_h * NumOfStates/2);
    pcoeff0i = CoeffVecI;
    pcoeff1i = CoeffVecI + (NumOfCoeff_h * NumOfStates/2);
    pdec1 = DecData;
    pdec2 = DecData + (NumOfStates/2);
    pdec  = DecData_h;

    if(SampTime == 1)
     {
       pdatar = DataR;
       pdatai = DataI;
```

```
       pdata1r = DataR + (NumOfStates * NumOfCoeff_h);
       pdata1i = DataI + (NumOfStates * NumOfCoeff_h);
      }
    else
     {
      pdatar = DataI;
      pdatai = DataR;
      pdata1r = DataI + (NumOfStates * NumOfCoeff_h);
      pdata1i = DataR + (NumOfStates * NumOfCoeff_h);
     }

/* compute coefficient : h = h + step * e * z* */

   i = 0;
   LOOP(NumOfStates/2)
     pstep = Step;
     if(*pgam1 <= *pgam2)
       {
        *pmet1++ = *pgam1++;
        pgam2++;
        *pdec = *pdec1;
        LOOP(NumOfCoeff_h)
          *pvecr = *pstep   * (*perr01r * *pdatar + *perr01i * *pdatai);
          *pveci = *pstep++ * (-(*perr01r) * *pdatai++
                              + *perr01i * *pdatar++);
          *pvecr++ += *pcoeff0r++;
          *pveci++ += *pcoeff0i++;
        ENDLOOP
          perr01r++;
          perr02r++;
          perr01i++;
          perr02i++;
          pcoeff0r -= NumOfCoeff_h;
          pcoeff0i -= NumOfCoeff_h;
          pdata1r += NumOfCoeff_h;
          pdata1i += NumOfCoeff_h;
       }
     else
       {
        *pmet1++ = *pgam2++;
        pgam1++;
        *pdec = *pdec2;
        LOOP(NumOfCoeff_h)
          *pvecr = *pstep   * (*perr02r * *pdata1r + *perr02i * *pdata1i);
          *pveci = *pstep++ * (-(*perr02r) * *pdata1i++
                              + *perr02i * *pdata1r++);
          *pvecr++ += *pcoeff1r++;
          *pveci++ += *pcoeff1i++;
        ENDLOOP
          perr01r++;
          perr02r++;
          perr01i++;
          perr02i++;
          pcoeff1r -= NumOfCoeff_h;
          pcoeff1i -= NumOfCoeff_h;
          pdatar += NumOfCoeff_h;
          pdatai += NumOfCoeff_h;
```

178

```c
      }

  *pdec++ = (*pdec << 1) + ((i & 2) && 1);
  i++;
  pstep = Step;
  if(*pgam1 <= *pgam2)
    {
      *pmet1++ = *pgam1++;
      pgam2++;
      *pdec = *pdec1++;
      pdec2++;
      LOOP(NumOfCoeff_h)
        *pvecr = *pstep    * ( *perr11r * *pdatar + *perr11i * *pdatai);
        *pveci = *pstep++ * (-(*perr11r) * *pdatai++
                              + *perr11i * *pdatar++);
        *pvecr++ += *pcoeff0r++;
        *pveci++ += *pcoeff0i++;
      ENDLOOP
        perr11r++;
        perr11i++;
        perr12r++;
        perr12i++;
        pcoeff1r += NumOfCoeff_h;
        pcoeff1i += NumOfCoeff_h;
        pdata1r += NumOfCoeff_h;
        pdata1i += NumOfCoeff_h;
    }
  else
    {
      *pmet1++ = *pgam2++;
      pgam1++;
      *pdec = *pdec2++;
      pdec1++;
      LOOP(NumOfCoeff_h)
        *pvecr = *pstep    * (*perr12r * *pdata1r + *perr12i * *pdata1i);
        *pveci = *pstep++ * (-(*perr12r) * *pdata1i++
                              + *perr12i * *pdata1r++);
        *pvecr++ += *pcoeff1r++;
        *pveci++ += *pcoeff1i++;
      ENDLOOP
        perr11r++;
        perr11i++;
        perr12r++;
        perr12i++;
        pcoeff0r += NumOfCoeff_h;
        pcoeff0i += NumOfCoeff_h;
        pdatar += NumOfCoeff_h;
        pdatai += NumOfCoeff_h;
    }

  *pdec++ = (*pdec << 1) + ((i & 2) && 1);
  i++;
ENDLOOP

pvecr = VectorR;
pveci = VectorI;
pcoeff0r = CoeffVecR;
```

179

```
   pcoeff0i = CoeffVecI;
   pdec = DecData;
   pdec1 = DecData_h;

   LOOP(NumOfStates)
    *pdec++ = *pdec1++;
   ENDLOOP

   LOOP(NumOfStates * NumOfCoeff_h)
     *pcoeff0r++ = *pvecr++;
     *pcoeff0i++ = *pveci++;
   ENDLOOP

   pcoeff0r = CoeffVecR;
   pcoeff0i = CoeffVecI;
   pcoeff1r = CoeffVecR;
   pcoeff1i = CoeffVecI;
   pmet1 = Metric;
   pdec = DecData;
   decdata = *pdec;
   gam_min = *pmet1;

/* select the path with minimum metric */

   LOOP(NumOfStates)
     if(gam_min > *pmet1)
       {
        gam_min = *pmet1++;
        decdata = *pdec++;
        pcoeff1r = pcoeff0r;
        pcoeff1i = pcoeff0i;
        pcoeff0r += NumOfCoeff_h;
        pcoeff0i += NumOfCoeff_h;
       }
     else
       {
        pmet1++;
        pdec++;
        pcoeff0r += NumOfCoeff_h;
        pcoeff0i += NumOfCoeff_h;
       }
   ENDLOOP

   pmet1 = Metric;

   LOOP(NumOfStates)
     *pmet1++ = *pmet1 - gam_min;
   ENDLOOP

/* output of decoded data */

   hilf_reg = decdata >> 31;
   *OutData++ = hilf_reg & 1;

/* output of the coefficient of the best path */

   LOOP(NumOfCoeff_h)
```

```
      *OutCoeffReal++ = *pcoeff1r++;
      *OutCoeffImag++ = *pcoeff1i++;
    ENDLOOP

