Accurate Modelling and Realisation of a 4th Generation Wireless Communication System

by

Shaun Schulze

Thesis presented at the University of Stellenbosch in partial fulfilment of the requirements for the degree of

Master of Science in Engineering (Electronic Engineering)

Department of Electrical and Electronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Study leaders:

Prof Keith D Palmer	**Christiane Kuhnert**
Department of Electrical	Institut für Höchstfrequenztechnik
and Electronic Engineering	und Elektronik
University of Stellenbosch	Universität Karlsruhe

December 2006

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: ....

S Schulze

Date: ....

Abstract

Accurate Modelling and Realisation of a 4th Generation Wireless Communication System

S Schulze

Department of Electrical and Electronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Thesis: MScEng (Electronic Engineering)

December 2006

A great demand exits for higher data rates and increased reliability of future consumer based mobile communication systems while being more bandwidth and power efficient. Orthogonal frequency division multiplexing (OFDM) in combination with multiple-input multiple-output (MIMO) schemes has become a promising candidate for fulfilling the demand of next generation communication systems.

The sensitivity of MIMO OFDM systems to physical impairments is of great interest and particularly the Alamouti space-time block code is under investigation in this thesis. Generic and comprehensive simulation models of an OFDM communication system incorporating the space-time block code are developed in a modular fashion and used in a performance evaluation with non-ideal component and channel behaviour.

Uittreksel

Akkurate Modellering en Realisasie van 'n 4de Generasie Draadlose Kommunikasie Stelsel

S Schulze

Departement Elektriese en Elektroniese Ingenieurswese
Universiteit van Stellenbosch
Privaatsak X1, 7602 Matieland, Suid Afrika

Tesis: MScIng (Elektroniese Ingenieurswese)

Desember 2006

Daar bestaan 'n groot aanvraag na vinnige datatempo's en beter betroubaarheid in toekomstige gebruikergerigte mobile kommunikasiestelsels wat terwylselfdetyd beskikbare bandwydte en drywing effektief moet benut. Ortogonale frekwensie deel multipleksering (OFDM) wat as Multi-Intree Multi-Uittree (MIMU) stelsels geïmplementeer word is tans 'n belowende oplossing vir volgende generasie kommunikasiestelsels.

Die sensitiwiteit van MIMU OFDM stelsels vir nie-ideale toestande is van groot belang en veral die Alamouti ruimte-tyd blok kode word ondersoek in hierdie tesis. Generiese simulasiemodelle van 'n OFDM kommunikasiestelsel wat hierdie ruimte-tyd blok kode gebruik word modulêr ontwikkel en dan geevalueer vir nie-ideale komponente en kanaalgedrag.

Acknowledgements

I would like to express my sincere gratitude to the following people:

Professor Keith Palmer for his support and supervision.
Professor Werner Wiesbeck for accommodating me in Karlsruhe at the IHE.
Christiane Kuhnert for her helpful insight and guidance.
André Young for proof reading this thesis.

Contents

List of Figures

List of Tables

Nomenclature

α_n	Angle of arrival of the n'th planewave
$\bar{V}$	Mobile receiver's velocity
β	$\frac{2\pi}{\lambda}$
Δf_D	Maximum Doppler shift
λ	Wavelength of the carrier frequency
$\mathcal{CN}(\mu, \sigma^2)$	Complex circular symmetric Gaussian process with mean μ and variance σ^2
B_d	Doppler spread of a radio channel
c_k	Complex modulating symbol for subcarrier k in the OFDM symbol
E_b/N_0	Bit energy to noise density
E_b	Energy per Bit
E_b/N_0	Bit Energy to Noise Spectral Density
E_D	distortion energy per bit present in the FFT period of one OFDM symbol
f_c	RF signal centre frequency
N_0	Noise power spectral density
N_{FFT}	FFT vector length
N_C	Number of subcarriers in one OFDM symbol
N_D	Number of data carriers
N_P	Number of pilots
n_r	Number of receive antennas
n_t	Number of transmit antennas
N_Z	Number of zeros used for padding the IFFT vector
T_{step}	Simulation time step
T_d	delay spread of a radio channel
T_g	Guard time of an OFDM symbol
T_s	OFDM symbol duration (with no guard time present)

T_t	Total OFDM symbol time (including the guard)
3GPP	3rd Generation Partnership Project
ADC	Analogue to Digital Converter
ADS	Agilent's Advanced Design System
AM-to-PM	Amplitude Modulation to Phase Modulation
ASK	Amplitude Shift Keying
BER	Bit Error Rate
bps	Bits per Second
BPSK	Binary Phase Shift Keying
CCDF	Complementary Cumulative Distribution Functions.
D-BLAST	Diagonal Bell Laboratories Layered Space-Time
DAC	Digital to Analogue Converter
DFT	Discreet Fourier Transform
DSP	Digital Signal Processing
DUT	Device Under Test
DVB	Digital Video Broadcasting
FDM	Frequency Division Multiplexing
GaAS	Gallium Arsenide
ICI	Inter-carrier Interference
IDFT	Inverse Discreet Fourier Transform
IF	Intermediate Frequency
IFFT	Inverse Fast Fourier Transform
ISI	Inter-Symbol Interference
LAN	Local Area Network
LO	Local Oscillator
MIMO	Multiple Input Multiple Output
MISO	Multiple-Input Single-Output
MMIC	Monolithic Microwave Integrated Circuit
MRC	Maximum Ratio Combiner
OFDM	Orthogonal Frequency Division Multiplexing

PCK	Perfect Channel Knowledge
PDF	Probability Density Function
PHEMT	Pseudomorphic High Electron Mobility Transistor
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SIMO	Single-Input Multiple-Output
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Power Ratio
STBC	Space Time Block Code
TOI	Third Order Intercept
TSDF	Timed Synchronous Data Flow
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
V-BLAST	Vertical Bell Laboratories Layered Space-Time

Chapter 1

Introduction

1.1 Motivation and Objectives

Consumer based mobile equipment has fuelled the development of reliable wireless communication systems over the past decade. An ever increasing demand exists for higher data rates and increased reliability while being more bandwidth and power efficient in order to provide more services to a growing user-base. Orthogonal frequency division multiplexing (OFDM) in combination with multiple-input multiple-output (MIMO) schemes has become a promising candidate for fulfilling the demand of next generation communication systems.

OFDM has gained increased popularity and has been adopted by several standards such as IEEE 802.11a wireless LAN [3], digital audio broadcasting (DAB) [4] and digital video broadcasting (DVB) [1]. The 3rd Generation Partnership Project (3GPP) has also proposed the use of OFDM for high speed asymmetric data transfer as an enhancement to the current UMTS Terrestrial Radio Access Network (UTRAN) and is currently under investigation [5].

OFDM in combination with MIMO schemes has gained much popularity in mainstream fourth generation mobile communication research. It is under investigation by several groups currently developing communication protocols including the Wireless Gigabit with Advanced Multimedia Support (WIGWAM) project [6], Wireless Metropolitan Area Networks IEEE 802.16 (WiMAX) working group [2] and the IEEE 802.11n wireless LAN task groups [7].

The sensitivity of MIMO OFDM systems to physical impairments is of great interest and under investigation in this thesis. Generic and comprehensive simulation models are developed in a modular fashion to help a system designer predict the effect of component behaviour on the performance of a MIMO OFDM communication system in terms of bit error rates.

1.2 Thesis Overview

The theory of orthogonal frequency division multiplexing is presented in Chapter 2 as well as the implementation of an OFDM transmitter and receiver. Chapter 3 introduces the concept of multiple-input multiple-output communication and particularly the Alamouti space-time block code. The integration of OFDM and the space-time block code is also discussed. General simulation and modelling techniques and the theory of a synchronous data flow simulation environment are discussed in Chapter 4 while the modelling of physical impairments and the effects on system performance are evaluated by means of bit error rate simulations in Chapter 5. Conclusions along with recommendations for future research are presented in Chapter 6.

Chapter 2

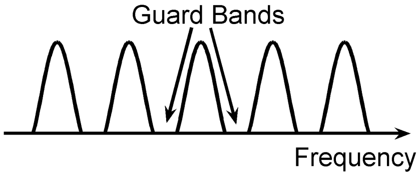
Orthogonal Frequency Division Multiplexing

Modern wireless communication systems that are robust and support high data rates while being spectrally efficient, easy to implement and computationally inexpensive are highly desirable for office mobility and mobile multimedia applications.

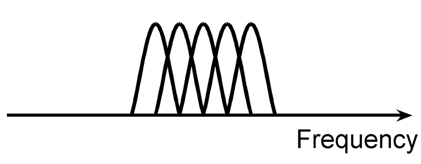
Multiple input multiple output (MIMO) communication is a promising field of research for achieving high mobile data rates in multipath communication channels. A suitable modulation technique is needed in order to implement conventional MIMO theory which is developed under the assumption of a narrow-band signal. Orthogonal frequency division multiplexing (OFDM) is a wideband modulation technique that possesses suitable narrowband properties and is investigated because of its great potential as a modulation scheme in combination with MIMO technology.

In classical digital modulation, where frequency division multiplexing (FDM) is used, the available frequency band is divided into channels. Independent data streams are transmitted over each channel with a guard frequency between channels in order to prevent spectral overlapping and eliminate inter-channel interference as illustrated in Figure 2.1(a).

Orthogonal frequency division multiplexing (OFDM) enables more efficient spectral usage, compared to conventional FDM, by allowing the subchannel spectra to overlap as illustrated in Figure 2.1(b). The spectra are however overlapped, as seen in Figure 2.3, in such a manner that



(a) Conventional frequency division multiplexing with guard bands.



(b) Orthogonal frequency division multiplexing with overlapping spectra.

Figure 2.1: Frequency domain representation of conventional FDM and OFDM.

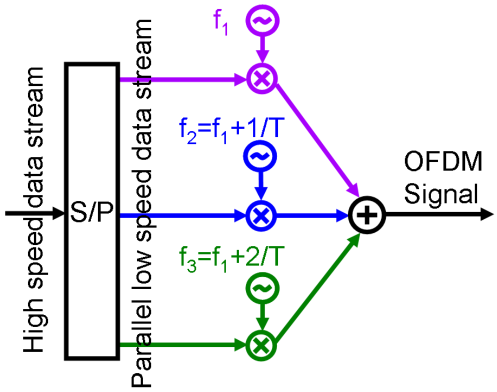


Figure 2.2: Conceptual OFDM signal generation.

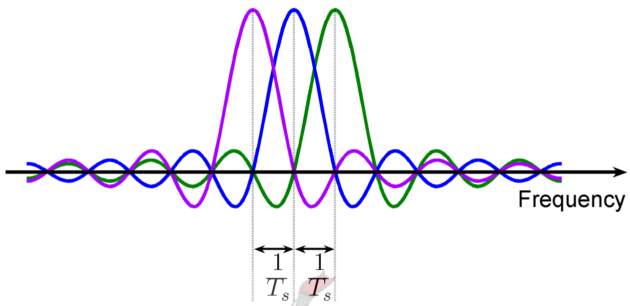


Figure 2.3: OFDM subcarrier's spectrum envelope.

channel orthogonality is preserved.

With OFDM a high speed bit stream is multiplexed into several parallel low bit rate streams. Each low bit rate stream independently modulates a carrier according to some desired digital modulation scheme such as quadrature amplitude modulation (QAM).

2.1 The Modulation of Orthogonal Subcarriers

An OFDM signal consists of a set of subcarriers each independently modulated by a digital data stream. A high data rate digital signal is multiplexed into several parallel low data rate streams which each modulate a separate subcarrier as depicted in Figure 2.2. The subcarriers are spaced in such a manner that their spectra overlap but nevertheless remain orthogonal. The operation is nothing more than an inverse Fourier transform and in Section 2.3.2 the use of a discrete Fourier transform for this purpose is discussed. The resulting signal envelope of each subcarrier viewed in the frequency domain is presented in Figure 2.3 with the peak of a subcarrier corresponding to a null of all other subcarriers.

Figure 2.4 illustrates a baseband equivalent of the OFDM subcarriers in the time domain, each modulated by an arbitrary digital modulation scheme. It is easily noted that the subcarriers are orthogonal as the phase transitions occur simultaneously and every subcarrier contains an integer number of cycles in the period T_s.

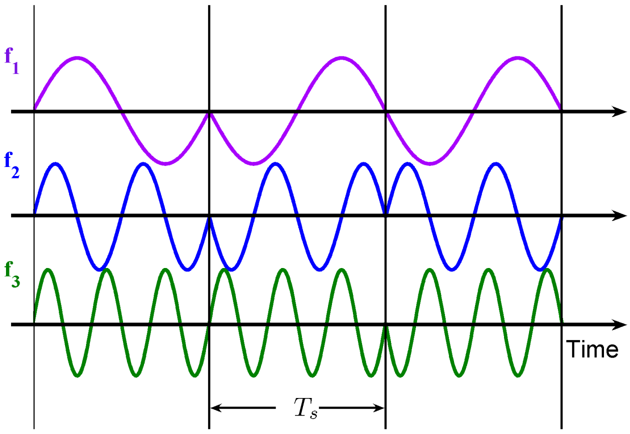


Figure 2.4: Subcarriers of an OFDM symbol in the time domain.

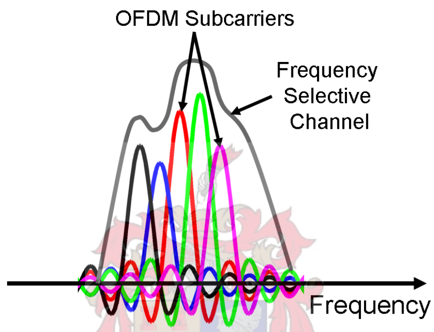


Figure 2.5: OFDM subcarrier spectral envelope and frequency selective channel.

A significant advantage of OFDM and one of the major reasons for its implementation is the effectiveness with which it combats the delay spread of a multipath channel. As explained later in Section 4.5.1.2, a multipath environment causes the transmitted signal to travel along different paths and arrive at multiple times, each with different phase and amplitude at the receiver. This is known as a delay spread and causes inter-symbol interference (ISI) as energy from the previous data symbol is spread into the next symbol. In the frequency domain it is equivalent to passing some frequencies while rejecting others. The channel is thus frequency selective as illustrated by Figure 2.5.

Modulating carriers with parallel low data rate streams has the advantage of increasing the immunity to the delay spread of a multipath channel. This is intuitively illustrated in both the time and frequency domain.

In the time domain, the period of the parallel low speed data streams is longer than that of the single high speed stream of Figure 2.2 by a factor of the amount of data carrying subcarriers. [1] Thus the relative delay spread is much shorter compared to the OFDM symbol period than that of

[1]In the simulations performed in this thesis the period of the parallel low speed substreams is 1705 times longer.

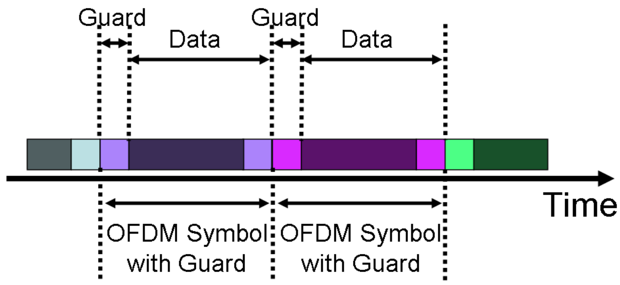


Figure 2.6: OFDM signal with cyclic extended guard interval.

a conventional single carrier system modulated by the high speed data.

From the perspective of the frequency domain it is argued that dividing the high speed data stream into parallel low speed streams effectively divides the signal bandwidth into orthogonal narrowband subcarriers as seen in Figure 2.5. This means that the frequency response of a frequency selective channel is quasi-linear or flat relative to each narrowband subcarrier. Channel correction, described in Section 2.4.3, is performed by simply estimating a phase and amplitude error for every subcarrier and does not involve complex equalization.

2.1.1 Guard Time

Single carrier systems with high data rates conventionally make use of equalization techniques which estimates the delay response of the channel and then corrects for it at the receiver [8, Section 7.8]. An OFDM signal has a long symbol time compared to a single carrier system due to the data parallelisation show in Figure 2.2. The delay spread is typically much shorter than the OFDM symbol time and may therefore be eliminated by inserting a guard interval before each OFDM symbol in order to protect it from the energy spilled over from the previous symbol.[2] This is done at the cost of lowering the data rate. The guard time is not simply a period in which no signal is sent as this would cause inter-carrier interference (ICI) as illustrated by [9]. Instead, a cyclic extension of the OFDM symbol is inserted in front of it (a copy of the last portion of each OFDM symbol is placed in front as seen in Figures 2.6 and 2.7). Using a cyclic extension has further significance as the OFDM symbol now contains repetition and is used for synchronization in Section 2.4.1.

Figure 2.8 shows the received subcarriers passed though an arbitrary two path delayed channel as and illustrates the effected of a delay spread channel. Note that a portion of the signal is distorted by the delay spread of the previous symbol. When the guard time is longer than the delay spread, part of the guard interval remains intact which is required for synchronization. The guard time is removed at the receiver and the signal can be demodulated without ISI.

[2] A guard time is not an option for high speed conventional single carrier systems as the symbol period is too short.

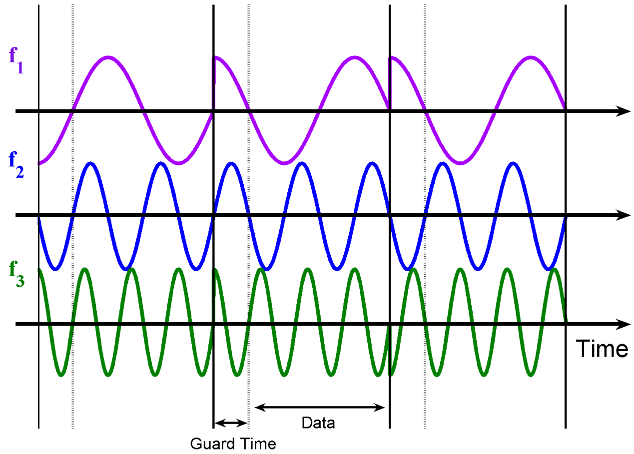


Figure 2.7: OFDM subcarriers with cyclic extended guard time.

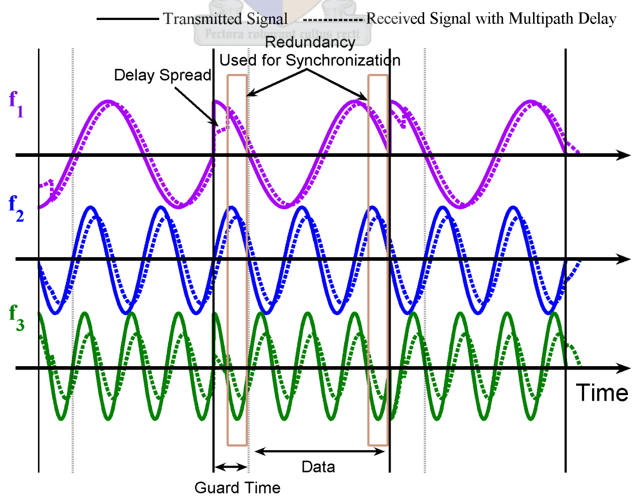


Figure 2.8: Received OFDM subcarriers after passing through a channel with delay spread.

2.2 Mathematical Description of an OFDM Symbol

The mathematical expression for a single OFDM symbol without a guard interval is given by (2.2.1).

$$s(t) = Re\left\{ e^{j2\pi f_c} \sum_{k=0}^{N_C-1} c_k e^{\frac{j2\pi(k-\frac{N_C-1}{2})}{T_s}(t-t_0)}; t_0 \leq t \leq t_0 + T_s \right\} \tag{2.2.1}$$

where:

$s(t)$	is the OFDM signal for one time period without the guard interval.
k	donates the subcarrier number within the OFDM symbol.
f_c	is the RF signal centre frequency.
N_C	is the number of subcarriers in one OFDM symbol.
c_k	is the complex modulating symbol for subcarrier k in the OFDM symbol.
T_s	is the OFDM symbol duration (with no guard time present).

c_k in (2.2.1) is any complex baseband symbol representing a bit or bit sequence that corresponds to an arbitrary digital modulation scheme (eg. ASK, PSK or QAM). Each complex symbol is used to modulate a corresponding subcarrier. An OFDM symbol of five subcarriers in Figure 2.9 serves to illustrate the manner in which parallel bit streams are multiplexed onto different subcarriers separated in frequency by $\frac{1}{T_s}$Hz.

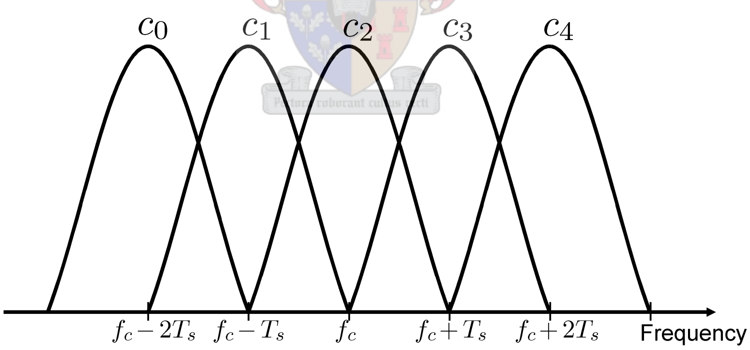


Figure 2.9: OFDM subchannel frequency positioning for corresponding parallel baseband symbol streams.

The inverse discreet Fourier transform (IDFT) may be employed to perform the multiplexing of each symbol. By rewriting (2.2.1) in the form

$$s(t) = Re\left\{ e^{j2\pi(f_c - \frac{N_C-1}{2})} \sum_{k=0}^{N_C-1} c_k e^{\frac{j2\pi k}{T}t}, \;\; 0 \leq t \leq T_s \right\}, \tag{2.2.2}$$

it is noted that the term $\sum_{k=0}^{N_C-1} c_k e^{\frac{j2\pi k}{T}t}$ from (2.2.2) corresponds to that of a continuous time Fourier series. Section 2.3.2 describes the use of a discrete Fourier transform for this purpose and how to arrange symbols in order to generate the signal in the digital domain.

2.3 Transmitter Implementation

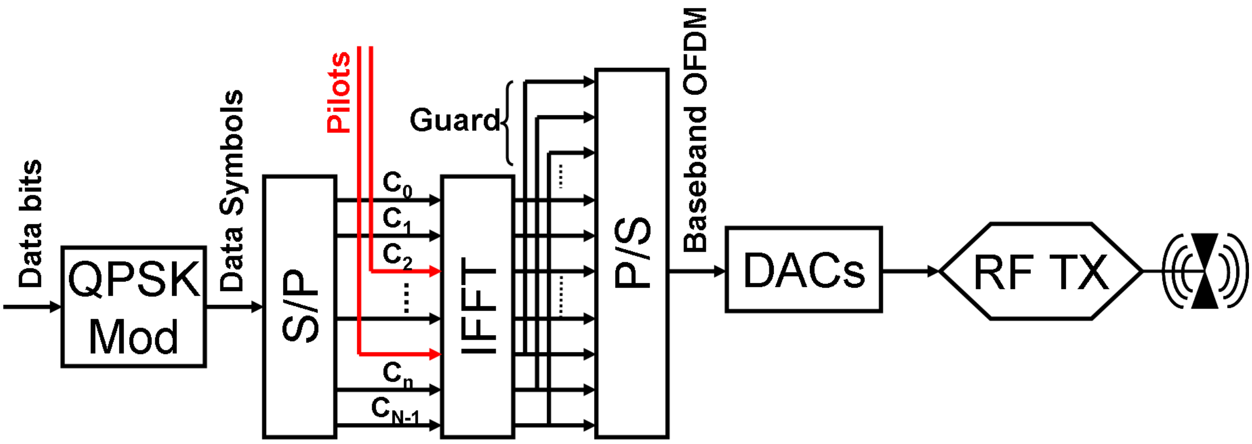


Figure 2.10: OFDM Transmitter Block Diagram.

This section describes the digital signal processing implementation of an OFDM transmitter in Figure 2.10 which is implemented in the C++ programming language for use in Agilent's ADS®Ptolemy simulator.

2.3.1 Quadrature Amplitude Modulation

The QPSK Mod block in Figure 2.10 transforms a group of data bits to a symbol in the form of a complex number. The amplitude and phase of this symbol represents that of the modulated carrier signal. Quadrature phase shift keying (QPSK) with Gray mapping is used in simulations throughout this thesis. Two bits are encoded into each data symbol according to the complex constellation diagram presented in Figure 2.11.

Throughout this thesis the terms *data symbol* and *OFDM symbol* are used.

Data symbol refers to the complex number in which several bits are encoded and represents the phase and amplitude of the subcarrier being modulated. For example a QPSK symbol as described above.

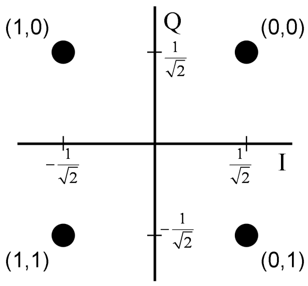


Figure 2.11: QPSK constellation diagram.

An *OFDM symbol* is the sum of all subcarriers in one IFFT period, after the IFFT operation is performed, and includes the guard time. Thus the period of an OFDM symbol equals the inverse of the subcarrier spacing plus the guard time.

2.3.2 The Discrete Fourier Transform

It was illustrated in Section 2.2 that a continuous time Fourier Series forms part of OFDM modulation. This section describes the use of the Discrete Fourier Transform in a realisable digital modulator.

The process by which parallel data streams modulate subcarriers $\frac{1}{T}$Hz apart, as depicted by Figure 2.2, can be performed in practical systems by an inverse fast Fourier transform (IFFT) followed by an up-conversion stage.

In Figure 2.10, a one dimensional IFFT is performed on a vector containing the modulating symbols in order to generate the I and Q baseband signals in the time domain.

Assuming there are N_C subcarriers, N_C symbols (one symbol for every subcarrier) symbols are arranged in a vector of length N_{FFT} before it is passed though the IFFT operation. The vector consists of N_D data symbols and N_P pilots while the rest of the vector is filled with N_Z zeros. Thus $N_C = N_D + N_P$ and $N_{\text{FFT}} = N_D + N_P + N_Z$.

In discreet terminology, the inter-carrier spacing is $1/N_{\text{FFT}}$ cycles per sample and the first null crossing bandwidth is $(N_D + N_P + 1)/N_{\text{FFT}}$ cycles per sample when ignoring the guard interval.

The IFFT vector has to be arranged as given in (2.3.1) so that the symbols are mapped to the corresponding subcarriers as illustrated in Figure 2.9. For simulation purposes 1705 subcarriers are generated by a 2048 point IFFT employing the radix-2 algorithm which is implemented by the build-in FFT component in ADS®.

Although the effects of digital to analogue converters are not considered in this thesis, a it is worth mentioning that the zeros in the OFDM generating IFFT vector provide excess unambiguous bandwidth which eases analogue filter requirements after the digital to analogue converter (DAC). Furthermore a zeroth-order hold DAC effectively multiplies the signal spectrum with a $\text{sinc}(\frac{f}{f_s})$ [10], where f_s is the sampling frequency. By zero padding the vector this effect is reduced.

$$[c_2 \;\; c_3 \;\; c_4 \;\; 0 \;\; \ldots \;\; 0 \;\; c_0 \;\; c_1] \tag{2.3.1}$$

$$y(m) = \begin{cases} x\left(\frac{N-1}{2} + m\right) & ; \quad m \in \left[0, \frac{N-1}{2}\right] \\ 0 & ; \quad m \in \left[\frac{N+1}{2}, M - \frac{N+1}{2}\right] \\ x\left(m - M + \frac{N-1}{2}\right) & ; \quad m \in \left[M + \frac{1-N}{2}, M - 1\right] \end{cases} \tag{2.3.2}$$

The *OFDMZeroPad* component (Appendix C.3) was developed for the purpose of arranging the subcarrier symbols and zero pad the IFFT vector accordingly. The complex signal produced by the IFFT is the sampled baseband in-phase and quadrature-phase signals.

2.3.3 Pilot Signals

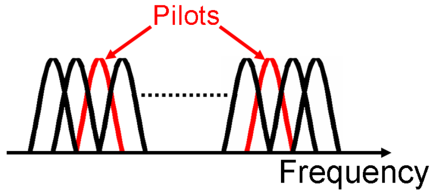


Figure 2.12: Illustration of pilot subcarriers scattered throughout the OFDM symbol.

A mobile communication channel is generally frequency selective, as described in Section 4.5.1, and pilot signals are used in order to estimate the channel response. The OFDM subcarriers are assumed to be very narrow band and the influence of the radio channel and hardware is approximated simply as an amplitude and phase error for every subcarrier.

Pilot signals are designated subcarriers that are scattered throughout the frequency band, as seen in Figure 2.12, and their transmitted amplitudes and phases are known to the receiver. The receiver calculates the amplitude and phase error occurring in each pilot and a linear extrapolation of the error is performed between pilots in order to estimate errors occurring in the data carrying subcarriers.

The same scattered pilot scheme used in terrestrial Digital Video Broadcasting (DVB-T) (2k mode) [1] was implemented and is used for channel estimation in all simulations. The red lines in Figure 2.10 illustrate that the pilot signals are inserted as known symbols before the IFFT is performed at the transmitter.

A combination of scattered and continual pilot positions are used within the OFDM symbol. The continual pilots are at fixed subcarrier positions in every OFDM symbol and any one continual pilot is always modulated by the same symbol for all time. The continual pilot subcarrier indices are given in Table 2.1 and the scattered pilots are distributed over an OFDM frame of 68 OFDM symbols at positions given by (2.3.3).

0	48	54	87	141	156	192
201	255	279	282	333	432	450
483	525	531	618	636	714	759
765	780	804	873	888	918	939
942	969	984	1050	1101	1107	1110
1137	1140	1146	1206	1269	1323	1377
1491	1683	1704				

Table 2.1: Continual pilot position index.

$$k = 3 \times (l\%4) + 12p \mid p \in \mathbb{N}_0, k \in [1, N_C] \tag{2.3.3}$$

with l the OFDM symbol index in the OFDM frame. The dark dots in Figure 2.13 illustrate the positions of scattered and continual pilots in an OFDM frame. In accordance to (2.3.3) the scattered pilots move by three places for every consecutive OFDM symbol. Each continual pilot coincides with a scattered pilot every fourth symbol while the total number of pilots for every OFDM symbol remain a constant 176. A pseudo random binary sequence (PRBS) , generated according to Figure

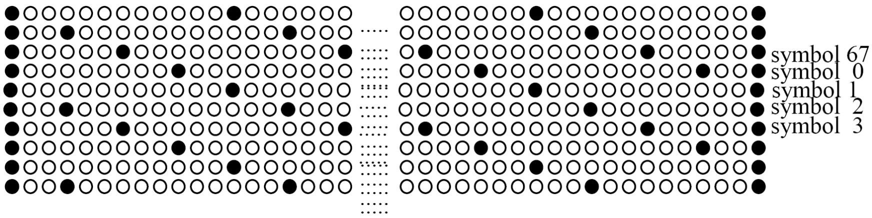


Figure 2.13: Pilot position in OFDM frame (reproduced from [1]).

2.14, determines the value of the continual and scattered pilot symbols. A pseudo random bit is generated for every subcarrier whether or not it is a data or pilot carrier. When a subcarrier is designated as a pilot it is modulated by its corresponding pseudo random bit at a boosted power level compared to a data carrier. The pilots are BPSK modulated as follows:

$$c_k = \frac{4}{3} \times 2(\frac{1}{2} - w_k) + j0 \tag{2.3.4}$$

where c_k is the modulating symbol of the pilot at subcarrier index k and w_k the corresponding pseudo random bit.

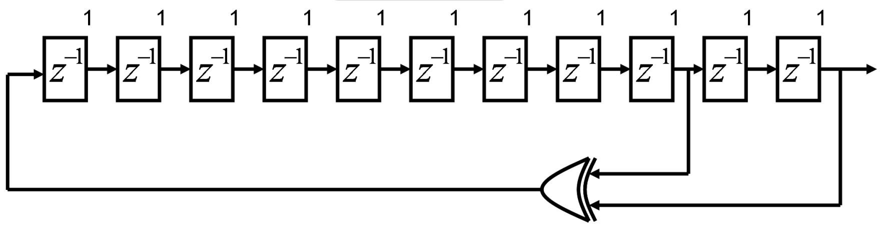


Figure 2.14: Pseudo random binary sequence generator.