    InReal++;
    InImag++;
    SampTime = SampTime * -1;
  ENDLOOP
}

/*  End of File                                 */
```

Appendix D

Published Work

1. S. Hischke, J. Habermann, R. Comley: "Receivers for Differentially Encoded OQPSK -A Comparison of Differential Demodulation with Joint Data and Channel Estimation-", International Conference on Telecommunication (ICT'97), April 1997, Melbourne/Australia

2. S. Hischke, J. Habermann: "Application of Per-Survivor Processing for Digital Mobile Radio", Vehicular Technology Conference (VTC'98), May 1998, Ottawa/Canada

3. S. Hischke, J. Habermann: "New Results on the Effects of Nonlinear Amplifiers on $DOQPSK$ and $\pi/4 - DQPSK$ Signals", Personal, Indoor and Mobile Radio Communications Conference (PIMRC'98), September 1998, Boston/USA

4. S. Hischke, J. Habermann: "A New Differential Demodulator for Offset-QPSK", International Journal of Electronics and Communications (AEÜ), 52 No.6, pages 375-383, December 1998

5. S. Hischke, J. Habermann: "Application of Offset-QPSK for Digital Mobile Radio", accepted for the January/February 2000 issue of the Journal of Telecommunications (Frequenz)

Receivers for Differentially Encoded OQPSK
-A Comparison of Differential Demodulation with Joint Data and Channel Estimation-

International Conference on Telecommunication (ICT'97)
April 1997, Melbourne/Australia

Receivers for Differentially Encoded OQPSK
- A Comparison of Differential Demodulation with Joint Data and Channel Estimation -

Sven Hischke*, Joachim Habermann*, Richard Comley**
*Fachhochschule Giessen-Friedberg Germany
** City University London England

*ABSTRACT: In this paper two new receiver struc-
tures for differentially encoded and filtered OQPSK
(DOQPSK) signals are investigated. In both struc-
tures the complex baseband signal is sampled twice
per symbol, i.e. once per bit. In the first receiver, the
differential demodulator, which is intended to be
used for time selective channels, the samples are
differentially preprocessed. This is followed by a
trellis decoder, which resolves the interference,
subsequently introduced between the I- and Q-
channel through differential preprocessing. In the
second receiver, to be designed for time and fre-
quency selective channels, no differential preproc-
essing is accomplished. Data detection and channel
estimation are jointly done in a trellis decoder. No
preambles are used for channel estimation, i.e. blind
estimation techniques are considered.*

1.   INTRODUCTION
Modulation formats for new mobile communication
systems should be bandwidth and spectrum efficient.
This requirement is often fulfilled by linear modula-
tion schemes. Secondly a modulation format should
support power efficient nonlinear amplifiers in the
mobile transmitters. In new mobile communication
standards (TETRA-Trans European Trunked Radio
System, JDC-Japan, ADC-U.S.) the modulation
format π/4-DQPSK was specified. One reason for
this choice is that the envelope of π/4-DQPSK has
no zeros. This reduces the generation of harmonics
and the distortion of the signal by non-linear ampli-
fiers. DOQPSK shows further reduced envelope
fluctuations and outperforms π/4-DQPSK on nonlin-
ear channels. However, up to now DOQPSK receiv-
ers for time and frequency selective mobile radio
channels are not available. A mobile radio receiver
should allow for

- fast tracking in time selective channels,
- short acquisition times in burst mode trans-
  mission,
- resolution of intersymbol interference gener-
  ated by the frequency selective mobile radio
  channel.

This paper gives two mobile radio receiver structures
for differentially encoded OQPSK signals. The first
receiver, which is based on differential demodula-
tion, has already been presented at a conference [1].
It is intended to be used for time selective mobile
radio channels. In this paper some new results based
on computer simulations for that receiver are pre-
sented. The second receiver structure is based on
blind equalisation techniques and can easily be ex-
tended to frequency and time selective channels. It
is further shown by computer simulations that the
above mentioned requirements for a mobile radio
receiver will be met by the blind equalisation re-
ceiver for DOQPSK signals.

2.   TRANSMITTER MODEL
Fig. 1 shows the model of the DOQPSK transmitter.
It differs from standard OQPSK by including a dif-
ferential encoder [1].
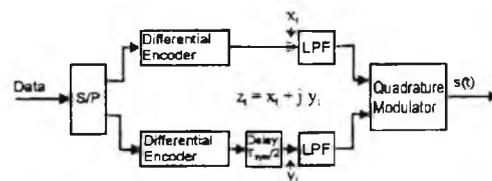
Fig. 1 DOQPSK Transmitter Model

3.   RECEIVER CONFIGURATIONS
3.1   Time Selective Channels
3.1.1   Differential Demodulation and Trellis
decoding (DiffTrell)
The following figure shows the block diagram of the
DOQPSK receiver (DiffTrell) that has already been
presented at the VTC 94 [1]. It is based on a differ-
ential preprocessing (demodulation) algorithm and a
trellis decoder to resolve cross-channel interference.
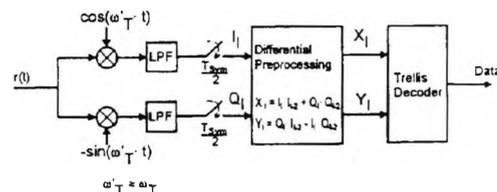
Fig. 2 DOQPSK Receiver Model (DiffTrell)

1055

The received signal r(t) is filtered with a root raised cosine pulse shaping filter and is sampled once per bit. The sampled signal at the input of the preprocessing unit is:

$$r_l = I_l + jQ_l = \left[z_l + \sum_{i=1}^{\infty} g_i \cdot \left(z_{l-i} + z_{l+i} \right) \right] \cdot e^{j\Phi_l} + n_l \quad (1)$$

$$g_i = 0 \quad \text{for } i = 2,4,6,..$$

In the above equation g_i gives the joint impulse response of the transmit and receive filter, and $z_l = x_l + jy_l$ represent the differentially encoded data bits in the transmitter (see Fig.1). Depending on the sampling time two equations for the received signal are found. For even sampling values l=2k one obtains:

$$r_{2k} = I_{2k} + jQ_{2k}$$

$$= \left[x_{2k} + j \cdot \sum_{i=1}^{\infty} g_i \cdot \left(y_{2k-i} + y_{2k+i} \right) \right] \cdot e^{j\Phi_{2k}} + n_{2k} \quad (2)$$

A similar equation is obtained for odd sampling values l=2k+1.

As a consequence of differential preprocessing

$$X_{2k} = \Re\left\{ r_{2k} \cdot r_{2k-2}^* \right\} = I_{2k} \cdot I_{2k-2} + Q_{2k} \cdot Q_{2k-2} \quad (3)$$

$$Y_{2k} = \Im\left\{ r_{2k} \cdot r_{2k-2}^* \right\} = Q_{2k} \cdot I_{2k-2} - I_{2k} \cdot Q_{2k-2} \quad (4)$$

an infinite number of symbols will mutually interfere. The interference between I- and Q-channel can be resolved using a trellis. If we choose a roll-off factor of r = 1.0, only two neighbouring symbols interfere with a given symbol. Considering differential preprocessing and exploiting symmetries it can be shown, that a trellis decoder with 8 states is sufficient. This has been demonstrated at the VTC 94 [1].

The following table shows the values of the impulse response and the number of states in the trellis for a more bandwidth efficient roll-off factor of r = 0.35, which is for example used for the TETRA system.

i	$g_i(r = 0.35)$	Number of states in the Trellis 2^{2i+1}
1	0.61858	8
3	-0.16244	128
5	0.05703	2048
7	-0.01382	32768
9	-0.00185	524288

Table 1 Values of the impulse response and number of states in the Trellis-Decoder

Theoretically an infinite number of samples of the impulse response g_i has to be considered. To reduce receiver complexity, the following investigations truncate the impulse response at i = 3. Eqn. 1 then changes to:

$$r_l = \left[z_l + g_1 \cdot \left(z_{l-1} + z_{l+1} \right) + g_3 \cdot \left(z_{l-3} + z_{l+3} \right) \right] \cdot e^{j\Phi_l} + n_l \quad (5)$$

The signals prior to the trellis decoder thus are:

$$X_{2k} = \Re\left\{ r_{2k} \cdot r_{2k-2}^* \right\}$$
$$= F\left\{ y_{2k-5}, y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k}, y_{2k+1}, y_{2k-3} \right\} \quad (6)$$

$$Y_{2k} = \Im\left\{ r_{2k} \cdot r_{2k-2}^* \right\}$$
$$= F\left\{ y_{2k-5}, y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k}, y_{2k+1}, y_{2k+3} \right\} \quad (7)$$

A roll-off factor r = 0.35 and a truncation after 3 samples (i = 3) result in a trellis decoder with 128 states. To further reduce the receiver complexity, the M algorithm [2] is implemented, which stores merely the M most probable paths through the trellis. The M algorithm with M=16 states reduces the number of states by a factor of 8 and gives a gain in computer simulation time by a factor of 2 (also considering the sorting algorithm). However, compared to an 8 state Viterbi algorithm the M algorithm is approximately twice as slow.

Fig. 3 and 4 show the simulation results of the algorithm DiffTrell on an AWGN and a Rayleigh fading channel. On the linear AWGN channel the curves are compared with π/4-DQPSK. For bit error rates less than 10^{-3} merely a small degradation is observed, when a 128 state Viterbi algorithm or the 16 state M algorithm is used. The Viterbi algorithm with 8 states, which has been designed for a roll-off factor of r = 1.0 (from [1]), degrades heavily. The small losses in bit error rate at lower signal to noise ratios compared to π/4-DQPSK are due to the implemented metrics in the trellis decoder, assuming white and uncorrelated noise samples, which is not the case due to differential preprocessing.

The positive results of differential demodulation for DOQPSK signals on an AWGN channel are preserved on the Rayleigh fading channel. The product of maximum Doppler frequency f_d and symbol period T_S has been chosen very high (here: $f_d*T_S = 0.0093$) in the simulations, to demonstrate the saturation effect of the residual bit error rate.
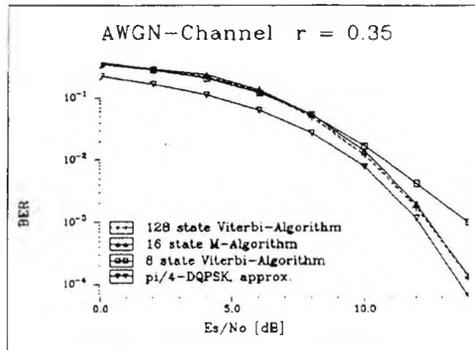
Fig. 3 DiffTrell on an AWGN channel
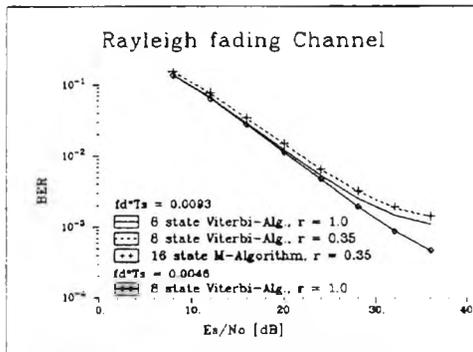
Fig. 4 DiffTrell on a Rayleigh channel

3.1.2 Differentially coherent decoding with channel estimation without preamble (BlindTrell)

The algorithm BlindTrell performs channel estimation and data detection jointly in a trellis decoder. For every transition an independent channel impulse response is calculated adaptively. In each state the path with the smallest metric and its channel impulse response are stored. The algorithm has been developed by Seshadri [3] for PAM signals and is here applied to DOQPSK signals. Note that now in the receiver (see Fig. 2) no differential preprocessing is accomplished.

The algorithm will be given for a roll off factor $r = 1.0$. The nominal value of the received signal r_l is:

$$r'_l = I'_l + j Q'_l = h \cdot \left(z_l + 0.5 \cdot \left(z_{l-1} + z_{l+1} \right) \right) \qquad (8)$$

$$h = h_{0R} + j h_{0I}$$

(' indicates a nominal value)

h denotes the complex and time variant channel coefficient.

If $l = 2k$, i.e. for even sampling values, one obtains:

$$I'_{2k} = h_{0R} \cdot x_{2k} - 0.5 \cdot h_{0I} \cdot \left(y_{2k-1} + y_{2k+1} \right) \qquad (9)$$

$$Q'_{2k} = h_{0I} \cdot x_{2k} + 0.5 \cdot h_{0R} \cdot \left(y_{2k-1} + y_{2k+1} \right) \qquad (10)$$

Since every sampling value depends on two more bits, a trellis with 4 states results. The metric J of a state j at a time instant 2k is calculated from the quadratic difference between the received values and the nominal values:

$$J_{2k}(j) = \min \Big\{ J_{2k-1}\left(\lfloor j/2 \rfloor \right) + \left[r_{2k} - r'_{2k}\left(\lfloor j/2 \rfloor \right) \right]^2 ;$$

$$J_{2k-1}\left(n + \lfloor j/2 \rfloor \right) + \left[r_{2k} - r'_{2k}\left(n + \lfloor j/2 \rfloor \right) \right]^2 \Big\};$$

$$(11)$$

with: j = state, n = number of states / 2

In contrast to [3] the individual impulse response h for every transition is not calculated, such that the impulse response minimises the difference between the received signal r_l and its nominal value, since ambiguities occur. Instead firstly the nominal values of the channel coefficients are calculated from the received signal and then the quadratic error between the calculated nominal values and the actual values of the impulse response is minimised applying the LMS algorithm. The nominal values of the impulse response can be calculated from the received signal components I and Q as:

$$h_{0Rnom,2k} = \left[x_{2k} \cdot I_{2k} + 0.5 \cdot Q_{2k} \cdot \left(y_{2k-1} + y_{2k+1} \right) \right]$$

$$\bullet \frac{1}{x_{2k}^2 + \frac{1}{4}\left(y_{2k-1} + y_{2k+1} \right)^2} \qquad (12)$$

$$h_{0Inom,2k} = \left[x_{2k} \cdot Q_{2k} - 0.5 \cdot I_{2k} \cdot \left(y_{2k-1} + y_{2k+1} \right) \right]$$

$$\bullet \frac{1}{x_{2k}^2 + \frac{1}{4}\left(y_{2k-1} + y_{2k+1} \right)^2} \qquad (13)$$

The LMS algorithm then gives:

$$e_{R,2k} = \frac{\partial \left[h_{0Rnom,2k} - h_{0R,2k-1} \right]^2}{\partial h_{0R}} \qquad (14)$$

$$e_{I,2k} = \frac{\partial \left[h_{0Inom,2k} - h_{0I,2k-1} \right]^2}{\partial h_{0I}} \qquad (15)$$

1057

188

$$h_{0R,2k}(j) = h_{0R,2k-1}\left(\lfloor j/2 \rfloor\right) - \Delta \cdot e_{R,2k} \qquad (16)$$

$$h_{0I,2k}(j) = h_{0I,2k-1}\left(\lfloor j/2 \rfloor\right) - \Delta \cdot e_{I,2k} \qquad (17)$$

Similarly to the above equations for even sampling values, the equations for odd sampling values are calculated.

An important aspect of an algorithm without a pre-amble is its acquisition performance. Table 2 gives the simulation results on a static AWGN channel with roll-off factor $r = 1.0$ and Es/No = 12 dB. Different seeds of the transmitter bit generator and starting values of the algorithm have been selected. The updating parameter Δ of eqns. 16 and 17 was set to 0.2.

$\Delta = 0.2$				
seed	channel	start value	# error	error position
9530128	0.707/0.707	0.707/0.707	0	
56473	0.707/0.707	0.707/0.707	2	1,3
9530128	0.707/0.707	0/0	2	2,3
56473	0.707/0.707	0/0	2	1,2

Table 2   Acquisition performance of the algorithm BlindTrell

The results show, that under the conditions mentioned above the algorithm BlindTrell needs only three bits for acquisition. In burst mode transmission, therefore, acquisition can be achieved with some tail bits.

Figures 5 - 9 show simulation results of the algorithm BlindTrell on an AWGN and a Rayleigh fading channel and compare with the differential demodulation algorithm DiffTrell. Fig. 5 gives results for a roll-off factor of $r = 1.0$ and Fig. 6 for 0.35.
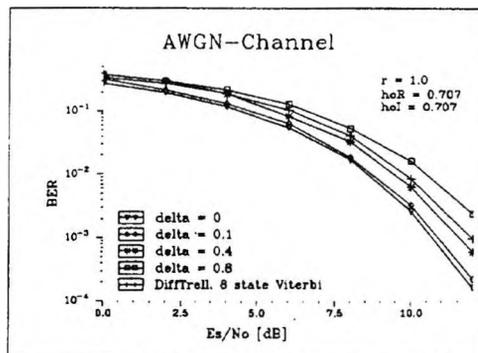
Fig. 5 BlindTrell on an AWGN channel

For small values of Δ (delta) a gain in bit error rate compared to DiffTrell is obtained, which is lost merely for very large values of Δ ($\Delta > 0.4$). If $\Delta < 0.1$, the results of a static channel estimation are nearly achieved.
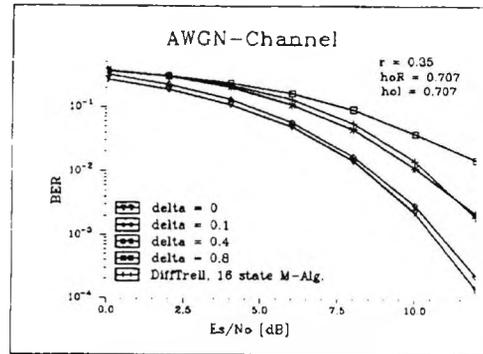
Fig. 6 BlindTrell on an AWGN channel

For the Rayleigh fading channel two products of symbol period and maximum Doppler frequency have been selected, $f_d * T_s = 0.0015$ and 0.0046. These products correspond to mobile velocities of 67km/h and 200 km/h in the European TETRA system; in the trans European GSM system the velocities are even higher by a factor of 7.5. Figs. 7 and 8 give results for a roll-off factor of 1.0. The results indicate that the performance of the algorithm DiffTrell can be met, if the adaptation speed (Δ) is chosen properly. Note that for a mere time selective channel, differential demodulation is a very effective algorithm.
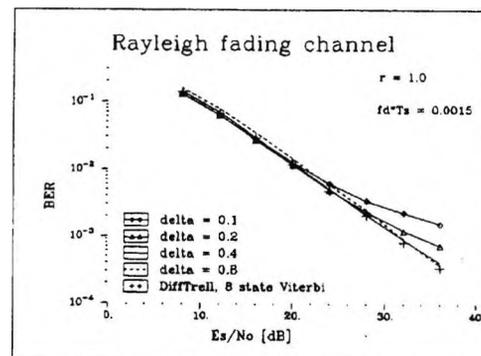

Fig. 7 BlindTrell on a Rayleigh channel

189
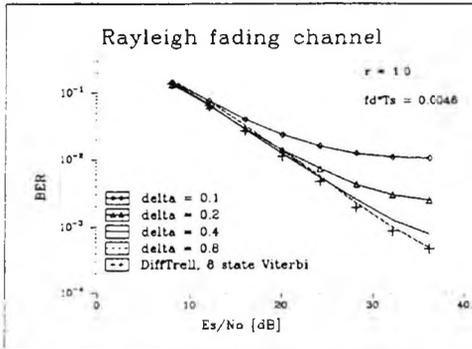
Fig. 8 BlindTrell on a Rayleigh channel

Similar results as for r = 1.0 are obtained for a roll off factor of r = 0.35, see Fig. 9.
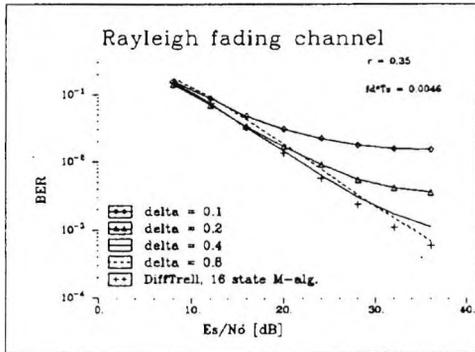

Fig. 9 BlindTrell on a Rayleigh channel

The advantages of the algorithm BlindTrell in comparison to DiffTrell emerge primarily in frequency selective channels, which is investigated next.

3.2 Receiver for time and frequency selective channels - BlindTrell on a frequency selective channel

Fig. 10 gives the model of the frequency selective channel, where the two coefficients h_0, h_1 are filtered complex Gaussian processes, thus defining a two path Rayleigh channel.
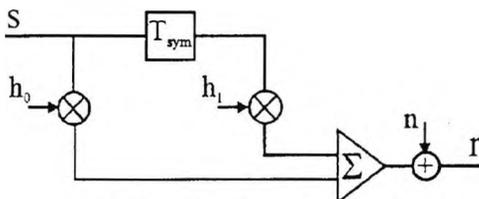
Fig. 10 Two path channel model

The derivation of the receiver equations is similar to the former in section 3.1.2. The nominal value of the received signal is:

$$r'_l = I'_l + jQ'_l = h_0 \cdot \left(z_l + 0.5 \cdot \left(z_{l-1} + z_{l+1}\right)\right)$$
$$+ h_1 \cdot \left(z_{l-2} + 0.5 \cdot \left(z_{l-3} + z_{l-1}\right)\right) \tag{18}$$

For even sampling values l=2k the nominal values of the complex input signal of the trellis decoder are:

$$I'_{2k} = h_{0R} \cdot x_{2k} + h_{1R} \cdot x_{2k-2} - 0.5 \cdot h_{0I} \cdot \left(y_{2k-1} + y_{2k+1}\right)$$
$$- 0.5 \cdot h_{1I} \cdot \left(y_{2k-3} + y_{2k-1}\right) \tag{19}$$

$$Q'_{2k} = h_{0I} \cdot x_{2k} + h_{1I} \cdot x_{2k-2} + 0.5 \cdot h_{0R} \cdot \left(y_{2k-1} + y_{2k+1}\right)$$
$$+ 0.5 \cdot h_{1R} \cdot \left(y_{2k-3} + y_{2k-1}\right) \tag{20}$$

Each sampling value depends on 4 more bits, thus resulting in a trellis with 16 states. The metric J of state j of the trellis is defined as:

$$J_{2k}(j) = \min\left\{ J_{2k-1}\left(\lfloor j/2 \rfloor\right) + \left[r_{2k} - r'_{2k}\left(\lfloor j/2 \rfloor\right)\right]^2; \right.$$
$$\left. J_{2k-1}\left(n + \lfloor j/2 \rfloor\right) + \left[r_{2k} - r'_{2k}\left(n + \lfloor j/2 \rfloor\right)\right]^2 \right\}, \tag{21}$$

with: j = state, n = number states / 2

The two eqns. 19 and 20 above are not sufficient to calculate the four channel coefficients (real and imaginary part of two coefficients). The preceding sampling value will not give two more required independent equations due to the modulation format and the filtering. The channel parameters are therefore determined recursively. During each sampling period in a first step two channel coefficients are calculated, considering the other two coefficients as known. In the next step the second pair of coefficients is calculated. For example, the two equations of the coefficient h_0 at an even sampling time l = 2k are given below:

$$h_{0Rnom,2k} = \left\{ x_{2k} \cdot \left[I_{2k} - x_{2k-2} \cdot h_{1R,2k-1} \right. \right.$$
$$+ 0.5 \cdot \left(y_{2k-3} + y_{2k-1}\right) \cdot h_{1I,2k-1} \right] + 0.5 \cdot \left(y_{2k-1} + y_{2k+1}\right)$$
$$\bullet \left[Q_{2k} - 0.5 \cdot \left(y_{2k-3} + y_{2k-1}\right) \cdot h_{1R,2k-1} - x_{2k-2} \cdot h_{1I,2k-1} \right] \right\}$$
$$/ \left(1.5 + 0.5 \cdot y_{2k-1} \cdot y_{2k+1}\right) \tag{22}$$

1059

$$h_{0Inorm,2k} = \Big\{ x_{2k} \cdot \Big[Q_{2k} - 0.5 \cdot \big(y_{2k-3} + y_{2k-1}\big) \cdot h_{1R,2k-1}$$

$$- x_{2k-2} \cdot h_{1I,2k-1} \Big] - 0.5 \cdot \big(y_{2k-1} + y_{2k+1}\big)$$

$$\bullet \Big[I_{2k} - x_{2k-2} \cdot h_{1R,2k-1} + 0.5 \cdot \big(y_{2k-3} + y_{2k-1}\big) \cdot h_{1I,2k-1} \Big] \Big\}$$

$$/ \big(1.5 + 0.5 \cdot y_{2k-1} \cdot y_{2k+1}\big)$$

$$(23)$$

The coefficient h_1 is calculated by similar equations. Updating of the coefficients h_0 and h_1 is done following the eqns. 14 to 17.

The last two figures show the results of the algorithm BlindTrell on a two path Rayleigh fading channel. The delay of the echo path is fixed to one symbol period. Therefore, the direct and echo signal are uncorrelated. The power of the echo signal is equal to the direct path. The upper and lower curves give the theoretical values of DQPSK without and with two fold diversity (L=2) in a maximal ratio receiver [4], where the receiver velocity in [4] is assumed to be zero. For lower Es/No values the gain of a diversity receiver can be achieved with the algorithm BlindTrell. However, at higher Es/No values a residual bit error rate is observed, which is due to the nonzero receiver velocity in the simulations, i.e. $f_d{}^*T_S > 0$. For a roll-off factor of r = 0.35 the 16 state trellis decoder of section 3.2 is used, which has been derived for r = 1.0. Though the receiver algorithm is not adapted to the pulse shaping filters, the performance loss is small.
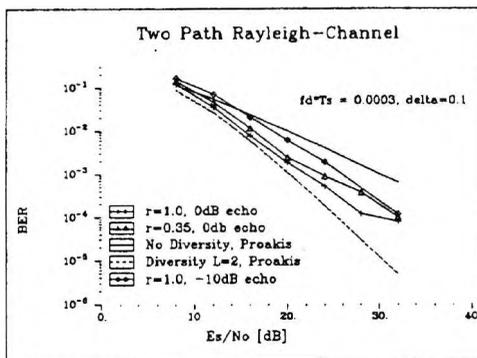

Fig. 11 BlindTrell on a two path Rayleigh channel
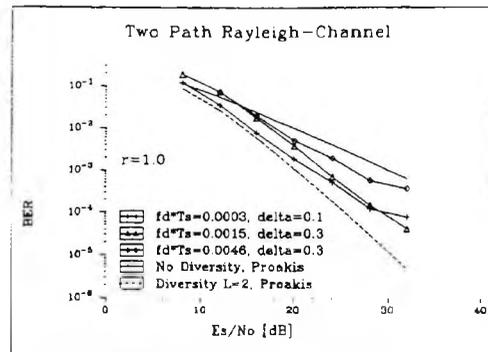
Fig. 12 BlindTrell on a two path Rayleigh channel

4. CONCLUSIONS

Two new receiver structures for differentially encoded and filtered OQPSK signals are analysed in this paper: a receiver applying differential demodulation and a receiver based on joint data and channel estimation. The two receiver algorithms are investigated by computer simulations (COSSAP) on a static linear AWGN channel and on a time variant Rayleigh fading channel, demonstrating a BER performance that can approach DQPSK. The second receiver is further investigated on a frequency selective two path Rayleigh fading channel. If both paths are uncorrelated, the diversity effect can fully be exploited by the algorithm.

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] C. Guenther, J. Habermann, "DOQPSK - Differential Demodulation of Filtered Offset QPSK", Proceedings IEEE VTC 94, pp. 1542-1546, Sweden, 1994.

[2] J. Huber, "Trelliscodierung", Springer-Verlag, 1992.

[3] N. Seshadri, "Joint Data and Channel Estimation Using Blind Trellis Search Techniques", IEEE Trans. on Communications, Vol. 42, No. 2/3/4, pp. 1000-1011, Feb./March/April 1994.

[4] J.G. Proakis, "Digital Communications", McGrawHill, 1995.

1060

191

Application of Per-Survivor Processing for Digital Mobile Radio

Application of Per-Survivor Processing
for Digital Mobile Radio

Sven Hischke* and Joachim Habermann**

The authors are with
* Department of Electrical, Electronic & Information Engineering, City University, London, England
**Department of Electrical Engineering, University of Applied Science, Giessen-Friedberg, Germany

Abstract -The bandwidth efficient modulation format OQPSK is implemented up to now merely in mobile radio systems which do employ spread spectrum techniques, i.e. it is not used in narrow band systems. This is due to the fact, that, in spite of its reduced performance degradation in non-linear channels in comparison to other linear modulation formats, for OQPSK no receivers were derived up to now, which satisfy the following requirements: fast tracking in time selective channels, short acquisition time in burst mode transmission, and resolution of intersymbol interference in frequency selective channels. In this paper a general approach for receiver implementations of differentially encoded OQPSK signals is given. Receiver structures for mobile radio applications are synthesised and investigated, which are all based on per-survivor processing (PSP).

I. INTRODUCTION

Data transmission over time and frequency selective mobile radio channels results both in a data sequence and channel estimation problem at the receiving side. It is sufficiently known from the literature that the common maximum likelihood estimation of the data s and of the channel impulse response h can not be carried out recursively. An optimal decoding requires a decoding of the entire message. However, the algorithm for data decoding and channel estimation can be split up into two steps:

$$[s_0, h_0] = \arg \left[\underset{h}{MIN} \left[\underset{s}{MIN} \sum_{i=0}^{N} \left| r_i - \sum_{i=0}^{L} h_i \cdot s_{i-i} \right|^2 \right] \right] \quad (1)$$

The above algorithm is optimal in slowly fading channels with respect to the maximum likelihood criterion. However, it is not recursive and has a high implementation complexity.

The classical, conventional, suboptimal MLSE-receiver initially performs a matched filtering that considers the channel impulse response and the impulse response of the transmission filter, followed by a sampling device with decorrelation and a Viterbi decoder. The unknown parameters of the different blocks of the receiver can e.g. be estimated from the decoded, but partly erroneous and delayed data (data aided), which results in a suboptimal implementation.

An alternative to the classical approach is per-survivor processing (PSP). PSP ([1], [2]) combines a group of algorithms which implement the data aided structure within the Viterbi algorithm itself. PSP uses the code sequence, that belongs to the survivor path to estimate the unknown parameters without delay.

The linear and bandwidth efficient modulation format OQPSK shows reduced amplitude fluctuations in comparison to other formats such as $\pi/4$-QPSK. OQPSK is therefore less sensitive to amplifier non-linearities, which produce inband distortions and out of band intermodulation products. From a transmitter point of view OQPSK, therefore, seems to be an attractive modulation format. However, for mobile radio applications receivers have to be found, in which decoding is done nearly optimally, with additional requirements for fast tracking in time selective channels, short acquisition time in burst mode transmission, and resolution of the intersymbol interference in frequency selective channels.

Based on the above requirements, in the following different adaptive MLSE algorithms for discrete time channel estimation based on PSP for differentially encoded OQPSK will be considered.

II. OFFSET-QPSK TRANSMITTER

The following figure gives the block diagram of the OQPSK transmitter [3], which is employed for all further investigations. The transmitter differs from the standard implementation by the differential precoding, in order to avoid ambiguities which may occur in the receiver.


Fig.1 DOQPSK Transmitter Model

430

VCT '98
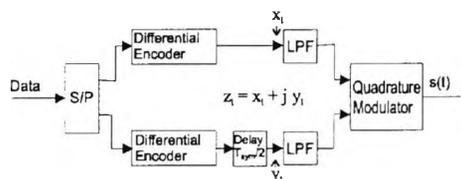
III. RECEIVER

The block diagram of the receiver, which combines Viterbi algorithm and discrete channel estimation within the trellis decoder, is given in Fig. 2. For reasons of a simplified receiver implementation, the standard channel matched filter with subsequent decorrelation device, which have to be adapted to the time varying channel, are replaced by a sub-optimal Nyquist-I root raised cosine roll-off low-pass filter, which precedes the trellis decoder. The twofold sampling per symbol considers the special feature of staggered (offset) QPSK.
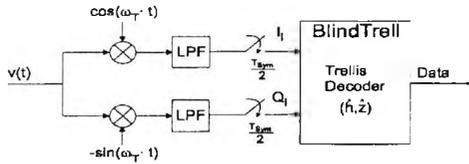
Fig. 2 DOQPSK Receiver Model

For the subsequent investigations it is useful to define a vector $\mathbf{z}$ that contains the filtered complex data symbols z. A Nyquist filtered OQPSK signal with a roll-off factor of $r = 1.0$ can be described as:

$$\mathbf{z}_l^T = \left[\tilde{z}_l, \ldots, \tilde{z}_{l-L} \right] \qquad (2)$$

with: $\tilde{z}_l = \left[0.5 \cdot z_{l-1} + z_l + 0.5 \cdot z_{l+1} \right]$, where $z_l = x_l + j\, y_l$ is the complex data symbol. The received and filtered signal r at time instant l is then determined as:

$$r_l = I_l + jQ_l = \mathbf{h}_l^T \cdot \mathbf{z}_l + n_l ` \qquad (3)$$

(n_l is the filtered noise signal)

The vector $\mathbf{h}$ represents the L+1 taps of the impulse response of the time variant and frequency selective channel:

$$\mathbf{h}_l^T = \left[h_l(0), \ldots h_l(L) \right] \qquad (4)$$

In the following different algorithms are investigated. The RLS algorithm is not considered in this paper due to its increased implementation complexity.

A. LMS ALGORITHM

At first the LMS algorithm is given for an iterative channel estimation within the trellis decoder. We have:

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} - \Delta \cdot \frac{\partial \left| e_{l-1} \right|^2}{\partial \hat{\mathbf{h}}_{l-1}^H} \qquad (5)$$

with: $e_{l-1} = r_l - \hat{r}_l = r_l - \hat{\mathbf{h}}_{l-1}^T \cdot \hat{\mathbf{z}}_l$. This leads directly to the well known solution:

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} + \Delta \cdot e_{l-1} \cdot \hat{\mathbf{z}}_l^* \qquad (6)$$

The estimates $\hat{\mathbf{h}}$ and the data values $\hat{\mathbf{z}}$ are obtained with PSP directly from the survivor path to each state in the trellis decoder. This is shown in Fig. 3 below.
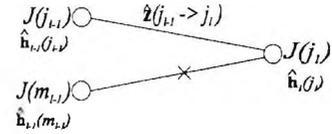

Fig. 3 Segment of a trellis with PSP and metrics $J(.)$

In order to avoid double indexing with the time index (see $\hat{h}$ in Fig. 3), the following definitions are used:
Generally the time is indicated as an index of the variable. If the variable is associated with a specific state in the trellis decoder, and state and variable do have the same time index, the time index is only indicated with the state. Arrows characterise state transitions in the trellis decoder.

The accumulated metrics J of state j at a time instant l is calculated with the quadratic norm. Selection of the minimum metrics gives:

$$J(j_l) = \mathrm{MIN}\left\{ J(j_{l-1}) + \left| r_l - \hat{\mathbf{h}}^T(j_{l-1}) \cdot \hat{\mathbf{z}}(j_{l-1} \to j_l) \right|^2 ; \right.$$
$$\left. J(m_{l-1}) + \left| r_l - \hat{\mathbf{h}}^T(m_{l-1}) \cdot \hat{\mathbf{z}}(m_{l-1} \to j_l) \right|^2 \right\} \qquad (7)$$

The marking (x) of the lower path in Fig. 3 assumes that the upper path is the survivor path. The channel which has been determined for this path so far, is used to update the channel estimate with the LMS algorithm:

$$\hat{\mathbf{h}}(j_l) = \hat{\mathbf{h}}(j_{l-1}) + \Delta \cdot e(j_{l-1} \to j_l) \cdot \hat{\mathbf{z}}^*(j_{l-1} \to j_l) \qquad (8)$$

B. MATRIX INVERSION

The vector $\mathbf{r}$ of the received signal can be expressed as:

$$\mathbf{r} = \mathbf{Z} \cdot \mathbf{h} + \mathbf{n}` \qquad (9)$$

With the aid of the above equation, the vector $\hat{\mathbf{h}}$ of the channel estimates can be calculated from the survivor path to each state. Given the vector of the received signal

$$\mathbf{r}_l^T = \left[r_l, \ldots r_{l-L} \right]. \qquad (10)$$

which contains L+1 complex samples (if the channel impulse response contains L+1 taps), and the data matrix estimate

$$\hat{\mathbf{Z}}_l = \begin{bmatrix} \hat{\mathbf{z}}^T(j_{l-1} \to j_l) \\ \hat{\mathbf{z}}^T(j_{l-2} \to j_{l-1}) \\ \vdots \\ \hat{\mathbf{z}}^T(j_{l-1-L} \to j_{l-L}) \end{bmatrix}, \qquad (11)$$

VCT '98

which is obtained from the data of the survivor path, an inversion of $\hat{\mathbf{Z}}$ is used to obtain an estimate $\hat{\mathbf{h}}$ of the channel:

$$\hat{\mathbf{h}} = \hat{\mathbf{Z}}^{-1} \cdot \mathbf{r} \qquad (12)$$

In order to suppress noise and interference, $\hat{\mathbf{h}}$ can be averaged over several estimation intervals. A major problem of this approach is, that on the one side $\hat{\mathbf{Z}}^{-1}$ may be singular and on the other side a computation of the inverse of a matrix will result in a higher receiver complexity. To reduce computational time, it is reasonable to compute only one single channel impulse response (and not one per state). The impulse response, however, has then to be obtained from the root path, which results in a delay of the channel estimate. Otherwise tentative decision have to be used, giving a BER degradation. The algorithm will not be applied in its original form, but will be modified to become the hybrid algorithm.

C. HYBRID ALGORITHM

The algorithm described in this section employs channel estimation approaches both from the LMS algorithm and from the matrix inversion algorithm. The adaptation procedure of the channel coefficients is based on the stochastic gradient algorithm:

$$\hat{h}(i)_l = \hat{h}(i)_{l-1} - \Delta \cdot \frac{\partial |e(i)_{l-1}|^2}{\partial \hat{h}(i)_{l-1}^*} \qquad (13)$$

The difference to (5) now is, that the error e is computed separately for every individual coefficient h. Note that the objective function now is the channel impulse response and not the received signal. The algorithm will thus give not an optimal BER performance but result in a more exact channel estimate which might be used in subsequent algorithms which rely on a channel side information. We have:

$$e(i, j_{l-1} \to j_l) = \tilde{h}(i, j_l) - \hat{h}(i, j_{l-1}) \qquad (14)$$

The current value $\tilde{h}(i, j_l)$ is obtained from the received signal r_l, by neglecting the noise term, solving (3) with respect to each estimated coefficient $\hat{h}(i, j_l)$, and by substituting the coefficients $\hat{h}(k, j_l)$ for $k \neq i$ through the estimated values $\hat{h}(k, j_{l-1})$ of the last sampling instant. We obtain:

$$\tilde{h}(i, j_l) = F\{r_l, \hat{h}(k, j_{l-1}), \hat{z}(j_{m-1} \to j_m)\}, k \neq i \qquad (15)$$

$\tilde{h}(i, j_l)$ is the current, possibly erroneous value of the channel coefficient i, which is obtained from the survivor path to state j_l. A smoothing of the coefficients can be achieved by application of the stochastic gradient algorithm:

$$\hat{\mathbf{h}}(j_l) = \hat{\mathbf{h}}(j_{l-1}) + \Delta \cdot \mathbf{e}(j_{l-1} \to j_l) \qquad (16)$$

Metric calculation is given by (7).

D. KALMAN ESTIMATOR

Fig. 4 shows the ARMA process model of the time and frequency selective fading channel vector h (see [4] and [5]).
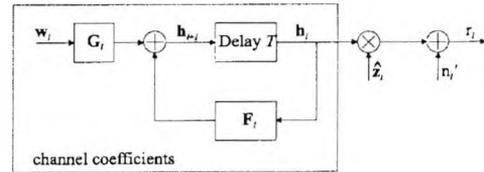

Fig. 4 ARMA Model

The AR part of the vector $\mathbf{h}_l$ in general considers also past samples of the channel coefficients, i.e. it is:

$$\mathbf{h}_l^T = [h_l(0), \ldots, h_{l-k}(0), \ldots, h_l(L), \ldots, h_{l-k}(L)] \qquad (17)$$

If several AR coefficients are used, a good approximation of the power spectral density function of a uniform direction Rayleigh fading process can be achieved. In a first order approximation we chose $h_{l-1}(i) \ldots h_{l-k}(i) = 0$. In this case the vector h is L+1-dimensional, $\mathbf{F}_l$ and $\mathbf{G}_l$ are $L+1 \times L+1$ matrices. If furthermore, the input process

$$\mathbf{w}_l^T = [w_l(0), \ldots, w_l(i), \ldots, w_l(L)]$$

with signal powers $\sigma_w^2(i)$ of its components $w(i)$ is not averaged, $\mathbf{G}_l$ becomes a unit matrix. Assuming that the channel coefficients are uncorrelated and unbiased, i.e.

$$E\{h(i) \cdot h^*(j)\} = E\{|h(i)|^2\} \cdot \delta_{i,j}, \ (\delta_{i,j} \triangleq \text{Kronecker Delta})$$

the matrix $\mathbf{F}_l$ is diagonal. We then obtain:

$$r_l = \mathbf{z}_l^T \cdot \mathbf{h}_l + n_l' = \mathbf{z}_l^T \cdot (\mathbf{F}_{l-1} \cdot \mathbf{h}_{l-1} + \mathbf{w}_{l-1}) + n_l' \qquad (18)$$

If the Rayleigh process is stationary, $\mathbf{F}_l = \mathbf{F}$. The elements of the matrix $\mathbf{F}$ can be described by a Bessel function $J_0(2\pi f_d \cdot T)$ and may further be simplified to $\beta = e^{-2\pi f_d T}$. For reasons of a simplified formulation $\hat{\mathbf{h}}_l$ and $\hat{\mathbf{z}}_l$ are used instead of $\hat{\mathbf{h}}(j_l)$ and $\hat{\mathbf{z}}(j_{l-1} \to j_l)$ in the following. The vector of the estimated channel coefficients, which is obtained with the aid of a Kalman filter adapted to the channel model eventually becomes ([5]):

$$\hat{\mathbf{h}}_{l+1} = \mathbf{F} \cdot \hat{\mathbf{h}}_l + \mathbf{k}_l \cdot (r_{l+1} - \hat{\mathbf{z}}_{l+1}^T \cdot \hat{\mathbf{h}}_l), \qquad (19)$$

where k is the Kalman gain factor:

$$\mathbf{k}_l = \beta \cdot \mathbf{R}_l \cdot \hat{\mathbf{z}}_l^* \cdot [\hat{\mathbf{z}}_l^T \cdot \mathbf{R}_l \cdot \hat{\mathbf{z}}_l^* + \sigma_N^2]^{-1} \qquad (20)$$

with: $E\{n_i \cdot n_k^*\} = \sigma_N^2 \cdot \delta_{ik}$. The equation of the Kalman gain can further be simplified if it is considered, that the values of

the main diagonal will dominate in the error covariance matrix $\mathbf{R}_l = E_{n',w}\left\{\left(\mathbf{h}_l - \hat{\mathbf{h}}_l\right)\cdot\left(\mathbf{h}_l - \hat{\mathbf{h}}_l\right)^H\right\}$. An averaging of $\mathbf{R}_l$ and of the denominator of the Kalman gain vector over the data results in the diagonal matrix $\tilde{\mathbf{R}}_l$ with elements $\rho_l(i)$ (see also [4]) and in the Kalman gain k as:

$$\mathbf{k}_l = \beta\cdot\tilde{\mathbf{R}}_l\cdot\hat{\mathbf{z}}_l^* \Big/ \left(c\cdot\sum_{i=0}^{L}\rho_l(i)+\sigma_{N'}^2\right) \qquad (21)$$

c is a constant which depends on the roll-off factor r of the Nyquist filters. Unlike the standard LMS algorithm, every channel coefficient is now multiplied by an individual gain factor

$$k_l(i) = \beta\cdot\rho_l(i)\cdot\hat{\tilde{z}}_{l-i}^* \Big/ \left(c\cdot\sum_{i=0}^{L}\rho_l(i)+\sigma_{N'}^2\right). \qquad (22)$$

By means of the Riccati equation it can be shown that for higher signal to noise ratios Es/No an approximation for the diagonal elements of the error covariance matrix can be found:

$$\rho_{l\to\infty}(i) \approx \sigma_w^2(i) = E\left\{|h(i)|^2\right\}\cdot\left(1-\beta^2\right) \qquad (23)$$

The Kalman gain $k(i)$, thus, is proportional to the average power $E\left\{|h(i)|^2\right\}$. In the steady state, consequently, the step size $k(i)$ for the adaptation of the channel coefficients is proportional to the average relative signal power of that specific path. This result can also be applied to the hybrid channel estimation algorithm and even to other adaptation algorithms, if the step size is chosen proportional to the signal power of a path.

IV. SIMULATION RESULTS

A. ACQUISITION PERFORMANCE

A short acquisition time and a fast tracking is essential in time selective channels. Fig. 5 shows the squared error of the channel coefficients of the hybrid algorithm on a two path Rayleigh fading channel. The signal-to-noise ratio in these simulations is fixed to 10 dB and the power of the echo path h(1) is reduced by 10 dB compared to the direct path h(0). Channel updating is performed with the step sizes Δ_0 of 0.13 and Δ_1 of 0.014. The mean squared error has been averaged over 30 independent simulations. Convergence of the direct path has been observed within 50 symbols. The rate of convergence of the echo path is slower, but the steady state squared error is in accordance to the reduced power. BER investigations show that no further bit errors occur after ten bits in the noise free case.
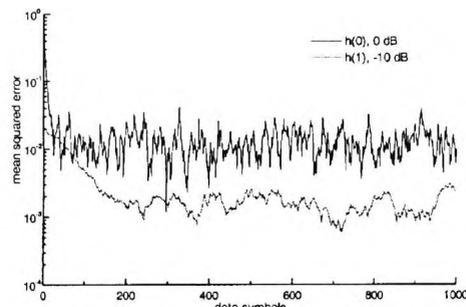

Fig. 5 Acquisition performance of the hybrid algorithm on a two path Rayleigh fading channel.

In order to get the same results with the LMS algorithm the step sizes have to be reduced by a factor of $\sqrt{2}$.

B. BIT ERROR RATE

Fig. 6 shows the simulation results of the hybrid algorithm on a two path Rayleigh fading channel. The delay of the echo path is fixed to one symbol period. Therefore, the direct and echo signal are uncorrelated. The two dotted lines give the well known theoretical bounds of DQPSK without and with two fold diversity in a maximal ratio receiver, where the receiver velocity is assumed to be zero. The straight lines show the performance for equal step sizes of 0.1 for direct and echo path, and for different powers of the echo signal. In case of equal signal powers the gain of a diversity receiver can almost be achieved with the hybrid algorithm. The residual bit error rate is due to the nonzero receiver velocity in the simulations. If the power of the echo signal is reduced, the bit error rate increases as expected.

The different symbols at Es/No = 20 dB in Fig. 6 show the bit error rate for the two reduced step sizes of 0.05 and 0.01 for the echo path. Corresponding to the results of the Kalman estimator, the BER for equal signal powers increases, whereas for an echo power of -20 dB the BER decreases. If the step size of the echo path is equal to 0.01 the results of the three power levels and the value of the theoretical curve without diversity are comparable, i.e. no diversity gain is achieved.

These results can be used to derive a strategy for the power estimation of the different paths. If a continuous data transmission is assumed, the step sizes can be adjusted adaptively, starting with very small values for the echo paths.

According to the Kalman estimator (22), the step sizes depend on the signal to noise ratio. In a practical receiver it is, however, not appropriate to compute the step size for every single value of Es/No. The curves in Fig. 7 give the results for three fixed step sizes, calculated from (22) for an Es/No of 10,15, and 20 dB in a range of Es/No from 10 to 20 dB. The curve which has been calculated for an Es/No of 15 dB shows, that for a wide range of Es/No the step size which has been calculated for the value in the middle of the Es/No-interval is sufficient.

VCT '98
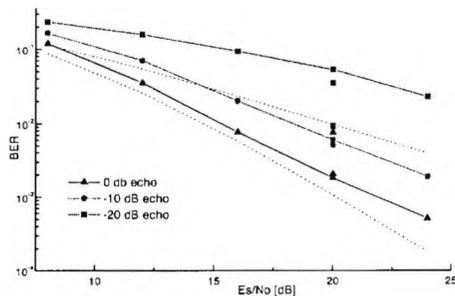
Fig. 6 Hybrid algorithm on a two path Rayleigh channel,
fd*Ts = 0.0003, roll-off = 1.0
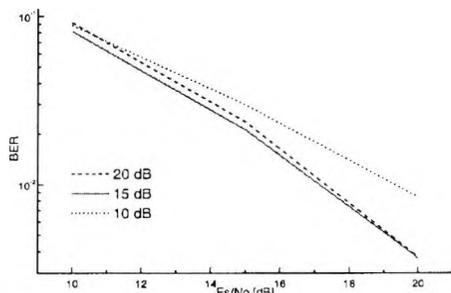
Fig. 7 Comparison of different step sizes, relative signal
power of the echo path is -10 dB, roll-off = 1.0
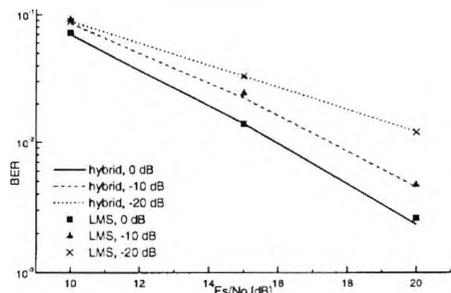
Fig. 8 LMS and hybrid algorithm on a two path Rayleigh
channel, roll-off = 0.5

Based on the Rayleigh fading channel, which is described
above, hybrid and LMS algorithm are compared in Fig. 8 for
different signal powers of the echo path. The step sizes of the
LMS algorithm are calculated with (22) for an Es/No of 15
dB. The step size of the hybrid algorithm are corrected with a
factor of $\sqrt{2}$. Under these assumption both algorithms result
in the same performance.

C. COMPLEXITY

In addition to the results with regards to the error rate,