The *OFDMMux* component was developed for multiplexing the data and pilot symbols in a vector corresponding to their subcarrier positions. It supports sweeping the pilot to data subcarrier power ratio used to demonstrate the effect of estimating the channel in the presence of noise. The C++ implementation is found in Appendix C.2.

2.3.4 Guard Time

As discussed in Section 2.1.1, the guard time is simply a copy of the last part of each OFDM symbol which is placed ahead of it. The guard time is chosen to be longer than the channel impulse response in order to circumvent inter-symbol interference. This operation is performed by the transmitter after the IFFT in Figure 2.10. The use of a guard interval has gained wide acceptance and used by several standards including DVB-T [1], DVB-H [11], WiMax [2] and 802.11a [3]. Alternate approaches to overcoming the delay spread with no guard interval has been investigated by [12]. For simulation purposes the guard interval is a $\frac{1}{16}$'th of the IFFT period with $R_p = \frac{T_G}{T_s} = \frac{1}{16}$.

2.3.5 RF Transmitter

The RF transmitter in Figure 2.10 would generally first consist of an I/Q up-conversion stage to the first intermediate frequency (IF) followed by further frequency conversion, amplification and filtering stages depending on the designer's implementation. It is also possible to generate a low IF signal in the digital domain with software radio techniques by for example packing the modulating symbols into the IFFT vector (which has to be at least $2N_C$ long) in a complex conjugate fashion in order to generate a real valued sampled signal. This would result in needing only one hardware mixer instead of the two required for I/Q up-conversion to the first IF. In this thesis the investigation is focused on the effects of single RF component phenomena on the complex modulating envelope forming part of a complete system.

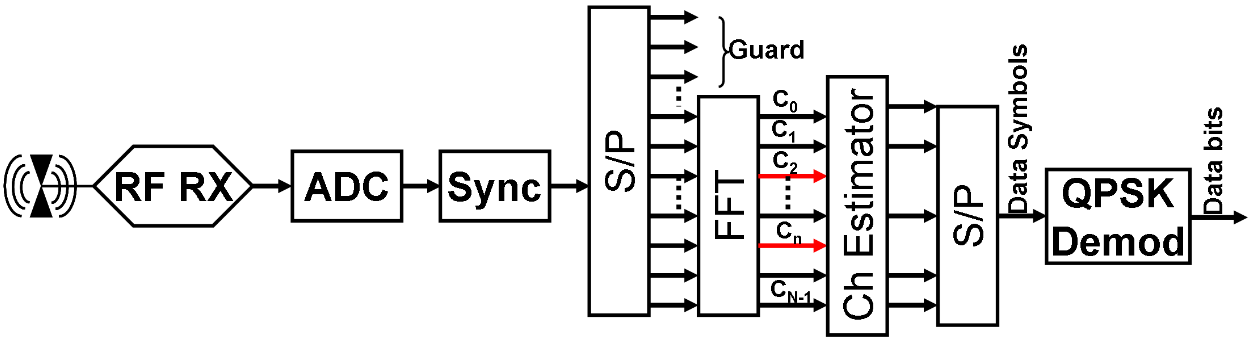


Figure 2.15: OFDM receiver block diagram.

2.4 Receiver Implementation

At the receiver the signal would pass through various amplification, filter and down-conversion stages either all the way down to I/Q baseband or to an IF frequency which is directly sampled by an analogue to digital converter (ADC) . The following parts of this section describe the actions performed on the sampled modulating envelope by the receiver in order to regain the transmitted data.

2.4.1 Synchronization

Timing and frequency synchronization at the receiver is a critical part of the data recovery process. The receiver has to determine where each OFDM symbol starts before performing the demodulating FFT as shown in the block diagram of a typical OFDM receiver in Figure 2.15

A further problem is the fact that the transmit and receive local oscillators used in the frequency conversion stages do not operating at exactly the same frequency. Such a frequency offset results in the loss of orthogonality between subcarriers and inter-carrier interference (ICI) occurs. A study of this is made in [13] where frequency offset is translated to an effective signal-to-noise ratio degradation. The effect of frequency offset is investigated in Section 5.4 and simulation results show that the frequency drift of practical oscillators can have a profound effect on data errors.

Synchronization thus has to be performed in terms of estimating the time and frequency offset. A simultaneous time and frequency estimation algorithm developed by Van de Beek et al [14] is employed. The algorithm performs a simultaneous maximum likelihood estimation of time and frequency offset by exploiting the repetition introduced by the cyclic extension of an OFDM symbol.

Maximum Likelihood (ML) Estimation is the statistical estimation of the parameters of a joint distribution function from observed values [15]. The joint distribution is assumed given as a function of unknown parameters.

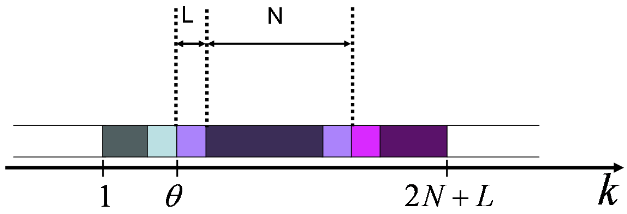


Figure 2.16: OFDM signal with cyclic extension.

Let $r_1, ..., r_n$ be observed values of the random variables $R_1, ..., R_n$ and $f(\mathbf{r})$ is the joint probability density function of the random variables. The maximum likelihood function $f(r_1, ..., r_n|\theta, \varepsilon)$ is calculated from the observed values $r_1, ..., r_n$ given their joint probability density function $f(\mathbf{r})$ as a function of the parameters θ and ε. $f(r_1, ..., r_n|\theta, \varepsilon)$ is thus the likelihood of observing the observed set $r_1, ..., r_n$, as a function of θ and ε.

OFDM Synchronization is performed by making use of a maximum likelihood (ML) estimation of timing and frequency offset as described in [14]. This can be done as we have an approximation of the joint probability density function of the received data.

The received signal is approximated as

$$r(k) = s(k-\theta)e^{2\pi\varepsilon k/N} + n(k) \tag{2.4.1}$$

where $s(k)$ is the transmitted signal and N the number of samples in an OFDM symbol without the guard interval. θ is the time delay and ε the frequency offset as a fraction of the OFDM inter-carrier spacing $\frac{1}{N}$.

The autocorrelation over the interval $k \in [\theta, \theta+L-1]$, with L the guard interval length shown in Figure 2.16, are central to the estimation technique and given as:

$$E\left[r(k)r^*(k+m)\right] = \begin{cases} \sigma_s^2 + \sigma_n^2, & m = 0 \\ \sigma_s^2 e^{j2\pi\epsilon}, & m = N \\ 0, & \text{elsewhere} \end{cases} \tag{2.4.2}$$

with $\sigma_s^2 = E\left[|s(k)|^2\right]$ and $\sigma_n^2 = E\left[|n(k)|^2\right]$.

Knowledge of the joint probability density function of the observed samples is required in order to construct the likelihood function. An OFDM signal is generated by adding independent modulated subcarriers as illustrated in Figure 2.2. When the number of subcarriers is large the received signal is approximated, by the central limit theorem, as a complex Gaussian process. The OFDM signal contains a cyclic extension which is spaced N samples apart. This results in the Gaussian process not being white in the bandwidth of concern. The likelihood function as a function of θ and ε,with k as illustrated by Figure 2.16, is

$$f(\mathbf{r}|\theta,\varepsilon) = \prod_{k=1}^{\theta-1} f(r(k)) \prod_{k=\theta}^{\theta+L-1} f(r(k), r(k+N)) \prod_{k=\theta+L}^{N+L} f(r(k)) \tag{2.4.3}$$

where

$$f(r(k)) = \frac{1}{\pi(\sigma_s^2+\sigma_n^2)} e^{-\frac{|r(k)|^2}{\sigma_s^2+\sigma_n^2}} \tag{2.4.4}$$

and using [8, Appendix B.1] the joint probability function is given as [14]

$$f\left(r(k), r(k+N)\right) = \frac{1}{\pi^2(\sigma_s^2+\sigma_n^2)^2(1-\rho^2)} e^{\frac{|r(k)|^2 - 2\rho\Re\left\{e^{j2\pi\varepsilon} r(k) r^*(k+N)\right\} + |r(k+N)|^2}{(\sigma_s^2+\sigma_n^2)(1-\rho^2)}}. \tag{2.4.5}$$

The likelihood function (2.4.3) is maximised by maximising its log and derived by [14] as

$$\begin{aligned}\Lambda(\theta,\varepsilon) &= log(f((r)|\theta,\varepsilon)) & (2.4.6)\\ &= \mid \gamma(\theta) \mid \cos(2\pi\varepsilon + \angle\gamma(\theta)) - \rho\Phi(\theta) & (2.4.7)\end{aligned}$$

where

$$\gamma(m) = \sum_{k=m}^{m+L-1} r(k) r^*(k+N) \tag{2.4.8}$$

$$\Phi(m) = \frac{1}{2}\sum_{k=m}^{m+L-1} \mid r(k) \mid^2 + \mid r(k+N) \mid^2 \tag{2.4.9}$$

$$\begin{aligned}\rho &= \left|\frac{E\{r(k) r^*(k+N)\}}{\sqrt{E\mid r(k)\mid^2 E\mid r(k+N)\mid^2}}\right| & (2.4.10)\\ &= \frac{\sigma_s^2}{\sigma_s^2+\sigma_n^2} & (2.4.11)\\ &= \frac{\text{SNR}}{\text{SNR}+1} & (2.4.12)\end{aligned}$$

Equation 2.4.7 is the log-likelihood of receiving data samples $r(k)$ as a function of the timing and frequency offset θ and ε. Thus the values for θ and ε that maximise $\Lambda(\theta,\varepsilon)$ are the most likely to have occurred and denoted by θ_{ML} and ε_{ML} respectively. The likelihood is maximised, as proved by [14], when

$$\theta_{ML} = \arg\max_{\theta} \left[|\gamma(\theta)| - \rho\Phi(\theta)\right] \tag{2.4.13}$$

and

$$\varepsilon_{ML} = -\frac{1}{2\pi}\angle\gamma(\theta_{ML}). \tag{2.4.14}$$

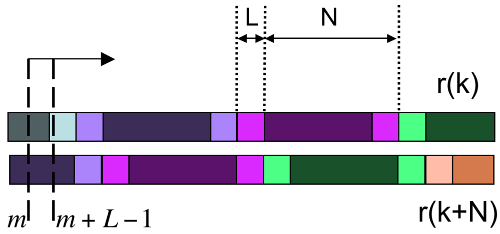


Figure 2.17: Moving sum for correlation- and power term calculations in ML synchronization.

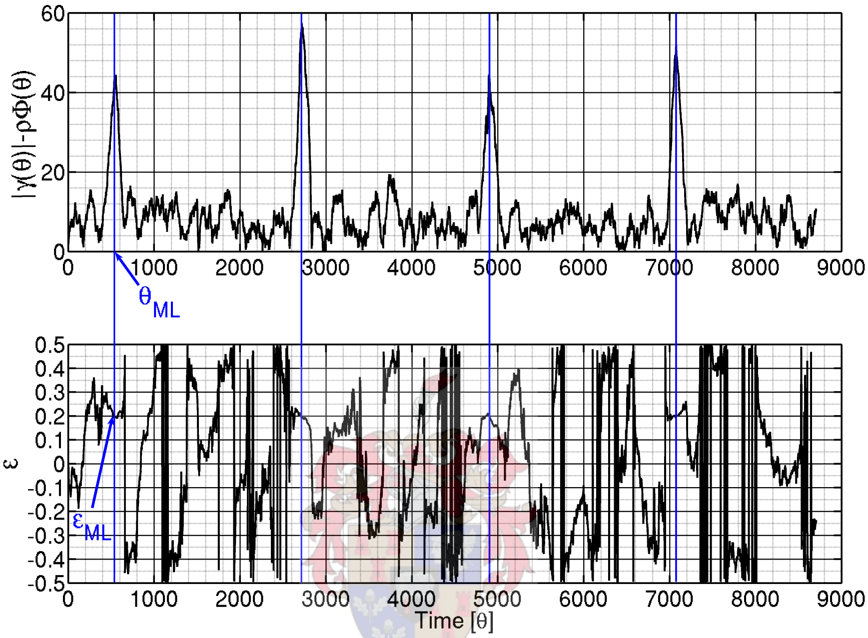


Figure 2.18: Maximum likelihood offset estimation for an OFDM signal with a timing offset of 554 samples and frequency offset of $\frac{0.2}{N}$ cycles per sample.

Figure 2.17 graphically illustrates the process by which (2.4.8) and (2.4.9) are calculated. Equation (2.4.8) takes the form of a running correlator while (2.4.9) is related to the average energy of the received signal and its delayed version over the same interval that the correlation is performed.

The ρ term in (2.4.13) which is related to the SNR (2.4.12) is not necessarily know to the receiver nevertheless it is shown by simulation in Section 5.1.3 that system performance is insensitive to not knowing its exact value.

Appendix C.5.1 contains the C++ implementation of the simultaneous timing and frequency offset estimation under the component name *OFDMMLSync*. Figure 2.18 illustrates the timing and frequency offset estimation, generated by the *OFDMMLSync* component, for a frequency-flat static channel in the presence of additive white Gaussian noise. The OFDM signal contains 1705 subcarriers, $N_{\text{FFT}} = 2048$, $L = 128$ and the SNR is $3dB$. The signal is delayed by 554 samples and frequency offset by $\frac{0.2}{N}$. The ML values are obtained by searching for peaks in $|\gamma(\theta)| - \rho\Phi(\theta)$

in blocks of $N + L$ samples as indicated in Figure 2.18 by the vertical blue lines. The sample at which the peak is found is the ML timing offset estimation (θ_{ML}). The frequency offset ε_{ML} is the value of ε at $\theta = \theta_{ML}$. Figure 2.18 roughly illustrates that the ML estimation is functioning well under very noisy conditions (3dB SNR).

A more thorough investigation of this timing and frequency offset estimation is performed in Section 5.4 where its performance is evaluated in a multipath channel with Doppler spread.

Care has to be exercised during the implementation especially when synchronising the very first received OFDM symbol. When a peak is detected in the first or last L samples of the $N+L$ block, a false peak may be generated by a partially received OFDM symbol or be caused by part of the next symbol peak while the first peak is present but smaller which is particularly likely in a fading channel. A solution to this problem is to move the peak searching algorithm on by $\frac{N+L}{2}$ samples. This ensures that the peaks occur in the middle of the $N + L$ block. The *OFDMMLPeakSearch* component, with source code in Appendix C.5.2, implements the peak searching.

It should be noted that the ML frequency offset estimation becomes ambiguous for offsets larger than $\pm 50\%$ of the OFDM subcarrier spacing. For this reason it is assumed by [14] that the frequency offset is known to within $\pm$half the subcarrier spacing either by making a coarse offset estimation or using local oscillators with sufficient precision.

The synchronisation algorithm described above searches for the ML start of each symbol and does not lose sync due to an offset in the sampling clock. A sampling clock that is running too fast or too slow is inherently compensated for by dropping or repeating samples respectively.

Although synchronisation is not lost, an analogue to digital converter with and erroneous clock produces a sampled OFDM symbol with subcarriers that no longer consist an integer amount of cycles. The subcarriers lose orthogonality and inter-carrier interference is introduced.

By monitoring the distance between the estimated timing offsets and knowing that they are supposed to be $N + L$ samples apart, the sampling clock offset can be estimated and tuned by appropriate feedback.

The static manner in which scheduling is implemented in the synchronous data flow simulation domain [16] renders it unsuited for modelling asynchronous sampling offsets and thus the sampling clock is assumed to be precise when the frequency offset estimation is evaluated in Chapter 5.

2.4.1.1 Alternative Frequency Offset Estimation Algorithms

Frequency synchronization can be achieved by making use of a training preamble as an alternative to using the cyclic prefix. Moose [13] has proposed a method by which two identical OFDM training symbols are transmitted without a guard interval. When the same amount of subcarriers are used in the training symbol as for the data symbols a maximum unambiguous offset estimation range of $\pm 50\%$ of the subcarrier spacing results. In order to resolve the ambiguity for offsets larger than $\pm 50\%$ of the inter-carrier spacing, a strategy is suggested where two shortened training symbols containing less subcarriers and spaced further apart in frequency are used. A larger frequency offset can be estimated but with inferior precision. This provides a rough estimate in order to resolve the above mention ambiguity, after which two longer training symbols are used for high precision.

Moose's idea of exploiting the correlation properties of training symbols is extended by Cox and Schmidl [17]. The first training symbol is transmitted with only even indexed subcarriers and results in a symbol with two identical parts in the time domain. A second pseudo-random training symbol is then used to resolve the estimation ambiguity.

Morelli and Mengali [18] describe a technique which extends [17] and renders similar precision. It employs only one training symbol with $L > 2$ identical parts in the time domain to give an unambiguous estimation range of $\pm\frac{L}{2}$ of the subcarrier spacing.

2.4.1.2 Coarse Frequency Offset Estimation

The alternatives described above offer solutions to estimating the frequency offset beyond $\pm 50\%$ of the subcarrier spacing but at the expense of introducing training symbols.

Synchronisation without special training symbols would be preferred as it decreases the signal complexity and simplifies transmit-receive coordination. Furthermore, OFDM is characterised by long symbol periods and the performance of estimation by periodic training symbols may deteriorate rapidly in a changing environment.

A technique is therefore required to extend the ambiguous range of [14] without increasing the complexity of the OFDM signal already proposed in Section 2.3. Resolving the frequency offsets larger than $\pm 50\%$ of the subcarrier spacing is facilitated by the fact that the pilots, used for estimating the channel frequency response, are transmitted at increased power. The demodulating FFT produces a "zero padded" vector due to signal oversampling. The frequency synchronisation technique previously discussed estimates the frequency offset to the nearest subcarrier and once this offset is removed the signal is effectively locked to a frequency which is offset by an integer multiple of the subcarrier spacing. As long as the received signal is sufficiently sampled the data can be recovered by the demodulating FFT while the symbols of the resulting vector are simply offset. The pilots are transmitted with increased power and the maximum likelihood estimate of the offset in a flat and static channel with AWGN is simply the correlation of the demodulated amplitudes with the known symbol and pilot subcarrier positions:

$$\Delta F_{ML} = \arg\max_{\Delta F} \left\{ \sum_{n=\Delta F}^{\Delta F + N_C - 1} |r_n| \times c_n \right\}; \quad \Delta F \in [0, N - N_C] \tag{2.4.15}$$

where r_n is the demodulated receive vector and c_n are the relative argument values of the transmit-

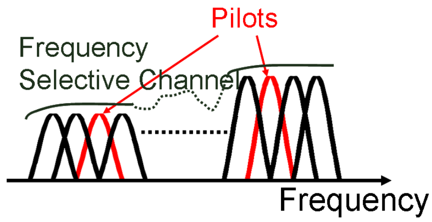


Figure 2.19: Received signal passed though a frequency selective channel.

ted subcarriers:

$$c_n = \begin{cases} 1 & , \text{ if data carrier} \\ \sqrt{R_p} & , \text{ if pilot carrier} \\ 0 & , \text{ elsewhere} \end{cases} \tag{2.4.16}$$

This will be referred to as *coarse frequency offset estimation* and would be perform after timing and fine frequency synchronisation.

2.4.2 Demodulation

Coherent demodulation occurs at the receiver after the timing and frequency synchronization is performed. The sampled baseband signal is passed through the FFT operation in Figure 2.15 which produces the data or pilot symbols for each subcarrier. These data symbols still contain phase and amplitude errors and correction occurs after channel estimation as described in the following section.

2.4.3 Channel Response Estimation

The same scattered pilot scheme used in [1] (2k mode) was implemented and discussed in Section 2.3.3.

After the FFT demodulation process in Figure 2.15, the red lines indicate the received pilot signals that are used to estimate the amplitude and phase errors of each subcarrier. The pilot symbols are known to the receiver and thus an estimation of the channel frequency response is produced at the frequency position of each pilot. The channel estimation for other data carrying symbols is performed by a linear 2D interpolation of the estimated amplitude and phase between pilot symbols. The channel estimations used in simulations are performed independently for each OFDM symbol and previous estimates do not form part of the algorithm.

Channel response estimation is performed by the *OFDMChEst* component in Appendix C.9 after which the data is extracted from the subcarrier vector.

Piloted channel estimation in OFDM which makes use of an FFT proves to be more computational efficient than a channel equalizer in conventional wide-band single carrier systems and is discussed in [9, Chapter 2.7] .

2.4.4 Symbol Decoding

Once the QPSK symbols' phase and amplitude are corrected they are decoded into a bit stream. It is assumed that the probability that one of the four possible symbols was transmitted is equal and thus the optimal decoding is given by a threshold detector displayed in Figure 2.20 and implemented by the *QPSKDeCoder* component in Appendix C.12. The decoded bits can then be compared to the bits originally transmitted to calculate a bit error rate.

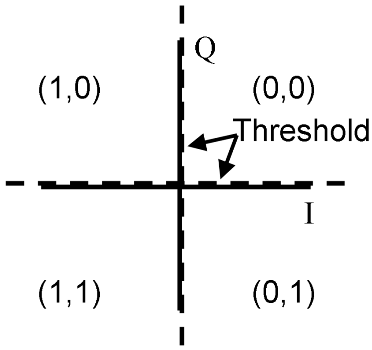


Figure 2.20: Symbol threshold decision for QPSK after demodulation and channel correction.

Chapter 3

Multiple-Input Multiple-Output Communication Systems

MIMO wireless communication technology has recently attracted much attention because of its promising increase in spectral efficiency. Multi-path wireless propagation phenomena have a deteriorating effect on conventional communication systems while MIMO systems exploit the multi-path as to increase data throughput. This chapter provides an introduction to the fundamentals of wireless communication with multi-element transmit and/or receive arrays.

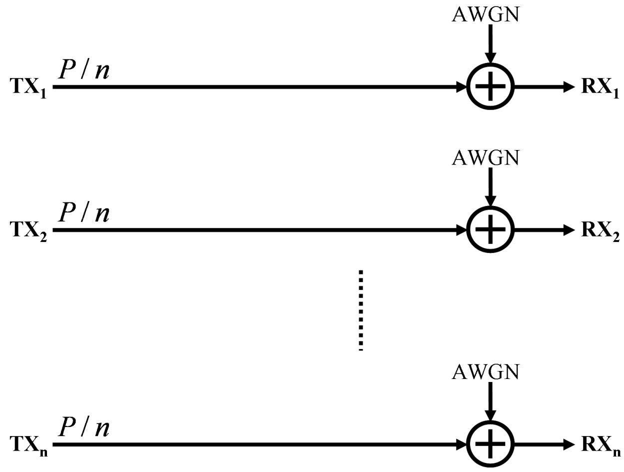
The advantage of transmitting information over parallel channels is illustrated from the classical Shannon capacity point of view for static channels by means of simplified examples in Section 3.1. Section 3.2 considers the general capacity of MIMO systems by decomposing the channel into equivalent eigenmodes which in turn represent orthogonal channels. Rayleigh fading is discussed in Section 3.3 and the potential superiority of MIMO techniques is highlighted.

The maximum ratio combiner and its implementation in OFDM is described in Sections 3.4 and 3.5 and an orthogonal transmit diversity scheme is described in Sections 3.6 and 3.7. An overview of other MIMO algorithms is given in Section 3.8.

3.1 Time-invariant Channel Capacity

The channel capacity of arbitrary transmit- and receive array combinations are discussed in order to provide an insight into the fundamental limits of transmitting and recovering reliable information in multiple-input multiple-output communication systems. Nyquist [19], Hartley [20] and most importantly Shannon [21] introduced the concept that information can be transmitted over a noisy channel with arbitrarily small probability of error provided that the rate of information does not exceed the channel capacity. Similarly, MIMO systems with multiple channel links are bound by an information theoretical optimal performance. Reference is made to the general capacity of MIMO systems including several examples to illustrate the effectiveness of exploiting spacial

(a) Single transmit and receive.



(b) Multiple parallel transmit and receive systems

Figure 3.1: Capacity illustration of signal power spread over various channels

diversity and the degree-of-freedom gain of the channel.

3.1.1 Parallel Channel Capacity with AWGN

In a single transmit single receive system with AWGN, as seen in Figure 3.1(a), the channel capacity in bits per second per hertz of signal bandwidth (bps/Hz) (or equivalently bits per cycle) is commonly known as the Shannon-Hartley law and given by [21]:

$$C = \log_2 (1 + \rho) \tag{3.1.1}$$

with ρ the signal-to-noise power ratio.

Now consider a system where the transmit power is divided among n transmitters each transmitting independent data over an isolated channel as shown in Figure 3.1(b) with a total system capacity of:

$$C = n \times \log_2(1 + \frac{\rho}{n}). \tag{3.1.2}$$

were ρ is the signal-to-noise power ratio at the receiver which would be present if the system had one transmit and one receive antenna. In the limit, as the number of parallel channels increase, the capacity becomes

$$\underset{n \to \infty}{C} = \frac{P}{N_0 \ln 2} \tag{3.1.3}$$

which results in a linear capacity increase with an increase in SNR. On the contrary for the single transmit and receive system the capacity increases logarithmically as given by (3.1.1). The capacity

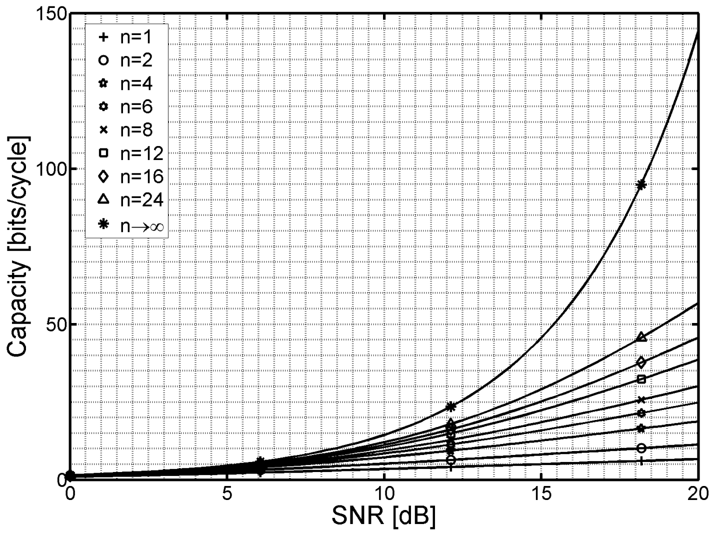


Figure 3.2: Capacity of system with n parallel channels.

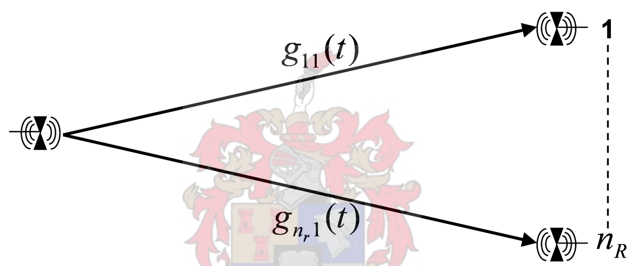


Figure 3.3: System with one transmitter and multiple receivers.

increase achieved by dividing the transmitted signal power among multiple channels is illustrated by Figure 3.2.

In wireless communication, multiple links may exist between antenna array elements and could be exploited to increase the system capacity. Every receive element will generally observe a transformed sum of every transmit element's signal. The basic technique for determining channel capacity is to set up a matrix describing the channel and perform a singular value decomposition in order to convert any channel matrix into an equivalent set of uncoupled parallel channels as in Figure 3.1(b).

3.1.2 Multiple Receive Elements with AWGN

A system with one transmitter and n_r receivers as depicted by Figure 3.3 has a channel matrix of

$$\mathbf{G} = \begin{bmatrix} g_{11} \\ g_{21} \\ \vdots \\ g_{n_r 1} \end{bmatrix} \tag{3.1.4}$$

while assuming the channel is frequency-flat and therefore the time dependence is omitted. The received signal vector is given by

$$\mathbf{r} = \mathbf{G}s + \mathbf{n} \tag{3.1.5}$$

where s is the transmitted symbol and $\mathbf{n}$ a vector of independent circular symmetric complex Gaussian noise. In order to formulate a function for capacity by which different systems can be compared directly, the definition of a normalised channel matrix $\mathbf{H} = \sqrt{\frac{\hat{P}}{P}}\mathbf{G}$ becomes useful. P is the average received signal power if there were only one transmit and one receive element and $\hat{P}$ the total transmitted power.

The channel capacity results when the received signals are optimally combined in order to maximise the mutual information between the input and output and is accomplished by the linear maximum ratio combiner [22] discussed in Section 3.4. Maximum ratio combining increases the effective signal-to-noise ratio and the capacity becomes

$$C = \log_2\left(1 + \rho \sum_{m=1}^{n_r} |h_{m1}|^2\right). \tag{3.1.6}$$

In the case of a static channel where every link experiences the same attenuation the normalised channel matrix is given by:

$$\mathbf{h} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \updownarrow n_r \tag{3.1.7}$$

resulting in a capacity of

$$\log_2(1 + \rho \times n_r) \tag{3.1.8}$$

Equation 3.1.8 demonstrates that the optimum combining linearly increases the signal-to-noise ratio with an increase in the amount of receivers. The capacity increase in a fading channel is more involved and is dealt with shortly.

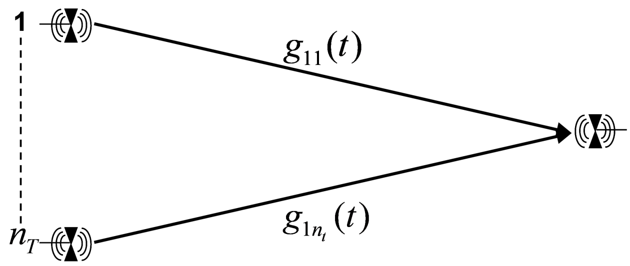


Figure 3.4: System with multiple transmitters and one receiver.

3.1.3 Multiple Transmit Elements with AWGN

A system with transmit diversity is shown in Figure 3.4 and again it is assumed that the total transmit power is independent of the number of transmit elements. The normalised channel matrix is constructed in a similar manner as described in the previous section and is given by

$$\mathbf{H} = \sqrt{\frac{\hat{P}}{P}} \left[g_{11} \; g_{12} \; \ldots \; g_{1n_r} \right]. \tag{3.1.9}$$

Optimum transmit beam-forming at the transmitter, provided the transmitter knows the channel, would result in the signals adding in-phase at the receiver and provide an effective SNR gain [23]. The capacity would then be

$$C = \log_2 \left(1 + \rho \sum_{m=1}^{n_t} |h_{1m}|^2 \right) \tag{3.1.10}$$

which is the same for the MRC with the same amount of receivers. In many cases it may not be realistic to assume that the transmitter has channel knowledge as this would require some form of feedback from the receiver. When independent data symbols are transmitted from all antennas the capacity becomes [22]

$$C = \log_2 \left(1 + \frac{\rho}{n_t} \sum_{m=1}^{n_t} |h_{1m}|^2 \right). \tag{3.1.11}$$

The legacy of transmitting independent symbols from each antenna becomes apparent when dealing with MIMO channels in the next section. If the channel is static with equal attenuation in each link the normalised channel becomes

$$\mathbf{H} = \underbrace{[1 \; \ldots \; 1]}_{n_t} \tag{3.1.12}$$

and the capacity

$$C = \log_2 (1 + \rho) \tag{3.1.13}$$

which is as expected, the same for the single-input single-output (SISO) case. The transmit diversity gain becomes evident when the channel links fade independently as will be revealed subsequently.

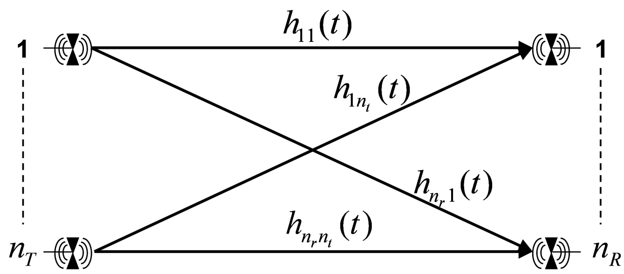


Figure 3.5: Channel matrix illustration.