complexity is also a decisive point when a receiver algorithm
for a new system has to be selected.

The two receiver algorithms for OQPSK are compared in
the number of multiplications (divisions) and additions
(subtractions). Tab. 1 gives the number of computations of a
component (real or imag.) of a channel coefficient for one state
in the trellis. The factor in the table indicates the simulation
time and shows the advantage of the LMS algorithm. Because
of one additional multiplication the hybrid algorithm is slower.

Tab. 1 Comparison of the LMS and the hybrid algorithm

	OQPSK LMS	OQPSK hybrid
factor	1.0	1.16
mult.	12	13
add.	13	11

V. CONCLUSION

A receiver structure for differentially encoded OQPSK
based on a trellis decoder with channel estimation through PSP
is derived in this paper. Three estimation principles are synthe-
sised and investigated. The simplified Kalman approach shows
the best results. A fast acquisition and a good tracking per-
formance of the algorithms in multipath Rayleigh fading chan-
nels is confirmed through simulations.

ACKNOWLEDGEMENT

REFERENCES

[1] R. Raheli, A. Polydoros, C.-K. Tzou, "Per-Survivor Proc-
essing: A General Approach to MLSE in Uncertain Environ-
ments", *IEEE Transactions on Communications*, VOL 43, pp.
354-364, February/March/April 1995.

[2] N. Seshadri, "Joint Data and Channel Estimation Using
Blind Trellis Search Techniques", *IEEE Transactions on
Communications*, VOL 42, pp.1000-1011, April 1994.

[3] S. Hischke, J. Habermann, R. Comley, "Receiver for Dif-
ferentially Encoded OQPSK - A Comparison of Differential
Demodulation with Joint Data and Channel Estimation", *Pro-
ceedings International Confetence on Telecommunications*,
pp. 1055-1060, Melbourne, Australia, 1997.

[4] W. Liu, H. Romanens, "Adaptive Channel Equalization for
High Speed Train", *Proceedings Vehicular Technology Con-
ference*, pp. 225-229, Sweden, 1994.

[5] B. Anderson, B. Moore, "Optimal Filtering", *Prentice-
Hall*, 1979.

New Results on the Effects of Nonlinear Amplifiers on $DOQPSK$ and $\pi/4 - DQPSK$ Signals

New Results on the Effects of Nonlinear Amplifiers on DOQPSK and π/4-DQPSK Signals

Sven Hischke* and Joachim Habermann**

* City University London, England
** University of Applied Sciences, Giessen-Friedberg, Germany
Sven.Hischke@e2.fh-friedberg.de and Joachim.Habermann@e2.fh-friedberg.de

ABSTRACT

One important feature of a modulation format for digital mobile radio should be the generation of an envelope, which shows only small fluctuations and therefore supports power efficient, nonlinear amplifiers. The bandwidth efficient linear modulation formats Offset-QPSK (OQPSK) and π/4-QPSK do fulfill this requirement, where OQPSK has further reduced envelope fluctuations compared to π/4-QPSK. The latter modulation format is attractive and has been implemented in several current mobile radio systems, since it can be demodulated differentially, which is a simple and efficient measure for fast phase synchronisation and tracking on time selective channels. However, on a former conference a new receiver structure for the differential demodulation of differentially encoded Offset-QPSK (DOQPSK) signals has been presented. In this paper the effects of nonlinear amplifiers on both modulation formats applying differential demodulation are investigated by computer simulations. The different effects of nonlinear distortions from power amplifiers and the resulting adjacent channel interference are investigated by introducing a generic nonlinear amplifier model. With the aid of the generic model, the out-of-band powers of the two linear modulation formats are compared as well as the BER in an additional AWGN and Rayleigh fading environment.

I. INTRODUCTION

For a series of mobile communication systems which will be introduced in the near future, an investigation of the modulation formats and the algorithms for channel estimation and equalisation is required in order to define the air interface. Two important requirements should be fulfilled by the modulation format. Firstly, it should be bandwidth and spectrum efficient. Secondly, since the power sources are limited in radio subscriber units, the generation of a modulated signal with an envelope, which shows only small fluctuations, is recommended. An envelope with reduced fluctuations supports power efficient, nonlinear amplifiers [1]. The linear modulation format π/4-QPSK fulfills the above mentioned requirements and is implemented in several current mobile radio systems. The choice for π/4-QPSK is motivated by two observations: Firstly, the envelope of π/4-QPSK has no zeros. This reduces the generation of harmonics and the distortion of the signal

by using nonlinear amplifiers. Secondly, a differential precoding of π/4-QPSK signals allows for a simple differential demodulation. Differential demodulation is an attractive combined channel estimation and demodulation technique on fast time selective fading channels, i.e., it should only be used for systems where the product of symbol rate and transmission channel coherence bandwidth is large against one. Differential demodulation does not resolve intersymbol interference introduced through a frequency selective channel. In addition to π/4-QPSK, OQPSK is also an attractive modulation format. Although OQPSK shows further reduced envelope fluctuations, in the past the non-availability of a differential demodulator for OQPSK led to a clear preference for π/4-QPSK in the TETRA (Terrestrial European Trunked Radio) system. Differential Demodulation of OQPSK generates a mutual interference between the I-channel and the Q-channel of the demodulated signal. In [2] a new algorithm for the differential demodulation of differentially encoded OQPSK (DOQPSK) signals has been presented which uses a trellis decoder to resolve the mutual interference of the I- and Q-channel after differential preprocessing.

In Section II some short remarks on the investigated receiver structures for π/4-DQPSK and DOQPSK are given. Section III and IV represent the main part of this paper. In Section III a generic nonlinear amplifier model is introduced, which allows for the investigation of both threshold and saturation effects. The generic model is used for all investigations on nonlinear channels. Based on the amplifier model the out-of-band power of DOQPSK and π/4-DQPSK are compared. Section IV, finally, gives the bit error rate simulation results of both modulation schemes under nonlinear amplification in an AWGN and an time selective mobile radio channel.

II. RECEIVER ALGORITHM

Fig.1 shows the block diagram of the DOQPSK receiver. The received signal $v(t)$ is down-converted and low pass filtered with square root raised cosine filters giving the I- and Q-channels of the resulting complex baseband signal. The differential preprocessing, which succeeds the receiver filter eliminates the unknown transmission channel phase and provides the input for the metric computation of the trellis decoding algorithm. For details on the receiver structure see [2, 3]. It has been shown that in the case

of Nyquist 1 pulse shaping filters with a roll off factor of $r = 1$ a trellis decoder with 8 states using the Viterbi algorithm is sufficient to resolve any mutual cross channel interference. If a roll off factor $r < 1.0$ (down to $r = 0.35$) is chosen, a trellis decoder with 16 states is used, applying the M-algorithm. In this case the cross channel interference can not be be resolved entirely, resulting in a small degradation, which will be shown with the simulation results. The receiver algorithm of $\pi/4$-DQPSK, on the other hand,
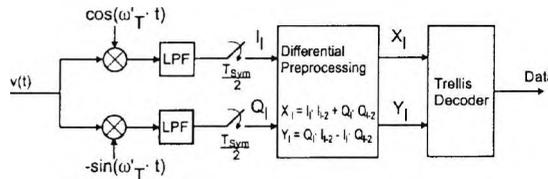

Figure 1: DOQPSK Receiver Model

uses the standard implementation. Decisions are made on a symbol by symbol basis. The algorithm is, e.g., described in [4] and the definition of the modulation format for the TETRA system is given in [5].

III. NONLINEAR AMPLIFIERS

A. Generic Amplifier Model

In nonlinear amplifiers typically two effects due to the nonlinear characteristic occur, AM-AM conversions and AM-PM conversions. With the generic amplifier model only the AM-AM conversion due to a nonlinear input output characteristic of the amplitude will be considered. The output phase versus input amplitude characteristic (AM-PM conversion) is not included in the model for the purpose of evaluating the dominant effect of the nonlinear amplitude factor alone. Two kinds of nonlinear amplitude distortions may appear on a power amplifier: saturation of the amplifier for high input voltage and blocking of the amplitude when the input signal is lower than the amplifier threshold (weak conduction region). To examine the different effects of nonlinearity, two separate amplifier models are used (Fig.2). The transfer characteristics are modelled by a third order spline function. The tables below the figures show the fixed points in the spline function. The weak conduction region in the left curve is influenced through the threshold paramter a. In the right curve the parameter c adjusts the edge-of-saturation amplitude. With the additional parameter e the amplitudes b and d have been fixed. All amplitudes are normalized to the average amplitude D, which is the root of the average power of the modulated signal.

B. Out-of-Band Power

The upper part of Fig.3 gives the complex phase plots of $\pi/4$-DQPSK and DOQPSK at the output of the transmitter filter in order to compare the envelope fluctuations of both modulation formats. In both cases a square root raised cosine filter with roll off factor $r = 0.35$, forming the transmitting part of a Nyquist 1

	in	out
p(0)	0	0
p(2a)	2a	a
p(1)	1	1-a
p(2)	2	2-a
p(i)	i	i-a

(a)

|	in	out
p(0)	0	0
p(0.3)	0.3	0.3
p(0.6)	0.6	0.6
p(b)	b=c-e	b
p(d)	d=c+2e	c

(b)
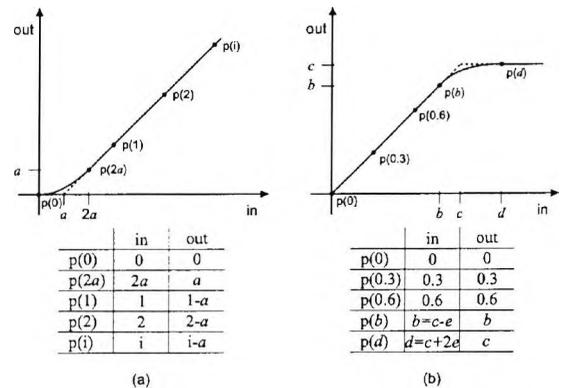
Figure 2: Amplitude transfer characteristic of a power amplifier; (a) weak conduction, (b) saturation

filter, is used. The inverse transmitter amplitude distributions of $\pi/4$-DQPSK and DOQPSK are shown in the lower part of Fig. 3. While $\pi/4$-DQPSK has a minimum amplitude of $0.18D$, the amplitude of DOQPSK is never less than $0.46D$. The peak amplitude of DOQPSK, however, is slightly larger, $1.51D$ instead of $1.48D$ for $\pi/4$-DQPSK. The peak-to-average power ratios are 3.56 dB and 3.41 dB for DOQPSK and $\pi/4$-DQPSK, respectively. With a larger r the conditions change to a better performance of DOQPSK. For $r = 0.5$ the values are $0.58D$, $1.41D$, and 3.01 dB for DOQPSK and $0.28D$, $1.44D$, and 3.15 dB for $\pi/4$-DQPSK. Fig.4

(a)   (b)

(c)
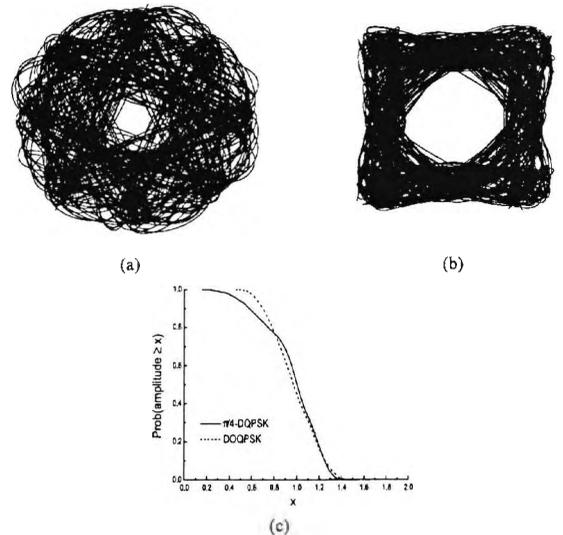
Figure 3: Complex phase plots of (a) $\pi/4$-QPSK and (b) OQPSK with a square root raised cosine Nyquist filter ($r = 0.35$); (c) Inverse amplitude distribution

shows the computed (fractional) out-of-band power as a function

of bandwidth B times symbol duration T_{Sym} for the amplifiers with amplitude characteristics from Fig.2, assuming a roll off factor of 0.35. The out-of-band power P_{out} is defined as:

$$P_{out} = 1 - \frac{\int_{-B/2}^{B/2} W(f)\, df}{\int_{-\infty}^{\infty} W(f)\, df} \qquad (1)$$

where $W(f)$ is the simulated power spectral density of the signal around the carrier frequency. Namely, P_{out} is the fraction of the total transmitted power that is found at frequencies separated more than $B/2$ from the carrier frequency. The abscissa of Fig.4 indicates the bandwidth B normalized by the symbol rate $f_S = 1/T_{Sym}$ as BT_{Sym}. Fig.4(a) gives the results for the effect
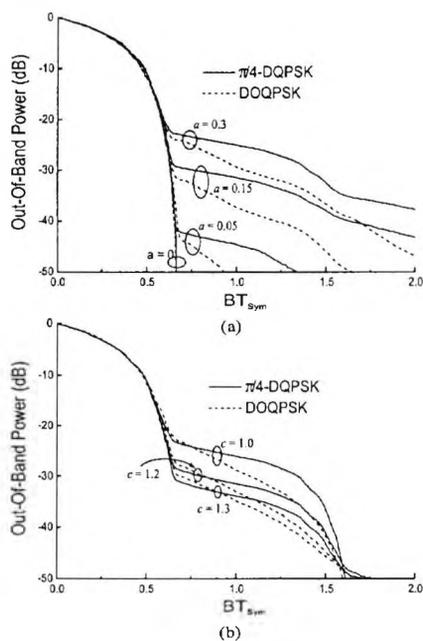

Figure 4: Out-of-band power ($r = 0.35$)

of a nonlinearity due to the weak conduction region. DOQPSK shows less spectral spreading than π/4-DQPSK, since the former has less energy in the deadband. However, it is interesting that even with small values of parameter a as 0.05 or 0.15, there is still some far-out spectrum radiation for both schemes, which have minimum amplitudes larger than these values. This is due to the fact that the presence of the threshold amplitude a in the transfer curve acts as a constant offset to the output amplitude and introduces an added noise like signal, which has a constant amplitude a and random phase, superimposed on the amplified distortion-free signal. Thus the out-of-band radiation is directly connected with a and occurs even for input signals with minimum amplitude above a. The effect of nonlinearity due to the saturation is

shown in Fig.4(b). Due to the similar amplitude distribution of both modulation schemes for higher amplitudes, the differences in the out-of-band power are smaller as in Fig.4(a). In spite of the fact that the peak amplitude of DOQPSK is larger than those of π/4-DQPSK, DOQPSK has less energy in the region around the center frequency of the adjacent channel. The gain of DOQPSK is increased, if pulse shaping filters with larger roll off factors are used.

IV. SIMULATION RESULTS

A. Linear Channel

Fig.5(a) compares the bit error rates BER of π/4-DQPSK and DO-QPSK with different roll off factors r as a function of the ratio of symbol energy to noise power density Es/No on an AWGN channel. The curves in Fig.5(a) show, that the optimal trellis de-
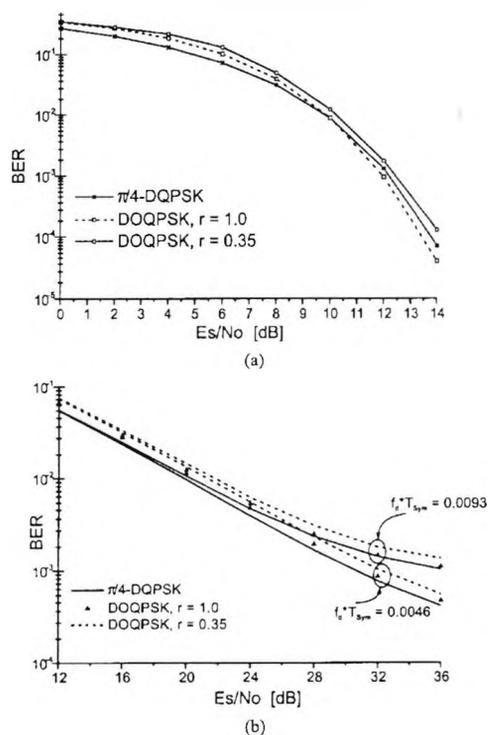

Figure 5: π/4-DQPSK and DOQPSK (a) on an AWGN channel and (b) on a Rayleigh fading channel

coder with 8 states for a roll off factor of $r = 1.0$ as well as the reduced complexity trellis decoding algorithm with 16 states for $r = 0.35$ lead, in the worst case, to a very small performance degradation compared to π/4-DQPSK for error rates less than 10^{-2}. The losses for smaller values of Es/No are a result of the implemented Euclidean metrics in the trellis decoder, which

are not optimal. In Fig.5(b) a Rayleigh fading channel is considered. The product of maximum Doppler frequency f_d and symbol duration T_{Sym} is chosen to be 0.0046 and 0.0093. At 450 MHz and a data rate of 36 kbit/s (18 ksym/s), which are the parameters of the TETRA system, the two values corresponds to a vehicle speed of 200 and 400 km/h. For a roll off factor of 1.0 and $f_d * T_{Sym} = 0.0093$ the degradation compared to π/4-DQPSK is negligible in the range of interest. For $r = 0.35$ DOQPSK shows only a small degradation over the full range. The results in Fig.5 have already been presented at a conference ([2]). These results are, however, a basis for comparison of the essential results on nonlinear channels.

B. Nonlinear Channel

Weak Conduction Region

A major motivation for the implementation of linear modulation formats with reduced envelope fluctuations is the reduction of the expenditure for linearization of nonlinear amplifier amplitude transfer curves. Fig.6 compares both modulation schemes DO-QPSK and π/4-DQPSK at a fixed bit error rate of 10^{-3} under relative adjacent channel power P_{AC}[1] and without noise for two values of the threshold parameter a. The two adjacent channels
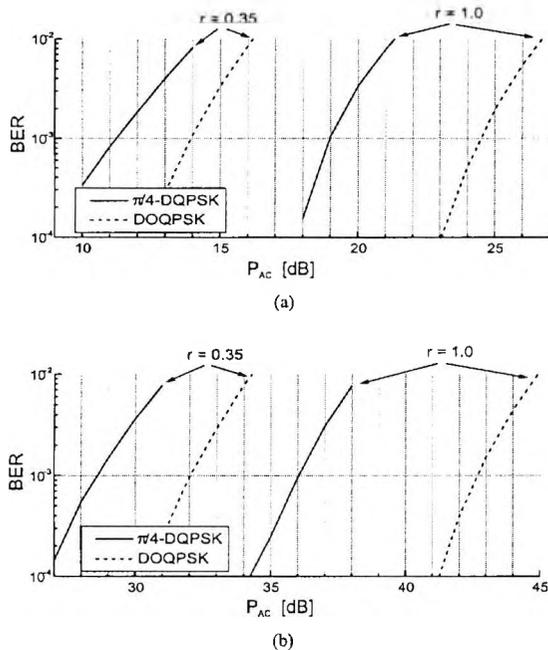

Figure 6: BER performance under ACI without noise using the amplifier model described in Fig.2a; (a) parameter $a = 0.3$, (b) $a = 0.05$

are of the same type as the original signal. Channel spacing is

---

[1] $P_{AC} = 10$ dB means that the average power of the adjacent channel is 10 dB above the average power of the information bearing channel.

$f_s * (1 + r)$, i.e., there is no overlap of the different spectra without the nonlinear amplifier. Independent of the chosen value of a, the use of DOQPSK results in a gain of more than 5.5 dB for a roll off factor of 1.0 and about 3.0 dB for $r = 0.35$. From another point of view DOQPSK can operate at a higher threshold value a to obtain the same error rate as π/4-DQPSK. Thus, the effort to linearize the amplifier characteristic is reduced, when DOQPSK is implemented. The simulation results in Fig.6 are obtained for adjacent channels which are synchronized to the measured channel. If this assumption is not fulfilled any more, e.g., if the adjacent channels are shifted by half a symbol period in the worst case, π/4-DQPSK shows a distinct degradation, especially for $r = 1.0$. The values of DOQPSK almost remain stable. Fig.7 gives the bit error performance under a fixed relative adjacent channel power of 10 dB in an AWGN environment. In the simulations the nonlinear amplifier characteristic with a parameter a of 0.3 is used. A P_{AC} of 10 dB has been chosen in Fig.7, since it follows from
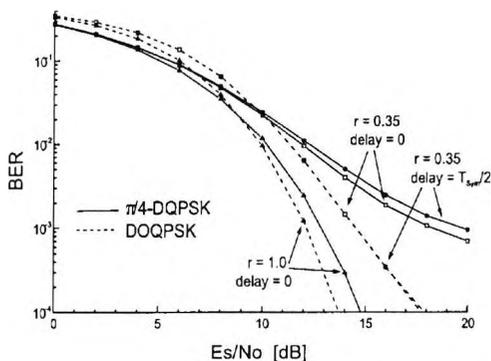

Figure 7: BER performance under ACI with $P_{AC} = 10dB$ and parameter $a = 0.3$ on an AWGN channel

Fig.6, that a BER less than 10^{-3} can be obtained for both modulation formats in the noise free case. Consequently, higher values of P_{AC} would shift the BER floor in Fig.7 to higher BER values. As expected, the gain of DOQPSK under nonlinear amplification is confirmed on an AWGN channel. If an error rate of 10^{-3} is considered, DOQPSK achieves a gain of 3.7 dB in the case of $r = 0.35$ and synchronized adjacent channels, and a gain of 5.1 dB, if the adjacent channels are shifted by $T_{Sym}/2$. The simulation results for a roll off factor of 1.0 and a delay of $T_{Sym}/2$, which are not drawn in Fig.7, show that DOQPSK has nearly the same performance as for zero delay, while π/4-DQPSK decreases to the values of $r = 0.35$. Fig.8 compares the algorithms in the time selective Rayleigh fading environment. The two bottom curves represent the results from Fig.5(b). In the linear case DOQPSK shows a small degradation compared to π/4-DQPSK. If the nonlinear amplifier is included in the simulations, the behavior is changed. Depending on the roll off factor r, the relative power of the adjacent channels P_{AC}, and the threshold parameter a, a performance gain of DOQPSK is obtained. Time shifts of the adjacent channels do have no significant influence on the BER in a Rayleigh fading environment.
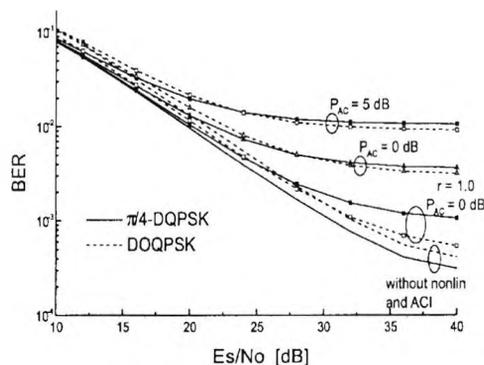
Figure 8: BER performance under ACI, $r = 0.35$, $a = 0.3$ and $f_d * T_{Sym} = 0.0046$ on a Rayleigh fading channel

Saturation Region

The investigations on amplitude distribution and out-of-band power in Section III have demonstrated, that in the case of saturation, differences in the error rate performance of both modulation schemes are hardly to be expected, when a roll off of 0.35 is chosen. In Fig.9 this assumption is confirmed. In opposite to an improvement of nearly 10 dB in the case of $r = 1.0$, only a gain of 1 dB is obtained, when $r = 0.35$. In the case of a higher value of the input saturation amplitude c, the gain will even be smaller. Simulations in an AWGN and Rayleigh fading environment yield no significant performance differences.
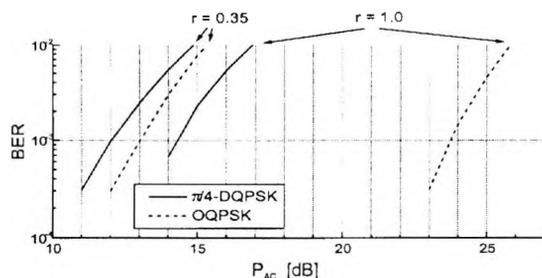

Figure 9: BER performance under ACI without noise using the amplifier model described in Fig.2b, parameter $c = 1.0$

V. CONCLUSION

Two receiver structures for differential demodulation of filtered DOQPSK and $\pi/4$-DQPSK signals have been investigated on nonlinear channels. The essential elements of the DOQPSK demodulator are a differential preprocessing unit and a trellis decoder, whose complexity depends on the chosen roll off factor of the pulse shaping filters. In neither case the number of states of the trellis decoder for DOQPSK exceeds 16 states. In linear channels both modulation formats behave the same. However, the

smaller envelope fluctuations of DOQPSK signals lead to a clear performance gain if nonlinear amplifiers are used. The expenditure for the linearization of the amplifier characteristic in the weak conduction region can be reduced considerably by the use of DOQPSK instead of $\pi/4$-DQPSK. The trade off between the reduced complexity of the transmitter amplifier and the increased complexity of the receiver baseband processing for DOQPSK may lead to a decision on a case by case basis. It is, however, certainly worthwhile to use this additional degree of freedom in the design of future mobile communication systems.

REFERENCES

[1] S. Ariyavisitakul and T.-P. Liu. Characterizing the effects of nonlinear amplifiers on linear modulation for digital portable radio communications. *IEEE Trans. Vehicular Technology*, 39(4):383–389, November 1990.