3.2 General Capacity Formulation

The general MIMO system is depicted in Figure 3.5 and will be described in the matrix notation of [22]. The transmitted signals are presented by the column vector

$$\mathbf{s(t)} = \begin{bmatrix} s_1(t) \\ \vdots \\ s_{n_t}(t) \end{bmatrix} \tag{3.2.1}$$

with $s_k(t)$ the signal transmitted from the k'th transmitter. Similarly, the received signals are described by the vector

$$\mathbf{r(t)} = \begin{bmatrix} r_1(t) \\ \vdots \\ r_{n_r}(t) \end{bmatrix} \tag{3.2.2}$$

The channel is described by $n_r \times n_t$ impulse responses in matrix notation as given in (3.2.3).

$$\mathbf{g(t)} = \begin{bmatrix} g_{11}(t) & g_{12}(t) & \dots & g_{1n_t}(t) \\ g_{21}(t) & g_{22}(t) & \dots & g_{2n_t}(t) \\ \vdots & \vdots & \ddots & \vdots \\ g_{n_r1}(t) & g_{n_r2}(t) & \dots & g_{n_rn_t}(t) \end{bmatrix} \tag{3.2.3}$$

where $g_{mn}(t)$ is the impulse response between transmit antenna n and receive antenna m. The normalised channel matrix is more convenient for formulating channel capacities and is given as:

$$\mathbf{h(t)} = \sqrt{\frac{P}{\hat{P}}}\mathbf{g(t)} \tag{3.2.4}$$

where P is the average received power as if there were only one transmit and one receive antenna and $\hat{P}$ the total transmit power independent of the number of transmitters. The received signals are given by

$$\mathbf{r(t)} = \sqrt{\frac{\hat{P}}{P}}\mathbf{h(t)} * \mathbf{s(t)} + \mathbf{n(t)} \tag{3.2.5}$$

where $*$ denotes convolution and **n(t)** represents the independent white Gaussian noise added at each receiver. When the channel is assumed to be frequency flat with respect to the signal bandwidth, which is equivalent to a negligible delay spread, the channel impulse response **h(t)** is modelled simply as complex random variables representing the gains and phase shifts of each link. The channel time dependence is eliminated from (3.2.5) and becomes

$$\mathbf{r(t)} = \sqrt{\frac{\hat{P}}{P}}\mathbf{h} \cdot \mathbf{s} + \mathbf{n}. \tag{3.2.6}$$

The normalised channel matrix represents a linear transformation of the transmitted signal vector. Any linear transformation can be represented by a vector rotation followed by a scaling of components and another rotation [23, 24] which is known as singular value decomposition. The channel matrix may be written as:

$$\mathbf{H} = \mathbf{U}\Lambda\mathbf{V}^{*} \tag{3.2.7}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and Λ a diagonal matrix of singular values. It was revealed by [24] that the capacity of a channel described by Λ is equivalent to that of $\mathbf{H}$ and is given by

$$C = \sum_{k} \log_2 \left(1 + \frac{P_k \lambda_k^2}{N_0}\right) \tag{3.2.8}$$

with λ_k the singular values of $\mathbf{H}$ or λ_k^2 the eigenvalues of $\mathbf{H}\mathbf{H}^{\dagger}$. Independent data has to be transmitted in every eigenmode in order to achieve capacity. The power allocated to each eigenmode is given by P_k with the constraint that $\sum_k P_k = P$. With transmit side channel information the power could be optimally allocated by way of the waterfilling algorithm [23] but for this study it is assumed that only the receiver has channel knowledge. This is a realistic assumption as transmitter channel knowledge would imply that a feedback path form the receiver to the transmitter is present. The existence of such a feedback path is generally impractical if the channel may change during the time taken to inform the transmitter of the channel state and in so doing invalidating the feedback information.

When the transmitter has no knowledge of the channel matrix with independent equally distributed random entries, the average mutual information between the transmitter and receiver is maximised when data transmitted by each antenna is statistically independent and power is allocated equally among all transmit elements [22, 24]. The capacity is given by

$$C = \log_2 \left(\det[\mathbf{I}_{\mathbf{n_r}} + (\rho / n_t \mathbf{H}\mathbf{H}^{\dagger})]\right). \tag{3.2.9}$$

with $\rho = P/N_0$.

3.3 Rayleigh Fading Channel Capacity

When the radio communication channel brings about fading to the received signal the capacity as defined in (3.2.9) has to be perceived in a manner depending on the channel coherence time. It should be noted that the maximum rate of error free information transfer approaches the channel capacity as the suitable error correcting code approaches infinite length. In practical systems the coding length is finite and the concept of channel capacity may be interpreted in two manners.

Firstly, consider the slow fading case where the channel coherent time is long compared to the code length. The channel may then be considered as a sequence of long quasi-static intervals, each associated with a particular SNR, during which the corresponding capacity can be approached with arbitrary small probability of error. The capacity given by (3.2.9) is therefore considered a random variable as a different capacity is associated with every channel realisation. This approach would be realistic when considering an office wireless LAN where communication occurs in bursts while changes in the channel are slow. For any given burst, a probability is associated to whether a certain capacity could be achieved or not depending on the SNR during that particular burst. When the particular capacity cannot be achieved the system is said to be in outage. A common measure of system performance, as used by [22], is to determine the capacity's complementary cumulative distribution functions (CCDF) for different SNRs. A figure of merit is then established by determining the maximum information rate that the system can run at for say 99% of the bursts.

The second scenario is when the error correction code extends over a large period during which the channel is changing. In [23, Section 5.4.5] it is deduced that as the codeword length tends to infinity a capacity of

$$C = E\left[\log_2\left(\det[\mathbf{I}_{\mathbf{n_r}} + (\rho/n_t\mathbf{H}\mathbf{H}^{\dagger})]\right)\right] \tag{3.3.1}$$

may be achieved and will be referred to as the *ergodic capacity*. In practical systems this capacity may be approached more rapidly with short codewords by employing adequate interleaving.

A few examples with Rayleigh fading is considered next while assuming that the array elements are positioned sufficiently far apart so that the links between the transmit and receive elements fade independently. This assumption is justified by the investigations of [25] where it was found that an antenna spacing of $\frac{\lambda}{2}$ results in roughly decorrelate channel links in a typical urban environment.

The presented capacities are calculated by simulation where 10^5 random channel realisation where found to produce adequate convergence. The examples are used to illustrate that systems with multiple receive or transmit elements only benefit from what will be referred to as *diversity gain*. The diversity gain in effect combats fading due to the presence of multiple independently fading links. Furthermore a system with multiple receive elements profits from a gain in effective SNR at the receiver while those with both multiple transmit and multiple receive antennas may additionally benefit from a *degree-of-freedom* gain.

3.3.1 Multiple Receive Antennas in a Rayleigh Fading Channel

In an independently Rayleigh fading channel with multiple receive elements, referred to as a single-input multiple-output (SIMO) system, the normalised matrix is given by

$$\mathbf{H} = [h_1 \ \ldots \ h_n] \, . \tag{3.3.2}$$

The channel link entries $h_k \in \mathcal{CN}(0,1)$ are independent circular symmetric complex Gaussian random processes with unit variance and zero mean. The capacity from (3.1.6) is given by

$$\begin{aligned} C &= \log_2\left(1 + \rho \sum_{m=1}^{n_r} |h_{m1}|^2\right) & (3.3.3)\\ &= \log_2(1 + \frac{\rho}{2}\chi^2_{2n_r}). & (3.3.4)\end{aligned}$$

χ^2_k is the Chi-square variate with k degrees-of-freedom and can be described by the sum of independent squared Gaussian variates as

$$\chi^2_k = \sum_{i=1}^{k} (X_i)^2 \ ; \ \ X_i \in \mathcal{N}(0,1) \tag{3.3.5}$$

The Chi-square probability density function is given by [26, Appendix F] as

$$f_{\chi^2_k}(x) = \frac{x^{(k/2)-1}}{2^{k/2}\Gamma(k/2)} e^{-x/2} u(x) \tag{3.3.6}$$

with mean k and variance $2k$.

In the quasi-static case the burst capacity is illustrated in Figure 3.6 as CCDFs which demonstrate the probability that a burst will succeed without error at a given capacity. As an example, with an average SNR of 18dB per receiver the communication bursts may meet a capacity of 1.1, 4.3 and 6.7 bits/s/Hz for the 1X1, 1X2, 1X4 respectively 99% of the time.

When error correction encoding occurs over multiple fades, as explained earlier, the resulting ergodic channel capacity as calculated by simulation is given in Figure 3.7. When the amount of receive antennas are small an increase in their number has a great effect on overcoming the fading channel while as the amount of receive antennas becomes large an increase merely increases the effective SNR and a logarithmic growth in capacity is observed. The commonly cited 1 bit/cycle approximate increase for every doubling in receive antennas (in the high SNR regime) is evident from Figure 3.8.

Although the multi-path combats fading and an increased amount of receivers improves the SNR, a SIMO system does not have the ability to construct parallel communication paths and thus exhibits no increased spacial degrees-of-freedom compared to a SISO system [23]. Therefore

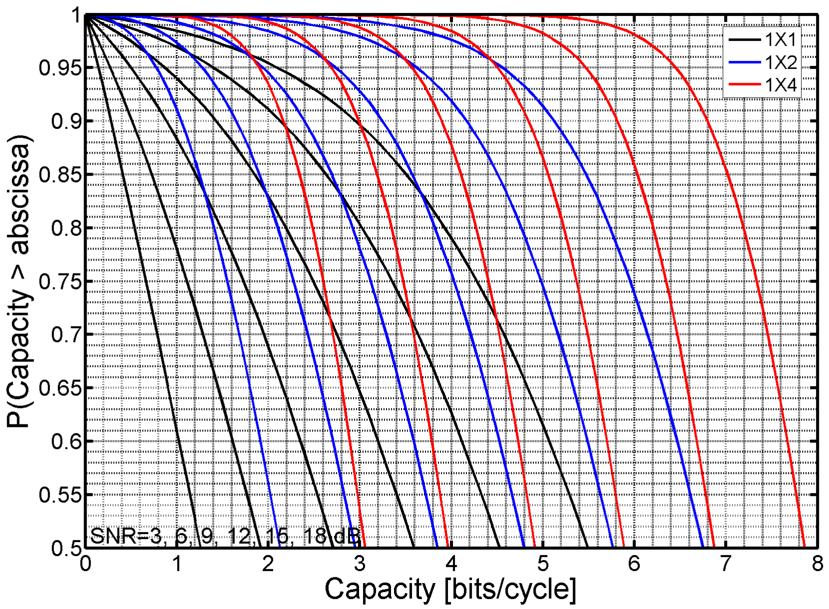


Figure 3.6: Complementary cumulative distribution function of burst capacity for systems with receive diversity.

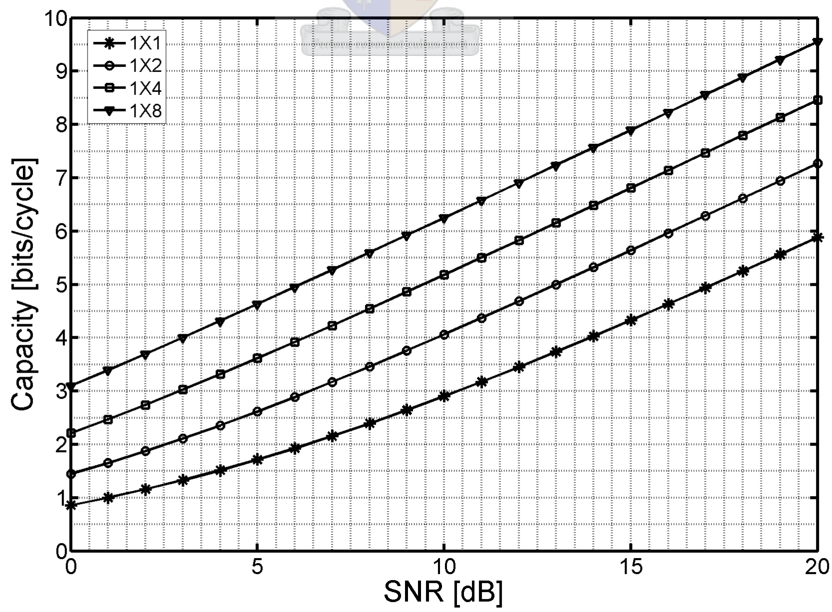


Figure 3.7: Ergodic capacity for systems with receive diversity.

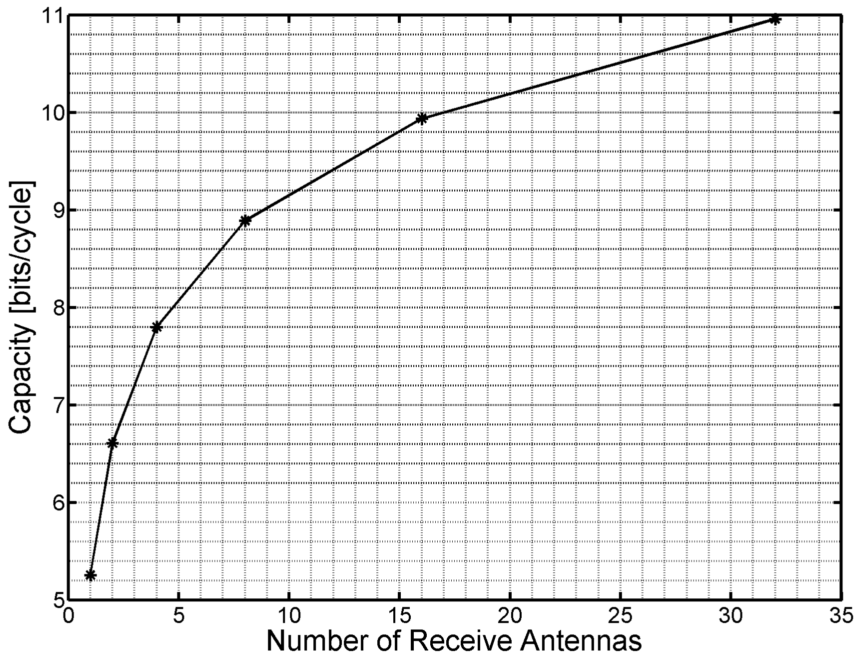


Figure 3.8: Ergodic capacity for systems with receive diversity at 18dB SNR.

the capacity growth is inevitably logarithmic in both the large SNR and/or large n_r regimes. A combination of transmit and receive diversity is required in order to accomplish a more linear growth in capacity as achieved by the parallel system depicted in Figure 3.1(b).

3.3.2 Multiple Transmit Antennas in a Rayleigh Fading Channel

The channel capacity of a system with multiple transmit antennas and one receiver, referred to as a multiple-input single-output system (MISO) , where the transmitter has no channel knowledge is calculated from (3.2.9) as [22]:

$$C = \log_2\left(1 + \frac{\rho}{n_t}\sum_{n=1}^{n_t}|h_{1n}|^2\right) \tag{3.3.7}$$

$$= \log_2\left(1 + \frac{\rho}{2n_t}\chi^2_{2n_t}\right) \tag{3.3.8}$$

The capacity increase due to increasing the transmit diversity while maintaining a constant total transmit power is illustrate by the burst capacity given in Figure 3.9. As an example, at an average SNR of 18dB, 99% of the communication bursts may meet a capacity of 1.1, 3.3 and 4.7 bits/s/Hz for the 1X1, 2X1, 4X1 systems respectively.

The ergodic capacity curves in Figures 3.10 and 3.11 suggest that the capacity converges as the number of transmit antennas increases. When the number of transmit antennas n_t is large the term

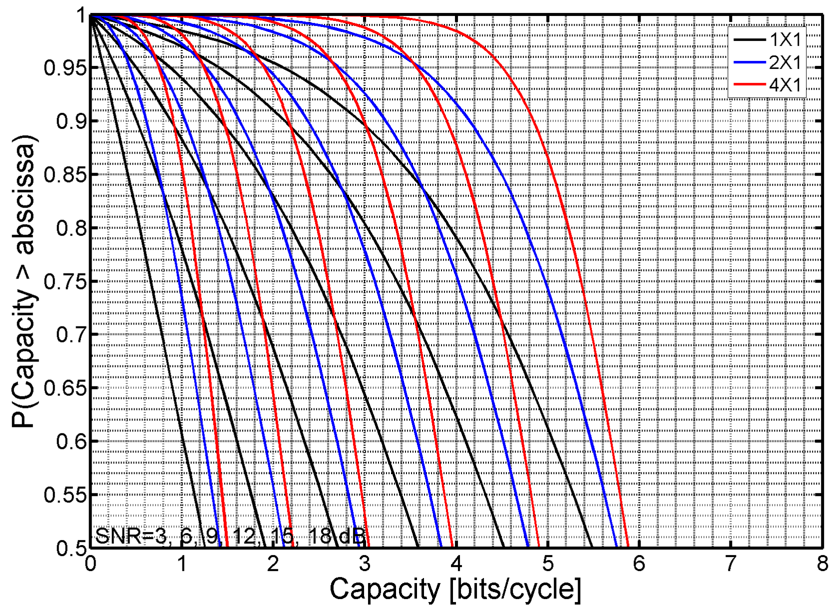


Figure 3.9: Complementary cumulative distribution function of burst capacity for systems with transmit diversity.

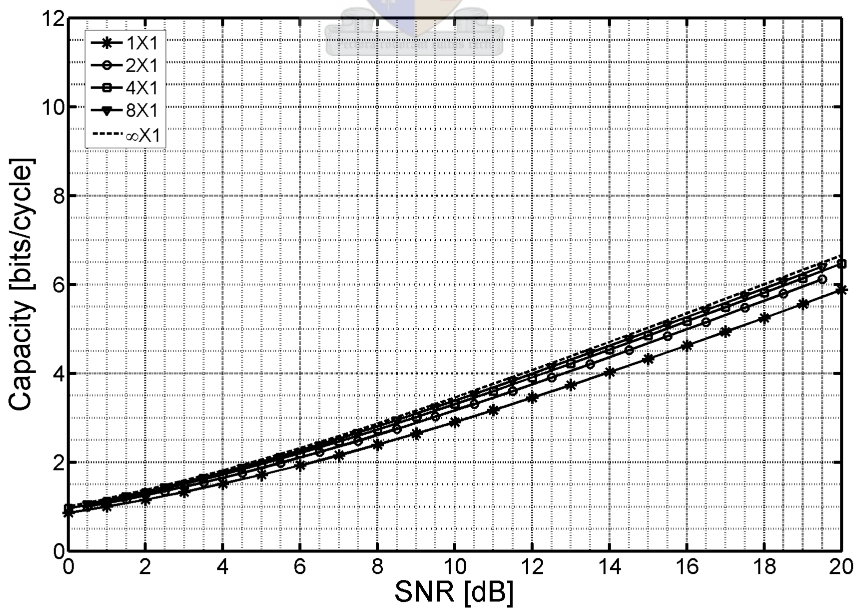


Figure 3.10: Ergodic capacity for systems with transmit diversity.

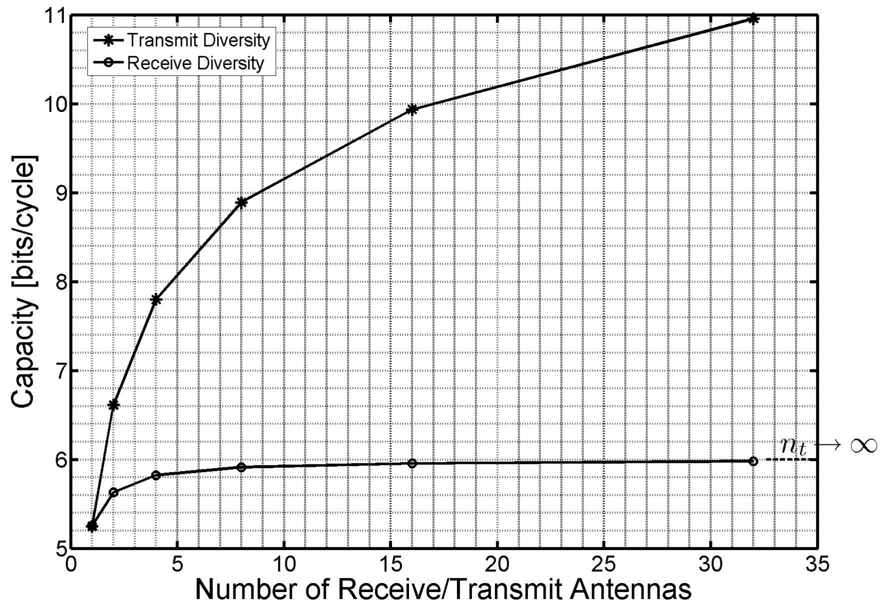


Figure 3.11: Ergodic capacity comparison between systems with either receive or transmit diversity at 18dB SNR. The total transmitted power is independent of the number of transmit front-ends and the transmitter has no channel knowledge.

$\frac{\rho}{2n_t}\chi^2_{2n_t}$ in (3.3.8) tends to a Gaussian process with mean ρ and variance $\frac{\rho^2}{n_t}$. Therefore

$$\lim_{n_t\to\infty} \frac{\rho}{2n_t}\chi^2_{2n_t} = \rho \tag{3.3.9}$$

and thus

$$\lim_{n_t\to\infty} C = \log_2(1+\rho) \tag{3.3.10}$$

meaning that as the amount of transmit antennas approaches infinity the system is able to combat the channel fading completely. This supports the intuitive limit that a MISO system in a fading channel, with AWGN and devoid of transmit channel knowledge can perform at best like a single input single output system where fading is absent.

The superiority of a receive diversity scheme over one with the same diversity but at the transmitter is revealed in Figure 3.11 and by comparing the results of Figure 3.6 to that of Figure 3.9.

The use of multiple transmit elements becomes more significant when used in combination with multiple receivers so as to facilitate the formation of parallel channels (multiple eigenmodes) and increase a communication system's so-called *degrees-of-freedom*.

3.3.3 MIMO Capacity in a Rayleigh Fading Channel

It was revealed from the preceding discussion that the growth in capacity for SIMO or MISO systems is inevitably logarithmic as the amount of antenna elements increase. A combination of

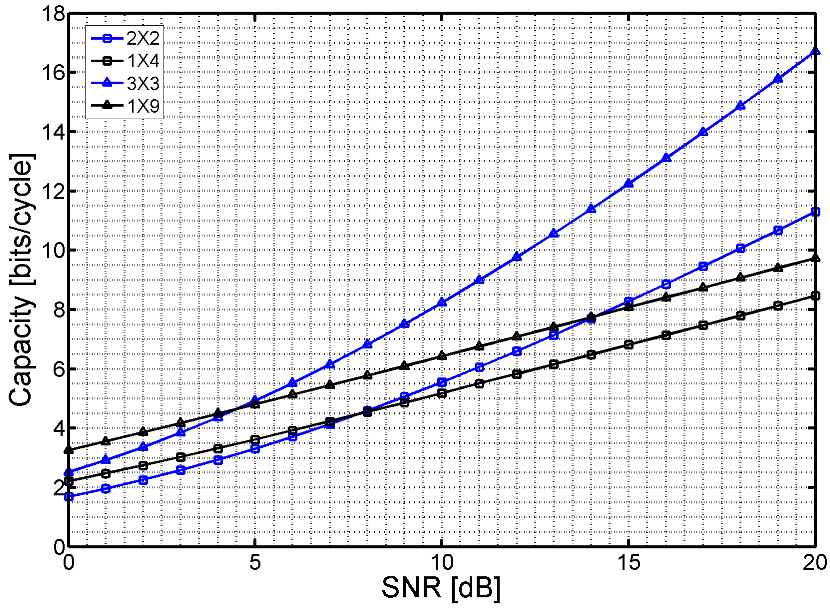


Figure 3.12: Ergodic capacity comparison between MIMO and SIMO systems in Rayleigh fading and AWGN. The total transmitted power is independent of the number of transmit front-ends and the transmitter has no channel knowledge.

transmit and receive diversity has to be utilized in order to achieve a more linear growth in capacity, as in the case of the parallel transmission system described by Figure 3.1(b). In linear algebra terminology the rank of channel matrix $\mathbf{H}$ has to increase with an increase in antenna elements. The general ergodic capacity for systems with no knowledge of the channel at the transmitter is given by [22]

$$C = E\left[\log_2\left(\det[\mathbf{I_{n_r}} + (\rho/n_t\mathbf{HH}^{\dagger})]\right)\right]. \tag{3.3.11}$$

The elements of $\mathbf{H}$ are independent complex circular symmetric Gaussian random processes.

3.3.3.1 Comparison of MIMO and SIMO systems with the same diversity.

The 2X2 and 1X4 systems are compared as they both have four independently fading paths. The ergodic capacities which were calculated by simulation and shown in Figure 3.12 illustrate the significance of increased degrees-of-freedom in the communication channel. In the low average SNR per receiver regime the 1X4 system outperforms the 2X2 system as a result of the 3dB increase in effective SNR due to the increased number of receive elements. The effect on capacity due to this increase becomes smaller as the SNR per receiver increases and in the large SNR regime the increased degrees-of-freedom renders the 2X2 system superior. The same is observed when comparing the 3X3 and 1X9 systems except that the effective SNR gain for the 1X9 system is less significant.

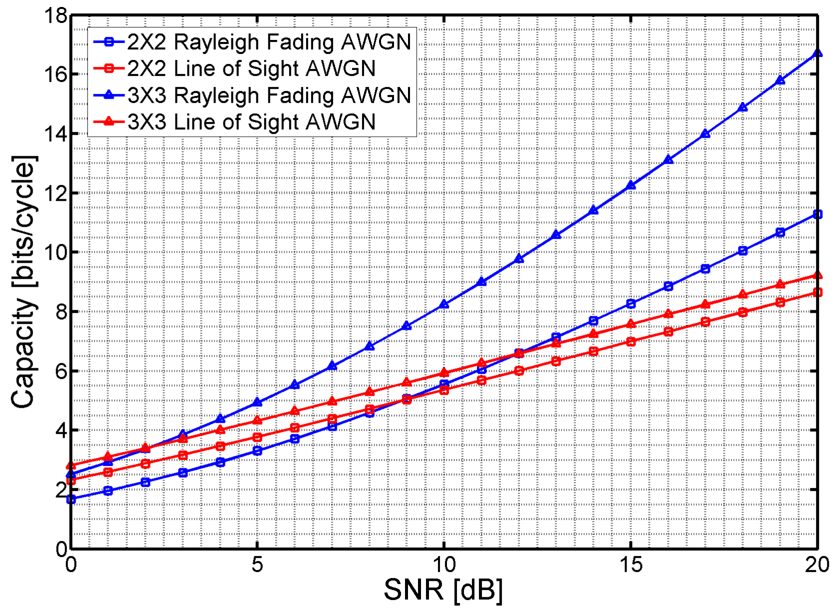


Figure 3.13: Ergodic capacity comparison between a static line of sight and Rayleigh fading channel.

3.3.3.2 Comparison of Line of Sight to Rayleigh Fading Channel.

Another interesting comparison is that of a fixed-link line-of-sight system with no multipath to that of an independently Rayleigh fading channel for omnidirectional transmit and receive antenna elements.

With the intention of making a realistic and fair comparison, it is assumed that the line-of-sight instance has the advantage of transmitter channel knowledge and optimal beam-forming is exploited. Optimal beam-forming essentially transpires to the traditional antenna array synthesis where each transmits the same information in such a manner that the signals add constructively at the receive array. This is essentially panning the transmit and receive array so that their main beams align. In the MIMO case with Rayleigh fading channel, the prior assumption of no transmit channel knowledge is maintained.

A direct line-of-sight MIMO channel contains a single eigenmode and presents no degree-of-freedom gain [23, Section 7.2.3]. Therefore its capacity is equivalent to that of a SISO channel with a power gain derived from the transmit and receive antenna gains.

Figure 3.13 illustrates the astounding capability of MIMO communication systems. The graphs are in fact revealing that the fundamental performance limit of MIMO communication in a fading channel may exceed that of a direct line-of-sight channel with no fading. However, inventing a coding scheme that can deliver the promising MIMO performance is an ongoing and challenging field of research whereas SISO codes running very close to capacity, namely turbo [27] and low density parity check [28] codes, are well established.

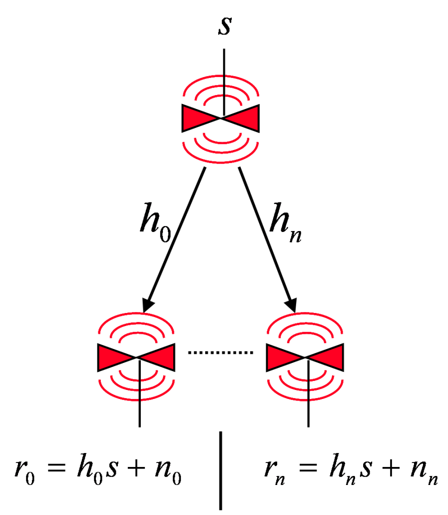


Figure 3.14: Maximum ratio combiner with two receivers

3.4 Maximum Ratio Combiner

It is widely known that channel fading severely decreases the performance of a single transmit single receive communication system. Fading is combated by employing multiple receive elements so that the transmitted data can be reconstructed even though some of the elements may experience a deep fade. When the receive antenna elements are placed more than half a wave length apart the fading experienced by each is typically decorrelated in a rich scattering environment [25]. The signal is detected at each receive element but affected by different channel link coefficients and given by

$$r_0 = h_0 s + n_0 \tag{3.4.1}$$
$$r_n = h_n s + n_n \tag{3.4.2}$$

The maximum likelihood detection of the transmitted signal involves processing analogues to that of a matched filter where the received signals are weighted according to their relative strengths and summed in-phase in order to maximise SNR [23]. The combining rule is given by [29]

$$\tilde{s} = h_0^* r_0 + \ldots + h_n^* r_n \tag{3.4.3}$$
$$= \left(|h_0|^2 + \ldots + |h_n|^2\right) s_0 + h_0^* n_0 + \ldots + h_n^* n_n \tag{3.4.4}$$

with the ML decision rule of

$$s_{0\text{ML}} = \underset{s_i \in \{s_1,\ldots,s_K\}}{\text{arg min}} \left[d^2(r_0, h_0 s_i) + \ldots + d^2(r_n, h_n s_i)\right] \tag{3.4.5}$$
$$= \underset{s_i \in \{s_1,\ldots,s_K\}}{\text{arg min}} \left[\left(|h_0|^2 + \ldots + |h_n|^2 - 1\right) |s_i|^2 + d^2(\tilde{s}, s_i)\right] \tag{3.4.6}$$

where s_i are all the possible transmitted symbols and $d^2(x_1, x_2) = (x_1 - x_2)(x_1^* - x_2^*)$ is the squared Euclidian distance between two complex signals.

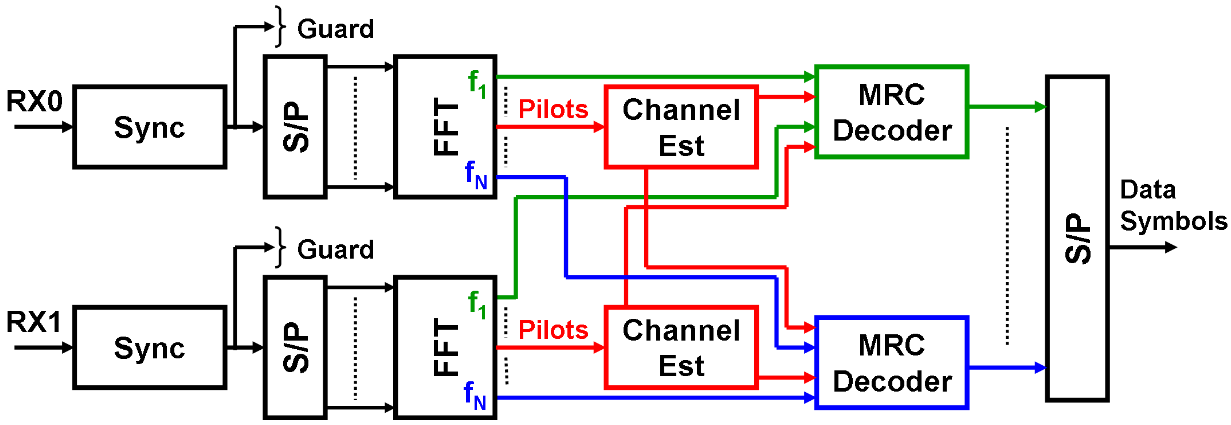


Figure 3.15: Receiver with maximum ratio combining used in conjunction with OFDM

3.5 Maximum Ratio Combiner in OFDM

Spatial maximum ratio combining is easily extended to OFDM. The signals are independently recovered at each receive element in exactly the same manner as for a SISO system described in Section 2.4. The corresponding subcarriers from every receive element contains the same information and MRC (described in Section 3.4) is applied independently to every subcarrier as illustrated in Figure 3.15.

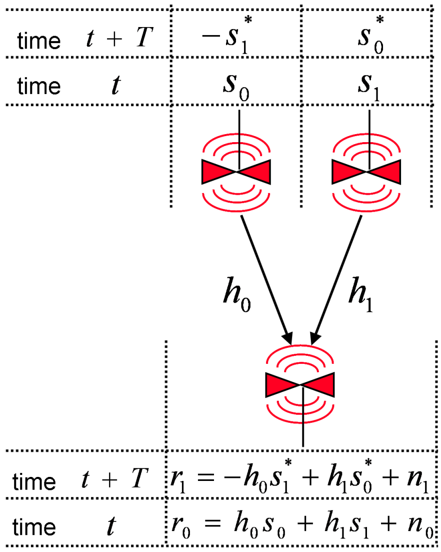


Figure 3.16: Alamouti 2X1 space time block code.

3.6 Alamouti Space-Time Block Code

Alamouti [29] introduced a scheme by which blocks of two complex data symbols of an arbitrary constellation are encoded over space and time. The code was devised to take full advantage of the channel diversity while maintaining orthogonality to facilitate low complexity ML detection at the receiver.

3.6.1 Transmitter Encoding

Data symbols are encoded across two transmit antennas and two time slots. During the first time slot, independent complex symbols are transmitted simultaneously at the same frequency from each antenna while in the second time slot a redundant version of the two symbols are transmitted as described by the encoding matrix:

$$C = \begin{bmatrix} s_0 & s_1 \\ -s_1^* & s_0^* \end{bmatrix} \tag{3.6.1}$$

The scheme encodes blocks of two data symbols over two time slots and across spatially separated antennas and is therefore known as a rate-1[1] space-time block code (STBC) . The encoding is illustrated by Figure 3.16

[1] An average of one information symbol is transmitted per symbol time.

3.6.2 Decoding with One Receive Antenna

Considering the system with one receiver in Figure 3.16, the receiver observes a weighted sum of two symbols in the form

$$\begin{aligned} r_0 &= r(t) = h_0(t)s_0 + h_1(t)s_1 + n_0 && (3.6.2)\\ r_1 &= r(t+T) = -h_0(t+T)s_1^* + h_1(t+T)s_0{*} + n_1 && (3.6.3)\end{aligned}$$

where T is the symbol time and n_0 and n_1 are independent circular symmetric complex white Gaussian noise processes. The channel is assumed to be frequency flat and have a sufficiently large coherence time so that it is virtually static over two symbol periods and hence

$$\begin{aligned} r_0 &= h_0 s_0 + h_1 s_1 + n_0 && (3.6.4)\\ r_1 &= -h_0 s_1^* + h_1 s_0{*} + n_1. && (3.6.5)\end{aligned}$$

Maximum likelihood decoding of the two transmitted symbols is achieved by linear processing due to the fact that the Alamouti code is orthogonal [30] and given by [29]

$$\begin{aligned} \tilde{s_0} &= h_0^* r_0 + h_1 r_1^* \\ &= (|h_0|^2 + |h_1|^2)s_0 + h_0^* n_0 + h_1 n_1^* && (3.6.6)\end{aligned}$$

and

$$\begin{aligned} \tilde{s_1} &= h_1^* r_0 - h_0 r_1^* \\ &= (|h_0|^2 + |h_1|^2)s_1 - h_0 n_1^* + h_1^* n_0 && (3.6.7)\end{aligned}$$

The ML decision rules result in the same as that for a maximum ratio combiner [29]:

$$s_{0\mathrm{ML}} = \underset{s_i \in \{s_1,\ldots,s_K\}}{\text{arg min}} \left[(|h_0|^2 + |h_1|^2 - 1)|s_i|^2 + d^2(\tilde{s_0}, s_i)\right] \quad (3.6.8)$$

and

$$s_{1\mathrm{ML}} = \underset{s_i \in \{s_1,\ldots,s_K\}}{\text{arg min}} \left[(|h_0|^2 + |h_1|^2 - 1)|s_i|^2 + d^2(\tilde{s_1}, s_i)\right] \quad (3.6.9)$$

where $d^2(x_1, x_2) = (x_1 - x_2)(x_1^* - x_2^*)$ is the squared Euclidian distance between two complex signals and s_i all possible transmitted symbols.

Equations (3.6.6) and (3.6.7) indicate that the Alamouti scheme with two transmit- and one receive antennas provides the same diversity order as a maximum-ratio combiner with one transmit and two receive antennas and the symbol recovery requires similar computational complexity.

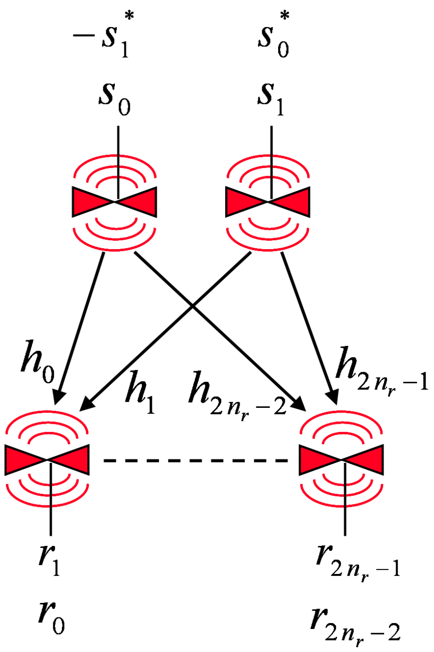


Figure 3.17: Alamouti $2Xn_r$ space time block code.

3.6.3 Decoding with n Receive Antennas

When extending the Alamouti scheme to an arbitrary amount of receive antennas as shown in Figure 3.17, the transmitted symbols are first recovered independently for each receive antenna according to (3.6.6) and (3.6.7) after which maximum ratio combining is applied [29]. The consequent receive decoding takes the form

$$\begin{aligned} \tilde{s_0} &= h_0^* r_0 + h_1 r_1^* + \ldots + h_{2n_r-2}^* r_{2n_r-2} + h_{2n_r-1} r_{2n_r-1}^* \\ &= (|h_0|^2 + |h_1|^2 + \ldots + |h_{2n_r-2}|^2 + |h_{2n_r-1}|^2) s_0 \\ &\quad + h_0^* n_0 + h_1 n_1^* + \ldots + h_{2n_r-2}^* n_{2n_r-2} + h_{2n_r-1} n_{2n_r-1}^* \end{aligned} \tag{3.6.10}$$

and

$$\begin{aligned} \tilde{s_1} &= h_1^* r_0 - h_0 r_1^* + \ldots + h_{2n_r-1}^* r_{2n_r-2} - h_{2n_r-2} r_{2n_r-1}^* \\ &= (|h_0|^2 + |h_1|^2 + \ldots + |h_{2n_r-2}|^2 + |h_{2n_r-1}|^2) s_0 + \\ &\quad - h_0 n_1^* + h_1^* n_0 - \ldots - h_{2n_r-2} n_{2n_r-1}^* + h_{2n_r-1}^* n_{2n_r-2} \end{aligned} \tag{3.6.11}$$

and the ML decision rule is given by

$$\begin{aligned} s_{0\mathrm{ML}} = \underset{s_i \in \{s_1,\ldots,s_K\}}{\text{arg min}} \; & [(|h_0|^2 + |h_1|^2 + \ldots + |h_{2n_r-1}|^2 - 1)\,|s_i|^2 \\ & + d^2(\tilde{s_0}, s_i)] \end{aligned} \tag{3.6.12}$$

and

$$s_{1\mathrm{ML}} = \underset{s_i \in \{s_1,\ldots,s_K\}}{\arg\min} \begin{array}{l} [(|h_0|^2 + |h_1|^2 + \ldots + |h_{2n_r-1}|^2 - 1)\,|s_i|^2 \\ +d^2(\tilde{s_1}, s_i)] \end{array} . \tag{3.6.13}$$

The scheme provides a $n_r \times n_t$ diversity gain and even though some of the air-links in the radio channel may fail due to a deep fade the signal may still be successfully recovered.

3.6.4 Ideal performance

Due to the orthogonal nature of the Alamouti space-time block code, ML decoding is achieved by linear processing at the receiver which effectively decouples the two transmitted symbols [30]. The capacity for uncorrelated channel links is therefore simple to derive [23, 31]:

$$C = \log_2 \left(1 + \frac{\rho}{2}||\mathbf{H}||^2\right) \tag{3.6.14}$$

where $||\mathbf{H}|| = \sum_{k=1}^{n_r}\sum_{l=1}^{n_t} h_{kl}$ is the Frobenius norm of the channel matrix. In a Rayleigh fading channel the capacity becomes

$$C = \log_2 \left(1 + \frac{\rho}{4}\chi^2_{2n_r n_t}\right) . \tag{3.6.15}$$

where χ^2_k is the Chi-square random variable discussed in Section 3.3. The ergodic capacity is plotted in Figure 3.18 by taking the expected value of (3.6.15). The Alamouti scheme provides a diversity gain of $n_t \times n_r$ but no spacial degree-of-freedom gain [23]. This is due to the fact that fully redundant encoding[2] is performed over the transmit antennas. Fading is combated by exploiting the maximum diversity gain the channel has to offer but the scheme does not construct parallel communication links, as illustrated by Figure 3.1(b). The theoretical general channel capacity for a 2X2 system therefore supersedes that of the Alamouti 2X2 scheme as seen in Figure 3.18. Systems which do exploit the spacial degrees of freedom tend to transmit independent information from each antenna, as in the case with BLAST architectures [32, 33], but do not necessarily exploit the maximum channel diversity.

The bit error probability of an Alamouti system using QPSK in an uncorrelated Rayleigh fading channel when the receiver has perfect channel knowledge is deduced from [34, C-18]:

$$P_e = \frac{1}{2}\left[1 - \sqrt{\frac{E_b/N_0}{2 + E_b/N_0}} \sum_{k=0}^{n_t n_r - 1} \binom{2k}{k} \left(\frac{1}{2(1 + E_b/N_0)}\right)^k\right] . \tag{3.6.16}$$

Figure 3.19 clearly illustrates the improved BER performance as the channel diversity increases.

[2]Fully redundant spacial encoding implies that every transmit antenna transmits every data symbol at least once.

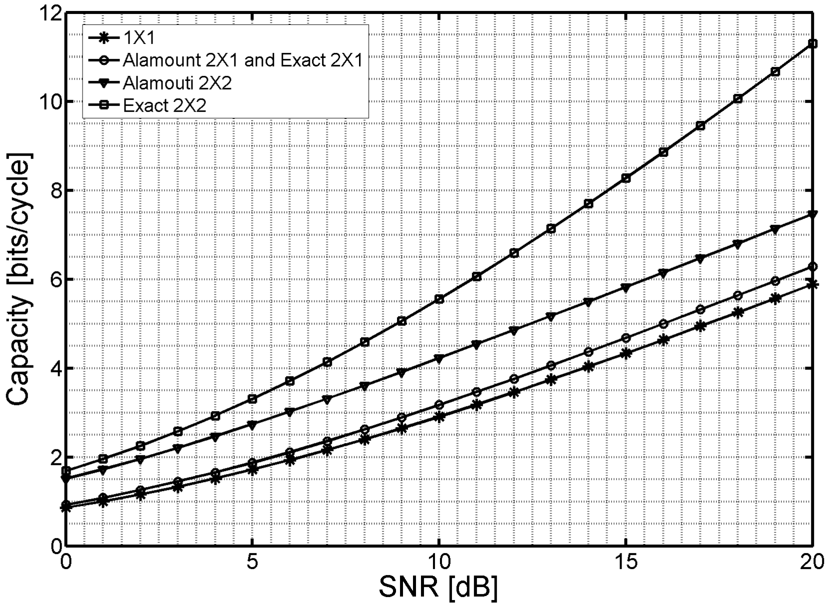


Figure 3.18: Ergodic channel capacity without transmit channel knowledge.

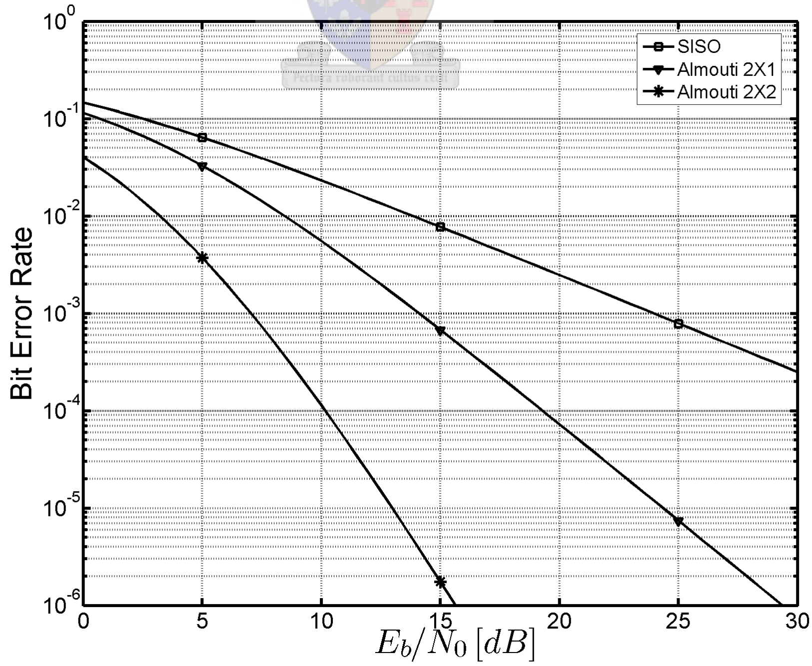


Figure 3.19: Theoretical QPSK bit error rate for an independent Rayleigh fading channel were the receiver has perfect channel knowledge while the total transmit power is conserved.

3.7 Alamouti Space-Time Block Coding and OFDM

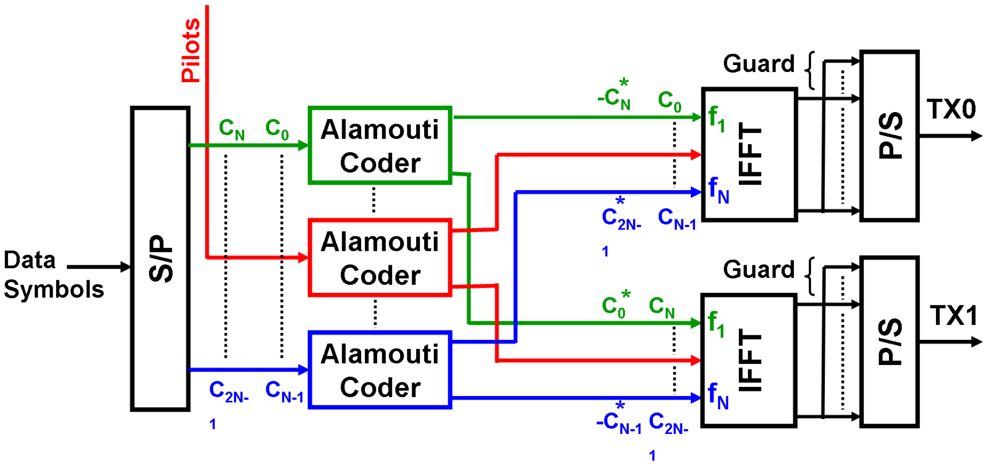


Figure 3.20: Transmitter with Alamouti space-time block code used in conjunction with OFDM.

Alamouti's space-time block code [29] assumes that the channel is frequency-flat which makes the use of OFDM attractive because it produces a wide-band signal with suitable narrow-band properties. Figure 3.20 provides an illustration of how the space-time block code and OFDM are integrated at the transmitter. Each space-time encoder in Figure 3.20 is described by (3.6.1) and acts on only one OFDM subcarrier and thus requires no knowledge of the overall modulation scheme. The receiver on the other hand is more complicated because of the channel estimation required.

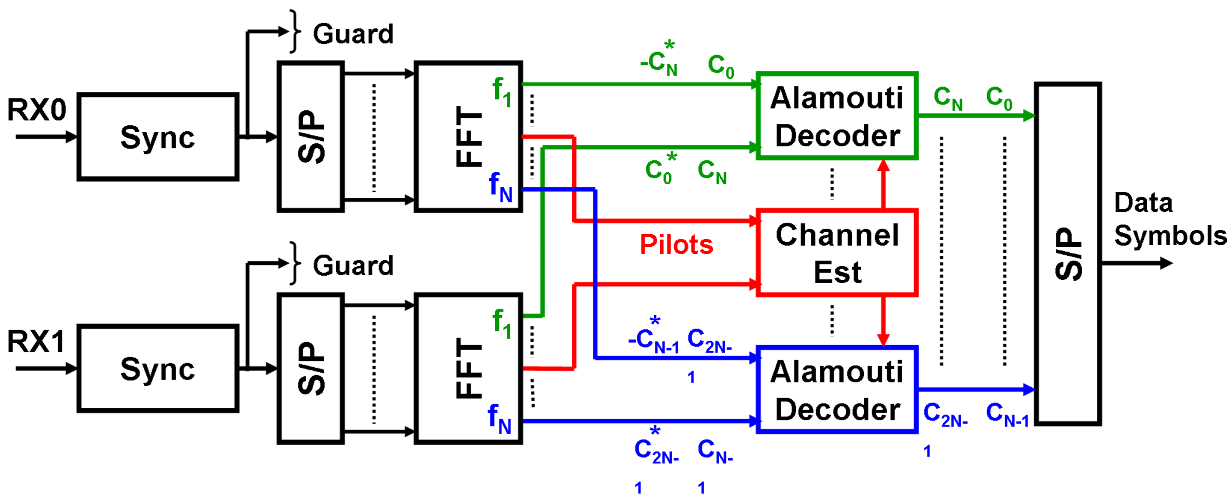


Figure 3.21: Receiver with Alamouti space-time block code used in conjunction with OFDM.

3.7.1 Channel estimation

The problem of channel response estimation needs special attention because with multiple transmit elements the receiver requires a means of distinguishing between pilots or training symbols from different radiating antenna elements. A solution for channel estimation by means of pilots scattered throughout the OFDM symbol is conceptualised in the following manners:

It would be possible to use an *Alternating pilot scheme* similar to that used by WiMax [2]. Different pilot frequency positions are allocated for each transmit antenna while nulls are assigned to the corresponding positions at the other. This ensures that both antennas never transmit pilots at the same frequency simultaneously and the receiver is able to update an estimate of the full channel response after very symbol period. The disadvantage is that in order to attain the same frequency resolution, n_t times more pilot positions (and therefore more bandwidth) are required compared to a system with only one transmit antenna.

Another suggestion stems from the fact that the Alamouti code is orthogonal. It was identified that the pilots could be transmitted simultaneously from the same frequency position while encoding them according to Alamouti's scheme. The same continuous and scattered pilot positions as described in Section 2.3.3 are used. At the transmitter the modulating pilot symbols are passed through Alamouti's space-time block coder as seen in red in Figure 3.20. Thus two symbol periods are required in order to extract any channel information at the receiver.

The functioning of the channel response estimator at the receiver, displayed in red in Figure 3.21, is described as follows: Assume S_{p0} and S_{p1} are the pilot symbols allocated to the same subcarrier at the two transmitters respectively. The channel is assumed quasi-static for two consecutive OFDM symbol periods during which the following signals are received at the pilot subcarrier:

$$r_{p0} = h_0 S_{p0} + h_1 S_{p1} + n_0 \tag{3.7.1}$$

$$r_{p1} = -h_0 S_{p1}^* + h_1 S_{p0}^* + n_1. \tag{3.7.2}$$

The pilot symbols s_{p0} and s_{p1} are known to the receiver and since symbol encoding is orthogonal, it is suspected that the optimal channel estimation would be a result of linear processing as was the case for decoding data symbols. The following linear combining is chosen to produce the channel estimate:

$$\begin{aligned} h_{0\ est} &= \frac{s_{p0}^* r_{p0} - s_{p1} r_{p1}}{|s_{p0}|^2 + |s_{p1}|^2} \\ &= h_0 + \frac{s_{p0}^* n_0 - s_{p1} n_1}{|s_{p0}|^2 + |s_{p1}|^2} \end{aligned} \tag{3.7.3}$$

$$\begin{aligned} h_{1\ est} &= \frac{s_{p1}^* r_{p0} + s_{p0} r_{p1}}{|s_{p0}|^2 + |s_{p1}|^2} \\ &= h_1 + \frac{s_{p1}^* n_0 + s_{p0} n_1}{|s_{p0}|^2 + |s_{p1}|^2}. \end{aligned} \tag{3.7.4}$$

The channel estimates of (3.7.3) and (3.7.4) are proven optimal by showing that

$$
\begin{gathered}
E[\Re(S_{p0}^{*}n_0 - S_{p1}n_1)\Re(S_{p1}^{*}n_0 + S_{p0}n_1)] = 0 \\
\cup \\
E[\Im(S_{p0}^{*}n_0 - S_{p1}n_1)\Im(S_{p1}^{*}n_0 + S_{p0}n_1)] = 0
\end{gathered} \tag{3.7.5}
$$

which implies that the noise added to the two channel estimates are uncorrelated and therefore the one channel estimate contains no information about the other. The conditions (3.7.5) are shown to hold for arbitrary complex pilot symbols in Appendix A.

3.7.2 Synchronisation

The OFDM symbols are radiated in sync from the two Alamouti transmit elements and it is assumed that the same reference oscillators are used for corresponding frequency translation stages. The signal at each receiver therefore has the same autocorrelation properties as for a SISO system. The joint maximum likelihood estimation of frequency and timing offset, which exploits the OFDM guard interval as described in Section 2.4.1, can be directly applied to the Alamouti receive elements as illustrated by the "Sync" block in Figure 3.21.

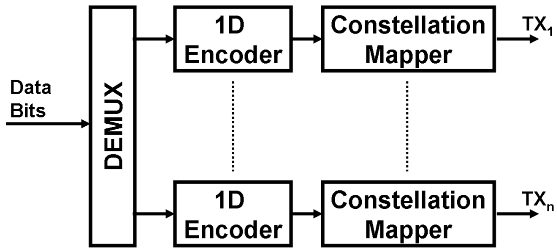


Figure 3.22: V-BLAST transmitter.

3.8 Other MIMO Algorithms

The Alamouti STBC achieves maximum transmit spacial diversity by encoding data symbols in space and time in an orthogonal manner. It was proved by [30] that a rate-1 orthogonal STBC for complex symbols exists only for a system with two transmit elements and is exactly the Alamouti scheme. Orthogonal STBC designs exist for systems with more than two transmit elements but at the price of a lower code rate [30].

Quasi-orthogonal STBCs with larger code rates than their orthogonal counterparts have been demonstrated by [35]. Even though such systems are no longer able to exploit the maximum transmit spatial diversity, they are shown to have superior performance at moderate to low SNR.

The Bell Laboratories Layered Space-Time (BLAST) architectures have recently attracted much interest. The system does not attempt to orthogonalise the transmitted symbols, as is the case with conventional STBCs, but instead relies on a rich multi-path environment to produce a channel matrix with sufficiently decorrelated columns in order to regain the transmitted data. The simplest of the two, known as Vertical BLAST (V-BLAST) [33] , performs no encoding at the transmitter except for conventional 1D encoding which is applied independently to the data at every transmit element as seen in Figure 3.22. Independent data symbol streams of the same constellation are simply transmitted simultaneously at the same carrier frequency from all n_t front-ends to increase the data throughput. The system exploits the available channel degrees of freedom but does not exploit the available transmit diversity because no encoding occurs across antenna elements [23, Section 3.3].

Suboptimal detection mechanisms are discussed in [33] which involves symbol interference cancellation in combination with with a nulling algorithm. Symbol interference cancellation is achieved by successively subtracting already detected code words to eliminate their disturbance while nulling entails algorithms such as minimum mean-squared error or zero-forcing to extract data symbols in the presence of the interference of other unknown symbols. Their investigation focuses on suboptimal detection because an ML detector would have to search through m^{n_t} possible transmit vectors and is this not viable when the amount of transmit antennas are large. Enormous capacities have been reported by Foschini et al. with a V-BLAST QAM16 laboratory prototype with 8 transmitters and 12 receivers.

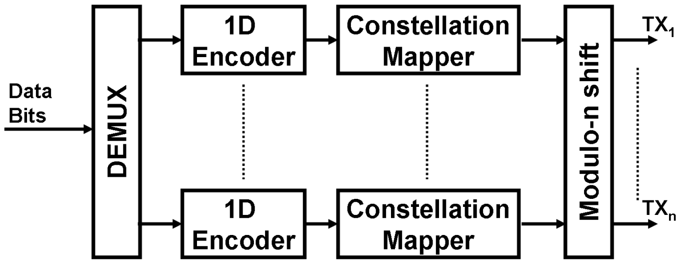


Figure 3.23: D-BLAST transmitter.

An investigation of different V-BLAST detection mechanisms was made by [36] and includes the maximum likelihood case.

A similar scheme is Diagonal BLAST (D-BLAST) [32] which performs staggered interleaving among the transmit antennas so that parts of every code word in a stream is spanned over the transmit antennas. Every stream containing independent data is independently encoded by a conventional coding scheme as is done for V-BLAST. D-BLAST is different from V-BLAST in that the symbols of independent code words are interleaved across the transmit antenna as shown in Figure 3.23. The code words are therefore staggered in diagonals over space and time enabling D-BLAST to take advantage of the available transmit diversity.

Chapter 4

Simulation and Modelling Techniques

With the aim of accurately predicting the performance of a communication system, the adequate modelling of various signals and systems for the purpose of numerical simulation is the focus of this chapter. The discussion commences with the fundamental representation of signals and system models in the simulation domain in Section 4.1. The manner in which noise is represented is attended to in Section 4.2 and the relation between bit energy, noise spectral density and signal-to-noise ratio is discussed in Section 4.3. Section 4.4 is concerned with estimating the error probability of a communication system and explains the confidence of the estimation. Wireless channel modelling, which is vital for evaluating communication systems, is discussed in Section 4.5

4.1 Timed Synchronous Data Flow Simulation

Synchronous data flow simulation refers to a manner in which an entire digital signal processing (DSP) system is modelled by separate components where the simulator handles the flow of data between these components in a computationally efficient manner. The Agilent *Advanced Design System®* Ptolemy simulator [16], a commercial package derived from the Berkeley Ptolemy simulator's [37] synchronous data flow domain, is used extensively for modelling the digital communication systems investigated in this thesis. The simulator implements a synchronous signal processing domain in a modular fashion where custom components are described in the C++ programming language.

A high frequency band-limited signal is represented in the simulator as a carrier modulated by a complex envelope in the time domain. Instead of over sampling the highest frequency it is sufficient to only sample the change in the carrier according to the Nyquist criterion. Thus a signal is entirely represented by sampling the modulating envelope adequately together with knowledge of the carrier frequency.

Simulation capabilities are further enhanced by the integration of the ADS Ptolemy® and the mixed time-frequency domain Circuit Envelope® simulator [38]. The approach performs a time domain procedure in addition to harmonic balance evaluations at every time sample of the envelope [39]. It enables the simulation of high frequency analogue hardware circuitry within a comprehensive communication system at higher computational efficiency than conventional transient SPICE-like simulations.

4.1.1 Complex Envelope Signal Representation

Band-pass RF communication signals are typically of the form:

$$v(t) = a(t)\cos(2\pi f_c t + \theta(t)) \tag{4.1.1}$$

where $a(t)$ and $\theta(t)$ are low-pass information containing signals and f_c the carrier frequency. Expanding (4.1.1):

$$\begin{aligned} v(t) &= a(t)\cos(\theta(t))\cos(2\pi f_c t) - a(t)\sin(\theta(t))\sin(2\pi f_c t) \\ &= \tilde{v}_I(t)\cos(2\pi f_c t) - \tilde{v}_Q(t)\sin(2\pi f_c t) \end{aligned} \tag{4.1.2}$$

with

$$\tilde{v}_I(t) = a(t)\cos(\theta(t))$$

and

$$\tilde{v}_Q(t) = a(t)\sin(\theta(t)).$$

It is evident that any communication signal of the form (4.1.1) can be represented by the sum of two quadrature carriers with low-pass modulating amplitudes $\tilde{v}_I(t)$ and $\tilde{v}_Q(t)$ known as the in-phase and quadrature-phase modulating envelopes.

In the timed synchronous data flow (TSDF) simulation domain signals are represented as a sampled voltage envelope ($\tilde{v}(t)$) in phasor notation. The real band-limited signal, provided the maximum frequency of $\tilde{v}(t)$ does not exceed f_c, is given by

$$\begin{aligned} v(t) &= \Re\left\{(\tilde{v}_I(t) + j\tilde{v}_Q(t))\, e^{j2\pi f_c t}\right\} \\ &= \Re\left\{\tilde{v}(t) e^{j2\pi f_c t}\right\} \end{aligned} \tag{4.1.3}$$

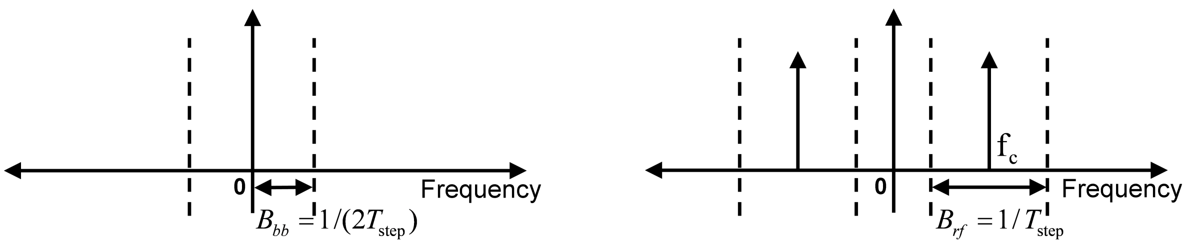
with

$v(t)$	the real band-limited RF signal in the time domain,
$\tilde{v}(t) = \tilde{v}_I(t) + j\tilde{v}_Q(t)$	the low-pass modulating complex envelope and
f_c	the RF characterisation frequency.

In the frequency domain the relation between the low-pass complex modulating envelope and the actual band-limited signal is as follows:

$$
\begin{aligned}
V(f) &= \mathcal{F}\left\{\tilde{v}_I(t)\cos(2\pi f_c t) - \tilde{v}_Q(t)\sin(2\pi f_c t)\right\} \\
&= \tilde{V}_I(f) * \frac{\delta(f+f_c)+\delta(f-f_c)}{2} - \tilde{V}_Q(f) * j\frac{\delta(f+f_c)-\delta(f-f_c)}{2} \\
&= \frac{1}{2}\left(\tilde{V}(f) * \delta(f-f_c) + \tilde{V}^*(f) * \delta(f+f_c)\right) \qquad (4.1.4)
\end{aligned}
$$

where $*$ donates convolution, $\mathcal{F}\{\}$ the Fourier transform and $\delta(f)$ the Dirac delta function.



(a) Baseband signal in the synchronous data flow domain

(b) Modulated RF signal in the synchronous data flow domain

Figure 4.1: Simulation bandwidth illustration in the synchronous data flow domain.

In the simulation domain, the complex envelope $\tilde{v}(t)$ with constant sampling period of T_{step} and results in a maximum unambiguous signal bandwidth which is referred to as the *simulation bandwidth*. Distinction is made between baseband signals where $f_c = 0$ and RF signals where $f_c > \frac{1}{2T_{step}}$. According to the definition of bandwidth depicted by Figures 4.1(a) and 4.1(b), $B_{bb} = \frac{1}{2T_{\text{step}}}$Hz for baseband signals and $B_{rf} = \frac{1}{T_{\text{step}}}$Hz when representing band-limited RF signals. The samples of a baseband signal are real while the sampled envelope of an RF signal may be complex.