[2] S. Hischke, J. Habermann, and R. Comley. Receivers for differentially encoded OQPSK - a comparison of differential demodulation with joint data and channel estimation -. In *Proc. IEEE Int. Conf. on Telecommunications*, pages 1055–1060, Melbourne, Australia, 1997.

[3] C. Guenther and J. Habermann. Process for transmitting digital signals which combines advantages of OQPSK and $\pi/4$-QPSK. United States Patent 5,615,230, March 1997.

[4] K. Feher. Modems for emerging digital cellular-mobile radio systems. *IEEE Trans. Vehicular Technology*, 40(2):355–365, May 1991.

[5] Radio equipment and systems; trans-european trunked radio; voice plus data part 2: Air interface. Technical Report ETS 300 392-2, European Telecommunications Standard Institute, 1995.

A New Differential Demodulator for Offset-QPSK

A New Differential Demodulator for Offset-QPSK

Sven Hischke and Joachim Habermann

Abstract One important feature of a modulation format for digital mobile radio should be the generation of an envelope, showing only small fluctuations and therefore supporting power efficient, nonlinear amplifiers. A nonlinear modulation format such as Gaussian Minimum Shift Keying (GMSK), see [1], which is implemented in the Global System for Mobile Communications (GSM), has a constant envelope. One disadvantage of such a nonlinear format is its reduced bandwidth efficiency. Another drawback is the increase in bit error rate, i.e., a nonlinear format is less power efficient than a linear format when the complexity of the receiver implementation is also considered. The linear modulation formats Offset-QPSK (OQPSK) and $\pi/4$-QPSK (QPSK: Quadrature Phase Shift Keying, 4-PSK) have a better bandwidth and power efficiency than GMSK and the envelope of both formats shows no zeros. OQPSK, which shows reduced envelope fluctuations compared to $\pi/4$-QPSK, has been implemented only in systems which employ spread spectrum techniques, i.e., it is not used in narrow band systems. This is due to the non-availability of an algorithm for differential demodulation of OQPSK. Differential demodulation is a simple and efficient measure for fast phase synchronisation and tracking. This paper presents a new receiver structure for differential demodulation of differentially encoded Offset-QPSK (DOQPSK) signals on time selective mobile radio channels. The bit error rate (BER) of the new receiver is compared with $\pi/4$-DQPSK on linear additive white Gaussian noise (AWGN) and Rayleigh fading channels, with transmitter and receiver square root raised cosine pulse shaping being assumed in all cases. The different effects of nonlinear distortions from power amplifiers and the resulting adjacent channel interference are investigated by introducing a generic nonlinear amplifier model. With the aid of the generic model the out-of-band powers of the two linear modulation formats are compared as well as the BER in AWGN and Rayleigh fading environments.

Keywords Mobile radio systems, modulation, offset-QPSK.

1. Introduction

For a number of mobile communication systems which will be introduced in the near future, investigations of modulation formats and algorithms for channel estimation and equalisation are required in order to define the air interface. Two important requirements should be fulfilled by the modulation format. Firstly, it should be bandwidth and spectrum efficient. Secondly, since the power sources are limited in subscriber radio units, the generation of a modulated signal with small envelope fluctuations is recommended. An envelope with reduced fluctuations supports power efficient, nonlinear amplifiers. The attempt to fulfill these two requirements with one modulation format has been and still is subject of many investigations. In the new mobile communication standard Terrestrial Trunked Radio (TETRA) the modulation format $\pi/4$-QPSK was specified. The same modulation format is also used by the Japanese Digital Cellular (JDC) System and for the second generation American Digital Cellular Standard (ADC-U.S. TIA 45.3). The choice of $\pi/4$-QPSK is motivated by two observations: Firstly, the envelope of $\pi/4$-QPSK has no zeros. This reduces the generation of harmonics and the distortion of the signal by nonlinear amplifiers [2]. Secondly, a differential precoding of $\pi/4$-QPSK signals allows for simple differential demodulation. Differential demodulation is an attractive combined carrier phase estimation and demodulation technique on fast time selective fading channels. Differential demodulation, however, does not resolve intersymbol interference introduced by a frequency selective channel. In addition to $\pi/4$-QPSK, OQPSK was also a candidate for the TETRA system. Although OQPSK shows further reduced envelope fluctuations, the non-availability of a differential demodulator for OQPSK led to a clear preference for $\pi/4$-QPSK. Fig. 1 gives the complex phase plots at the output of the transmitter filter in order to compare the envelope fluctuations of both formats. In both cases a square root raised cosine filter, forming the transmitting part of a Nyquist 1 filter, is used. The difficulty of differential demodulation is due to the time overlap of the I- and Q-signals, because in OQPSK the I- and Q-channels are staggered, i.e., time shifted by half a symbol period.

Fig. 2 shows the block diagram of the DOQPSK transmitter. From the serial data stream two bits are mapped in parallel to the two branches of the quadrature modulator. The bits of each branch are differentially encoded and filtered with square root raised cosine Nyquist 1 filters. The delay of half a symbol period $T_S/2$ in one branch gives the desired offset of one component in the modulated signal $s(t)$. It should be pointed out that the DOQPSK transmitter in Fig. 2 differs from the standard OQPSK transmitter by merely including a differential encoder in each channel. Applying a coherent demodulation, the I- and Q-components can be decoded separately without crosscomponent interference, if the constant time shift between the

---

Dipl.-Ing.(FH) S. Hischke, Department of Electrical, Electronic and Information Engineering, City University, London, England, E-mail: S.Hischke@city.ac.uk.
Prof. Dr.-Ing. J. Habermann, Department of Electrical Engineering II, Fachhochschule Giessen-Friedberg, Germany, E-mail: haber@e2.fh-friedberg.de.
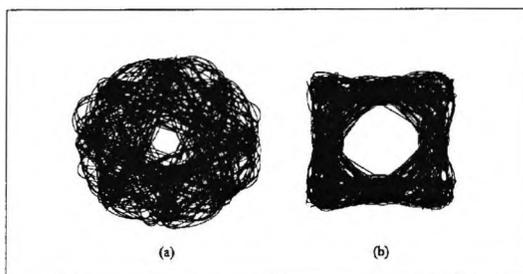
Fig. 1. Complex phase plots of (a) $\pi/4$-QPSK and (b) OQPSK with a square root raised cosine Nyquist filter ($r = 0.35$).
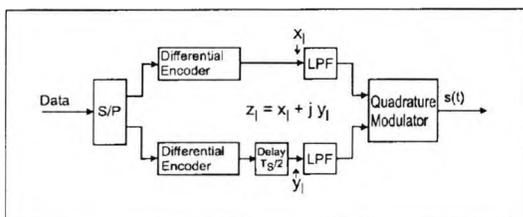

Fig. 2. DOQPSK transmitter model.
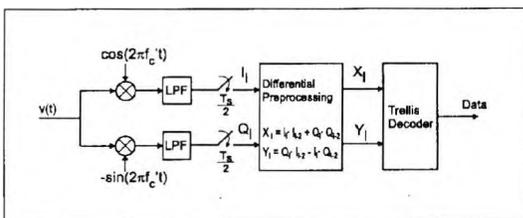
Fig. 3. DOQPSK receiver model.

two component signals is taken into account. The problem, then, is to derive an efficient carrier phase synchronisation algorithm. In a differential scheme this time shift leads to a mutual interference of the I- and Q-components. Conventional differential demodulators, like those described in [1], can not be applied anymore. Motivated by the assumption that OQPSK modulated signals require coherent demodulation [3], some authors ([4] and [5]) have investigated the use of differential demodulation. In both analyses, however, only unfiltered signals are considered. In [6] a new algorithm was presented which uses a trellis decoder to resolve the mutual interference of the I- and Q-components after differential preprocessing. The described algorithm is suitable only for raised cosine filtered signals with a roll-off factor $r = 1.0$.

In Section 2 of this paper the basic structure of the new algorithm is discussed and an advanced algorithm suitable for general roll-off factors and with reduced complexity is given. The new algorithm has partly been presented at

the ICT'97 [7]. In Section 3 a generic nonlinear amplifier model is introduced. The model is used for all investigations on nonlinear channels. Based on the amplifier model the out-of-band powers of DOQPSK and $\pi/4$-DQPSK are compared. Section 4, finally, gives the bit error rate simulation results of both modulation schemes on several linear and nonlinear channels.

2. Receiver Algorithm

Fig. 3 shows the block diagram of the receiver. The received signal $v(t)$ is down- converted with a synthesiser frequency f_c' slightly mistuned to the carrier frequency f_c and low pass filtered with square root raised cosine filters resulting in the signal $r(t)$. The I- and Q-components of the complex baseband signal $r(t)$ are sampled twice per symbol interval T_S, i.e., once per bit interval. The discrete signal r_l at epoch l for an AWGN channel becomes:

$$
\begin{aligned}
r_l &= I_l + \mathrm{j}\,Q_l \\
&= \left[z_l + \sum_{i=1}^{\infty} g_i(z_{l-i} + z_{l+i}) \right] \mathrm{e}^{\mathrm{j}\,\phi_l} + n_l', \\
g_i &= 0, \quad i = 2, 4, 6, \ldots,
\end{aligned}
\tag{1}
$$

where g_i designates the overall impulse response of transmitter and receiver filters, which has zeros at multiples of a symbol period due to the Nyquist 1 criterion. ϕ_l denotes the slowly time varying phase (slow in comparison to a symbol or bit interval), $z_l = x_l + \mathrm{j}\,y_l$ gives the encoded, complex data at the transmitter (see Fig. 2), and n_l' is the filtered noise signal. As a consequence of the time offset in the transmitter and the bit-by-bit sampling in the receiver, z_l contains only the data of the I-channel for even sampling times and the data of the Q-channel for odd sampling times. We obtain:

$$
z_l = \begin{cases} x_l, & l = 2k \\ \mathrm{j}\,y_l, & l = 2k + 1. \end{cases}
\tag{2}
$$

Depending on the binary and differentially encoded data x_l and y_l at the transmitter side, the equations at even $(2k)$ and odd sampling times $(2k + 1)$ can be formulated as:

$$
\begin{aligned}
r_{2k} &= I_{2k} + \mathrm{j}\,Q_{2k} \\
&= \left[x_{2k} + \mathrm{j} \sum_{i=1}^{\infty} g_i(y_{2k-i} + y_{2k+i}) \right] \mathrm{e}^{\mathrm{j}\,\phi_{2k}} \\
&\quad + n_{2k}',
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
r_{2k+1} &= I_{2k+1} + \mathrm{j}\,Q_{2k+1} \\
&= \left[\sum_{i=1}^{\infty} g_i(x_{2k+1-i} + x_{2k+1+i}) + \mathrm{j}\,y_{2k+1} \right] \mathrm{e}^{\mathrm{j}\,\phi_{2k+1}} \\
&\quad + n_{2k+1}'.
\end{aligned}
\tag{4}
$$

The differential preprocessing after the receiver filters carries out the mathematical operation $r_l r_{l-2}^*$, where r^* is the conjugate complex of r. It eliminates the unknown phase

ϕ_l and provides the input for the metric computation of the trellis decoding algorithm. The real component X_l and the imaginary component Y_l of the signal at the output of the differential preprocessing unit are:

$$X_l = \Re\left\{r_l r_{l-2}^*\right\} = I_l I_{l-2} + Q_l Q_{l-2}, \qquad (5)$$

$$Y_l = \Im\left\{r_l r_{l-2}^*\right\} = Q_l I_{l-2} - I_l Q_{l-2}, \qquad (6)$$

where $\Re\{\}$ takes the real part and $\Im\{\}$ the imaginary part of an operand. The trellis decoder in Fig. 3 compensates for the intersymbol interference introduced through differential demodulation.

The structure of the demodulator is the same for any type of Nyquist filtering. Depending on the chosen roll-off factor only the trellis decoder has to be changed, since symbols may interfere mutually due to differential preprocessing. Let us examine the I-channel, for example. Due to the Nyquist 1 criterion, there is no interference from other I-channel bits. However, there may be a number of Q-channel bits which interfere with the I-channel bit considered.

In the case of a roll-off factor $r = 1.0$, only the preceding and succeeding Q-channel bit interferes on a given I-channel bit since the signal also satisfies the Nyquist 2 criterion (see Fig. 4). Therefore, the impulse response g_i in eq. (1) is 1.0 for $i = 0$, 0.5 for $i = 1$, and zero for all other values of i. Let us first investigate the output of the preprocessing unit at the sampling intervals $l = 2k$, when no noise is considered in r_l (see [6], [8]), i.e., the nominal values $\overline{X}_{2k}$ and $\overline{Y}_{2k}$ of the trellis decoder are calculated as:

$$\begin{aligned}\overline{X}_{2k} &= \Re\left\{r_{2k} r_{2k-2}^*\right\} \\ &= x_{2k} x_{2k-2} \\ &\quad + \frac{1}{4}(y_{2k-1} + y_{2k+1})(y_{2k-3} + y_{2k-1}), \quad (7)\end{aligned}$$

$$\begin{aligned}\overline{Y}_{2k} &= \Im\left\{r_{2k} r_{2k-2}^*\right\} \\ &= \frac{1}{2} x_{2k-2}(y_{2k-1} + y_{2k+1}) \\ &\quad - \frac{1}{2} x_{2k}(y_{2k-3} + y_{2k-1}) \quad (8)\end{aligned}$$

and then at the sampling intervals $l = 2k + 1$:

$$\begin{aligned}\overline{X}_{2k+1} &= \Re\left\{r_{2k+1} r_{2k-1}^*\right\} \\ &= y_{2k+1} y_{2k-1} \\ &\quad + \frac{1}{4}(x_{2k} + x_{2k+2})(x_{2k-2} + x_{2k}), \quad (9)\end{aligned}$$

$$\begin{aligned}\overline{Y}_{2k+1} &= \Im\left\{r_{2k+1} r_{2k-1}^*\right\} \\ &= \frac{1}{2} y_{2k+1}(x_{2k-2} + x_{2k}) \\ &\quad - \frac{1}{2} y_{2k-1}(x_{2k} + x_{2k+2}). \quad (10)\end{aligned}$$

The construction of a trellis is now easy. In the case of $l = 2k$, both $\overline{X}_{2k}$ and $\overline{Y}_{2k}$ are functions of $y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k}$, and y_{2k+1}. Therefore, transitions from an initial state $(y_{2k-3}, x_{2k-2}, y_{2k-1}, x_{2k})$ to a final state $(x_{2k-2},$

$y_{2k-1}, x_{2k}, y_{2k+1})$ have to be considered. To these transitions values of $\overline{X}_{2k}$ and $\overline{Y}_{2k}$ as specified in (7) and (8) are associated. Similarly, transitions from an initial state $(x_{2k-2}, y_{2k-1}, x_{2k}, y_{2k+1})$ to the final state $(y_{2k-1}, x_{2k}, y_{2k+1}, x_{2k+2})$ have to be taken into account at the odd sampling intervals. From the decoded bits x_l and y_l an estimate of the input data sequence of the transmitter is obtained with standard differential decoding.

Now, it is interesting to note that $\overline{X}_{2k+1} = \overline{X}_{2k}$ and $\overline{Y}_{2k+1} = -\overline{Y}_{2k}$ when the initial and final states are considered at even and odd sampling intervals. This means that the resulting 16 state trellis with two transitions leaving and entering each node is essentially time invariant. In Table 1 the transitions and their associated values of $\overline{X}_l$ and $\overline{Y}_l$ are given. Since both $\overline{X}_l$ and $\overline{Y}_l$ are invariant when the initial and final state are complemented, only the first 8 states are included. Applying this symmetry to the trellis decoder and bearing in mind that the bits are decoded differentially an 8 state trellis results.

Table 1. Trellis for DOQPSK with $r = 1.0$.

Initial state	Final state	$\overline{X}_{2k}$ $= \overline{X}_{2k+1}$	$\overline{Y}_{2k}$ $= -\overline{Y}_{2k+1}$
1 1 1 1	1 1 1 1	2	0
1 1 1 1	1 1 1 −1	1	−1
1 1 1 −1	1 1 −1 1	0	2
1 1 1 −1	1 1 −1 −1	−1	1
1 1 −1 1	1 −1 1 1	1	0
1 1 −1 1	1 −1 1 −1	1	−1
1 1 −1 −1	1 −1 −1 1	−1	0
1 1 −1 −1	1 −1 −1 −1	−1	−1
1 −1 1 1	−1 1 1 1	0	−2
1 −1 1 1	−1 1 1 −1	−1	−1
1 −1 1 −1	−1 1 −1 1	2	0
1 −1 1 −1	−1 1 −1 −1	1	1
1 −1 −1 1	−1 −1 1 1	−1	0
1 −1 −1 1	−1 −1 1 −1	−1	1
1 −1 −1 −1	−1 −1 −1 1	1	0
1 −1 −1 −1	−1 −1 −1 −1	1	1

If a roll-off factor $r < 1.0$ is chosen, the second Nyquist criterion is not fulfilled any more and an infinite number of Q-channel bits interfere with a certain I-channel bit, since the impulse response of a Q-channel bit has nonzero values at the sampling times of all preceding and succeeding I-channel bits. Consequently an infinite number of states in the trellis decoder would be needed to resolve the intersymbol interference from the Q-channel into the I-channel and vice versa. To overcome this problem, the impulse response considered in the trellis decoder to calculate the nominal values in the trellis table, is truncated after i bit intervals at the cost of a performance degradation. Table 2 gives the number of states in the trellis decoder, when the receiver considers an impulse response of length $2i + 1$ bits. The roll-off factor of the pulse shaping filters has been fixed to $r = 0.35$. This roll-off factor is, for example, implemented in the TETRA system. To reduce the number of states and hence the receiver complexity, the considered
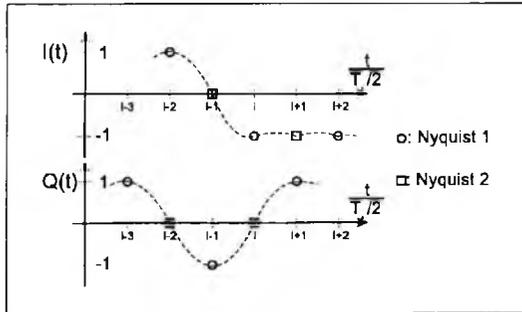
Fig. 4. I- and Q-channel signals at the receiver filter output which satisfy Nyquist criteria 1 and 2.

impulse response is truncated at $i = 3$ in the following investigations. Equation 1 then simplifies to:

$$r_l = [z_l + g_1(z_{l-1} + z_{l+1})$$
$$+ g_3(z_{l-3} + z_{l+3})]e^{j\phi_l} + n_l'. \qquad (11)$$

Table 2. Values g_i of the impulse response and number of states in the trellis decoder.

Samples i	g_i $(r = 0.35)$	No. of states in trellis decoder 2^{2i+1}
1	0.61858	2
3	−0.16244	128
5	0.05703	2048
7	−0.01382	32768
9	−0.00185	524288

Note that the sampling value g_2 of the impulse response is omitted in eq. (11), since for all even sampling values the impulse response is zero due to the Nyquist 1 criterion and the sampling at twice the symbol rate. A roll-off factor $r = 0.35$ and a truncation of the impulse response after 3 samples ($i = 3$) result in a trellis decoder with 128 states, as shown in the following. Eqs. (12) and (14) show that the input signals X_l and Y_l of the trellis decoder are functions of the transmitted bits $y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}$ and y_{l+3}:

$$X_l = \Re\left\{r_l r_{l-2}^*\right\} \qquad (12)$$
$$= F\left\{y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n_l'\right\},$$
$$Y_l = \Im\left\{r_l r_{l-2}^*\right\} \qquad (13)$$
$$= F\left\{y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n_l'\right\}.$$

The detailed yet lengthy equations relating the bits x_l and y_l to the nominal signals $\overline{X}_l$ and $\overline{Y}_l$ can be found similarly to eqs. (7)–(10) when the additive noise is omitted. Here transitions from an initial state $(y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1})$ into a final state $(y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3})$ occur. It is observed that the bits x_{l-4} and x_{l+2} do

not have any influence on X_l and Y_l since the impulse response is equal to zero at these specific sampling intervals. Consequently, four transitions always have the same pairs of nominal values in the trellis. However, x_{l-4} and x_{l+2} define a state and therefore the number of states in the trellis, since all bits must be used to store the trellis path to each state. Without this, the history information of the different paths would be lost and a Viterbi decoding is not possible. Both, initial and final state depend on eight bits resulting in a trellis with 256 states. Since lower and upper part of the trellis are symmetric, as has been shown for $r = 1.0$, the number of states can be reduced to 128.

Data decoding in the trellis decoder is implemented using the Viterbi algorithm. To simplify computational complexity, the metric in the Viterbi algorithm is obtained from the squared Euclidean distance between the input vector and the nominal value of each state, i.e., it is assumed that the additive noise is white and Gaussian. However, the differential preprocessing together with the two samples per symbol in principle invalidate this assumption. The accumulated metric $\gamma_{j,l}$ of state j at epoch l and the branch metric λ are calculated from the well-known equations:

$$\gamma_{j,l} = \min\{\gamma_{n,l-1} + \lambda_{nj,l-1}, \gamma_{m,l-1} + \lambda_{mj,l-1}\},$$
$$\lambda_{uj,l} = \left(X_l - \overline{X}_{uj,l}\right)^2 + \left(Y_l - \overline{Y}_{uj,l}\right)^2,$$
$$u = n, m. \qquad (14)$$

$\overline{X}_{uj,l}$ and $\overline{Y}_{uj,l}$ indicate the nominal values from the trellis table. n and m represent the two states in the trellis leading to the common state j and $\min\{y_1, y_2\}$ is the minimum function between y_1 and y_2. Due to the differential preprocessing the Euclidean metrics are not more optimal as indicated above. However, at higher signal-to-noise ratios the resulting performance degradation can be neglected, as will be shown by computer simulations.