It should be noted that f_c in (4.1.3) is not necessarily the signal's *carrier frequency* in a conventional sense but rather the *characterisation frequency*. The *characterisation frequency* indicates the frequency centre of the simulation bandwidth while the *carrier frequency* refers to the frequency centre of the information content.

It is possible to represent an RF signal at a specific carrier f_1 relative to a different characterisation frequency f_c as portrayed in Figure 4.2. The new modulating envelope $\tilde{v}_c(t)$ is derived as follows:

$$
\begin{aligned}
v(t) &= \Re\left\{\tilde{v}_1(t) \times e^{j2\pi f_1 t}\right\} \\
&= \Re\left\{\tilde{v}_1(t)e^{j2\pi(f_1-f_c)t} \times e^{j2\pi f_c t}\right\} \\
&= \Re\left\{\tilde{v}_c(t) \times e^{j2\pi f_c t}\right\} \qquad (4.1.5)
\end{aligned}
$$

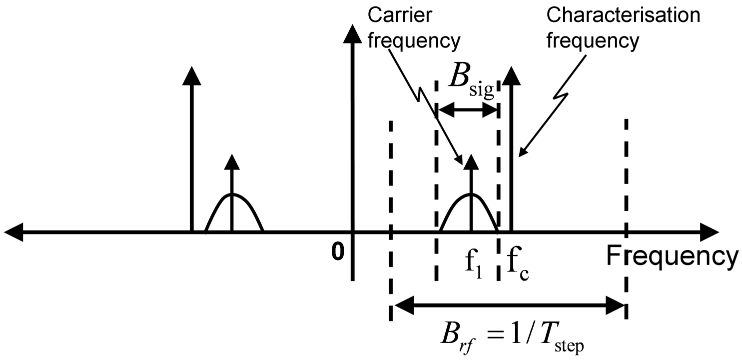


Figure 4.2: Modulated carrier f_1 may be expressed in terms of a different characterisation frequency f_c provided the signal falls within the simulation bandwidth.

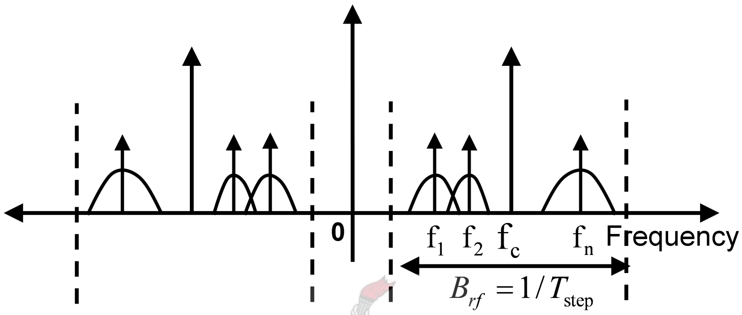


Figure 4.3: Multiple modulated carriers expressed in terms of the same characterisation frequency f_c.

with $\tilde{v}_c(t) = \tilde{v}_1(t)e^{j2\pi(f_1-f_c)t}$, the modulating envelope relative to the frequency f_c. Expressing a signal relative to a different characterisation frequency may be done provided that the Nyquist condition is not violated, leading to $(|f_c - f_1| + \frac{B_{\text{sig}}}{2}) < \frac{1}{2T_{\text{step}}}$ with B_{sig} the RF signal bandwidth and T_{step} the simulation time step.

The sum of multiple modulating envelopes operating on different carriers can be represented by a single modulating envelope referred to a single characterisation frequency as illustrated by Figure 4.3 and given by

$$v(t) = \sum_{n=1}^{N} v_n \times e^{j2\pi(f_n-f_c)t} \times e^{j2\pi f_c t}. \tag{4.1.6}$$

OFDM is in principle the independent modulation of multiple carriers and for simulation purposes the signal is represented by summing the complex envelopes of each carrier expressed relative to the same characterisation frequency.

4.1.2 Complex Envelope System Representation

An RF information carrying signal is typically band-limited and systems acting on the signal are described in the pass-band around a centre frequency. A band-limited signal was described in the previous section in terms of an equivalent low-pass modulating envelope. In a similar manner, a linear system's action on a band-limited RF signal can be described as an equivalent low-pass complex envelope transfer function acting on the signal's modulating envelope.

In order to derive the low-pass transfer function envelope for a linear system with transfer function $h(t)$, consider a band-limited input signal $x(t)$ and output $y(t)$ with the relation:

$$y(t) = h(t) * x(t) \tag{4.1.7}$$

We choose

$$\tilde{H}(f) = H(f + f_c)U(f + f_c) \tag{4.1.8}$$

where $H(f)$ is the transfer function of the actual linear system in the frequency domain and $U()$ the Heaviside step function defined as

$$U(x) = \begin{cases} 0 & , \quad x < 0 \\ 1 & , \quad x > 0. \end{cases} \tag{4.1.9}$$

In the time domain

$$\tilde{h}(t) = \mathcal{F}^{-1}\left\{\tilde{H}(f)\right\} \tag{4.1.10}$$

with $\mathcal{F}^{-1}\{\}$ denoting the inverse Fourier transform.

From (4.1.4) the input signal's low-pass complex envelope spectrum is related to the actual signal spectrum by

$$\tilde{X}(f) = 2X(f + f_c)U(f + f_c). \tag{4.1.11}$$

The negative frequency components of any real signal is the complex conjugate of the positive frequencies. The properties of the Hilbert transform [40, Section 3.4] are utilised to reshape the output signal as to eliminate the redundant negative components and facilitate in the derivation of a system's low-pass equivalent envelope transfer function. The output signal (4.1.7) is becomes

$$\begin{aligned} y(t) &= h(t) * x(t) \\ &= \Re\left\{h(t) * x(t) * \left(1 + \frac{j}{\pi t}\right)\right\}. \end{aligned} \tag{4.1.12}$$

The system response and input signal in (4.1.12) is transformed to the frequency domain revealing that the negative frequency components are redundant:

$$\begin{aligned} y(t) &= \Re\left\{\mathcal{F}^{-1}\left\{H(f)X(f)\left(1 + sgn(f)\right)\right\}\right\} \\ &= \Re\left\{\mathcal{F}^{-1}\left\{2H(f)X(f)U(f)\right\}\right\} \qquad (4.1.13) \\ &= \Re\left\{\mathcal{F}^{-1}\left\{\left[2H(f + f_c)X(f + f_c)U(f + f_c)\right] * \delta(f - f_c)\right\}\right\} \qquad (4.1.14) \end{aligned}$$

Substituting (4.1.11) and the choice made in (4.1.8) into (4.1.14) results in:

$$\begin{aligned} y(t) &= \Re\left\{\mathcal{F}^{-1}\left\{\left[\tilde{H}(f)\tilde{X}(f)\right] * \delta(f - f_c)\right\}\right\} \\ &= \Re\left\{\tilde{h}(t) * \tilde{x}(t)e^{j2\pi f_c t}\right\} \end{aligned} \tag{4.1.15}$$

We can now define $\tilde{y}(t) = \tilde{h}(t) * \tilde{x}(t)$ such that from (4.1.15)

$$y(t) = \Re\left\{\tilde{y}(t)e^{j2\pi f_c t}\right\}. \tag{4.1.16}$$

The preceding derivation demonstrates that $\tilde{H}(f)$ from (4.1.8) is indeed the spectrum of the equivalent low-pass transfer function envelope and is calculated as

$$\tilde{h}(t) = \mathcal{F}^{-1}\left\{H(f)\right\} \tag{4.1.17}$$

$$= \int_{-B_{rf}/2}^{B_{rf}/2} H(f + f_c)U(f + f_c)e^{j2\pi f t}dt. \tag{4.1.18}$$

When a signal and linear system are both described by low-pass complex envelopes and a common characterisation frequency f_c, the response of the system is attained by convolving their complex envelopes. In the discrete simulation domain, this implies that only the low-pass complex envelopes, for both the system and signal, require oversampling and not the actual highest frequency components.

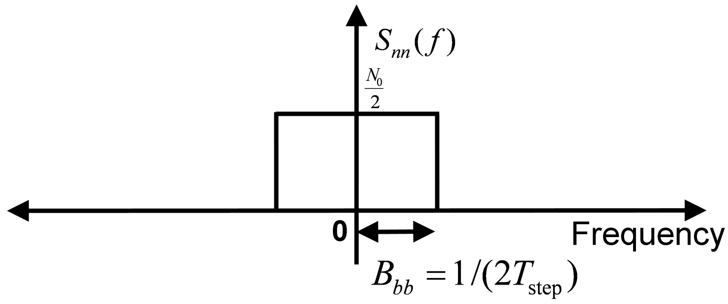


Figure 4.4: Power spectral density of band-limited white noise.

4.2 Simulating Noise

The simulation of additive noise plays a vital role in evaluating the performance of a communication system. The concept of sampled band-limited noise where consecutive samples are independent despite its band-limited nature is introduced in Section 4.2.1 while the representation of band-passed noise as a sampled complex modulating envelope is described in Section 4.2.2.

4.2.1 Discrete Low-Pass White Noise

By the central limit theorem noise in a communication system is expected to be Gaussian distributed and is commonly modelled by a stationary white Gaussian process. The discrete domain inherently implies that a maximum unambiguous bandwidth can be represented and in effect the simulated noise is band-limited. Irrespective of this band-limited nature it is shown that the sampled noise has the desired autocorrelation properties.

Low-pass white Gaussian noise refers to a Gaussian noise process with a constant power spectral density over the bandwidth of interest as given by (4.2.1) and shown in Figure 4.4 with $\frac{N_0}{2}$ the two-sided power spectral density. The corresponding autocorrelation function is of the form (4.2.2) and depicted by the solid line in Figure 4.5.

$$S_{nn}(f) = \begin{cases} \frac{N_0}{2} & , \quad |f| < B_{bb} \\ 0 & , \quad \text{elsewhere} \end{cases} \tag{4.2.1}$$

$$R_{nn}(\tau) = N_0 B_{bb} \frac{\sin(2\pi B_{bb}\tau)}{2\pi B_{bb}\tau} \tag{4.2.2}$$

When the noise is sampled at the Nyquist frequency ($\frac{1}{T_{\text{step}}} = 2B_{bb}$), the discrete sample instants are donated by $\tau = kT_{\text{step}}$ and the autocorrelation function of (4.2.2) becomes

$$R_{nn} = \begin{cases} 1 & , \quad \tau = 0 \\ 0 & , \quad \text{elsewhere} \end{cases} . \tag{4.2.3}$$

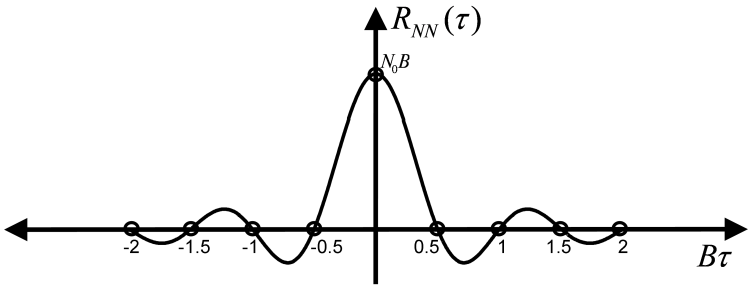


Figure 4.5: Autocorrelation of low-pass white noise.

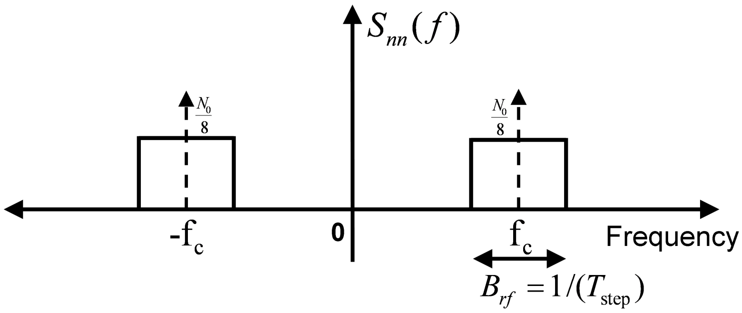


Figure 4.6: Power spectral density of band-passed white noise.

This demonstrates that when low-pass white noise is sampled at the Nyquist frequency the noise samples are independent.

4.2.2 Complex Representation of Band-Limited Noise

For simulation purposes RF signals are represented by a characterisation frequency f_c and its sampled complex modulating envelope, as described in Section 4.1.1. The bandwidth is typically small compared to f_c and therefore in-band noise is similarly represented by a low-pass complex modulating envelope:

$$\tilde{n}(t) = \Re\left\{(\tilde{n}_I(t) + j\tilde{n}_Q(t))\, e^{j2\pi f_c t}\right\} \tag{4.2.4}$$

The low-pass quadrature components $\tilde{n}_I(t)$ and $\tilde{n}_Q(t)$ are related to the actual band-pass noise process $n(t)$ by their power spectral densities, as derived in [8, Section 5.5], as follows:

$$S_{\tilde{n}_I\tilde{n}_I}(f) = S_{\tilde{n}_Q\tilde{n}_Q}(f) = (S_{nn}(f - f_c)U(f - f_c) + S_{nn}(f + f_c)U(f + f_c)) \tag{4.2.5}$$

and

$$S_{\tilde{n}_I\tilde{n}_Q}(f) = j(S_{nn}(f - f_c)U(f - f_c) - S_{nn}(f + f_c)U(f + f_c)). \tag{4.2.6}$$

When the noise $n(t)$ is wide-sense stationary and its power spectral density is symmetrical about $f = f_c$ where $f > 0$, as presented by Figure 4.6, it implies from (4.2.5) and (4.2.6) that $\tilde{n}_I(t)$ and $\tilde{n}_Q(t)$ are identically distributed and uncorrelated. The complex noise envelope $\tilde{n}(t) =$

$\tilde{n}_I(t) + j\tilde{n}_Q(t)$ is said to be *complex circular symmetric*, meaning that $\tilde{n}_c(t)e^{j\theta}$ has the same distribution as $\tilde{n}_c(t)$ for any θ [23, Section 2.2.4].

The communication systems are evaluated in the next chapter in the presence of zero-mean complex circular symmetric Gaussian noise with variance σ^2 and donated by $\mathcal{CN}(0, \sigma^2)$. In the discrete simulation domain, a random number generator models the band-limited white noise by producing the uncorrelated samples of the complex modulating envelope $\tilde{n}(t) = \tilde{n}_I(t) + j\tilde{n}_Q(t)$ where the bandwidth depends on the simulation time step.

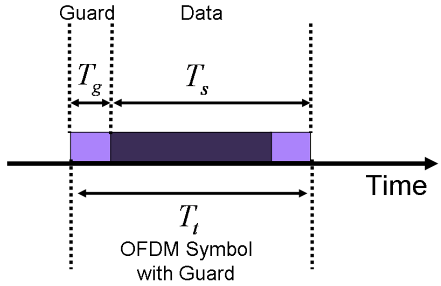


Figure 4.7: OFDM symbol with guard and symbol time.

4.3 Relation Between Bit Energy to Noise Power Spectral Density and Signal-to-Noise Ratio

It is common to present the performance results of a communication system relative to some ratio of signal-to-noise present in the receiver. The signal-to-noise power ratio (SNR) may be used but is not necessarily adequate when direct comparisons of different digital communication protocols are desired. An equivalent approach which is often more useful is to portray the performance relative to the effective energy per bit to noise power spectral density. The relation between E_b/N_0 and SNR is a constant ratio and discussed for an OFDM signal.

The noise power in base-band representation with a power spectral density as defined in Figure 4.4 is given by

$$P_n = N_0 B_{bb}. \tag{4.3.1}$$

In the simulation domain the noise is sampled at the Nyquist frequency and therefore

$$P_n = \frac{N_0}{2T_{\text{step}}}. \tag{4.3.2}$$

The average energy per OFDM symbol including the guard interval for subcarriers that are modulated by an m-ary modulation scheme with constant amplitude is given by

$$E_s = E_b \log_2(m) N_D + E_p N_P + E_G \tag{4.3.3}$$

where E_b is the bit energy used directly in conveying one bit of information and E_p is the energy per pilot subcarrier during the useful symbol time T_s illustrated in Figure 4.7. The number of data and pilot subcarriers are given by N_D and N_P respectively and E_G is the energy in the guard interval.

The pilots are transmitted at a power ratio of R_P compared to the data subcarriers and (4.3.3) becomes

$$E_s = E_b \log_2(m)(N_D + R_P N_P) + E_G. \tag{4.3.4}$$

The cyclic extended guard interval is a copy of the first part of the useful signal and therefore on average

$$E_G = R_G E_b \log_2(m)(N_D + R_P N_P) \tag{4.3.5}$$

with $R_G = \frac{T_g}{T_s}$ the guard interval to useful symbol time ratio. Equation (4.3.4) becomes

$$E_s = E_b \log_2(m)(N_D + R_P N_P)(1 + R_G). \tag{4.3.6}$$

The average signal power is

$$\begin{aligned} P_s &= \frac{E_t}{T_t} \\ &= \frac{E_b \log_2(m)(N_D + R_P N_P)(1 + R_G)}{T_s(1 + R_G)} \\ &= \frac{E_b \log_2(m)(N_D + R_P N_P)}{T_s} \end{aligned} \tag{4.3.7}$$

with T_t the total OFDM symbol time as shown in Figure 4.7. From (4.3.2) and (4.3.7), the relation between SNR and E_b/N_0 is

$$\text{SNR} = \frac{P_s}{P_n} = \frac{\log_2(m)(N_D + R_P N_P)}{T_s B_{bb}} \frac{E_b}{N_0} \tag{4.3.8}$$

or

$$\frac{E_b}{N_0} = \frac{T_s B_{bb}}{\log_2(m)(N_D + R_P N_P)} \text{SNR}. \tag{4.3.9}$$

In the sampled domain, $B_{bb} = \frac{1}{2T_{\text{step}}}$ and $T_s = N_{FFT} T_{\text{step}}$ resulting in

$$\frac{E_b}{N_0} = \frac{N_{FFT}}{2\log_2(m)(N_D + R_P N_P)} \text{SNR}. \tag{4.3.10}$$

It should be noted that the *bit energy* (E_b) in (4.3.9) and (4.3.10) is the actual useful energy that represents one bit of information in an OFDM symbol. To express the bit error rate of a communication system relative to E_b/N_0 does not account for the energy lost in the pilot subcarriers and during the guard interval. A more meaningful approach is to define the average energy per bit as

$$E_{bt} \equiv \lim_{T \to \infty} \frac{\int_0^T P_s(t)dt}{\int_0^T r_b(t)dt} \tag{4.3.11}$$

where $r_b(t)$ is the bit rate. E_{bt} is the total energy (including the energy expended to estimate the channel and that which is present in the guard interval) per bit conveyed. The energy per OFDM symbol remains constant when constant amplitude m-ary modulation is used for the subcarriers and (4.3.11) becomes

$$E_{bt} = \frac{E_s}{N_D \log_2(m)} \tag{4.3.12}$$

$$= (1 + R_P \frac{N_P}{N_D})(1 + R_G)E_b \tag{4.3.13}$$

and the total signal power can be expressed in terms of E_{bt} as

$$P_s = \frac{N_D \log_2(m) E_{bt}}{T_t}. \tag{4.3.14}$$

Using (4.3.1) and (4.3.14), the relation between $\frac{E_{bt}}{N_0}$ and SNR is then given by

$$\frac{E_{bt}}{N_0} = \frac{B_{bb} T_s (1 + R_P)}{N_D \log_2(m)} \text{SNR} \tag{4.3.15}$$

and in the discrete simulation domain

$$\frac{E_{bt}}{N_0} = \frac{N_{FFT}(1 + R_P)}{2 N_D \log_2(m)} \text{SNR}. \tag{4.3.16}$$

4.4 Monte Carlo Bit Error Rate Simulation

The rate at which bit errors occur is inevitably the measure of performance of a digital communication system. The Monte Carlo bit error rate analysis is a technique for estimating the probability of error by a sequence of Bernoulli trials. Pseudo random data is passed through a communication system and after synchronisation and demodulation at the receiver the amount of errors are counted and divided by the number of bits transmitted to determine an estimate of the bit error rate (BER) .

This thesis is concerned with the modelling of a comprehensive communication system, including imperfections, in a simulation environment in order to predict performance by means of the Monte Carlo method.

The behaviour may alternatively be evaluated by analytical methods and are extensively used to verify initial simulation results. As the effects of complex hardware and the radio channel on a sophisticated digital communication system are considered, the analytical evaluation becomes exceedingly complicated and the simulated performance is heavily relied upon.

4.4.1 BER Estimation

The estimated BER

$$r_e = \frac{1}{N}\sum_{n=1}^{N} b_n \tag{4.4.1}$$

is the result of a sequence of Bernoulli trials with N the number of bits observed and b_n the n'th trial in the sequence where $b_n = 1$ when an error occurs and $b_n = 0$ for no error.

The BER expressed as a random variable R_e, is described by the binomial probability mass function [26, Section 2.5]

$$f_{R_e}(r_e|(N,p)) = \frac{1}{N}\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}\delta(r_e-k) \tag{4.4.2}$$

and corresponding distribution

$$F_{R_e}(r_e|(N,p)) = \frac{1}{N}\sum_{k=0}^{N}\binom{N}{k}p^k(1-p)^{N-k}u(r_e-k) \tag{4.4.3}$$

where $p = E[b_n]$, the probability of error. The expected value of the BER estimate

$$E[R_e] = E\left[\frac{1}{N}\sum_{n=1}^{n=N} b_n\right] = p \tag{4.4.4}$$

and therefore the estimator is said to be unbiased. When assuming that errors occur independently, the variance is

$$E[(R_e - E[R_e])^2] = \frac{p(1-p)}{N}. \tag{4.4.5}$$

The variance relative to the squared mean is of special interest as it is an indication of the confidence by which the BER is known.

$$\frac{E[(R_e - E[R_e])^2]}{(E[R_e])^2} = \frac{(1-p)}{pN} \tag{4.4.6}$$

4.4.2 Simulation Confidence

The confidence that a random variable is observed between two values r_{e1} and r_{e2}, with $r_{e2} > r_{e1}$, is defined as

$$\begin{aligned} C_{R_e}(r_{e1}, r_{e2}) \equiv P(r_{e1} < R_e < r_{e2}|(N,p)) &= P(R_e < r_{e2}|(N,p)) - P(R_e < r_{e1}|(N,p)) \\ &= F_{R_e}(r_{e2}|(N,p)) - F_{R_e}(r_{e1}|(N,p)). \end{aligned} \tag{4.4.7}$$

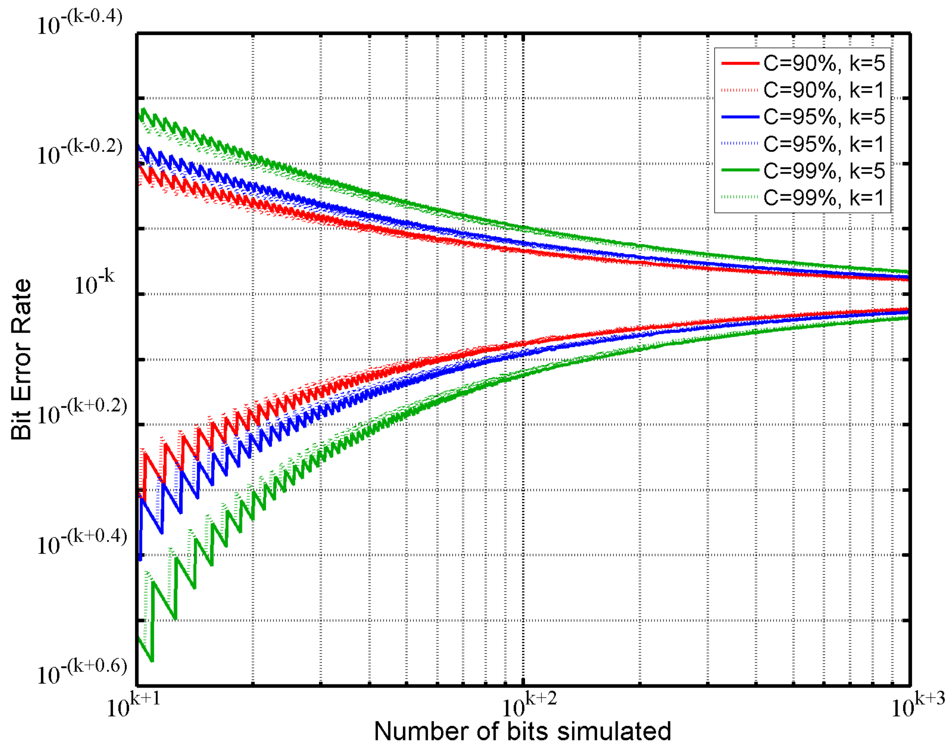


Figure 4.8: BER confidence bands for Monte Carlo analysis for a probability of error $p = 10^{-k}$.

For a given confidence, the limits r_{e1} and r_{e2} are

$$r_{e1} = F^{-1}_{R_e}\left(\frac{C}{2}|(N,p)\right) \tag{4.4.8}$$

$$r_{e2} = F^{-1}_{R_e}\left((1+\frac{C}{2})|(N,p)\right). \tag{4.4.9}$$

where F^{-1} is the inverse binomial distribution. As and example, for a probability of error of $p = 10^{-5}$ with $N = 10^8$ bits simulated results in a 95% certainty that the simulated bit error rate lies between 0.938×10^{-5} and 1.05×10^{-5}.

The reliability of a BER estimation is illustrated by the confidence bands in Figure 4.8.[1] The results illustrate that the confidence by which the BER is known, given that $p < 0.1$ and $Np > 10$, tends to be primarily dependent on the product Np, the number of bits simulated times the probability of error.

Similar confidence bands concurring with that of Figure 4.8 are given by Jeruchim [40, page 501] by approximating (4.4.2) as Gaussian for large N. The curves in Figure 4.8 are for a probability of error $p = 10^{-5}$ and $p = 10^{-1}$ and found to deviate insignificantly which supports Jeruchim's assumption that (4.4.2) tends to Gaussian quite rapidly.

[1]Calculating the binomial distribution directly from (4.4.3) is tedious and tabulated values do not necessarily exist for a large range. Figure 4.8 is derived with the help of Matlab's® *binoinv* function which utilises the natural logarithm of the gamma function approximation as described by [41].

4.4.3 Bit Error Correlation

Channel phenomena and system memory may result in bit errors that are not uncorrelated and would reduce confidence in the BER estimation. As described in [40, page 503] and [42], if bit errors occur in bursts of length $m+1$ without a change in the overall error probability, the relative variance of the BER estimation increases by a factor of approximately $(2m+1)$ resulting in

$$\frac{E[(R_e - E[R_e])^2]}{(E[R_e])^2} \approx \frac{1-p}{pN}(2m+1). \qquad (4.4.10)$$

The implication is that $(2m+1)$ more bits need to be evaluated in order to attain the same relative variance as for the uncorrelated case. Special attention is required for systems where correlated bit errors are expected for example in a slow fading channel.

4.5 Radio Channel Modelling

In order to predict the performance of a communication system we need to successfully reproduce the behaviour of the transmission medium. Extreme variation in the received signal may be expected when transmitting to a mobile that is moving in a rich scattering environment. While the mobile receiver or objects in the channel are moving the signal is altered by a frequency shift due to the Doppler effect and large fades occur as signals add out of phase almost every half wavelength. The received signal may be modelled as a superposition of planewaves arriving with random phase, amplitude and direction of arrival. The mobile movement introduces a Doppler shift in each planewave that depends on the mobile's velocity relative to the planewave's angle of arrival and the carrier frequency.

Channel models derived from that introduced by Jakes [25] have been widely accepted for the evaluation of communication systems [5, 43] and an improved model proposed by Zheng and Xiao [44] is used in this study.

A general description and related terminology of the radio channel are presented in Section 4.5.1 and the channel model is derived in Sections 4.5.2 and 4.5.3.

4.5.1 General Channel Description

The mobile radio channel refers to the interface between transmitter and receiver and may exhibit a wide range of effects on a transmitted signal. A multitude of signal paths may be available, each having random delay, amplitude and frequency shift. In general the channel is unpredictable and changes with time and is thus described statistically. A short intuitive description of common terminology is given while detailed mathematical description can be found in [34].

4.5.1.1 Channel Impulse Response

The channel impulse response refers to the received signal in the time domain after transmitting a infinitely short pulse at the transmitter. In a multipath environment the impulse response is typically time varying and for the purpose of this study assumed to be a discrete train of pulses spread in time.

4.5.1.2 Delay Spread and Coherence Bandwidth

The transmitted signal is spread out in time before arriving at the receiver due multiple signal paths of different lengths. When a short pulse is transmitted, the *delay spread* (T_d) refers to the time it takes for the majority of the transmitted energy to arrive at the receiver relative to the time when the first energy arrives. The delay spread translates to a transfer function in the frequency domain that varies in amplitude and phase as a function of frequency. The channel is thus said to be frequency selective. The *coherence bandwidth* ($B_c \approx \frac{1}{T_d}$) is the frequency range over which correlated fading may be expected. If $B_c \ll B$, where B is the signal bandwidth, the channel is said to be *frequency flat*.

4.5.1.3 Doppler spread and Coherence Time

A frequency shift occurs when the mobile receiver is moving relative to reflectors, scatterers and the transmitter. The signal is "unfolded" in a frequency band referred to as the *Doppler spread* (B_d). The resulting *channel coherence time* ($T_c \approx \frac{1}{B_d}$) is an indication of how rapidly the channel changes. The channel is considered quasi-static when the signal symbol time $T \ll T_c$.

4.5.2 Fading Model Derivation

The mathematical description of a channel model with Doppler shift, as described in [25] and [44], is presented. This channel model is further extended by using multiple uncorrelated faders and a tapped delay line filter as described in Section 4.5.3.

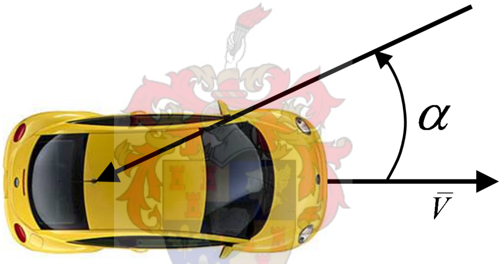


Figure 4.9: Plane wave angle of arrival at a mobile receiver.

It is assumed that the received signal is the sum of planewaves with randomly varying amplitude and phase. Figure 4.9 illustrates a mobile receiver moving at velocity $\bar{V}$ where the n'th planewave is received with a Doppler shift of ω_n defined as

$$\omega_n = \beta v \cos(\alpha_n) \tag{4.5.1}$$

where $\beta = \frac{2\pi}{\lambda}$, α_n is the planewave's angle of arrival and v the mobile's speed. Equation (4.5.1) indicates that the Doppler shift is confined to a maximum of $\pm\beta v$ rad/s.

The signal received from the n'th planewave is given by:

$$E_n(t) = \Re\left\{E_0 c_n e^{j(\omega_c t + \omega_m t \cos\alpha_n + \phi_n)}\right\}. \tag{4.5.2}$$

with

$E_0 c_n$	the amplitude at the receiver,
ω_c	the carrier frequency in rad/s,
$\omega_m = \beta v$	the maximum frequency shift in rad/s due to Doppler and
ϕ_n	the random phase shift of the n'th signal.

For the following derivation the relative time delay of each signal path is ignored except for the phase shift (ϕ_n) it introduces. The time delay will be accounted for by a tapped delay line described in Section 4.5.3.

The resulting received signal is obtained from a linear superposition of N planewaves and is given by:

$$E(t) = E_0 \Re \left\{ T(t) e^{j\omega_c t} \right\} \tag{4.5.3}$$

where

$$T(t) = \sqrt{2} \sum_{n=1}^{N} c_n e^{j(\omega_n t + \phi_n)} \tag{4.5.4}$$

is the complex envelope of the received signal with $\omega_n = \omega_m \cos \alpha_n$.