A complexity reduction compared to the Viterbi algorithm can be achieved in the case of trellis decoding if not all paths of the trellis are considered. The number of states according to Table 1 can then be reduced dramatically. The M-algorithm [9] (in [10] referred to as SA(B)) stores merely the M most probable paths through the trellis. Table 3 gives a comparison between the required simulation times, indicated by the time factor, and the achieved bit error rates for an AWGN channel at a signal-to-noise ratio per symbol (E_s/N_0) of 10 dB. For all investigations the M-algorithm has been obtained from the original trellis with 128 states, i.e., the filter impulse response in the trellis decoder has been truncated after 3 samples ($i = 3$). The Viterbi algorithm with 8 states, which is adapted to a roll-off factor of 1.0, leads to the shortest simulation time. Assuming $r = 0.35$, the trellis decoder is mismatched and, therefore, results in the highest bit error rate. The M-algorithm with 32 states requires less than half of the simulation time of the Viterbi algorithm, at almost identical bit error rate. Sorting of the M most probable paths from $2M$ possible paths at any time interval is implemented with the Quicksort algorithm. If the M-algorithm with 16 states is used, only a small degradation

Table 3. Comparison between simulation time and bit error rate; AWGN channel with $E_s/N_0 = 10$ dB.

| Algorithm	Time factor	BER
128 state Viterbi	1.00	$1.2142 \cdot 10^{-2}$
32 state M-algorithm	0.45	$1.2168 \cdot 10^{-2}$
16 state M-algorithm	0.26	$1.3826 \cdot 10^{-2}$
14 state M-algorithm	0.24	$1.5481 \cdot 10^{-2}$
8 state Viterbi	0.19	$1.6981 \cdot 10^{-2}$

Fig. 5. Amplitude transfer characteristic of a power amplifier: (a) Weak conduction, (b) saturation.

Fig. 6. Inverse amplitude distribution ($r = 0.35$). Amplitudes are normalised with D.

compared to the Viterbi algorithm with 128 states is observed. However, a gain in computer simulation time by a factor of 4 is achieved. Therefore, for all further investigations the Viterbi algorithm with 8 (for $r = 1.0$) and 128 (for $r = 0.35$) states and the M-algorithm with 32 and 16 states (for $r = 0.35$) will be used.

3. Nonlinear Amplifiers

3.1 Generic Amplifier Model

In nonlinear amplifiers typically two effects due to the nonlinear characteristic occur, AM-AM conversion and AM-PM conversion. With the generic amplifier model only the AM-AM conversion due to a nonlinear input output characteristic of the amplitude will be considered. The output phase versus input amplitude characteristic (AM-PM conversion) is not included in the model, since the purpose was to evaluate the dominant effect of the nonlinear amplitude factor alone. Two kinds of nonlinear amplitude distortions may appear at a power amplifier: saturation of the amplifier for high input voltage and blocking of the amplitude when the input signal is lower than the amplifier threshold (weak conduction region). To examine the different effects of nonlinearity, two separate amplifier models are used (Fig. 5). The transfer characteristics are modelled by a third order spline function. The tables below the figures show the fixed points in the spline function. The weak conduction region in the left curve is mainly influenced through the threshold parameter a. In the right curve the parameter c adjusts the edge-of-saturation amplitude. The parameter e defines the shaping of the curve in the saturation region. For the following investigations e will be fixed to 0.3. All amplitudes are normalised to the average amplitude D, which is the root of the average power of the modulated signal.
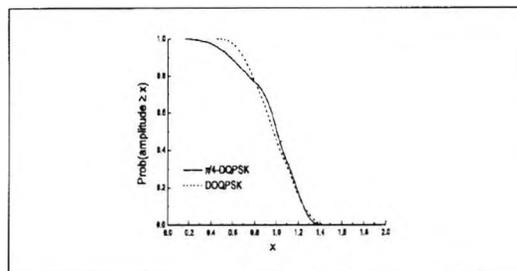
3.2 Out-of-Band Power

The inverse amplitude distributions of $\pi/4$-DQPSK and DOQPSK at the transmitter side are shown in Fig.6. Both modulation schemes use a square root raised cosine Nyquist filter with roll-off factor $r = 0.35$. While $\pi/4$-DQPSK has a minimum amplitude of $0.18D$, the amplitude of DOQPSK is never less than $0.46D$. The peak amplitude of DOQPSK, however, is slightly larger, $1.51D$ instead of $1.48D$ for $\pi/4$-DQPSK. The peak-to-average power ratios are 3.56 dB and 3.41 dB for DOQPSK and $\pi/4$-DQPSK, respectively. With a larger r the conditions change to a better performance of DOQPSK. For $r = 0.5$ the values are $0.58D$, $1.41D$, and 3.01 dB for DOQPSK and $0.28D$, $1.44D$, and 3.15 dB for $\pi/4$-DQPSK.

Fig. 7 shows the computed (fractional) out-of-band power as a function of bandwidth B times symbol duration T_S for the amplifiers with amplitude characteristics from Fig. 5, assuming a roll-off factor of 0.35. The out-of-band power P_{out} is defined as:

$$P_{out} = 1 - \frac{\int_{-B/2}^{B/2} W(f)\,\mathrm{d}f}{\int_{-\infty}^{\infty} W(f)\,\mathrm{d}f}, \tag{15}$$

where $W(f)$ is the simulated power spectral density of the signal around the carrier frequency. P_{out} is the fraction of the total transmitted power that is found at frequencies separated more than $B/2$ from the carrier frequency. Fig. 7(a) shows the effect of a nonlinearity due to the weak conduction region. DOQPSK shows less spectral spreading than $\pi/4$-DQPSK, since the former has less energy in the deadband. However, it is interesting that even with small values of the threshold parameter a as 0.05 or 0.15, there
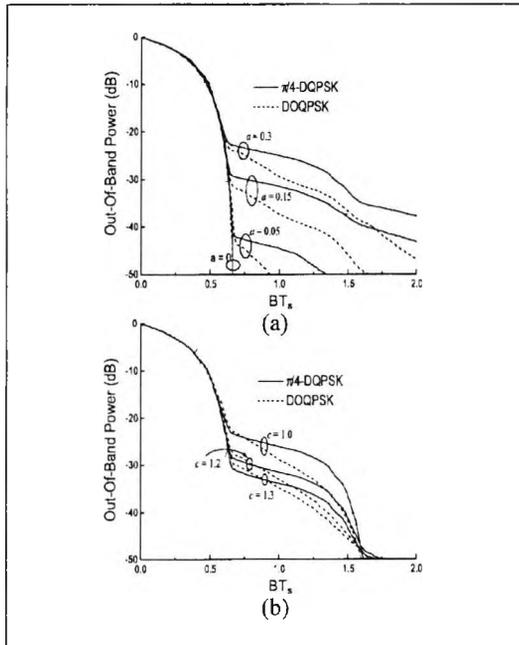
Fig. 7. Out-of-band power ($r = 0.35$).

is still some far- out spectrum radiation for both schemes, which have minimum amplitudes larger than these values. This is due to the fact that the presence of the threshold amplitude a in the transfer curve acts as a constant off-set to the output amplitude and introduces an additional noise-like signal with constant amplitude a and random phase, superimposed on the amplified distortion-free signal. Thus the out-of-band radiation is directly connected with a and occurs even for input signals with minimum amplitude above a.

The effect of a nonlinearity caused by saturation is shown in Fig. 7b. Due to the similar amplitude distributions of both modulation schemes for higher amplitudes, the differences in the out-of-band power are smaller than in Fig. 7a. In spite of the fact that the peak amplitude of DOQPSK is larger than that one of $\pi/4$-DQPSK, DO-QPSK has less energy in the region around the center frequency of the adjacent channel. The gain of DOQPSK compared to $\pi/4$-DQPSK is further increased for pulse shaping filters with larger roll-off factors.

4. Simulation Results

4.1   Linear Channel

Figs. 8 and 9 compare the bit error rates (BER) of $\pi/4$-DQPSK and different DOQPSK receiver algorithms as a function of the ratio of symbol energy-to-noise power den-
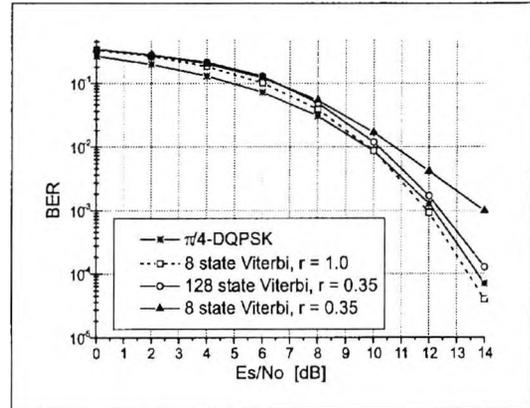

Fig. 8. $\pi/4$-DQPSK and DOQPSK on an AWGN channel.
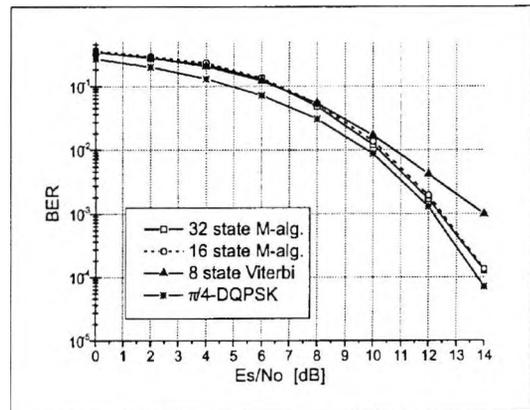
Fig. 9. M-algorithm on an AWGN channel ($r = 0.35$).

sity E_s/N_0 for an AWGN channel. The curves in Fig. 8 show that the Viterbi algorithm for DOQPSK with 8 states for a roll-off factor of 1.0 and the algorithm with 128 states for $r = 0.35$ lead to a very small degradation compared to $\pi/4$-DQPSK for error rates less than 10^{-2}. The losses for smaller values of E_s/N_0 are a result of the implemented non-optimal Euclidean metrics. The Viterbi algorithm with 8 states, which is mismatched for $r = 0.35$, shows a noticeable performance degradation.

In Fig. 9 the results for the different M-algorithms are shown. The M-algorithm with 32 states achieves the same performance as the Viterbi algorithm with 128 states. The loss due to a further reduction to 16 states in the M-algorithm is negligible. On the other hand, with 16 states a gain in simulation time of a factor of four is achieved (see Tab.3) compared to the Viterbi algorithm with 128 states.

In a narrowband mobile radio communication system, Rayleigh fading with Jakes Doppler spectrum [11] is com-
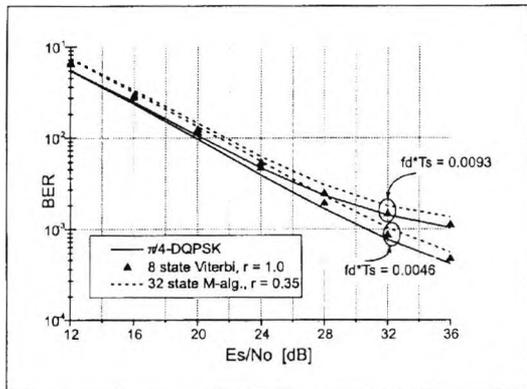
Fig. 10. π/4-DQPSK and DOQPSK on a Rayleigh fading channel.

monly considered. Results for such a type of channel are given in Fig. 10. The product of maximum Doppler frequency f_d and symbol duration T_S is chosen to be 0.0046 and 0.0093. At 450 MHz and a data rate of 36 kbit/s (18 ksym/s), which are the parameters of the TETRA system, the two values corresponds to a vehicle speed of 200 and 400 km/h. As expected, the good results of DO-QPSK in the AWGN channel are also confirmed in the Rayleigh fading channel. For a roll-off factor of 1.0 and $f_d T_S = 0.0093$ the degradation compared to π/4-DQPSK is negligible in the range of interest. The M-algorithm with 32 states and $r = 0.35$ shows only a small degradation over the full range of E_s/N_0.

4.2 Nonlinear Channel

Weak Conduction Region

A major motivation for the implementation of linear modulation formats with reduced envelope fluctuations is the cost reduction for linearization of nonlinear amplifier amplitude transfer curves. Fig. 11 compares both modulation schemes DOQPSK and π/4-DQPSK at a bit error rate of 10^{-3} under relative adjacent channel power P_{AC}[1] and without noise for two values of the threshold parameter a. The two adjacent channels are of the same type as the original signal. Channel spacing is $f_s(1 + r)$, i.e., there is no overlap of the different spectra without the nonlinear amplifier. Independent of the chosen threshold value a, the use of DOQPSK results in a gain against π/4-DQPSK of more than 5.5 dB for a roll-off factor of 1.0 and about 3.0 dB for $r = 0.35$. From another point of view DOQPSK can operate at a higher threshold value a to obtain the same error rate as π/4-DQPSK. Thus, the effort to linearise the amplifier characteristic is reduced, when DOQPSK is implemented. The simulation results in Fig. 11 are obtained

[1] $P_{AC} = 10$ dB means that the average power of the adjacent channel is 10 dB above the average power of the information bearing channel.

for adjacent channels which are synchronised to the measured channel. If this assumption is not fulfilled, e.g., if the adjacent channels are shifted by half a symbol period in the worst case, π/4-DQPSK shows a distinct degradation, especially for $r = 1.0$. The values of DOQPSK remain almost stable.

Fig. 12 gives the bit error performance under a fixed relative adjacent channel power of 10 dB in an AWGN environment. In the simulations the nonlinear amplifier characteristic with a parameter a of 0.3 is used. A P_{AC} of 10 dB has been chosen in Fig. 12, since it follows from Fig. 11, that a BER of less than 10^{-3} can be obtained for both modulation formats in the noise free case. Consequently, higher values of P_{AC} would shift the BER floor in Fig. 12 to higher BER values. As expected, the gain of DOQPSK under nonlinear amplification is confirmed on an AWGN channel. If an error rate of 10^{-3} is considered, DOQPSK achieves a gain of 3.7 dB in the case of $r = 0.35$ and synchronised adjacent channels, and a gain of 5.1 dB, if the adjacent channels are shifted by $T_S/2$. The simulation results for a roll-off factor of 1.0 and a delay of $T_S/2$ (not included in Fig. 12), show that DOQPSK has nearly the same performance as for zero delay, while π/4-DQPSK decreases to the values of $r = 0.35$.

Fig. 13 compares the algorithms in the time selective Rayleigh fading environment. The two bottom curves represent the results from Fig. 10. In the linear case DOQPSK shows a small degradation compared to π/4-DQPSK. If the nonlinear amplifier is included in the simulations, the behavior is changed. Depending on the chosen parameters r and the relative power of the adjacent channels, a performance gain of DOQPSK is obtained. Time shifts of the adjacent channels have no significant influence on the BER in a Rayleigh fading environment.

Saturation Region

The investigations on amplitude distribution and out-of-band power in Section 3 indicate that in the case of saturation, differences in the error rate performance of both modulation schemes are hardly to be expected, when a roll-off of 0.35 is chosen. In Fig. 14 this assumption is confirmed by the BER results. In contrast to an improvement of nearly 10 dB in the case of $r = 1.0$, only a gain of 1 dB is obtained, when $r = 0.35$. In the case of a higher value of the input saturation amplitude d, the gain will even be smaller. Simulations in AWGN and Rayleigh fading environments yield no significant performance differences.

5. Conclusion

A receiver structure for the differential demodulation of filtered DOQPSK signals has been derived. Its essential elements are a differential preprocessing unit and a trellis decoder, whose complexity depends on the chosen roll-off factor of the pulse shaping filters. A complexity reduction of the receiver is achieved if the M-algorithm is implemented for decoding instead of the Viterbi algorithm. A comparison of the new receiver structure with
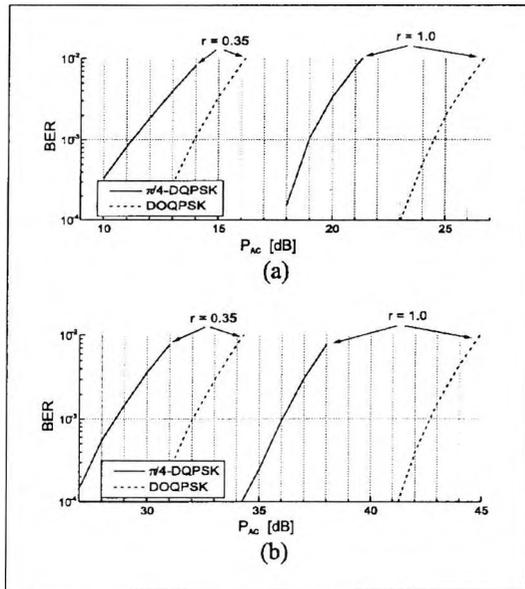
Fig. 11. BER performance under ACI without noise using the amplifier model described in Fig. 5a: (a) Parameter $a = 0.3$, (b) $a = 0.05$.
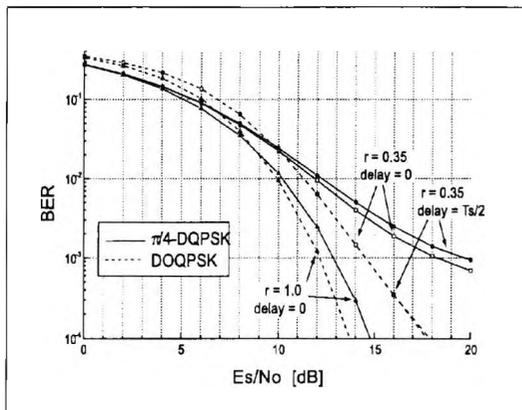
Fig. 12. BER performance under ACI with $P_{AC} = 10$ dB and parameter $a = 0.3$ on an AWGN channel.

$\pi/4$-DQPSK on different channels has been carried out. In the specific case of raised cosine pulse shaping with a roll- off factor $r = 1.0$ the performance degradation of DOQPSK on linear channels is found to be insignificant. When a roll-off factor less than 1.0 is used, only a small degradation with respect to $\pi/4$-DQPSK is obtained. The losses are due to the fact that the trellis decoder considers only a finite number of states and, therefore, is not optimal.
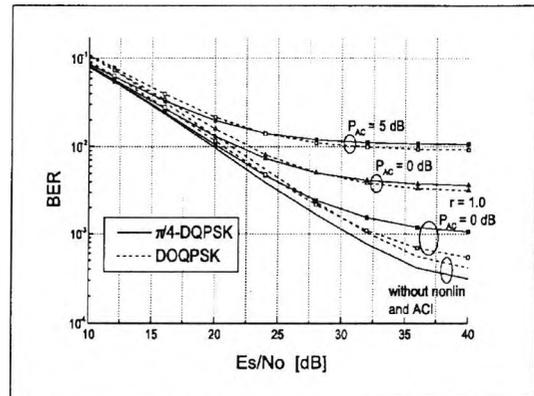

Fig. 13. BER performance under ACI, $r = 0.35$, $a = 0.3$, and $f_d T_s = 0.0046$ on a Rayleigh fading channel.
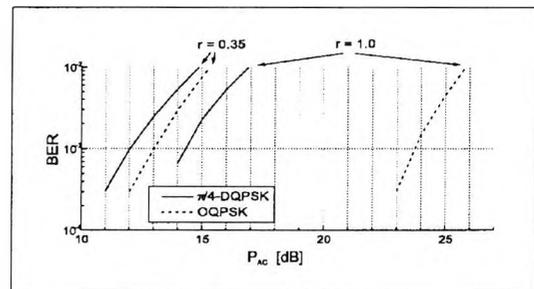

Fig. 14. BER performance under ACI without noise using the amplifier model described in Fig. 5b. Parameter $c = 1.0$.

However, the smaller envelope fluctuations of DOQPSK signals lead to a clear performance gain if nonlinear amplifiers are used. The linearization cost for the amplifier characteristic in the weak conduction region can be reduced considerably by DOQPSK instead of $\pi/4$-DQPSK. The trade off between the reduced complexity of the transmitter amplifier and the increased complexity of the receiver baseband processing for DOQPSK may lead to a decision on a case by case basis. It is, however, certainly worthwhile to use this additional degree of freedom in the design of future mobile communication systems.

Acknowledgement

References

[1] Proakis, J. G.: Digital communications. 3rd ed. New York: McGraw-Hill, 1995.

[2] Ariyavisitakul, S.; Liu, T.-P.: Characterizing the effects of non-linear amplifiers on linear modulation for digital portable radio communications. IEEE Trans. Vehicular Technology VT– 39 (1990), 383–389.

[3] Feher, K.: MODEMS for emerging digital cellular-mobile radio systems. IEEE Trans. Vehicular Technology VT– 40 (1991), 355–365.

[4] Kaleh, G. K.: Differential detection via the Viterbi algorithm for offset modulation and MSK-type signals. IEEE Trans. Vehicular Technology VT– 41 (1992), 401–406.

[5] Defly. K.; Lecours, M.; Boutin, N.: Differential detection of the OQPSK signal: Coding and decoding. Proc. IEEE Int. Conf. Commun., Boston, MA, USA, 1989. 1660–1664.

[6] Guenther, C. G.; Habermann, J.: DOQPSK-differential demodulation of filtered offset QPSK. Proc. IEEE Vehicular Technology Conf., Stockholm, Sweden, 1994. 1542–1546.

[7] Hischke, S.; Habermann, J.; Comley, R.: Receivers for differentially encoded OQPSK — A comparison of differential demodulation with joint data and channel estimation. Proc. IEEE Int. Conf. Telecommun., Melbourne, Australia, 1997. 1055–1060.

[8] Guenther, C.; Habermann, J.: Process for transmitting digital signals which combines advantages of OQPSK and $\pi/4$-QPSK. United States Patent 5,615,230, Mar. 1997.

[9] Huber, J.: Trelliscodierung. Heidelberg, Germany: Springer-Verlag, 1992.

[10] Aulin, T.: A new trellis decoding algorithm. Tech. Rept. 2, Goeteborg, Sweden: Chalmers University of Technology, 1985.

[11] Jakes, W. C.: Microwave mobile communications. New York: Wiley. 1974.

Sven Hischke was born in Bad Homburg, Germany in 1971. He received the Dipl.-Ing.(FH) degree from the Fachhochschule Giessen-Friedberg, Germany in 1996. Since 1996, he is a Ph.D. student at the Department of Electrical, Electronic and Information Engineering at City University London. He is currently involved in a research project, supported by the German Research Council (DFG). His research interests are the development and investigation of new receiver algorithms for digital mobile radio systems. He is a member of the Association of German Electrical Engineers (VDE) and a student member of IEEE. In 1996 he received an VDE award (first prize) for his diploma thesis.

Joachim Habermann was born in Wetzlar, Germany in 1953. He received the Dipl.-Ing. and Dr.-Ing. from Technical University Darmstadt in 1979 and 1984, respectively. In 1984 he joined the ABB research centre in Baden, Switzerland, where he worked in the Telecommunications department in the area of mobile communications. He was engaged in physical layer aspects for future mobile communications systems. Additionally he was a member of the international working group, defining the physical layer aspects of the European GSM system. Since 1990 he is Professor at the Fachhochschule Giessen-Friedberg, where he is head of the Telecommunications Laboratory. During a sabbatical leave from 1992 to 1993 he performed research on spread spectrum communications at Ascom Tech in Switzerland for the Universal Mobile Telecommunications System UMTS. He has served as a reviewer for various international conferences and for IEEE Trans. on Communications for several years. Since 1989 he is a member of IEEE.

Application of Offset-QPSK for Digital Mobile Radio

Prof. Dr.-Ing. Bernhard Rembold · Institut für Hochfrequenztechnik RWTH Aachen
Melatener Str. 25 · D-52072 Aachen

Herrn
Prof. Dr.-Ing. J. Habermann
Fachhochschule Giessen-Friedberg
Fachbereich Elektrotechnik II
Wilhelm Leuschner Str. 13

61169 Friedberg

Prof. Dr.-Ing. Bernhard Rembold
Chefredakteur

Institut für Hochfrequenztechnik
RWTH Aachen
Institutsgebäude
Melatener Str. 25
52072 Aachen
Telefon: 0241/80 79 31-32
Telefax: 0241/88 88 213
E-Mail: rembold@ihf.rwth-aachen.de

Unser Zeichen
Datum
Betrifft   5.7.1999
Veröffentlichung in der FREQUENZ - Application of Offset-QPSK for Digital Mobile Radio

Sehr geehrter Herr Habermann!

Haben Sie herzlichen Dank für die Zusendung o. g. Beitrages zur Veröffentlichung in der FREQUENZ, dessen Eingang (5.7.1999) per Post und per e-mail ich Ihnen hiermit bestätigen möchte.

Es ist geplant, die Arbeit in Heft 1-2/2000 (Widmungsheft für Professor Zschunke) zu drucken.

Mit freundlichen Grüßen

(Prof. Dr.-Ing. B. Rembold)

Application of Offset-QPSK for Digital Mobile Radio

Einsatz des Modulationsverfahrens Offset-QPSK im digitalen Mobilfunk

By Sven Hischke and Joachim Habermann

July 1, 1999

Abstract

The bandwidth efficient modulation format Offset-QPSK (OQPSK) is implemented up to now merely in mobile radio systems which do employ spread spectrum techniques, i.e., it is not used in narrow band systems. This is due to the fact, that, in spite of its reduced performance degradation in nonlinear channels in comparison to other linear modulation formats, for OQPSK no receivers were derived up to now which satisfy the following requirements: fast tracking in time selective channels, short acquisition time in burst mode transmission, and resolution of intersymbol interference in frequency selective channels. This paper presents new receiver structures for differentially encoded Offset-QPSK (DOQPSK) signals both for purely time selective and time and frequency selective mobile radio channels. The receiver for purely time selective channels is based on differential demodulation succeeded by a trellis decoder. The general receiver approach for channels, which are also frequency selective, applies per-survivor processing in a trellis decoder. The bit error rates of the receivers are given for linear additive white Gaussian noise, Rayleigh fading, and frequency selective channels. The effects of nonlinear distortions from power amplifiers and the resulting adjacent channel interference are investigated by introducing a generic nonlinear amplifier model.

Übersicht

Ein attraktives Modulationsverfahren für den digitalen Mobilfunk ist die Offset-QPSK (OQPSK). Zur Zeit wird die OQPSK nur in Spreizbandsystemen eingesetzt, da bisher noch keine Schmalbandempfänger für OQPSK Signale abgeleitet wurden, die folgende Bedingungen erfüllen: schnelle Auf- und Nachsynchronisation in zeitselektiven Kanälen und Auflösung der Intersymbolinterferenz als Folge von frequenzselektiven Kanälen. In diesem Artikel werden zwei neue Empfängerstrukturen für differentiell codierte OQPSK (DOQPSK) präsentiert. Der erste Empfänger basiert auf einer differentiellen Demodulation gefolgt von einem Trellis Decoder und ist nur für rein zeitselektive Kanäle geeignet. Der zweite Empfänger kann für zeit- und frequenzselektive Kanäle verwendet werden und führt die gemeinsame Kanalschätzung und Datendecodierung in einem Trellis Decoder durch. Für beide Empfänger werden die Fehlerraten im additiven Gaußkanal, im reinen Schwundkanal und in Mehrwegekanälen durch Computersimulationen ermittelt. Die Effekte durch nichtlineare Verstärker werden mit einem allgemeinen Verstärkermodell untersucht.

1

Für die Dokumentation

Mobile radio, Modulation, Equalisation, Differential Demodulation, Offset-QPSK

1 Introduction

Investigations of modulation formats and algorithms for channel estimation and equalisation are still essential for future mobile communication systems in order to define the air interface. Two important requirements should be fulfilled by a modulation format. Firstly, it should be bandwidth and spectrum efficient. Secondly, since the power sources are limited in mobile subscriber radio units, the generation of a modulated signal with small envelope fluctuations is recommended. An envelope with reduced fluctuations supports power efficient, nonlinear amplifiers. These two requirements can, however, not jointly and completely be fulfilled with one modulation format. In several mobile communication standards the modulation format $\pi/4$-QPSK was specified. The choice of $\pi/4$-QPSK was motivated by two observations: Firstly, the envelope of $\pi/4$-QPSK has no zeros. This reduces the generation of harmonics and the distortion of the signal by nonlinear amplifiers [1]. Secondly, a differential precoding of $\pi/4$-QPSK signals allows for a simple differential demodulation. Differential demodulation is an attractive combined carrier phase estimation and demodulation technique on fast time selective fading channels. Differential demodulation, however, does not resolve intersymbol interference introduced by a frequency selective multipath channel.

It will be shown in this article that for the attractive modulation format DOQPSK, which is even more power efficient than the latter format, receivers both for time and time and frequency selective mobile radio channels can be derived.

In section 2 the DOQPSK transmitter model is given. In Section 3 the principles of the receiver structures are discussed. The first algorithm which is based on differential demodulation has yet been given in [2], whereas the coherent trellis decoding algorithm for time and frequency slective channels has only partly been presented at the VTC'98 [3]. In Section 4 a generic nonlinear amplifier model is introduced in order to investigate the out-of-band power of DOQPSK. Section 5, finally, gives the bit error rate simulation results of both receiver algorithms on several linear and nonlinear channels.

2 Transmitter Model

Fig. 1 shows the block diagram of the DOQPSK transmitter. From the serial data stream two bits are mapped in parallel to the two branches of the quadrature modulator. The bits of each branch are differentially encoded and filtered with square root raised cosine Nyquist 1 filters. The delay of half a symbol period in one branch gives the desired offset of one component in the modulated signal $s(t)$. It should be pointed out that the DOQPSK transmitter in Fig. 1 differs from the usual OQPSK transmitter by merely including a differential encoder in each

2

channel.

To demonstrate the limited amplitude fluctuations, the inverse amplitude distributions of the two modulation formats DOQPSK and π/4-DQPSK at the transmitter side are shown in Fig. 2. Both modulation schemes use a square root raised cosine Nyquist filter with roll-off factor (r) of 0.35. While π/4-DQPSK has a minimum amplitude of $0.18D$, where D is the average amplitude, the amplitude of DOQPSK is never less than $0.46D$. Thus, in contrast to standard QPSK the envelope of both modulation formats has no zeros. The peak amplitude of DOQPSK is slightly larger than π/4-DQPSK, i.e., $1.51D$ instead of $1.48D$. The peak-to-average power ratios are $3.56dB$ and $3.41dB$ for DOQPSK and π/4-DQPSK, respectively. If r is increased, the fluctuations of DOQPSK are even more reduced than those of π/4-DQPSK.

3 Receiver Models

In a coherent demodulation structure for Offset-QPSK signals the I- and Q-components can in principle be decoded separately, if the constant time shift between the two component signals is taken into account and if the channel is not frequency selective. The problem, then, is to derive an efficient carrier phase synchronisation algorithm. Going beyond this estimation problem, in the following a coherent channel estimation algorithm for frequency selective mobile radio channels based an per-survovor processing (PSP) is given. On the other hand, if the channel is purely time selective and if a simple differential demodulation algorithm is employed, the time shift in the offset signal leads to a mutual crosscomponent interference of the I- and Q-components. Conventional differential demodulators, like those described in [4] can no more be applied. To resolve the mutual interference of the I- and Q-components after differential preprocessing, an algorithm is presented which uses an additional trellis decoder. At first the differential demodulation receiver will be presented. Both the differential and coherent receiver structure, however, are designed for raised cosine filtered signals with any roll-off factor.

3.1 Differential Demodulation with Subsequent Trellis Decoding

Fig. 3 shows the block diagram of the receiver for purely time selective mobile radio channels. The received signal $v(t)$ is down-converted with a synthesiser frequency f'_c slightly mistuned to the carrier frequency f_c and low pass filtered with square root raised cosine filters resulting in the signal $r(t)$. The I- and Q-components of the complex baseband signal $r(t)$ are sampled twice per symbol interval T_S, i.e., once per bit interval. The discrete signal r_l at epoch l for an additive white Gaussian noise (AWGN) channel becomes:

$$
\begin{aligned}
r_l &= I_l + \mathrm{j}\,Q_l \\
&= \left[z_l + \sum_{i=1}^{\infty} g_i \cdot (z_{l-i} + z_{l+i}) \right] \cdot \mathrm{e}^{\mathrm{j}\,\phi_l} + n'_l \\
g_i &= 0, \quad \text{for} \quad i = 2, 4, 6, \ldots
\end{aligned}
\tag{1}
$$

3

where g_i designates the overall impulse response of transmitter and receiver filters, which has zeros at multiplies of a symbol period due to the Nyquist 1 criterium. ϕ_l denotes the slowly time varying phase (slow in comparison to a symbol or bit interval), $z_l = x_l + j\,y_l$ gives the encoded, complex data at the transmitter (see Fig. 1), and n'_l is the filtered noise signal. As a consequence of the time offset in the transmitter and the bit-by-bit sampling in the receiver, z_l contains only the data of the I-channel for even sampling times and the data of the Q-channel for odd sampling times. We obtain:

$$z_l = \begin{cases} x_l & ; \quad l = 2k \\ j\,y_l & ; \quad l = 2k + 1 \end{cases} \tag{2}$$

Depending on the binary and differentially encoded data x_l and y_l at the transmitter side, the equations at even $(2k)$ and odd sampling times $(2k + 1)$ can be formulated as:

$$\begin{aligned} r_{2k} &= I_{2k} + j\,Q_{2k} \\ &= \left[x_{2k} + j \cdot \sum_{i=1}^{\infty} g_i \cdot (y_{2k-i} + y_{2k+i}) \right] \cdot e^{j\,\phi_{2k}} + n'_{2k} \end{aligned} \tag{3}$$

$$\begin{aligned} r_{2k+1} &= I_{2k+1} + j\,Q_{2k+1} \\ &= \left[\sum_{i=1}^{\infty} g_i \cdot (x_{2k+1-i} + x_{2k+1+i}) + j \cdot y_{2k+1} \right] \cdot e^{j\,\phi_{2k+1}} + n'_{2k+1} \end{aligned} \tag{4}$$

The differential preprocessing after the receiver filters carries out the mathematical operation $r_l r^*_{l-2}$. It eliminates the unknown phase ϕ_l and provides the input for the metric computation of the trellis decoding algorithm. The real component X_l and the imaginary component Y_l of the signal at the output of the differential preprocessing unit are:

$$X_l = \Re\left\{ r_l \cdot r^*_{l-2} \right\} = I_l \cdot I_{l-2} + Q_l \cdot Q_{l-2} \tag{5}$$

$$Y_l = \Im\left\{ r_l \cdot r^*_{l-2} \right\} = Q_l \cdot I_{l-2} - I_l \cdot Q_{l-2} \tag{6}$$

The trellis decoder in Fig. 3 compensates for the intersymbol interference introduced through differential demodulation.

The structure of the demodulator is the same for any type of Nyquist filtering. Depending on the chosen roll-off factor only the trellis decoder has to be changed. Let us examine the I-channel, for example. Due to the Nyquist 1 criterium, there is no interference from other I-channel bits. However, there may be a number of Q-channel bits which interfere with the I-channel bit considered. In the case of a roll-off factor of 1.0, only the preceding and succeeding Q-channel bit interferes on a given I-channel bit, since the signal also satisfies the Nyquist 2 criterium, resulting in a trellis decoder with 8 states. If a roll-off factor $r < 1.0$ is chosen, the second Nyquist criterium is not fulfilled any more and an infinite number of Q-channel bits interfere with a certain I-channel bit, since the impulse response of a Q-channel bit has nonzero values at the sampling times of all preceding and succeeding I-channel bits. Consequently an

4

infinite number of states in the trellis decoder would be needed to resolve the intersymbol interference from the Q-channel into the I-channel and vice versa. To overcome this problem, the impulse response considered in the trellis decoder to calculate the nominal values in the trellis table, is truncated after 3 bit intervals at the cost of a performance degradation. A roll-off factor of 0.35 and a truncation of the impulse response after 3 samples result in a trellis decoder with 128 states (see [2]). (7) and (8) indicate that in this case the input signals X_l and Y_l of the trellis decoder are functions of the transmitted bits y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1} and y_{l+3}:

$$X_l = \Re\left\{ r_l \cdot r_{l-2}^* \right\} = F\left\{ y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n'_l \right\} \tag{7}$$

$$Y_l = \Im\left\{ r_l \cdot r_{l-2}^* \right\} = F\left\{ y_{l-5}, y_{l-3}, x_{l-2}, y_{l-1}, x_l, y_{l+1}, y_{l+3}, n'_l \right\} \tag{8}$$

Data decoding in the trellis decoder is implemented using the Viterbi algorithm. To simplify computational complexity, the metric in the Viterbi algorithm is obtained from the squared Euclidean distance between the input vector and the nominal values of each state, i.e., it is assumed that the additive noise is white and Gaussian. However, the differential preprocessing together with the twofold sampling per symbol in principle invalidate this assumption, thus the Euclidean metric is not optimal. The accumulated metric $J(j_l)$ of state j at epoch l and the branch metric λ are calculated from the well-known equations:

$$J(j_l) = \min\left[J(n_{l-1}) + \lambda(nj_{l-1}) \quad , \quad J(n_{l-1}) + \lambda(mj_{l-1}) \right] \tag{9}$$

$$\lambda(uj_l) = \left(X_l - \overline{X}_{uj,l} \right)^2 + \left(Y_l - \overline{Y}_{uj,l} \right)^2 \quad ; u = n, m$$

$$(\overline{X}, \overline{Y} \text{ indicate the nominal values without noise})$$

where n and m represent the states in the trellis leading to state j and $\min[y_1, y_2]$ is the minimum function between y_1 and y_2.

A complexity reduction compared to trellis decoding with the Viterbi algorithm applying (9) can be achieved if not all paths of the trellis are decoded. The M-algorithm [5] stores merely the M most probable paths through the trellis. If the M-algorithm with 32 states is used, only a small degradation compared to the Viterbi algorithm with 128 states is observed. However, a gain in computer simulation time by a factor of 2 is achieved. Therefore, for all further investigations the Viterbi algorithm with 8 (for $r = 1.0$) and 128 (for $r = 0.35$) states and the M-algorithm with 32 states (for $r = 0.35$) will be used.

3.2 Coherent Demodulation and Channel Estimation with Per-Survivor Processing (PSP)

Data transmission over time and frequency selective mobile radio channels results both in a data sequence and channel estimation problem at the receiving side. It is sufficiently known

from the literature that the common maximum likelihood estimation of the data and of the channel impulse response $h(t)$ can not be carried out recursively. The classical, conventional, suboptimal MLSE-receiver initially performs a matched filtering that considers the channel impulse response and the impulse response of the transmission filter, followed by a sampling device with decorrelation and a Viterbi decoder. The unknown parameters of the different blocks in the receiver can, e.g., be estimated from the decoded, but partly erroneous and delayed data (data aided), which results in a suboptimal implementation. An alternative to the classical approach is per-survivor processing (PSP see [6]). PSP combines a group of algorithms which implement the data aided structure within the Viterbi algorithm itself. PSP uses the code sequence, that belongs to the survivor path to estimate the unknown parameters without delay.

The block diagram of the receiver, which combines trellis decoding with the Viterbi algorithm and discrete channel estimation within the trellis decoder is given in Fig. 4. For reasons of a simplified receiver implementation, the standard channel matched filter with subsequent decorrelation device, which have to be adapted to the time varying channel, are replaced by a sub-optimal Nyquist 1 root raised cosine roll-off low-pass filter, which precedes the trellis decoder. The twofold sampling per symbol considers the special feature of staggered (offset) QPSK.

For the subsequent investigations it is useful to define a vector $\mathbf{z}$ that contains the filtered complex data symbols z. A Nyquist filtered OQPSK signal with a roll-off factor of $r = 1.0$ can then be described as:

$$\mathbf{z}_l^{\mathrm{T}} = [\tilde{z}_l, \ldots, \tilde{z}_{l-N}] \tag{10}$$

$\mathbf{z}^{\mathrm{T}}$ is the transpose of $\mathbf{z}$, and:

$$\tilde{z}_l = [0.5 \cdot z_{l-1} + z_l + 0.5 \cdot z_{l+1}] \tag{11}$$

where $z_l = x_l + j y_l$ is the complex data symbol. The received and filtered signal r at time instant l is then determined as:

$$r_l = I_l + jQ_l = \mathbf{h}_l^{\mathrm{T}} \cdot \mathbf{z}_l + n_l' \tag{12}$$

with the filtered noise signal n_l'. The vector $\mathbf{h}$ represents the $L + 1$ taps of the impulse response of the time variant and frequency selective channel:

$$\mathbf{h}_l^{\mathrm{T}} = [h_l(0), \ldots, h_l(L)] \tag{13}$$

The estimates $\hat{\mathbf{h}}$ and the decoded data $\mathbf{z}$ are obtained with PSP directly from the survivor path to each state in the trellis decoder. This is shown in Fig. 5. Arrows characterise state transitions in the trellis decoder. The deletion (marking with x) of the lower path assumes that the upper path is the survivor path. The accumulated metrics J of state j at a time instant l is calculated with the quadratic norm. Selection of the minimum metrics gives:

$$J(j_l) \;\; = \;\; \min \left[J(j_{l-1}) + \left| r_l - \hat{\mathbf{h}}^{\mathrm{T}}(j_{l-1}) \cdot \mathbf{z}(j_{l-1} \to j_l) \right|^2 \right;$$

6

$$J(m_{l-1}) + \left| r_l - \hat{\mathbf{h}}^{\mathrm{T}}(j_{m-1}) \cdot \mathbf{z}(m_{l-1} \to j_l) \right|^2 \right] \tag{14}$$

The number of states in the trellis is 2^{L+2}, where L is the number of bits which interfere as a consequence of the frequency selective channel. If the intersymbol interference due to the channel extends over one symbol, i.e., two bits, then $2^{2+2} = 16$ states are required. Due to the overlap of the I- and Q-channel bits in the offset signal, the trellis decoder needs still four states when the channel is purely time selective ($L = 0$). This is because three bits interfere mutually, when the roll-off factor is $r = 1.0$. Then 2 bits have influence on the third bit, leading to $2^2 = 4$ states. For smaller roll-off factors the trellis, which is adapted to $r = 1$, can still be used as has been shwon in [7].

Based on the above assumptions two adaptive MLSE algorithms for discrete time channel estimation based on PSP for differentially encoded OQPSK will be considered. Firstly the LMS algorithm is given and secondly an improvement with a simplified Kalman algorithm is derived.

LMS Algorithm

At first the LMS algorithm is investigated. Estimation of the channel impulse response in the classical form [8] is given by:

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} - \Delta_{LMS} \cdot \frac{\delta |e_l|^2}{\delta \hat{\mathbf{h}}_{l-1}^{\mathrm{H}}} \tag{15}$$

with $e_l = r_l - \hat{\mathbf{h}}_{l-1}^{\mathrm{T}} \cdot \mathbf{z}_l$. This leads directly to the well known solution:

$$\hat{\mathbf{h}}_l = \hat{\mathbf{h}}_{l-1} + \Delta_{LMS} \cdot e_l \cdot \mathbf{z}_l^* \tag{16}$$

If PSP is implemented, the decoded data of the survivor path is used to obtain an estimate of the channel impulse response for that specific path within the trellis. The channel estimate which has been determined for every survivor path so far, is updated with the LMS algorithm:

$$\hat{\mathbf{h}}(j_l) = \hat{\mathbf{h}}(j_{l-1}) + \Delta_{LMS} \cdot e_l(j_{l-1} \to j_l) \cdot \mathbf{z}^*(j_{l-1} \to j_l) \tag{17}$$

Simplified Kalman Estimator

The major advantage of the described LMS algorithm is the computational simplicity. However, the price paid for the simplicity is slow convergence, especially when the autocorrelation matrix of the input signal of the estimator has a large spread of the eigenvalues. The slow convergence is due to the fact, that the gradient algorithms have only a single parameter for controlling the convergence rate, namely the step-size parameter Δ. In order to obtain faster convergence, it is necessary to apply algorithms involving additional parameters. Therefore, based on the iterative channel estimation with the LMS algorithm, an optimisation of the estimate with the aid of a simplified Kalman estimator is developed. The Kalman estimator requires an ARMA process model of the frequency selective fading channel. The derivation of the algorithm, which is not given in this paper, is based on a first order Markov process model of the channel and follows the steps in [7], [9], and [8].

7

Averaging over the data and introducing further simplifications on the Kalman gain factor and the error covariance matrix (see [7]) finally leads to a simplified Kalman estimator. Normalising the received average signal power $\sigma^2 = \sum_{i=0}^{L} \sigma_i^2 = 1$, where σ_i^2 is the average signal power of each channel tap (echo path), gives:

$$\Delta(i) = \frac{\beta \cdot \sigma_i^2 \cdot (1 - \beta^2)}{1.5 \cdot (1 - \beta^2) + \sigma_{N'}^2} \tag{18}$$

with the noise power $\sigma_{N'}^2 = 10^{-\frac{E_b/N_0}{10}}$. The variable $\beta = e^{-2\pi f_d T_S}$ depends on the maximum Doppler frequency f_d and the symbol duration T_S.

The main result of (18) is, that in contrast to the LMS algorithm every coefficient of the vector h, i.e., every echo, is adjusted with a seperate step size parameter. The step size depends mainly on the average power of the echo. The Doppler frequency, which is included in the parameter β, has only a minor influence due to the simplifications made to obtain (18). However, the algorithm is very simple and gives nearly optimal performance in contrast to the LMS algorithm, as will be shwon by the simulation results.