Equation (4.5.4) is known as Clarke's reference model with statistical properties discussed in [34, 45]. The arrival angles (α_n) and phase shifts (ϕ_n) are both uniformly distributed random variables. If N is large the real and imaginary components of T(t) tend to independent Gaussian random processes and the resulting amplitude is Rayleigh distributed when no line of sight is present.

The next step is to create a model of (4.5.4) which will accurately simulate its statistical properties. Furthermore, the ability to generate multiple fading signals which are uncorrelated is required in order to realise the complete channel model as the sum of independently fading delay spread signals described in Section 4.5.3.

Jakes [25] has proposed a deterministic technique by employing the sum of low frequency sinusoidal oscillators and choosing each oscillator's phase so that multiple uncorrelated fading signals are created.

The Jakes model is adequate for simulating frequency flat fading but not for simulating multiple faders for a channel with delay spread. The technique has come under much scrutiny, from among others [46, 47, 48, 49] and they have shown it to exhibit inadequate performance while proposing improved models. Zheng and Xiao [44] have noted that (4.5.4) can be entirely described by its first- and second-order statistical properties while approximating it as a random complex Gaussian process. They have further indicated that the improved models [46, 47, 48] fail to meet the desired second-order statistical properties. While making use of some of these modifications, Zheng and Xiao [44] have created a further improved model. Their model is still based on the sum of sinusoids concept and is the focus of this discussion.

The normalisation factor c_n in (4.5.4) is chosen so that $c_n^2 = \frac{1}{N}$ resulting in the average received power in (4.5.3) equalling E_0^2. Uniform angles of arrival and quadrantal symmetry is assumed,

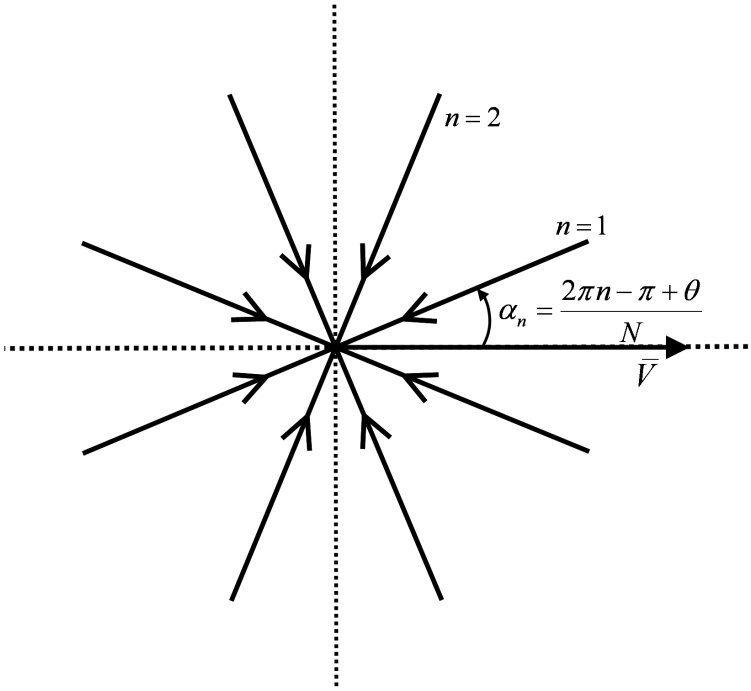


Figure 4.10: Uniform angels of arrival configuration.

which is a good approximation provided N is large [25], and (4.5.4) becomes:

$$T(t) = \sqrt{\frac{2}{N}} \sum_{n=1}^{M} \left\{ e^{j(\omega_n t + \phi_n)} + e^{j(-\omega_n t + \phi_{n+M})} + e^{j(-\omega_n t + \phi_{n+2M})} + e^{j(\omega_n t + \phi_{n+3M})} \right\}. \tag{4.5.5}$$

with $M = \frac{N}{4}$. The four terms in (4.5.5) represent the respective signals received from each quadrant in 2D Cartesian space as illustrated by Figure 4.10. The dotted lines represent the axes of symmetry and the random angle θ in Figure 4.10 and (4.5.7) is necessary for creating multiple uncorrelated faders. While the choice is not unique, [44] proposes that we choose

$$\begin{aligned}
\phi_{n+M} &= -\varphi + \tfrac{\pi}{2} \\
\phi_{n+2M} &= -\phi_n \\
\phi_{n+3M} &= \varphi + \tfrac{\pi}{2} \\
\alpha_n &= \tfrac{(2\pi n - \pi + \theta)}{N}
\end{aligned}$$

where θ, ϕ_n and φ_n are mutually independent, uniformly distributed random variables between $(-\pi$ and $\pi]$.

Equation (4.5.5) is simplified to

$$T(t) = 2\sqrt{\frac{2}{N}} \sum_{n=1}^{M} \cos(\omega_m t \cos(\alpha_n) + \phi_n) + j\cos(\omega_m t \sin(\alpha_n) + \varphi_n). \tag{4.5.6}$$

Multiple uncorrelated Rayleigh faders, with index k, are generated by the equation:

$$T_k(t) = 2\sqrt{\frac{2}{N}}\sum_{n=1}^{M}\cos(\omega_m t\cos(\frac{2\pi n - \pi + \theta_k}{4M}) + \phi_{n,k}) + j\cos(\omega_m t\sin(\frac{2\pi n - \pi + \theta_k}{4M}) + \varphi_{n,k}). \tag{4.5.7}$$

The angles θ_k,$\phi_{n,k}$ and $\varphi_{n,k}$ are mutually independent random variables for all n and k and uniformly distributed between $(-\pi, \pi]$.

It is reported by [44] that good convergence is achieved by summing as little as 8 sinusoids.

4.5.3 Implementation of the Radio Channel Model

In the previous section a derivation of a fading model is given for a frequency flat channel. It is extended to a channel with delay spread by passing the complex envelope of the transmitted signal though a tapped delay line as seen in Figure 4.11. Each tap is multiplied by a gain factor

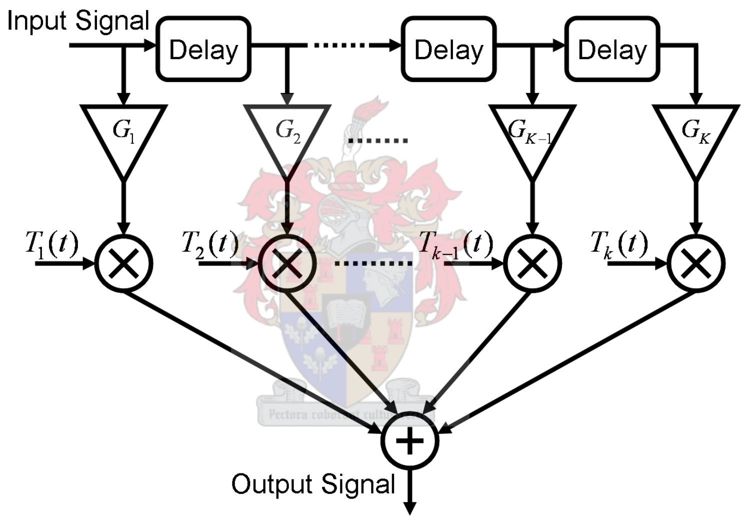


Figure 4.11: Multipath channel model with tapped delay line and modified Jakes' fading generators.

which represents the multipath intensity profile. The Doppler shift and fading characteristics are implemented by means of (4.5.7) which introduces uncorrelated fading at each tap. The use of a tapped delay line has been widely accepted in many standards for evaluating communication systems [1, 43, 50, 51]. These standards recommend typical tapped delay line profiles for different channel scenarios. The specific channel profile used for simulations is described in Section 5.3.1.

Chapter 5

Evaluation of Communication Systems

Modern modulation techniques are becoming exceedingly complex and so does the prediction of their performance. Finding analytic solutions which describe the performance of a communication system, including the effect of non-ideal hardware is extremely challenging, especially when nonlinear components are involved. The process is further complicated by the fact that a vast range of component modelling techniques exists and it becomes impractical to find analytic performance descriptions when several complex component models are employed simultaneously and interchangeably.

In this chapter system performance is evaluated by setting up a comprehensive communication system simulator in software that is able to predict the bit error rate by a Monte Carlo analysis. Figure 5.1 illustrates the concept where random data bits are modulated and passed though hardware and channel models after which the data is demodulated and compared to the originally transmitted data in order to calculate a bit error rate (BER). A complete system implementation, including channel estimation and correction algorithms, is vital for predicting the effects of non-ideal components and to compare different communication schemes.

In a traditional sense the design of RF hardware involves meeting classical figures of merrit, for example spurious response levels for amplifiers or phase noise profiles and frequency drift

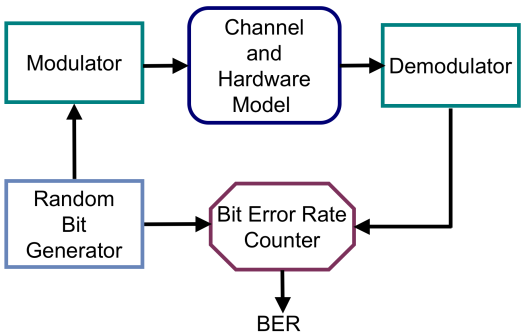


Figure 5.1: Simplified conceptual illustration for the evaluation of a communication system through simulation.

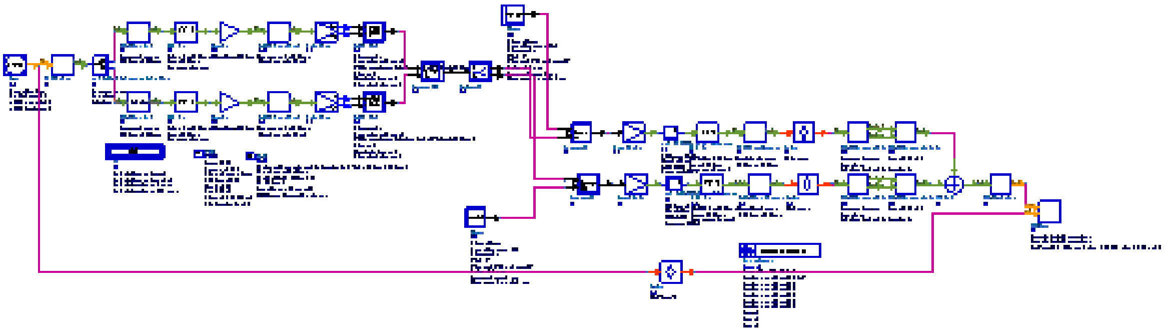


Figure 5.2: ADS schematic with custom communication system components.

Table 5.1: OFDM parameters used in simulation.

OFDM-generating IFFT length (N_{FFT})	2048
Number of subcarriers (N_C)	1705
Number of data subcarriers (N_D)	1529
Number of pilots (N_P)	176
Number of guard interval samples (N_G)	128
Guard to useful symbol time ratio ($R_G = \frac{N_G}{N_{FFT}}$)	$\frac{1}{16}$

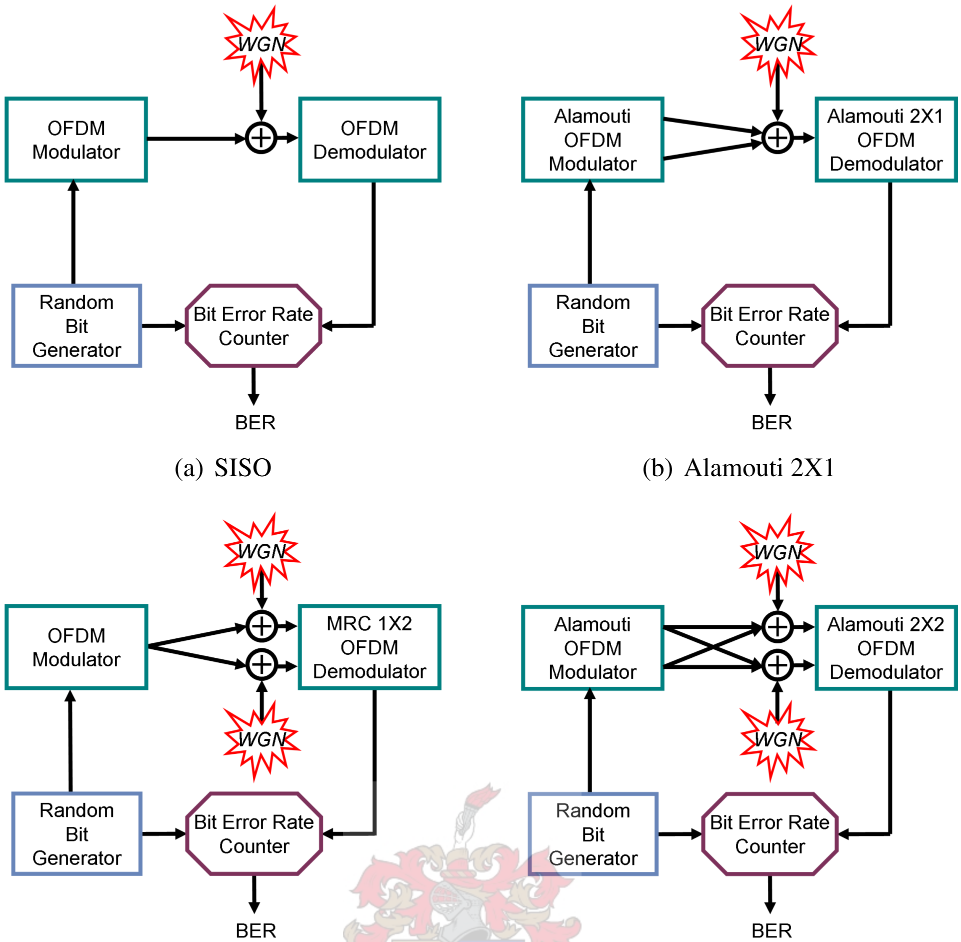
for oscillators. This investigation involves trying to establish the exact effects these non-ideal components have on the bit error rate.

Closed form mathematical solutions serve a very important purpose throughout this investigation in order to verify simulated results. Throughout this work the setups are first simulated under ideal conditions which are directly compared to closed form mathematical solutions in order to verify that the simulators are functioning as expected.

The communication systems evaluation commences by simply considering AWGN with a flat and static channel in Section 5.1. Simulation results are presented in Section 5.2 for an idealised slow fading channel, where the channel is considered static over one OFDM symbol period, and without delay spread. These simplified channel models provide valuable insight as they define the best-case performance to which more realistic channel modelling and estimation algorithms can be compared. In Section 5.3 the channel model is extended to one with more realistic fading properties, including delay spread and Doppler shift. Section 5.4 addresses frequency offset and the estimation thereof while Section 5.5 evaluates the effect of nonlinear amplifiers to provide a guideline as to how far they may be driven into saturation. The performance of the synchronisation and channel estimation techniques, discussed in Sections 2.4.1 and 2.4.3, are also evaluated in all the above mentioned scenarios.

The communication systems' components are implemented in such a manner that the OFDM parameters can be changed effortlessly in a list of variables on the schematic as shown in Figure 5.2. The parameters used in simulations in this chapter are somewhat arbitrarily chosen and given in Table 5.1 The pilot to data subcarrier power ratio (R_P), the amount of data and pilot subcarriers

(N_D and N_P) and the pilot positions are chosen to coincide with the 2k DVB-T [1] standard.



(a) SISO (b) Alamouti 2X1

(c) MRC 1X2 (d) Alamouti 2X2

Figure 5.3: Simulation setups in the presence of white Gaussian noise

5.1 Simulation with a Static Channel and Additive White Gaussian Noise

The communication systems discussed in Chapter 3 are evaluated in the presence of zero mean white Gaussian noise in a static and frequency-flat channel with equal attenuation in the link between every element of the transmit and receive arrays.

Initially, perfect channel knowledge at the receiver is assumed to produce simulated results that match ideal-case theoretical performance predictions. The performance of channel estimation and synchronisation is evaluated subsequently.

5.1.1 Evaluation with AWGN and Perfect Channel Knowledge

The simulation setups for a static frequency-flat channel with uncorrelated AWGN at each receiver are illustrated in Figure 5.3 and it is assumed that the receiver has perfect knowledge of the channel.

The approximate theoretical probability of bit error versus bit energy to noise spectral density

(E_b/N_0) [1] for coherent QPSK demodulation in AWGN, as commonly published [52, 53], is given by

$$p = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{5.1.1}$$

where $\mathrm{erfc}()$ is the complimentary error function defined as

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt.$$

The theoretical BER of (5.1.1) is derived for single carrier QPSK with additive white Gaussian noise. Provided orthogonality is preserved among subcarriers, the BER performance for OFDM depends on the modulation scheme of each subcarrier [52, Section 4.3.2]. The theoretical BER curves are plotted against the average bit energy to noise spectral density (E_b/N_0) per receiving front-end in Figure 5.4. The theoretical plot for a maximum ratio combiner (MRC) compared to a SISO system improves (shifts to the left) by 3dB every time the amount of receivers is doubled. The MRC adds the received signals coherently as explained in Section 3.4, and thus the effective E_b/N_0 increases by 3dB when doubling the amount of receivers while assuming that the noise in each receiving front-end is uncorrelated.

The simulated BER[2] results with perfect channel knowledge (PCK) are presented as markers in Figure 5.4. It should be noted that the plot is for the actual bit energy in the useful OFDM symbol period. When comparisons are made to other modulation schemes the discarded energy in the guard interval would have to be accounted for and the resulting BER graph would shift to the right (deteriorate) by $10\log_{10}\left(1+\frac{T_g}{T_s}\right)$ dB as described in Section 4.3.

The demodulated constellations with AWGN in Figure 5.5 illustrate the effect of multiple receivers employing maximum ratio combining. The received signals are added in-phase providing a signal power increase by a factor of n_r^2 while the noise in different receive paths are uncorrelated and its power effectively increases by a factor of n_r, hence the effective 3dB improvements in E_b/N_0 exhibited in Figure 5.4.

From Figures 5.4 and 5.5 it seems as if the Alamouti schemes provide no improvement over a scheme with one transmitter but it should be noted that the simulation setups contain no channel fading and thus the improvement is simply dependent on the amount of receivers. The advantage of using Alamouti becomes evident in Section 5.2 were the increase in transmit spacial diversity assists in combatting channel fading.

The simulated bit error rates in Figure 5.4 correspond very well to theoretical predictions and are of importance as these simulations verify that the overall OFDM modulation and demodulation process, combined with the Alamouti STBC and MRC, is functioning as expected.

[1]Refer to Section 4.3 for a discussion on the relation between *bit energy to noise spectral density* and *signal-to-noise ratio* for OFDM systems.

[2]The number of bits simulated $> \frac{10^3}{\mathrm{BER}}$.

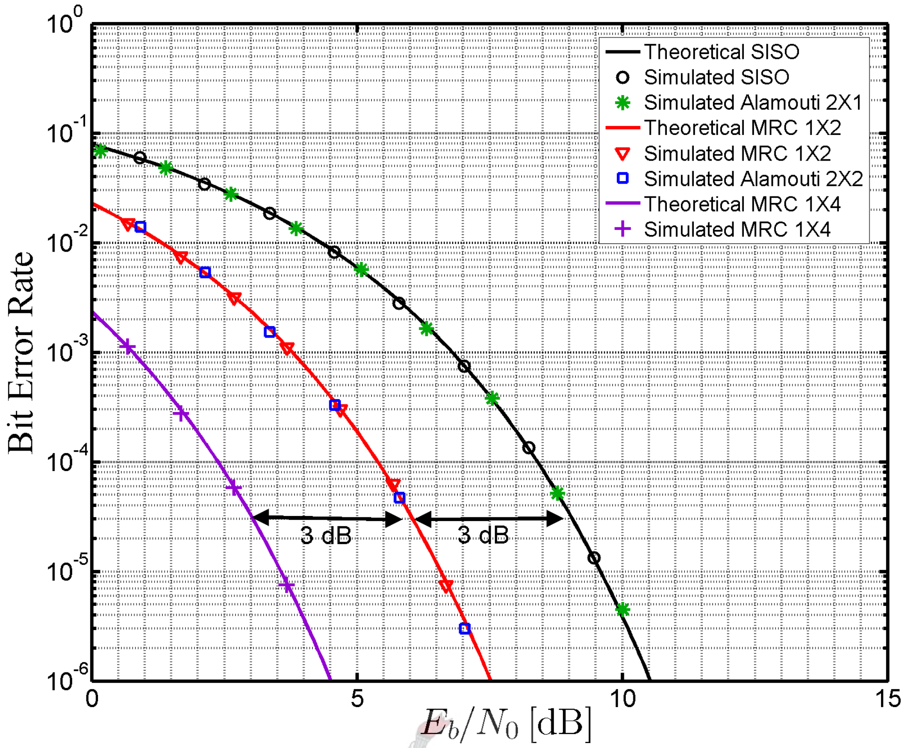
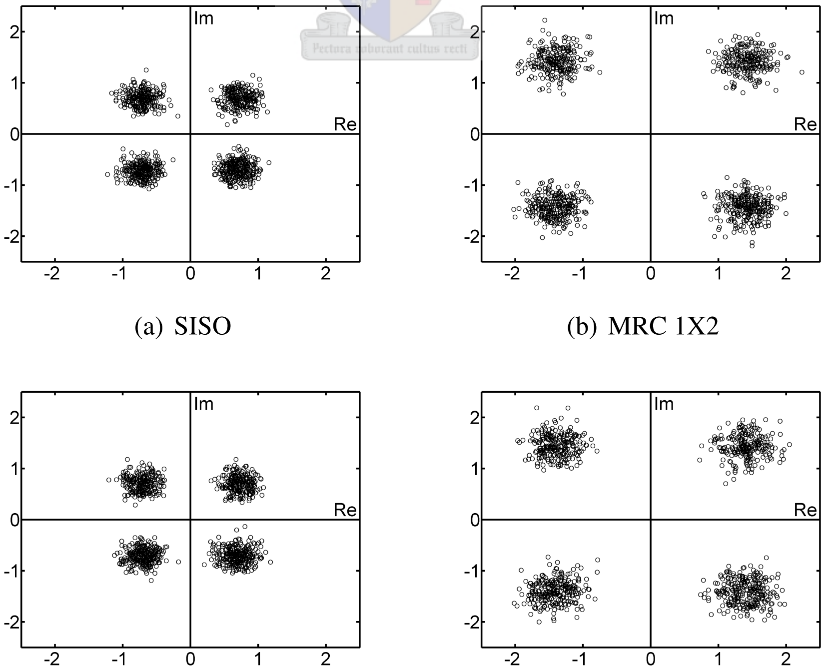


Figure 5.4: BER versus average E_b/N_0 per receiver with additive white Gaussian noise and perfect channel knowledge (PCK).



(a) SISO

(b) MRC 1X2

(c) Alamouti 2X1

(d) Alamouti 2X2

Figure 5.5: Constellation diagrams at 10dB E_b/N_0.

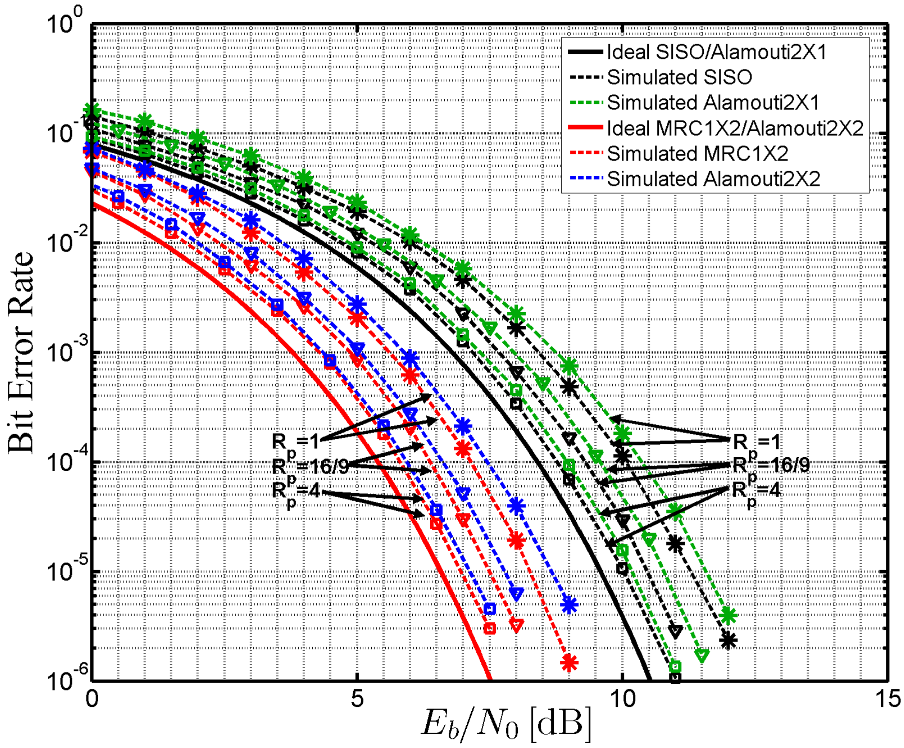


Figure 5.6: Simulation results with continuous channel estimation in AWGN. Alamouti systems transmit encoded pilots simultaneously.

5.1.2 Estimating the Channel Response in AWGN

In practical systems the communication channel is estimated in the presence of noise which results in a BER deterioration compared to that portrayed by Figure 5.4. The performance of communication systems are dependent on the channel estimation and synchronisation algorithms and in this section this dependence is investigated. The channel estimation by pilots scattered among the subcarriers, as described in Section 2.3.3 are evaluated by simulation for different pilot to data subcarrier power ratios. Both the alternating and simultaneous orthogonally encoded pilot schemes for the Alamouti systems, discussed in Section 3.7.1, are considered.

Similar simulation setups are implemented as described in Section 5.1.1 where the channel is frequency-flat and static but with the difference that the channel response is not known to the receiver and is estimated continuously and independently for every OFDM symbol (no averaging occurs over consecutive symbols) by means of the piloting schemes described in Sections 2.3.3 and 5.1.2.2.

5.1.2.1 Pilot to Data Subcarrier Power Ratio

Simulations were conducted for four pilot to data symbol power ratios, namely $R_p = 1, \frac{16}{9}, 3$ and 4. $R_p = \frac{16}{4}$ is the ratio used in DVB-T [1] and DVB-H [11] while $R_p = 3$ minimises the recovered symbol SNR, as explained later, and the other two ratios were chosen to illustrate the general effect of changing the pilot power. The BER deterioration due to estimating the channel in the presence of noise as well as the increase in channel estimation accuracy due to an increase in pilot power are evident from Figure 5.6. The smallest deterioration is observed for the SISO case as only the channel phase information is of importance for SISO-QPSK symbol correction. The MRC 1X2 system experiences slightly more degradation while the Alamouti schemes deteriorate the most.

In Figure 5.6 the BER curves are plotted against E_b/N_0 and they are not adequate for a fair comparison of the performance of the different pilot power ratios. It would be logical to expect that the total transmit power is constant and therefore the power allocated to the pilots is taken away from that which could have been allocated to the data. A fair comparison is made by plotting the BER against E_{bt}/N_0 which considers the total average energy per bit conveyed as discussed in Section 4.3.

Some optimal power distribution between the data and pilot subcarriers is expected and is derived in an approximate manner for a SISO OFDM system in Appendix B as the following:

$$\begin{aligned} R_P &= \sqrt{\frac{N_D}{N_P}} \qquad (5.1.2) \\ &= \sqrt{\frac{1529}{176}} \\ &= 2.95 \end{aligned}$$

The simulation results for the various pilot powers are compared in Figures 5.7 and 5.8. The DVB-T [1] standard uses a pilot power ratio of $R_P = 1.778$ while the calculation above suggests $R_P = 2.95$. The results from Figures 5.7 and 5.8 indicate that similar performance is attained for the ratios. When the R_P becomes too small or too large the performance starts deteriorating more rapidly as indicated by the ratios $R_P = 1$ and $R_P = 4$.

The estimation of a static channel may be improved by simply averaging over consecutive estimations. In a multipath and rapidly changing mobile environment, evaluated in Section 5.3, the optimum channel estimation by means of averaging over consecutive symbols becomes more involved especially because of the inherently long OFDM symbol time. This work focuses on estimating the channel independently for every OFDM symbol.

5.1.2.2 Alternative Alamouti Piloting Schemes

In Section 3.7.1 two piloting schemes are proposed for estimating the channel response for Alamouti systems. One suggestion is to use the *Alternating pilot scheme* similar to that used by WiMax [2]. The same amount of pilot subcarrier positions are allocated as for the SISO system but pilot transmission from a specific subcarrier is alternated between the transmitting antennas. At any given time the active pilot subcarrier of one transmit antenna corresponds to a null on the other antenna. The concept is described by Figure 5.9.

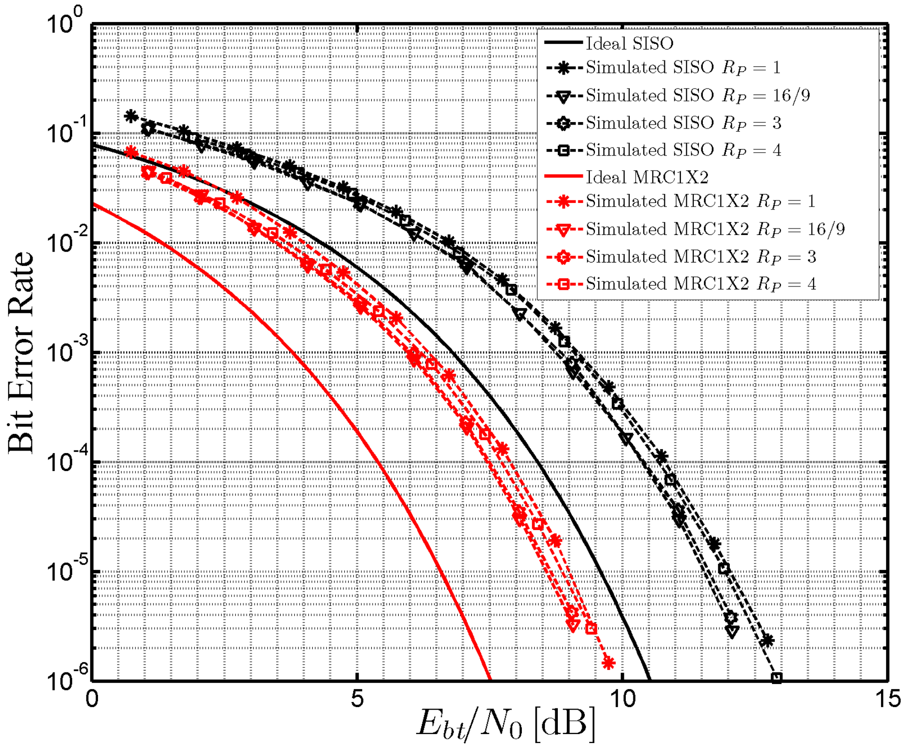


Figure 5.7: Simulation results with continuous channel estimation in AWGN for different pilot to data subcarrier power ratios and a guard to useful symbol time ratio $R_G = \frac{1}{16}$.

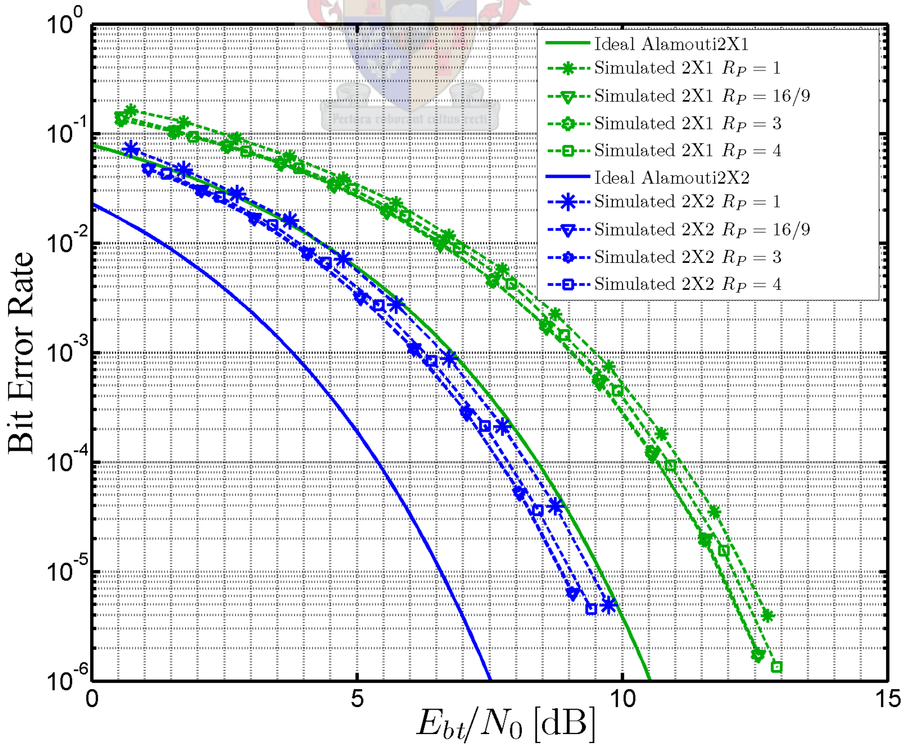


Figure 5.8: Simulation results for Alamouti systems with continuous channel estimation in AWGN for different pilot to data subcarrier power ratios and a guard to useful symbol time ratio $R_G = \frac{1}{16}$. Orthogonal pilots are transmitted simultaneously.

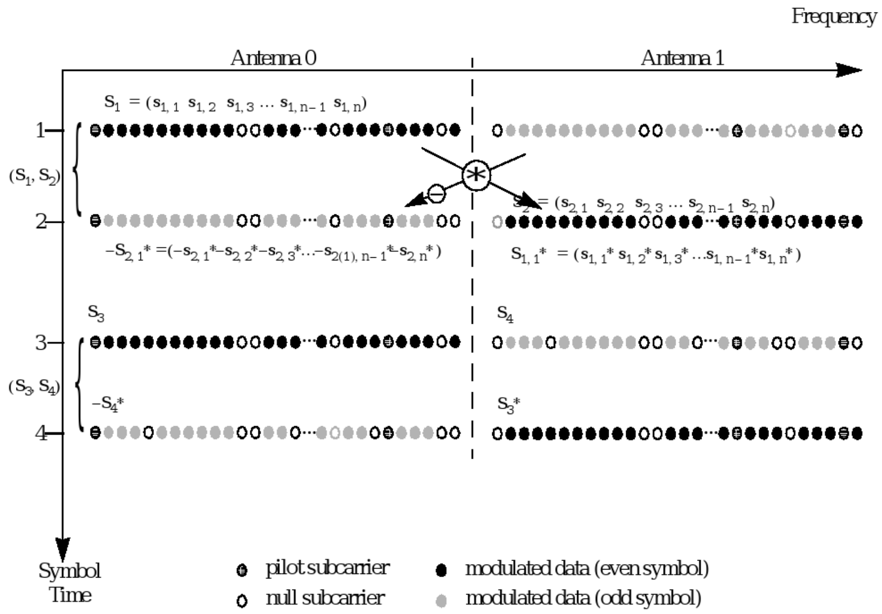


Figure 5.9: Pilot during Space-Time Coding in WiMax (reproduced from [2]).

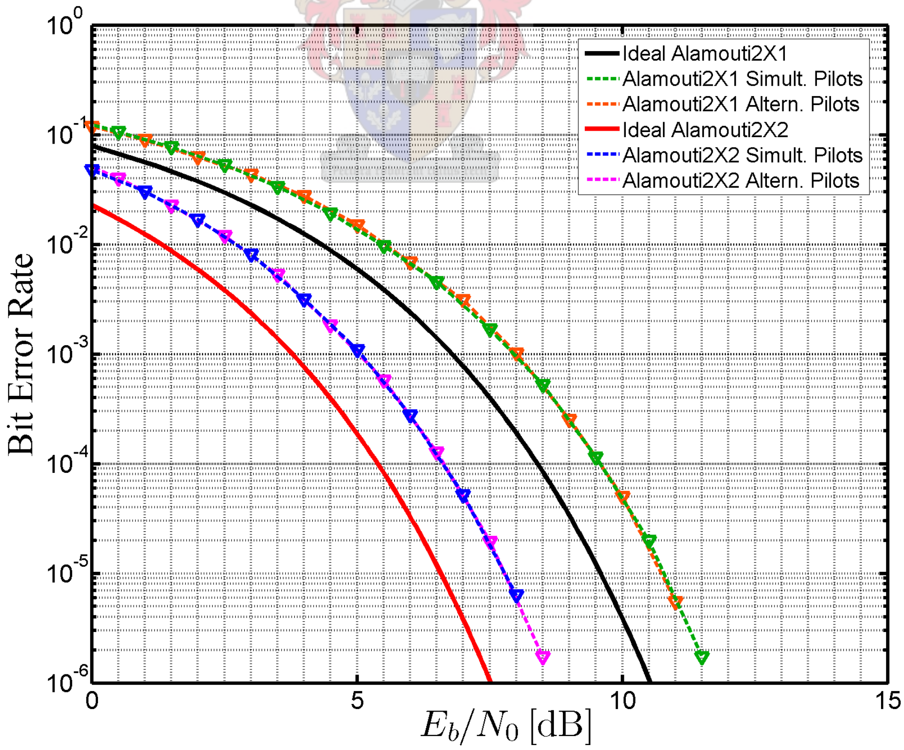


Figure 5.10: Simulation results with channel estimation for simultaneous and alternating pilots in AWGN. $R_p = \frac{16}{9}$.

The other is the *Alamouti encoded pilot scheme* which uses the same amount of pilot subcarriers as the SISO system but encodes the pilots orthogonally according to the Alamouti STBC and transmits them simultaneously from the same subcarrier positions for both antennas. All channel links are estimated simultaneously after observing two related symbol periods.

Alternating the pilots at the two transmit front-ends for every OFDM symbol provides a simple means of estimating the entire channel after every two symbols. Half of the channel is estimated independently during the first symbol period and the other half during the second in contrast to the simultaneous piloting scheme where the entire channel is estimated after observing two corresponding symbol periods. The amount of pilot subcarriers allocated in the OFDM structure is fixed and in order to conserve the total transmitted pilot power the alternating scheme must transmit its active pilots at twice the power while the corresponding pilot at the other antenna is nulled.

The simulation results of the two schemes are compared in Figure 5.10 and as predicted in Section 3.7.1, they have identical performance.

5.1.3 Timing and Frequency Synchronisation Implementation Loss

The synchronisation technique [14] discussed in Section 2.4.1 which exploits the redundancy of the cyclic extended guard interval in order to correct for timing and frequency offset is assessed in a frequency-flat static channel while the algorithm's effectiveness in a delay spread Doppler-fading channel will be investigated in Section 5.4. Due to the presence of noise it is expected that the offset estimation will produce small synchronisation errors even though no actual offset exists. Simulations are conducted in order to determine whether these errors deteriorate the overall BER. The problem that the ML synchronisation algorithm does not know the signal-to-noise ratio at the receiver is also addressed and a critical facet of timing synchronisation for OFDM Alamouti systems is revealed. A small difference in the timing estimation error between two corresponding Alamouti symbol periods is shown to result in substantial break-down while a simple scheme is proposed to combat it.

The channel frequency response is continuously estimated by making use of the simultaneous pilot scheme, described in Section 2.3.3 with a pilot power ratio $R_P = \frac{16}{9}$. No averaging of timing and frequency synchronisation nor of the channel response estimation is performed over consecutive OFDM symbols.

Figure 5.11 provides simulation results which clearly illustrate that the loss due to implementing the synchronisation algorithm may be considered negligible for all systems, provided they have exact knowledge of the SNR at the receiver (or $\rho = \frac{\text{SNR}}{\text{SNR}+1}$). In a real communication system this may only be possible in an isolated channel when the receiver hardware is designed for a specific SNR. The SNR may also be estimated by the receiver by exploiting the autocorrelation properties of the cyclic extension over several OFDM symbols to improve accuracy or by making use

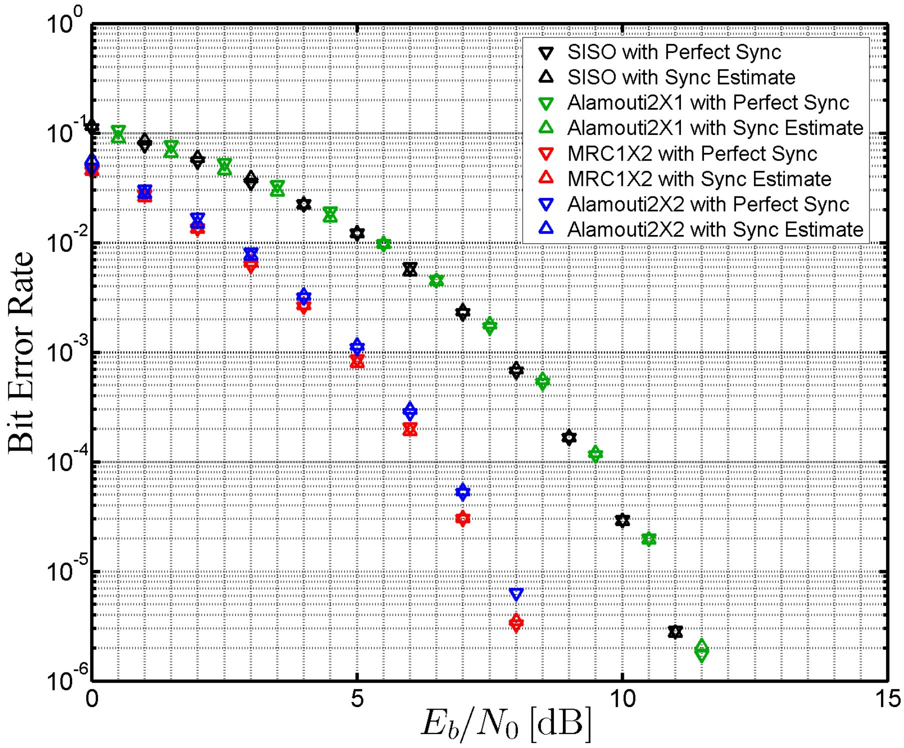


Figure 5.11: Simulation results with frequency and timing offset estimation algorithm with known SNR compared to perfect synchronisation. The channel is flat and static with AWGN while no frequency offset is present.

of training symbols. Neither are desired as in a mobile environment the SNR may vary rapidly from symbol to symbol due to the characteristically long OFDM symbol period. Training symbols would further result in a loss in data rate.

Another option is to set ρ in equation (2.4.13) to a fixed value and determine whether knowing it exactly affects system performance. The estimator would be less optimal but if the degradation is minimal this approach would be preferred as it requires no extra computation. Figure 5.12 illustrates the robustness of the ML estimation when the SNR is assumed to be an arbitrary value of 5dB irrespective of the true SNR.

The loss caused by this practically implementable timing and frequency offset estimation in AWGN is considered negligible.

5.1.3.1 The Effect of Timing Offset Estimation Errors

A timing synchronisation error causes inter-symbol interference as well as a phase shift to every OFDM subcarrier. This section focuses on the effect of the phase shift while the inter-symbol interference is considered negligible as was illustrated by the simulation results in Figure 5.12. It will be demonstrated that a SISO or MRC system's piloted channel response estimation corrects for this phase error as if it was caused by the channel while special care has to be taken for the

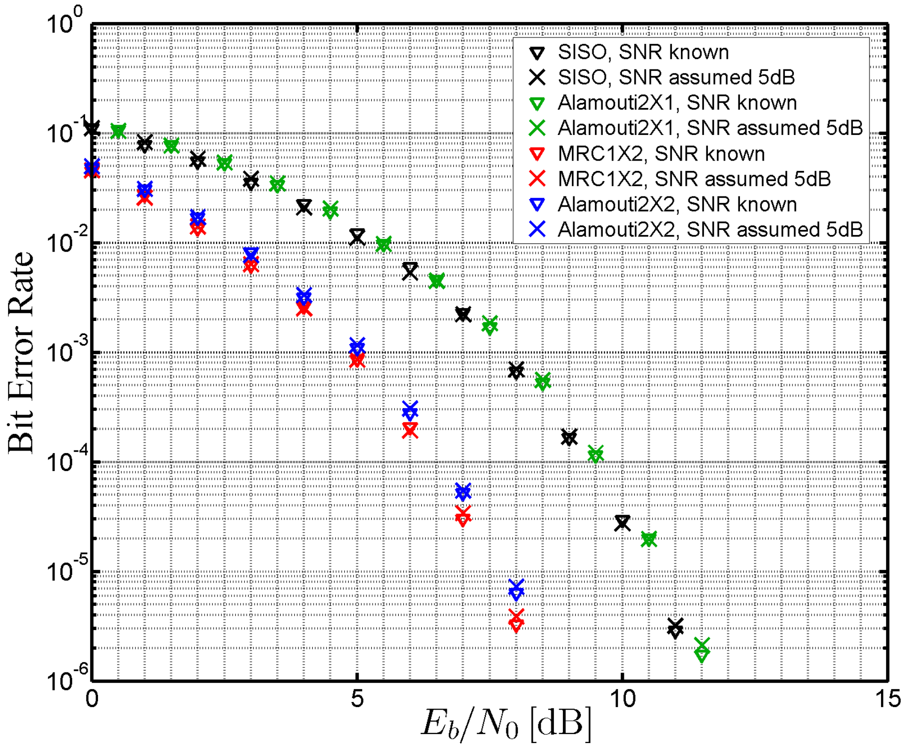


Figure 5.12: Simulation results with frequency and timing offset estimation while assuming SNR=5dB compared to perfect Synchronisation. The channel is flat and static with AWGN while no frequency is present.

Alamouti case.

A baseband timing error in an OFDM system causes a phase shift to every subcarrier depending on the amount of cycles the corresponding subcarrier makes in one FFT period. The phase shift is given by

$$\psi_i = 2\pi \frac{n_e i}{N} \text{ rad} \tag{5.1.3}$$

where n_e is the timing offset error in samples and i is a subcarrier index corresponding to the amount of cycles a specific baseband subcarrier makes in one FFT period of N samples. The phase shift (5.1.3) due to a timing error is thus a linear progression over the frequency band. The channel response is estimated by means of pilot symbols scattered among the subcarriers while a linear extrapolation of the phase is made in order to estimate the channel frequency response at data carrying subcarriers as described in Section 2.3.3.

Phase Error due to a Timing Error in SISO Eystems

For a noiseless narrow-band SISO system the received symbol from any given OFDM subcarrier is

$$r_i = h_i s_i \tag{5.1.4}$$

where h_i is the narrow-band complex channel coefficient and s_i the transmitted symbol modulating the i'th subcarrier. When a timing error occurs and ignoring inter-symbol interference the received symbol becomes

$$r_i = h_i s_i e^{j\psi_i} \tag{5.1.5}$$

where ψ_i is a phase shift due to the timing offset and is given by (5.1.3). The transmitted symbol is recovered by multiplying r_i by the conjugate of the channel coefficient estimate ($\tilde{h}_i$):

$$\tilde{s}_i = \tilde{h}_i^* r_i. \tag{5.1.6}$$

The pilots used to determine $\tilde{h}_i$ occur in the same OFDM symbol as the data s_i and therefore the channel estimate also contains the phase shift ψ_i and the channel estimate is given as

$$\tilde{h}_i = h_i e^{j\psi_i}. \tag{5.1.7}$$

Substituting (5.1.7) and (5.1.5) into (5.1.6), the symbol is recovered as

$$\begin{aligned} \tilde{s}_i &= (h_i e^{j\psi_i})^* \times h_i s_i e^{j\psi_i} \\ &= |h_i|^2 s_i \end{aligned} \tag{5.1.8}$$

The phase shift due to the timing error is corrected for by the piloting scheme at the receiver as if it had occurred in the channel while the receiver is unable to tell the difference as shown by the equation above. In the SISO case the piloting channel estimation scheme corrects the phase error due to a timing offset error superbly while the implication for the Alamouti scheme is subtly different.

Phase Error due to a Timing Error in Alamouti Systems with Alternating Pilots

As described in Section 3.6 the Alamouti space-time block code encodes two independent symbols in space (over two antennas) and time (over two consecutive symbol periods). In the simplest case and ignoring noise to simplify matters, the weighted sum of two symbols are received during the first symbol period:

$$r_{0i} = h_{0i} s_{0i} + h_{1i} s_{1i}. \tag{5.1.9}$$

If a difference in the timing error occurs during the second Alamouti period a phase rotation ψ_i, given by (5.1.3), occurs at subcarrier i resulting in

$$r_{1i} = (-h_{0i} s_{1i}^* + h_{1i} s_{0i}^*) e^{-j\psi_i}. \tag{5.1.10}$$

If the channel is estimated by alternating pilots and h_{0i} is estimated during the first period then

$$\tilde{h_{0i}} = h_{0i} \tag{5.1.11}$$

while h_{1i} is estimated during the second period resulting in

$$\tilde{h_{1i}} = h_{1i}e^{j\psi_i}. \tag{5.1.12}$$

The symbol recovery (3.6.6) and (3.6.7) is done as follows:

$$\begin{aligned}
\tilde{s_{0i}} &= \tilde{h}_{0i}^* r_0 + \tilde{h}_{1i} r_{1i}^* \\
&= h_{0i}^*(h_{0i}s_{0i} + h_{1i}s_{1i}) + h_{1i}e^{j\psi}(-h_{0i}^* s_{1i}e^{-j\psi} + h_{1i}^* s_{0i}e^{-j\psi}) \\
&= (|h_{0i}|^2 + |h_{1i}|^2)s_{0i}
\end{aligned} \tag{5.1.13}$$

and

$$\begin{aligned}
\tilde{s_{1i}} &= \tilde{h}_{1i}^* r_{0i} - \tilde{h}_{0i} r_{1i}^* \\
&= h_{1i}^* e^{-j\psi}(h_{0i}s_{0i} + h_{1i}s_{1i}) + h_{0i}(-h_{0i}^* s_{1i}e^{-j\psi} + h_{1i}^* s_{0i}e^{-j\psi}) \\
&= (|h_{1i}|^2 + |h_{0i}|^2)s_{1i}e^{-j\psi_i} \\
&= (|h_{1i}|^2 + |h_{0i}|^2)s_{1i}e^{-j2\pi n_e i/N}.
\end{aligned} \tag{5.1.14}$$

Equation (5.1.14) indicates that a change as small as one sample in the timing offset estimation error from the first to second Alamouti symbol periods introduces a catastrophic phase error which is unknown to the receiver and is not corrected for by the channel estimation.

Phase Error due to a Timing Offset Error for Alamouti Systems with Simultaneous Orthogonal Pilots

The channel may also be estimated by orthogonal simultaneous pilots as described in Section 3.7.1. With a change in timing offset error, the channel estimation is described by (3.7.3) and (3.7.4) becomes

$$\begin{aligned}
\tilde{h}_{0i} &= \frac{s_{p0i}^* r_{p0i} - s_{p1i} r_{p1i}}{|s_{p0i}|^2 + |s_{p1i}|^2} \\
&= h_{0i}\frac{(|s_{p0i}|^2 + |s_{p1i}|^2 e^{j\psi_i})}{|s_{p0i}|^2 + |s_{p1i}|^2} - h_{1i}\frac{s_{p0i}^* s_{p1i}(1 - e^{j\psi_i})}{|s_{p0i}|^2 + |s_{p1i}|^2}
\end{aligned} \tag{5.1.15}$$

$$\begin{aligned}
\tilde{h}_{1i} &= \frac{s_{p1i}^* r_{p0i} + s_{p0i} r_{p1i}}{|s_{p0i}|^2 + |s_{p1i}|^2} \\
&= h_{1i}\frac{(|s_{p0i}|^2 e^{j\psi_i} + |s_{p1i}|^2)}{|s_{p0i}|^2 + |s_{p1i}|^2} - h_{0i}\frac{s_{p0i} s_{p1i}^*(1 - e^{j\psi_i})}{|s_{p0i}|^2 + |s_{p1i}|^2}.
\end{aligned} \tag{5.1.16}$$

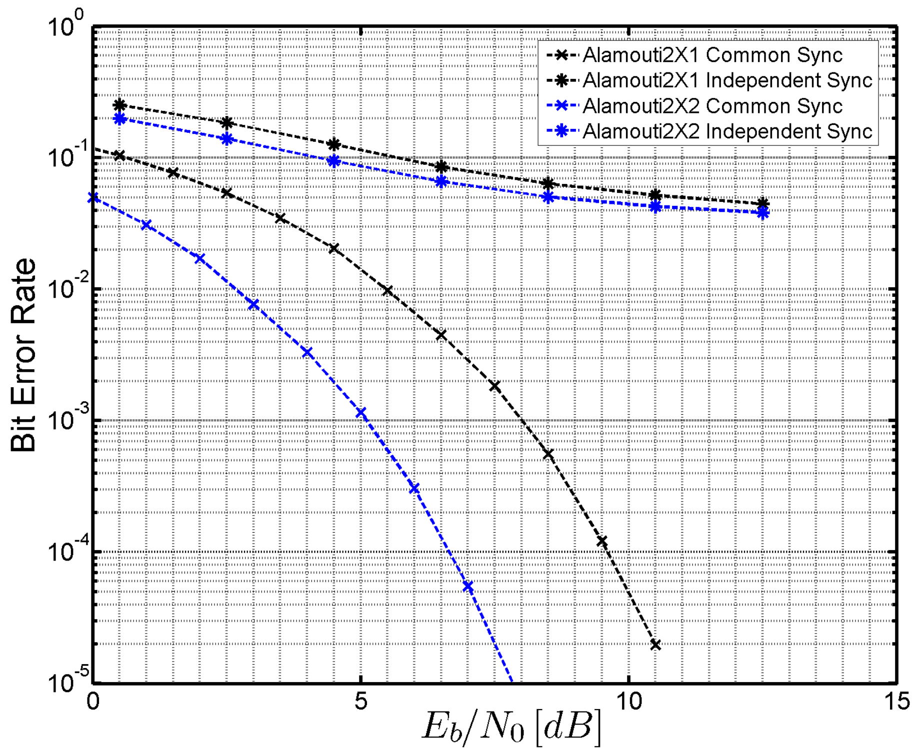


Figure 5.13: Simulation results with frequency and timing offset estimation for Alamouti receivers that use a common timing offset for both symbol periods compared to estimating the timing offset independently. The receivers assume a fixed SNR=5dB. The channel is flat and static with AWGN while no frequency or timing offset is present.

The first terms of (5.1.15) and (5.1.16) demonstrate a catastrophic phase shift, as was the case for the alternating scheme, that is not accounted for. Additionally, the change in timing offset error causes a loss in the estimation orthogonality as shown by the second terms. Figure 5.13 illustrates the performance decline caused by a change in the timing error when synchronising the two symbol periods independently compared to using the first symbol's synchronisation point for both symbols.

A timing offset error is not the predicament as long as the error is the same for both Alamouti symbol periods. Therefore the channel estimation for Alamouti systems needs to be consistent, meaning that the error in the estimation offset must be constant over two corresponding symbol periods. When this happens, the resulting phase shift is corrected for by the piloted channel response estimation scheme as if it had occurred in the channel in a similar fashion to that described for a SISO system above.

The simplest solution is to use the estimated synchronisation point of the first symbol for both. It would also be possible to construct an ML estimate running over two corresponding Alamouti periods to produce a single and more accurate "averaged" synchronisation point for the symbol pair. This added complexity is not desired as the common offset error occurring from using only the first symbol offset estimation was demonstrated to have negligible effect in Figure 5.12 and an increased accuracy would result in insignificant performance gains.

5.1.4 Conclusions

The performance of several OFDM communication systems where evaluated in a static channel with AWGN. The evaluation commenced with the simplified assumption that the receiver has perfect channel knowledge while consequent evaluations were conducted for more realistic conditions where the channel response and synchronisation had to be estimated.

The power ratio of the pilots in the linear channel estimation scheme was assessed as well as different pilot strategies for the Alamouti space-time block code scheme. Performance loss due to the implementation of timing and frequency offset estimation was shown to be negligible while important insight was given with regards to a difference in the timing offset estimation error over the two engaged Alamouti symbol periods.

The communication systems were evaluated by simulations and compared to an analytic assessments where feasible. In the rest of this chapter more complicated hardware and channel effects are investigated and the performance evaluation relies more heavily on simulation results as analytic formulations become exceedingly problematic.

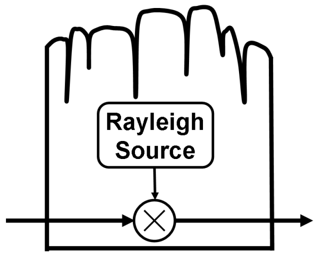


Figure 5.14: Slow fading channel implementation with additive white Gaussian noise.

5.2 Idealised Slow Fading Channel

As discussed in Section 4.5, fading occurs as a result of a changing multi-path environment. In this section a simplified fading model is used as a starting point to which more realistic channel modelling simulations are compared.

Rayleigh fading typically occurs in a wireless channel where no direct line-of-sight is present and therefore an idealised Rayleigh fading scenario is considered. The channel is frequency-flat (no delay spread) and changes occur at the beginning of each transmitted OFDM symbol and remains static for the duration of a symbol period and thus there is no inter-carrier interference. Simulation results are presented for receivers that have perfect knowledge of the fading channel and for those that have to estimate the channel and synchronise in the presence of noise.

5.2.1 Idealised Fading Model

Channel fading is implemented by a Rayleigh random variable generator depicted in Figure 5.14 which generates one fading coefficient for every OFDM symbol. Although this is not a realistic channel model, the simulation results are for the "ideal case" and serve as the ground work to which other channel implementations are compared.

The long-term average signal power is preserved after passing it though the multiplier in Figure 5.14 by setting the average power of the Rayleigh channel source to unity. This stems from the fact that the signal and channel fading are uncorrelated and thus $E[(SX)^2] = E[S^2]E[X^2]$ with E the expectation operator, S the transmitted signal and X the fading channel coefficient expressed as independent random processes. The Rayleigh probability density function is given by

$$f_X(x) = \frac{x}{\alpha^2} e^{\left(\frac{-x^2}{2\alpha^2}\right)} \tag{5.2.1}$$

with $E[X^2] = 2\alpha^2$ and the signal power passing through the multiplier remains unchanged when $\alpha = \sqrt{\frac{1}{2}}$.

The idealised fading causes a change in the channel after every OFDM symbol which in turn causes bit errors to occur in bursts and retard the BER simulation convergence. The certainty by

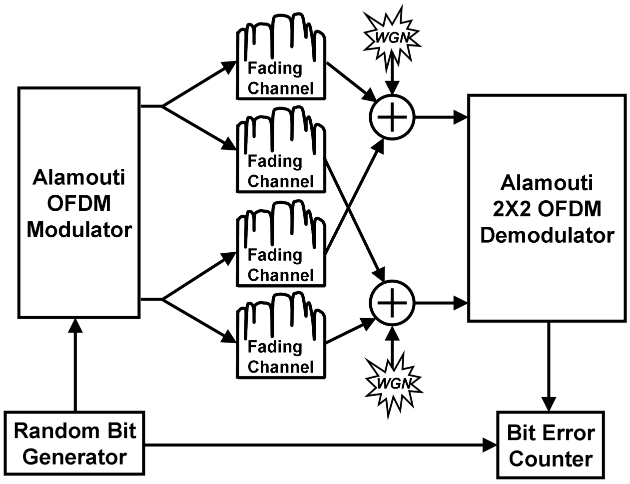


Figure 5.15: Alamouti 2X2 fading setup with independent fading in every channel.

which the bit error rate is known deteriorates substantially, as explained in Section 4.4.2, for the same amount of data passing though the system.

The simulation time increases significantly by a factor in the order of $2N_D$ in order to obtain precision comparable to that of independent bit errors. This computational burden is reduced when frequency selective fading is introduced in Section 5.3 which allows for subcarriers to fade more independently and results in smaller bit error bursts.

5.2.2 Idealised Slow Rayleigh with Perfect Channel Knowledge

Simulations are performed with idealised Rayleigh fading channel links while the fading coefficients, generated by the source in Figure 5.2, are fed to the receiver so that it can be corrected for after noise is added. The simulation setup for the Alamouti 2X2 system, where every channel link fades independently, is shown in Figure 5.15.

Figure 5.16 confirms that the simulated BER with perfect channel knowledge compares well to analytic solutions and results published by [29]. The probability of error for MRC and Alamouti systems using QPSK in an uncorrelated Rayleigh fading channel when the receiver has perfect channel knowledge is deduced from [34, Appendix C-18] as

$$P_{e\ \text{SISO}} = \frac{1}{2}\left[1 - \frac{1}{\sqrt{1 + \frac{1}{E_b/N_0}}}\right] \tag{5.2.2}$$

$$P_{e\ \text{Alamouti2X}n_r} = \frac{1}{2}\left[1 - \sqrt{\frac{E_b/N_0}{2 + E_b/N_0}} \sum_{k=0}^{2n_r-1} \binom{2k}{k} \left(\frac{1}{2(1 + E_b/N_0)}\right)^k\right] \tag{5.2.3}$$

$$P_{e\ \text{MRC1X}n_r} = \frac{1}{2}\left[1 - \sqrt{\frac{E_b/N_0}{1 + E_b/N_0}} \sum_{k=0}^{n_r-1} \binom{2k}{k} \left(\frac{1}{2(1 + 2E_b/N_0)}\right)^k\right]. \tag{5.2.4}$$

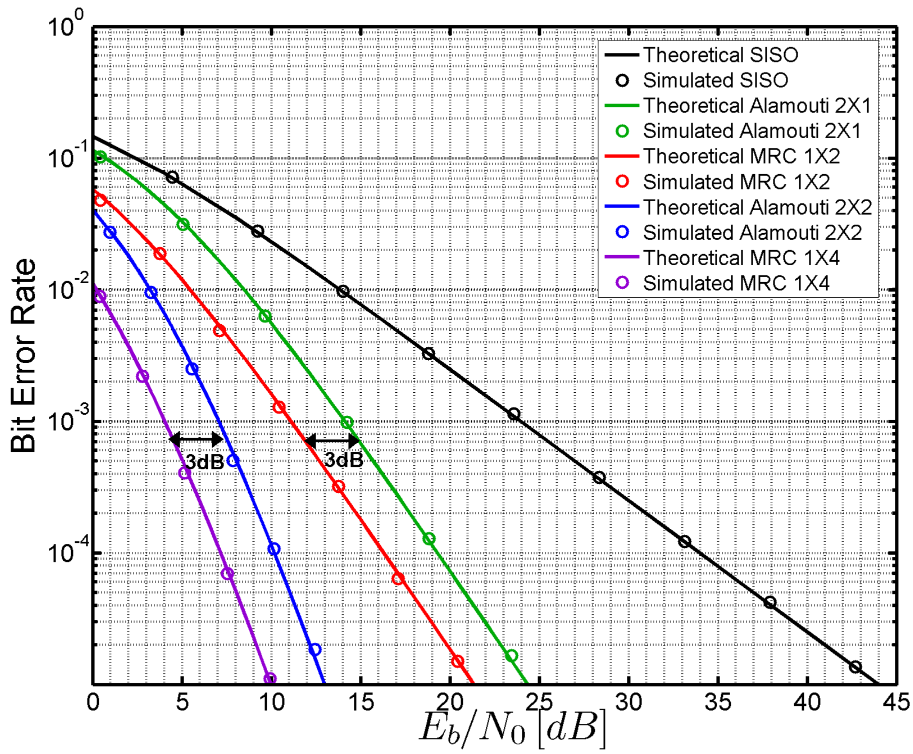


Figure 5.16: Simulation results with additive white Gaussian noise, slow Rayleigh fading and perfect channel knowledge.

Figure 5.16 illustrates how the increase in spacial diversity improves the BER by overcoming channel fading. The characteristic 3dB difference between the MRC 1X2 and the Alamouti 2X1, as mentioned in [29], is evident from these simulations. Both systems have two path diversity and thus exhibit similar behaviour while the MRC system has the advantage of an increased amount of receivers which increases the effective SNR as described in Section 3.4.

5.2.3 Idealised Slow Fading with Continuous Channel Response Estimation

A more realistic system to the one discussed in the previous section would be where the receiver has to estimate the channel in the presence of noise. The same idealised Rayleigh fading channel is used as before but the receiver has to estimate the channel response by means of pilots with a power ratio $R_p = \frac{16}{9}$ as discussed in Section 2.3.3 as well as estimate the timing and frequency offset by means of the technique discussed in Section 2.4.1.