4   Nonlinear Amplifiers

4.1   Generic Amplifier Model

In nonlinear amplifiers typically two effects due to the nonlinear characteristic occur, AM-AM conversions and AM-PM conversions. With the generic amplifier model only the AM-AM conversion due to a nonlinear input output characteristic of the amplitude will be considered. The output phase versus input amplitude characteristic (AM-PM conversion) is not included in the model, since the purpose was to evaluate the dominant effect of the nonlinear amplitude factor alone. Two kinds of nonlinear amplitude distortions may appear at a power amplifier: saturation of the amplifier for high input voltage and blocking of the amplitude when the input signal is lower than the amplifier threshold (weak conduction region). Since the effects at the weak conduction region dominate the bit error rate (BER), they will only be considered. Fig. 6 gives the amplitude transfer characteristic of the generic power amplifier. The transfer characteristic is modelled by a third order spline function. The table below the figure shows the fixed points in the spline function. The weak conduction region is mainly influenced through the threshold parameter a. All amplitudes are normalised to the average amplitude D.

4.2   Fractional Out-of-Band Power

Fig. 7 shows the computed (fractional) out-of-band power for the two modulation formats DOQPSK and $\pi/4$-DQPSK with a roll-off factor of 0.35. The out-of-band power P_{out} is defined as:

$$P_{out} = 1 - \frac{\int\limits_{-B/2}^{B/2} W(f)\, df}{\int\limits_{-\infty}^{\infty} W(f)\, df} \tag{19}$$

where $W(f)$ is the simulated power spectral density of the signal around the carrier frequency. P_{out} is the fraction of the total transmitted power that is found at frequencies separated more than $B/2$ from the carrier frequency. The abscissa of Fig. 7 gives the bandwidth B normalised by the symbol rate $1/T_S$ as BT_S. DOQPSK shows less spectral spreading than $\pi/4$-DQPSK, since the former has less energy in the deadband (see also Fig. 2). However, it is interesting to note, that even with small values of the threshold parameter a as 0.05 or 0.15, there is still some far-out spectrum radiation for both schemes, which have minimum amplitudes larger than these values. This is due to the fact that the presence of the threshold amplitude a in the transfer curve acts as a constant offset to the output amplitude and introduces an additional noise-like signal with constant amplitude a and random phase, superimposed on the amplified distortion-free signal. Thus the out-of-band radiation is directly connected with a and occurs even for input signals with minimum amplitude above a.

5 Simulation Results

5.1 Linear Channels

Receiver with Differential Demodulation

Investigations of the receiver with differential demodulation are only meaningful for non-frequency selective channels, since the receiver can not cope with echos.

Figs. 8 compares the bit error rates of $\pi/4$-DQPSK and of the different DOQPSK receiver algorithms as a function of the ratio of symbol energy-to-noise power density Es/No on an AWGN channel. The curves in Fig. 8 show that the Viterbi algorithm for DOQPSK with 8 states for a roll-off factor of 1.0 and the algorithm with 128 states for $r = 0.35$ lead to a very small deviation from the curve of $\pi/4$-DQPSK for error rates less than 10^{-2}. The losses for smaller values of Es/No are a result of the implemented non-optimal Euclidean metrics. The Viterbi algorithm with 8 states, which is mismatched for $r = 0.35$, shows a noticeable performance degradation. The M-algorithm with 32 states, which is not included in the figure, achieves nearly the same performance as the Viterbi algorithm with 128 states.

In a narrowband mobile radio communication system, transmission channels subject to Rayleigh fading with Jakes Doppler spectrum are commonly considered. Results for such a type of channel are given in Fig. 9. The product of maximum Doppler frequency f_d and symbol duration T_S is chosen to be 0.0046 and 0.0093. At 450 MHz and a data rate of 36 kbit/s (18 ksym/s), which are the parameters of the TETRA system, the two values corresponds to a vehicle speed of 200 and 400 km/h. As expected, the good results of DOQPSK in the AWGN channel are also confirmed in the Rayleigh fading channel. For a roll-off factor of 1.0 and

9

$f_d * T_S = 0.0093$ the degradation compared to $\pi/4$-DQPSK is negligible in the range of interest. The M-algorithm with 32 states and $r = 0.35$ shows only a small degradation over the full range of Es/No.

Receiver with Coherent Demodulation

The coherent receiver can cope with both time and frequency selective channels. Due to fast motions of the mobile, the receiver should have a good acquisition and tracking performance.

Acquisition and Tracking Performance

Fig. 10 shows the squared error of the channel coefficients of the simplified Kalman algorithm on a two path Rayleigh fading channel. The signal-to-noise ratio in these simulations is fixed to 10 dB and the power of the echo path $h(1)$ is reduced by 10 dB compared to the direct path $h(0)$. Channel updating is performed with the step sizes Δ_0 of 0.13 and Δ_1 of 0.014. The mean squared error has been averaged over 30 independent simulations. Convergence of the direct path is observed within 50 symbols. The rate of convergence of the echo path is slower, but the steady state squared error is in accordance to the reduced power. BER investigations show that no further bit errors occur after ten bits in the noise free case.

Detailed investigations of the tracking performance in [7] have shown that PSP based on the LMS algorithm may lead to bit slip events. An error event may occur, when the echo path has a small power compared to the direct path, but the step sizes are identical. The simplified Kalman algorithm avoids bit slips.

Bit Error Rate

It is fairly known from the literature, that on an AWGN channel the performance difference between coherent demodulation with differential detection and differential demodulation is approximately 3 dB for DQPSK signals. This result is confirmed for DOQPSK. For a roll-off factor of 0.35 a gain of 2 dB can be achieved with the coherent receiver in the E_s/N_0 region of interest. In contrast to the AWGN channel, on fast and purely time selective channels the phase estimation in the differential demodulator, which is based on the phase difference between the current and the last symbol, leads to a better performance than it can be obtained with the slower adaptive channel estimator in the coherent receiver. The gain of the receiver with differential demodulation then is 2 dB, when the step size Δ of the coherent receiver is calculated from 18. A major drawback of the non-coherent receiver is, that the better tracking capability of differential demodulation can only be used in non-frequency selective channels.

In [7] detailed investigations on the BER performance in frequency selective channels can be found. As an example in Figure 11 the simulation results of the DOQPSK receiver on a two path Rayleigh fading channel are given. The delay of the echo path is fixed to one symbol period. The lower curve gives the theoretical bound of DQPSK with two fold diversity in a maximal ratio receiver. In case of equal signal powers and a small Doppler frequency the

10

gain of a diversity receiver can almost be achieved. The residual bit error rate is due to the nonzero receiver velocity in the simulations. An increase of the Doppler frequency leads to a clear degradation in the BER. If the power of the echo path is reduced compared to the direct path and the step-sizes are adapted to this new environment, the BER also increases. However, depending on the remaining echo power a gain compared to the theoretical bound without diversity can be achieved. This effect is illustrated by the curve for a relative echo power of -10 dB. Another interesting effect, which is not shown in the figures, has been found by further simulations. If the power of the echo path is equal to the direct path, but the step size of the echo path in the receiver is adapted to a smaller power value (e.g. -20 dB), the result is comparable to the theoretical bound without diversity. In [7] it is shown, that a misalignment of the equalisation algorithm, i.e., a wrong number of channel taps (echos) and wrong timing offsets, does not cause problems, if the echo amplitudes are set to very small values in an initial phase and then increased properly.

5.2 Nonlinear Channels

A major motivation for the implementation of linear modulation formats with reduced envelope fluctuations is the cost reduction for linearisation of nonlinear amplifier amplitude transfer curves. Fig. 12 compares the coherent receivers of the modulation schemes DOQPSK and $\pi/4$-DQPSK at a fixed bit error rate of 10^{-3} under relative adjacent channel power P_{AC}[1] and without noise for a threshold parameter $a = 0.3$. The two adjacent channels are of the same type as the original signal. Channel spacing is $f_s \cdot (1 + r)$, i.e., there is no overlap of the different spectra without the nonlinear amplifier. The use of DOQPSK results in a gain against $\pi/4$-DQPSK of more than 4 dB for a roll-off factor of 1.0 and about 1.5 dB for $r = 0.35$. This gain holds even for smaller threshold values a. From another point of view DOQPSK can operate at a higher threshold value a to obtain the same error rate as $\pi/4$-DQPSK. Thus, the effort to linearise the amplifier characteristic is reduced, when DOQPSK is implemented. The simulation results in Fig. 12 are obtained for adjacent channels which are synchronised to the measured channel. If this assumption is not fulfilled, e.g., if the adjacent channels are shifted by half a symbol period in the worst case, $\pi/4$-DQPSK shows a distinct degradation, especially for $r = 1.0$. The values of DOQPSK remain almost constant.

Similar results as for the coherent receiver on nonlinear channels can be obtained with the differential receiver (see [2]).

6 Conclusion

Offset QPSK is an attractive modulation format, since it is bandwidth efficient and shows only small amplitude fluctuations. It can thus be used for transmission over nonlinear power efficient

---

[1]$P_{AC} = 10$ dB means that the average power of the adjacent channel is 10 dB above the average power of the information bearing channel.

amplifiers. Two receiver structures for differentially encoded Offset-QPSK signals are compared. The first structure is based on differential demodulation, whereas the second structure applies joint coherent demodulation and channel estimation.

Firstly, the differential demodulation scheme is considered, which is only suitable for time selective mobile radio channels. Its essential elements are a differential preprocessing unit and a trellis decoder, whose complexity depends on the chosen roll-off factor of the pulse shaping filters. A complexity reduction of the receiver is achieved if the M-algorithm is implemented for decoding instead of the Viterbi algorithm. A comparison of the new receiver structure with a standard receiver for $\pi/4$-DQPSK, which is used in several mobile radio systems, has been carried out on different linear channels. The performance degradation of DOQPSK on linear channels is found to be very small, independent of the cut-off frequency of the pulse shaping filters.

Secondly, a receiver structure based on a trellis decoder with channel estimation through per-survivor processing is derived and investigated. The receiver can be used both for time and time and frequency selective channels. A fast acquisition and a good tracking performance of the algorithm in multipath Rayleigh fading channels is confirmed through computer simulations. Compared with the receiver based on differential demodulation a small degradation in purely time selective channels is observed.

The smaller envelope fluctuations of DOQPSK compared to $\pi/4$-DQPSK signals lead to a clear performance gain if nonlinear amplifiers are used. The linearisation cost for the amplifier characteristic in the weak conduction region can be reduced considerably by DOQPSK instead of $\pi/4$-DQPSK. The trade off between the reduced complexity of the transmitter amplifier and the increased complexity of the receiver baseband processing for DOQPSK may lead to a decision on a case by case basis.

Acknowledgement

References

[1] S. Ariyavisitakul and T.-P. Liu. Characterizing the effects of nonlinear amplifiers on linear modulation for digital portable radio communications. *IEEE Trans. Vehicular Technology*, 39(4):383–389, November 1990.

[2] S. Hischke and J. Habermann. A new differential demodulator for offset-qpsk. *Int. Journal Electron. Commun. (AE)*, 52(6):375–383, December 1998.

[3] S. Hischke and J. Habermann. Application of per-survivor processing for digital mobile radio. In *Proc. IEEE Vehicular Techn. Conf.*, pages 430–434, Ottawa, Canada, 1998.

12

[4] J.G. Proakis. *Digital Communications*. McGrawHill, New York, 3rd edition, 1995.

[5] J. Huber. *Trelliscodierung*. Springer-Verlag, Heidelberg, Germany, 1992.

[6] R. Raheli, A. Polydoros, and C.-K. Tzou. Per-survivor processing: A general approach to MLSE in uncertain environments. *IEEE Trans. Communications*, 43(2/3/4):354–364, 1995.

[7] S. Hischke. *Receivers for Differentially Encoded Offset-QPSK*. PhD thesis, City University, London, England, submitted July 1999.

[8] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, London, 2nd edition, 1991.

[9] W. Liu and H. Romanens. Adaptive channel equalization for high speed trains. In *Proc. IEEE Vehicular Tech. Conf.*, pages 225–229, Stockholm, Sweden, 1994.

Dipl.-Ing.(FH) Sven Hischke
Department of Electrical,
Electronic & Information Engineering
City University
London, England
sven.hischke@e2.fh-friedberg.de

Prof. Dr.-Ing. Joachim Habermann
Department of Electrical Engineering II
Fachhochschule Giessen-Friedberg
Wilhelm-Leuschner-Str. 13
61169 Friedberg, Germany
joachim.habermann@e2.fh-friedberg.de

13
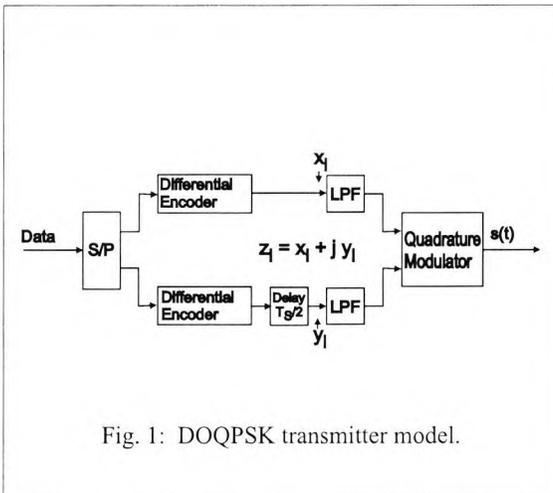
Fig. 1: DOQPSK transmitter model.
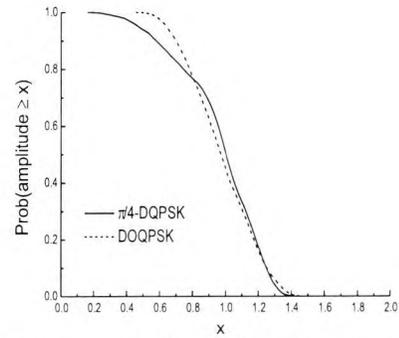
Fig. 2: Inverse amplitude distribution ($r = 0.35$). Amplitudes are normalised with D.
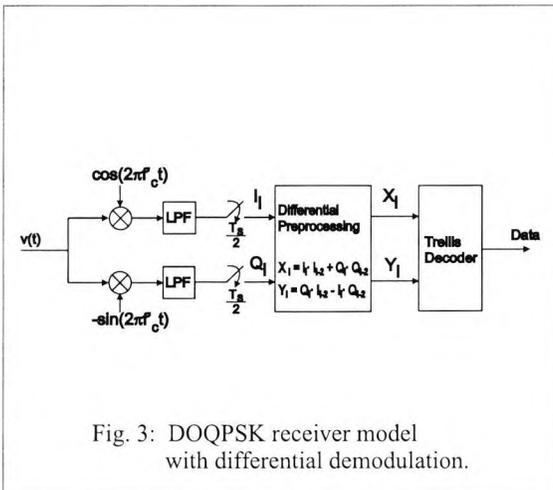
Fig. 3: DOQPSK receiver model with differential demodulation.
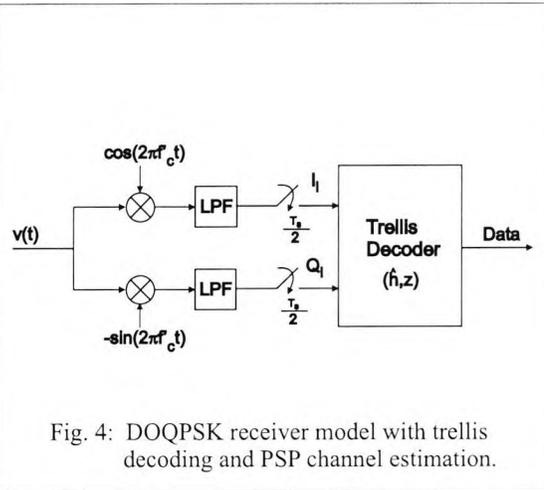
Fig. 4: DOQPSK receiver model with trellis decoding and PSP channel estimation.
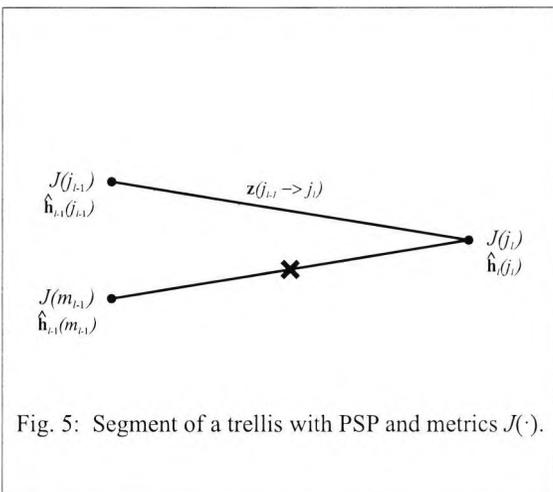
Fig. 5: Segment of a trellis with PSP and metrics $J(\cdot)$.
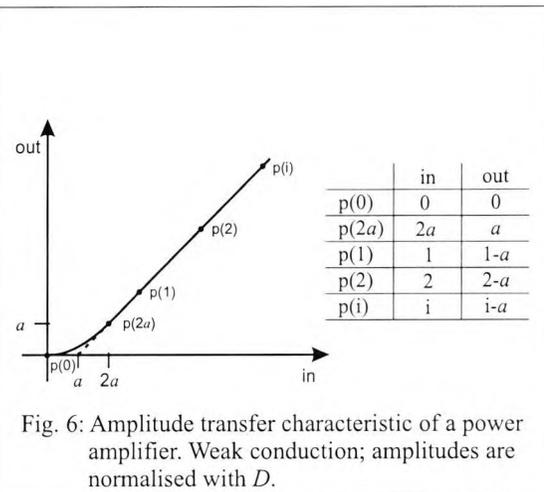
Fig. 6: Amplitude transfer characteristic of a power amplifier. Weak conduction; amplitudes are normalised with D.
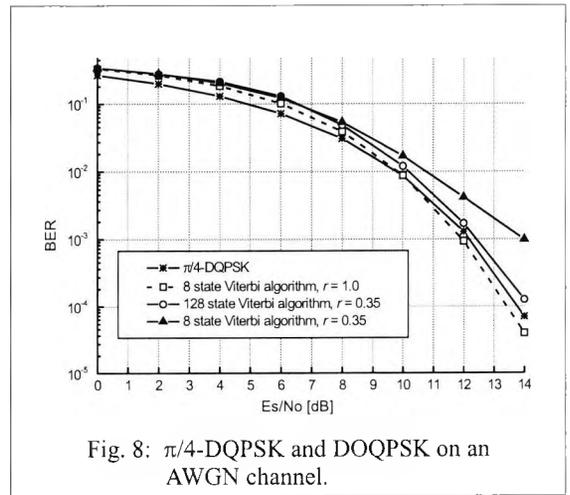
230

Fig. 7: Out-of-band power ($r = 0.35$).

Fig. 8: π/4-DQPSK and DOQPSK on an AWGN channel.
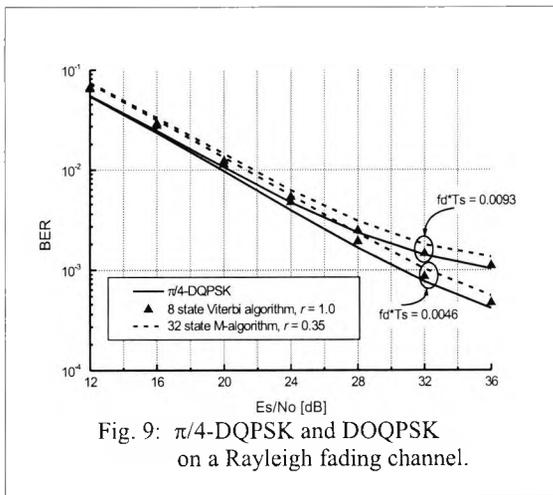
Fig. 9: π/4-DQPSK and DOQPSK on a Rayleigh fading channel.
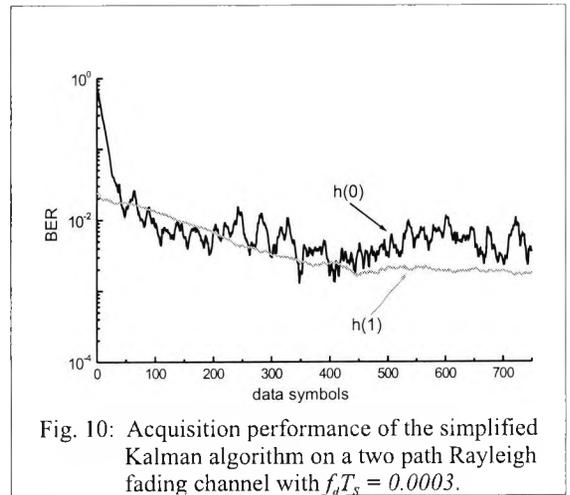
Fig. 10: Acquisition performance of the simplified Kalman algorithm on a two path Rayleigh fading channel with $f_d T_s = 0.0003$.
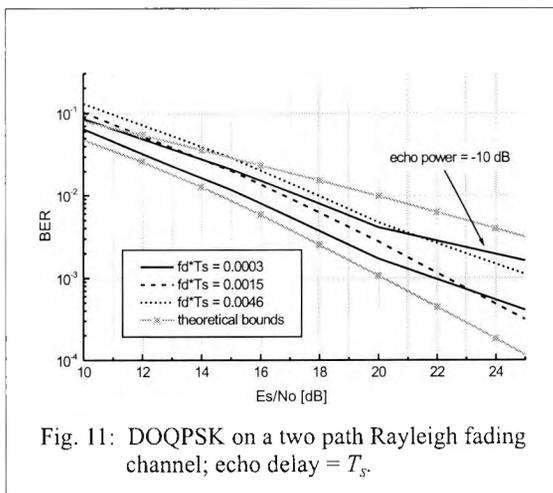
Fig. 11: DOQPSK on a two path Rayleigh fading channel; echo delay = T_s.
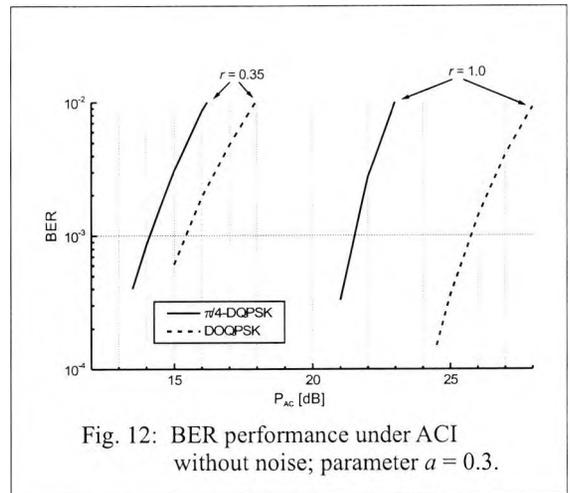
Fig. 12: BER performance under ACI without noise; parameter $a = 0.3$.

Bibliography

[1] Y. Akaiwa and Y. Nagata. Highly Efficient Digital Mobile Communications with a Linear Modulation Method. *IEEE Journal SAC*, SAC-5(5):890–895, 1987.

[2] S. Ariyavisitakul and T.-P. Liu. Characterizing the Effects of Nonlinear Amplifiers on Linear Modulation for Digital Portable Radio Communications. *IEEE Trans. Vehicular Technology*, 39(4):383–389, November 1990.

[3] T. Aulin. A New Trellis Decoding Algorithm. Technical Report 2, Chalmers University of Technology, Goeteborg, Sweden, 1985.

[4] T. Aulin. Simultaneous Data Detection and Channel Estimation with the Viterbi Algorithm. Technical Report 157, Chalmers University of Technology, Goeteborg, Sweden, 1993.

[5] K. David and T. Benkner. *Digitale Mobilfunksysteme*. Informationstechnik. B.G. Teubner, Stuttgart, 1996.

[6] K. Defly, M. Lecours, and N. Boutin. Differential detection of the OQPSK Signal: Coding and Decoding. In *Proc. IEEE Int. Conf. on Communications*, pages 1660–1664, Boston, MA, USA, 1989.

[7] U. Dersch, C. Guenther, and C. Schlegel. Spectral Efficiency of Digital Communications Systems, Volume A. Final report, Ascom Tech Ltd, Baden, Switzerland.

[8] U. Dersch, C. Guenther, and C. Schlegel. Spectral Efficiency of Digital Communications Systems, Volume B. Final report, Ascom Tech Ltd, Baden, Switzerland.

[9] M.C. Jeruchim et al. *Simulation of Communication Systems*. Plenum Press, New York, 1992.

[10] K. Feher. MODEMS for Emerging Digital Cellular-Mobile Radio Systems. *IEEE Trans. Vehicular Technology*, 40(2):355–365, May 1991.

[11] G.D. Forney. Maximum-Lkelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference. *IEEE Trans. Information Theory*, IT-18(3):363–378, 1972.

[12] G.D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1978.

[13] S.G. Glisic and P.A. Leppaenen. *Wireless Communications, TDMA versus CDMA*. Kluwer Academic Publishers, Netherland, 1997.

[14] C.G. Guenther and J. Habermann. DOQPSK - Differential Demodulation of Filtered Offset QPSK. In *Proc. IEEE Vehicular Tech. Conf.*, pages 1542–1546, Stockholm, Sweden, 1994.

[15] J. Habermann and P. Rauber. The Land Mobile Radio Channel in Northern Switzerland at 146 Mhz: Measurements, Evaluation, and Interpretations. Research Report CRB 88-03 C, ABB Corporate Research, Baden, Switzerland, 1988.

[16] E. Haensler. *Grundlagen der Theorie statistischer Signale*. Nachrichtentechnik 10. Springer Verlag, Berlin, Heidelberg, 1983.

[17] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, London, 2nd edition, 1991.

[18] J. Huber. *Trelliscodierung*. Springer-Verlag, Heidelberg, Germany, 1992.

[19] Editor in Chief R.C. Dorf. *The Electrical Engineering Handbook*. CRC Press, Boca Raton, Florida, 1993.

[20] G.K. Kaleh. Differential Detection via the Viterbi Algorithm for Offset Modulation and MSK-Type Signals. *IEEE Trans. Vehicular Technology*, 41(4):401–406, November 1992.

[21] K.D. Kammeyer. *Nachrichtenuebertragung*. Informationstechnik. B.G. Teubner, Stuttgart, 2nd edition, 1996.

[22] W. Liu and H. Romanens. Adaptive Channel Equalization for High Speed Trains. In *Proc. IEEE Vehicular Tech. Conf.*, pages 225–229, Stockholm, Sweden, 1994.

[23] H. Meyr, M. Moeneclaey, and St.A. Fechtel. *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*. Wiley Series in Telecommunications and Signal Processing. Wiley, New York, 1998.

[24] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing*. Prentice Hall, London, 1975.

[25] J.-E. Porath and T. Aulin. Simultaneous Data Detection and Channel Estimation with Viterbi Algorithm. In *Proc. IEEE Vehicular Tech. Conf,.*, pages 1254–1258, Stockholm, Sweden, 1994.

[26] J.G. Proakis. *Digital Communications*. McGrawHill, New York, 3rd edition, 1995.

[27] R. Raheli, A. Polydoros, and C.-K. Tzou. Per-Survivor Processing: A General Approach to MLSE in Uncertain Environments. *IEEE Trans. Communications*, 43(2/3/4):354–364, 1995.

[28] T.S. Rappaport. *Wireless Communications: Principle & Practice*. Prentice Hall, New Jersey, 1996.

[29] N. Seshadri. Joint Data and Channel Estimation Using Blind Trellis Search Techniques. *IEEE Trans. Communications*, 42(2/3/4):1000–1011, 1994.

[30] N. Seshadri and C.-E.W. Sundberg. List Viterbi Decoding Algorithms with Applications. *IEEE Trans. Communications*, 42(2/3/4):313–323, 1994.

[31] B. Sklar. Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization. *IEEE Comm. Magazine*, 35(7):90–100, July 1997.

[32] B. Sklar. Rayleigh Fading Channels in Mobile Digital Communication Systems Part II: Mitigation. *IEEE Comm. Magazine*, 35(7):102–109, July 1997.

[33] G. Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, Orlando, Florida, 2nd edition, 1980.

[34] T.S.Rappaport. *Cellular radio & Personal Communications, Study Guide*. IEEE Press, New Jersey, 1996.