The degradation due to estimating the channel response is illustrated in Figure 5.17.

Figure 5.18 verifies that the timing and frequency offset estimation implementation loss is negligible for all as was found for a static channel in Section 5.1.3. This is expected because the channel is static over any given symbol period and no delay spread occurs which could degrade the offset estimation algorithm described in Section 2.4.1.

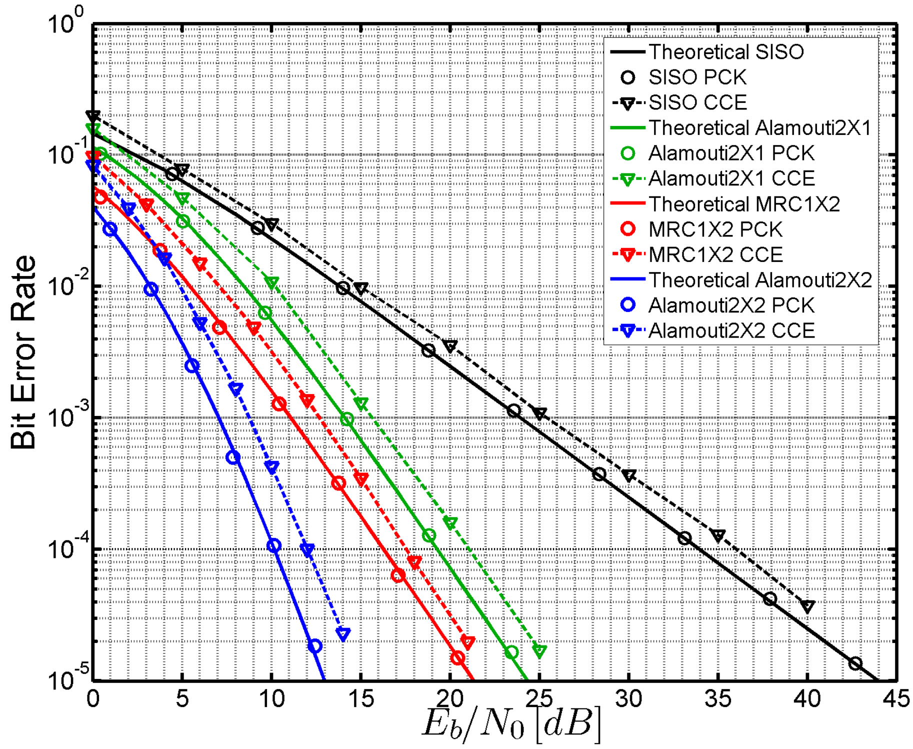


Figure 5.17: Comparison of systems with perfect channel knowledge (PCK) and continuous channel response estimation (CCE). The channel exhibits frequency-flat idealised slow Rayleigh fading with AWGN.

The results obtained in this section serve as a reference for the delay spread channel with Doppler shift which is investigated next.

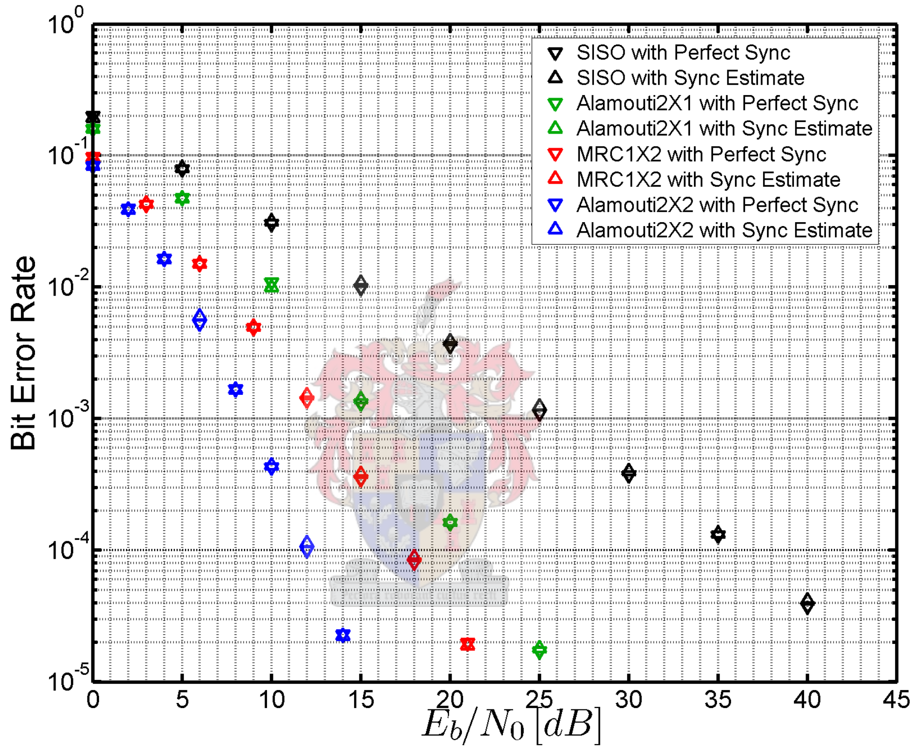
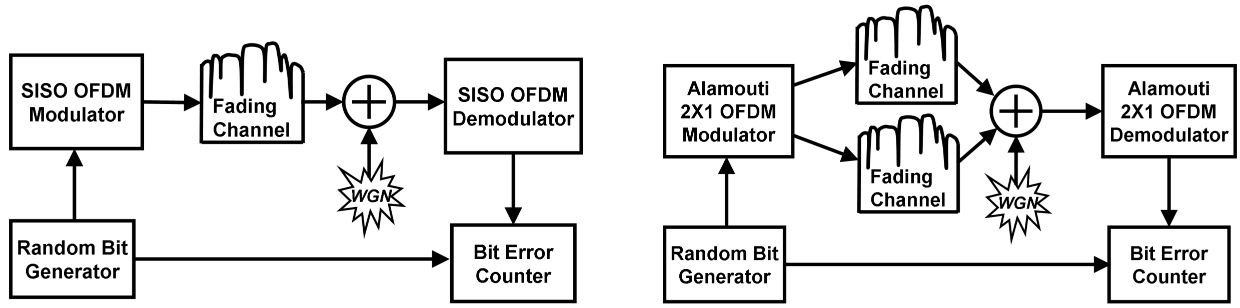
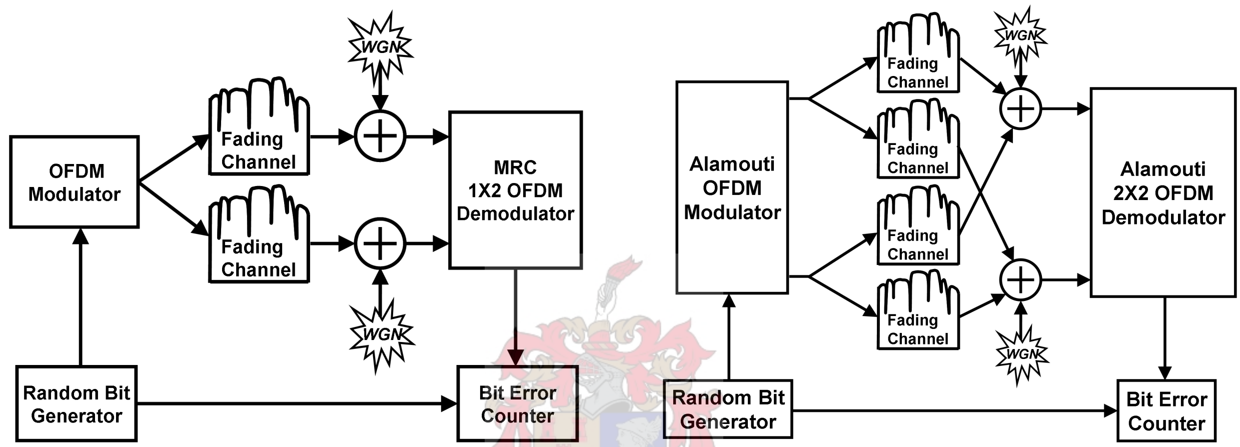


Figure 5.18: Simulation results with perfect synchronisation at the receiver(s) compared to receiver(s) that estimate the frequency and timing offset estimation while assuming the SNR=5dB. The channel exhibits idealised Rayleigh fading with AWGN.



(a) SISO simulation setup. (b) Alamouti 2X1 simulation setup.

Figure 5.19: Single receiver simulation setups with channel fading and additive white Gaussian noise.



(a) MRC 1X2 simulation setup. (b) Alamouti 2X2 simulation setup.

Figure 5.20: Multiple receiver simulation setups with channel fading and additive white Gaussian noise.

5.3 Delay Spread Doppler Fading Channel

Simulation results for a more realistic channel model than the idealised Rayleigh fading in Section 5.2 are presented. The channel model under evaluation is described in Section 4.5 and implements a delay spread and Rayleigh fading with Doppler shift. The delay spread causes frequency selectivity and inter-symbol interference (ISI) while the Doppler shift results in inter-carrier interference (ICI). The effect these phenomena have, as well as the performance of the estimation algorithms employed by the receivers are illustrated.

5.3.1 Simulation Setups

The simulation setups are shown in Figures 5.19 and 5.20 and illustrate that each channel link experiences independent fading while uncorrelated noise is added at every receiver.

The channel delay spread is modelled with the tapped delay profile illustrated by Figure 5.21 and Table 5.2 and is similar to the "*Typical case for urban area (TUx)*" as published by GSM 05.05 [50].

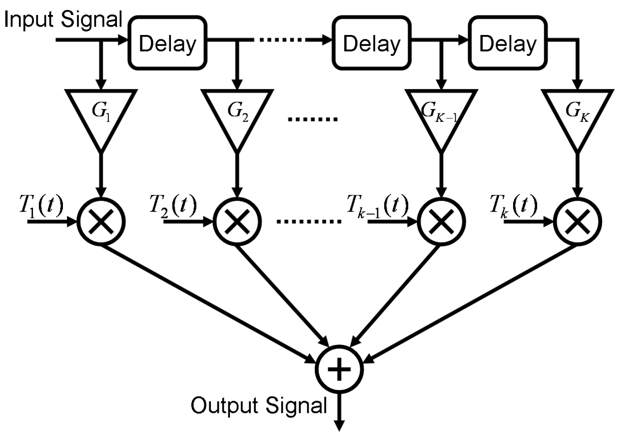


Figure 5.21: Channel model with tapped delay line and Doppler spectrum.

Table 5.2: Tapped delay line channel profile.

Tap No.	Delay [samples]	Delay [% of guard interval]	Weight [dB]
1	0	0	-3
2	3	3.9	0
3	7	5.5	-2
4	21	16	-6
5	30	23	-8
6	64	50	-10

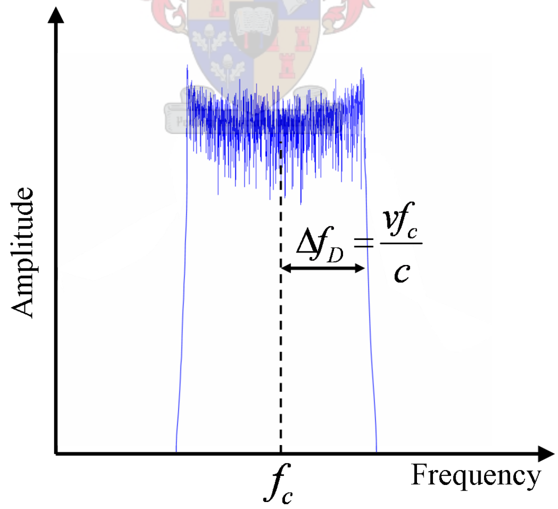


Figure 5.22: Power spectral density of a single tone which has passed though a multi-path Doppler spread channel.

The tap weights of Table 5.2 are relative and the true gain at each tap is normalized to produce an overall channel gain of 0dB. Each tap is then multiplied by an uncorrelated fading processes as described in Section 4.5. The OFDM guard interval consists of 128 samples, as discussed in Section 5, and is twice as long as the maximum delay spread.

Assuming the modulated signal is narrowband relative to the centre frequency, the maximum

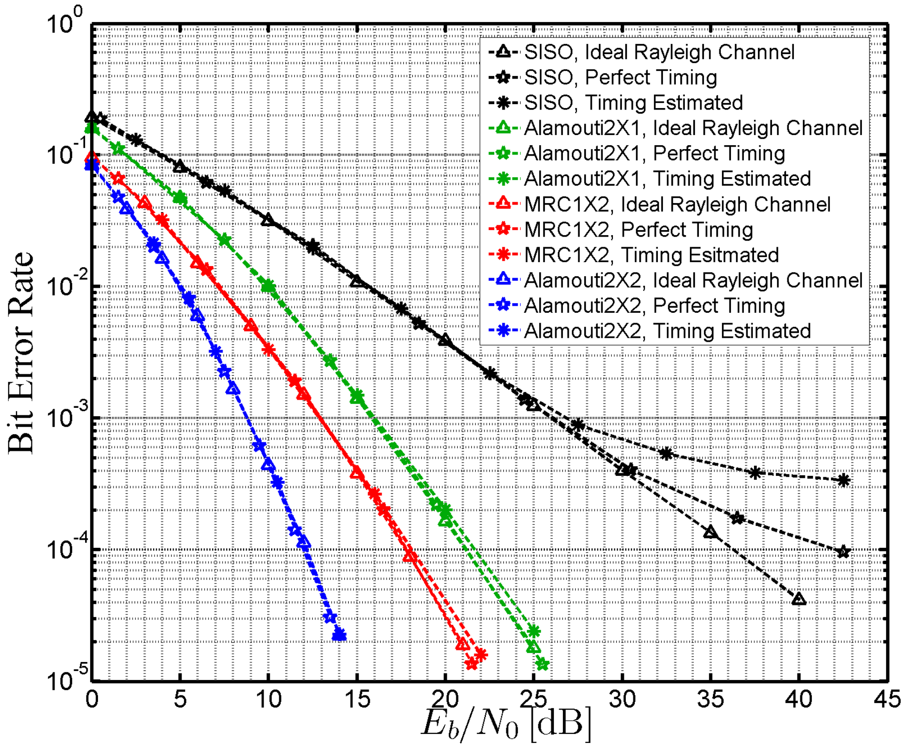


Figure 5.23: Bit error rate simulation of delay and Doppler spread fading channel with- and without timing offset estimation compared to ideal Rayleigh fading. The maximum Doppler shift is 0.37% of the inter-carrier spacing. Channel frequency response estimation is performed by means of pilots ($R_P = \frac{16}{9}$) for all systems.

Doppler shift experienced at all frequencies by the receiver is approximated as the same and given by

$$\Delta f_D = \frac{v f_c}{c} \tag{5.3.1}$$

with v the mobile's speed relative to the base station, f_c the centre frequency and c the wave propagation speed. In a dynamic multi-path environment with no line-of-sight and a receive antenna which is omnidirectional in the angle of arrival the unmodulated signal carrier is typically spread in the frequency domain producing the Clark or Jakes power spectral density [25], [54]:

$$S(f) = \begin{cases} \frac{\sigma_0^2}{\pi f_D \sqrt{1-((f-f_c)/f_D)^2}} & , \quad f_c - f_D < f < f_c + f_D \\ 0 & , \quad \text{elsewhere} \end{cases} . \tag{5.3.2}$$

The typical spectral shape as a result of the relative Doppler shift of the multiple arrival angles is given in Figure 5.22 as produced by the simulation model.

5.3.2 Simulation Results

Simulations are performed for a delay spread channel with different maximum Doppler shifts relative to the subcarrier spacing and compared to the idealised Rayleigh fading case.

Due to the delay spread, the timing and frequency offset estimation is no longer optimal [14]. The maximum channel delay spread is half of the cyclic extended guard interval, as described by Table 5.2, and therefore no ISI occurs if the timing of the guard interval from the shortest path is known exactly. ISI is introduced by an error in estimating the timing offset.

Simulation results for three scenarios are presented in Figure 5.23. Firstly the idealised Rayleigh fading case from Section 5.2, where no inter-symbol nor inter-carrier interference occurs. Secondly the multi-path Doppler and delay spread channel is employed while the timing offset is known exactly to the receiver and thus no ISI occurs while ICI is present. Thirdly the timing offset is estimated for the multi-path Doppler and delay spread channel and therefore both inter-symbol and inter-carrier interference occur. In all the above mentioned scenarios the channel frequency response is estimated by means of the piloting system described in Section 2.3.3 with a pilot to data subcarrier power ratio of $\frac{16}{9}$.

The effect of ICI is observed from the BER deterioration, in Figure 5.23, by comparing the second to the first scenarios. A minor BER deterioration, caused by ICI of a channel with a maximum Doppler spread of 0.37% of the subcarrier spacing, is observed for the SISO case at large SNR. The BER deterioration is unnoticeable for the systems with spacial diversity larger than one while considering a BER down to 10^{-5}.

Similarly the performance of the timing offset estimation algorithm is evaluated from the BER deterioration between scenarios three and two. Figure 5.23 illustrates that the algorithm performs exceptionally well in a multi-path delay spread Doppler fading channel. The systems with higher spacial diversity demonstrate increased immunity to the small synchronisation errors.

Simulation results for the SISO, MRC and Alamouti systems with the receiver travelling at different speeds are presented in Figures 5.24, 5.25 and 5.26. The idealised slow fading case, where the channel is static for the duration of each OFDM symbol, from Figure 5.17, is also plotted to serve as the “ideal case” reference.

ICI is introduced by the channel which is changing during an OFDM symbol and the degradation in BER as the mobile receiver’s speed increases is evident. The simulation results in Figures 5.24, 5.25 and 5.26 indicate that the ICI causes the BER to converge as E_b/N_0 increases. The receiver has limited mechanism by which to correct for the ICI and it will thus produce a bit error rate floor which would be reached when the system is completely noiseless.

The systems with the same amount of receive front-ends are compared in Figures 5.25 and 5.26 and it is evident that at low speeds the system with the larger transmit diversity performs significantly better. A field of great interest is the effects caused by a receiver moving at higher

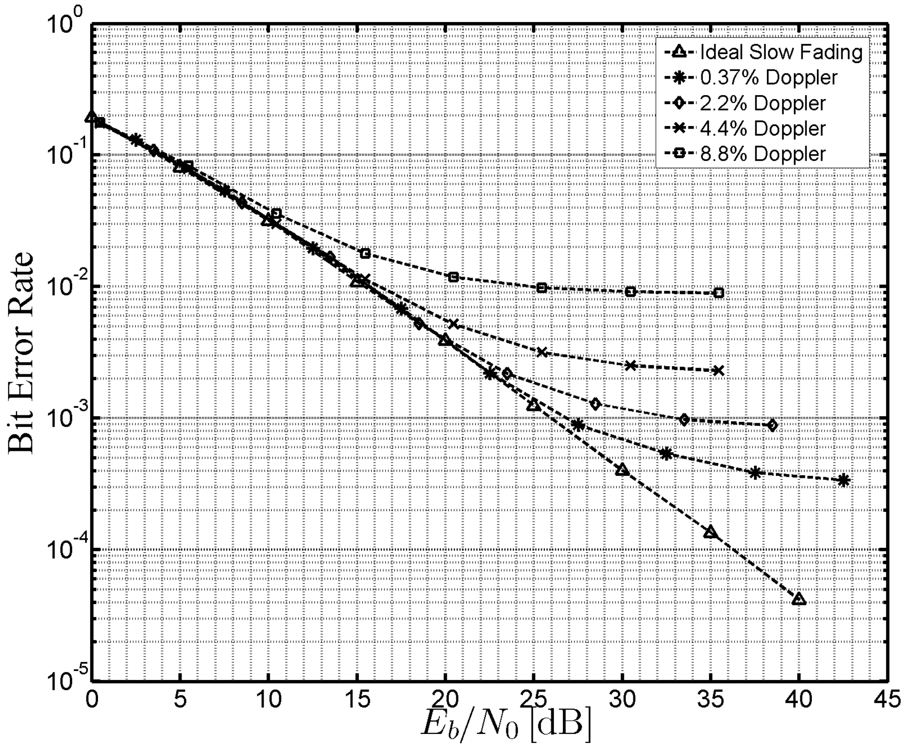


Figure 5.24: Bit error rate simulation of a SISO system with delay spread channel for different maximum Doppler shifts relative to the subcarrier spacing. Continuous estimation of the channel response as well as timing and frequency offset is employed.

speeds in an urban environment. From the simulation results it is evident that the Alamouti STBC systems are substantially more sensitive to fast-fading.

Consider Figure 5.25 where both the SISO and Alamouti 2X1 systems have one receiver. The Alamouti 2X1 system has two transmitters and thus 2-path diversity and demonstrates superior immunity to channel fading at low mobility. When the speed at which the receiver is moving in the channel is increased the Alamouti 2X1 system performance deteriorates considerably and more rapidly than that of the SISO scheme. The same is observed for the MRC 1X2 and Alamouti 2X2 systems in Figure 5.26. In a multi-path environment the Alamouti systems, compared to a maximum ratio combiner with the same amount of receivers, are superior provided the maximum Doppler shift is less than approximately 4.5% of the subcarrier spacing.

An intuitive reason for this rapid deterioration in the Alamouti systems at higher speeds is partly due to the fact that it has two transmitters. When the amount of transmitters increase so too does the time needed to estimate the channel links and for a STBC the amount of symbol periods that need to be observed before the data can be decoded increases. The channel estimation and Alamouti symbol recombination algorithms require the channel to be static over two corresponding symbol periods which is not the case at higher receiver speeds and results in a less accurate estimation of the channel. Furthermore, it was shown by the simulation results in Section 5.1 and by [55] that the Alamouti STBC systems are more sensitive to channel estimation errors than SISO or MRC

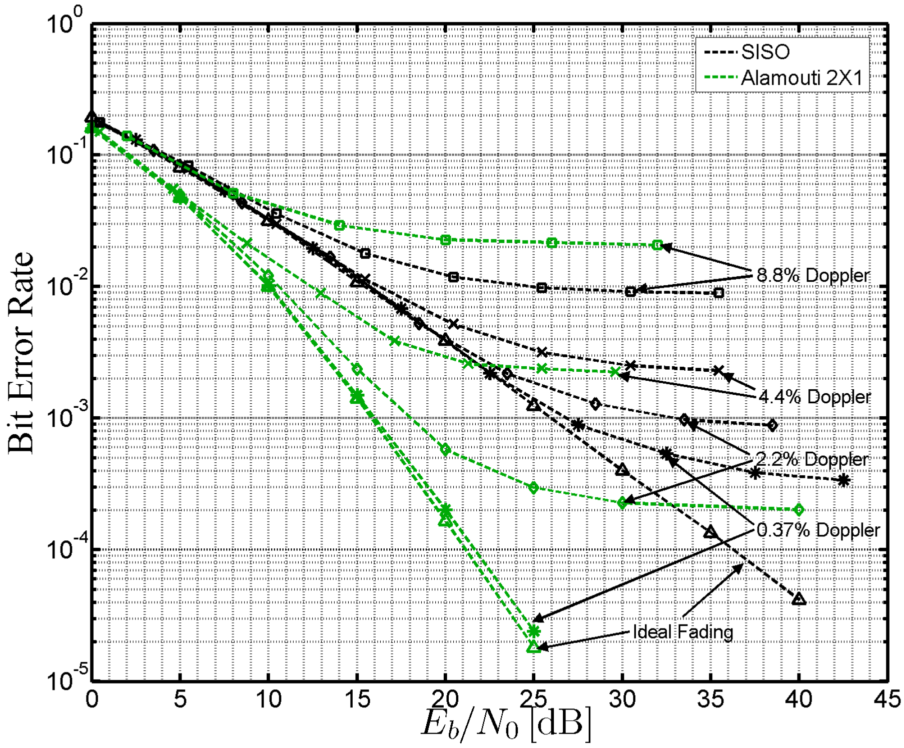


Figure 5.25: Bit error rate simulation of a SISO and Alamouti 2X1 system for different maximum Doppler shifts relative to the subcarrier spacing. Continuous estimation of the channel response as well as timing and frequency offset is employed.

systems and this sensitivity contributes to the increased deterioration observed in Figures 5.25 and 5.26.

5.3.3 Fading Example

In order to place the fading in perspective the simulation results are related to OFDM parameters. The 3rd Generation Partnership Project (3GPP) has launched a feasibility study for the use of OFDM for UTRAN enhancement [5] and as an example the OFDM parameters are chosen to suite the UMTS bandwidth and a sample rate that is compatible with the UMTS chip rate at a carrier frequency of 2.14 GHz.

The system has 1705 subcarriers spaced 2.6786 kHz apart with a total bandwidth of 4.567 MHz and a guard time of 23.33 μs ($\frac{1}{16}$'th the OFDM symbol).

The transmitted signal is narrow-band, meaning that Doppler shift for all frequency components of the signal is calculated relative to the carrier frequency. A small relative bandwidth of $\frac{5\times10^6}{2.14\times10^9} = 0.23\%$ justifies this approximation.

For these parameters, the maximum Doppler shift relative to the subcarrier spacing of 0.37% translates to a mobile travelling at 5km/h which is comparable to that of a person walking. The 4.4% maximum Doppler shift relative to the subcarrier spacing translates to the receiver travelling

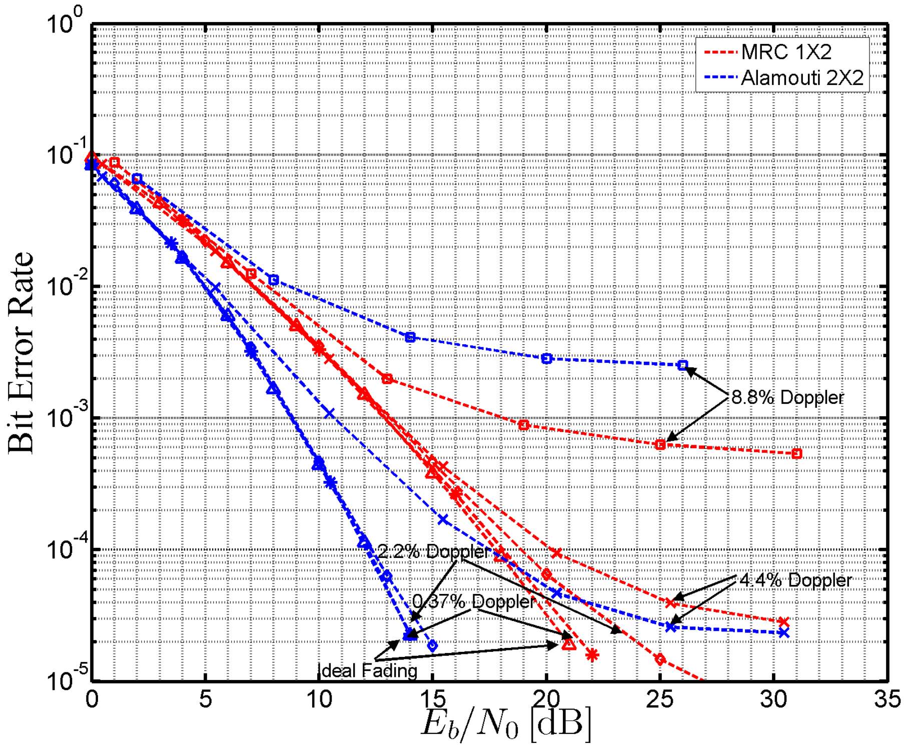


Figure 5.26: Bit error rate simulation of a SISO and Alamouti 2X1 system for different maximum Doppler shifts relative to the subcarrier spacing. Continuous estimation of the channel response as well as timing and frequency offset is employed.

at 60km/h, the maximum speed anticipated for a "*typical urban environment*". From the simulation results in Figures 5.25 and 5.26 a 5km/h Doppler fading channel for these specific OFDM parameters is comparable to the ideal slow fading case. On the other hand, travelling at 60 km/h would deteriorate system performance significantly.

5.3.4 Avoiding Aliasing in the Simulation Domain

The fading channels are simulated by multiplying the transmitted OFDM signal with the fading signal created by the sum of low frequency sinusoids as described in Section 4.5. The multiplication in the time domain causes a "smearing" in the frequency domain as the signal spectrum is convolved with a function of the form as depicted by Figure 5.22. The simulation domain sampling should be adequate in order to avoid aliasing.

The zero-padding used by the IFFT in order to generate the OFDM signal, provides excess simulation bandwidthto eliminate the effect of aliasing while simulating Doppler effects. The Jakes spectrum with a maximum Doppler shift of 8.8% has an RF bandwidth of

$$\begin{aligned} 2\Delta f_d &= \frac{2 \times 0.088}{N_{FFT}} \\ &= 8.59 \times 10^{-5} \text{ cycles/sample.} \end{aligned}$$

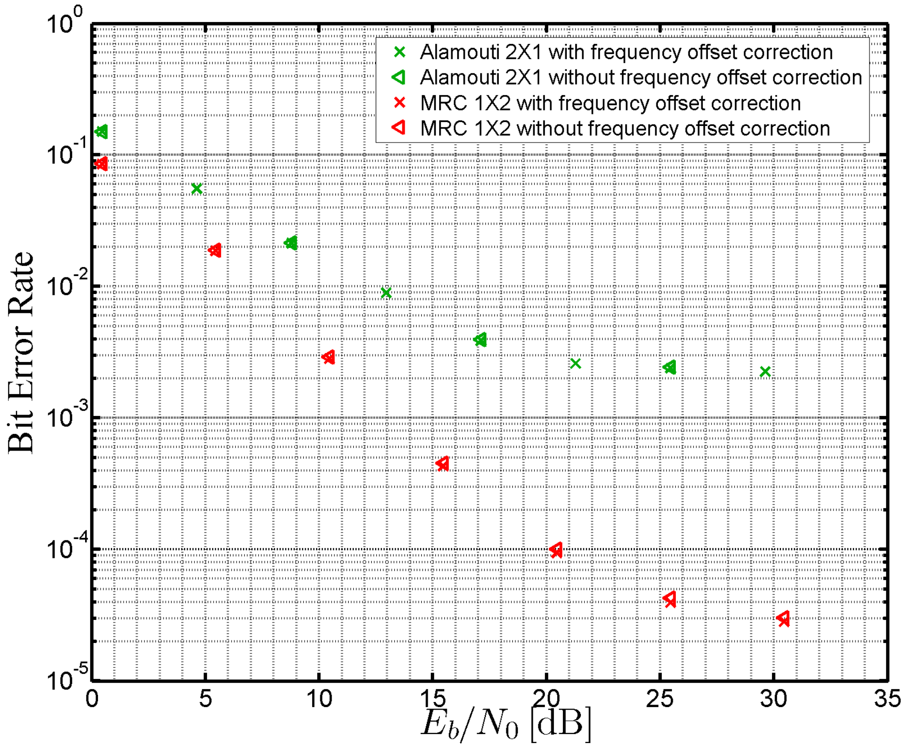


Figure 5.27: Comparison of BER simulations with and without frequency synchronization in a fading channel with Doppler spread $\Delta f_D = 4.4\%$.

No aliasing is therefore expected when convolving its spectrum with the OFDM signal.

In Section 5.5, which investigates nonlinear amplifiers, the effect of simulation domain aliasing is significant and discussed further.

5.3.5 Frequency Offset Estimation in a Doppler Spread Fading Channel

The maximum likelihood frequency offset estimation [14] is derived from the axiom that a single frequency offset exists while the Doppler spread channel model, used in the simulations, is effectively the sum of multiple transmitted signals each with a different frequency and time offset. Nevertheless, Figure 5.27 shows that the frequency offset estimation does not deteriorate system performance in a Doppler spread environment.

5.3.6 Conclusions

Channel effects which include Doppler and delay spread multi-path fading were investigated. The results provide a system designer with an important understanding of the effects of multi-path fading and guidelines on the choice of OFDM parameters such as the subcarrier spacing required for the largest anticipated receiver velocity. It was revealed that when a certain speed is exceeded the Alamouti is out performed by a system with only one transmitter despite the Alamouti scheme possessing twice the spatial diversity.

The timing and frequency synchronization algorithm [14], as described in Section 2.4.1, was evaluated in a realistic channel model. The algorithm was derived for a single frequency offset without considering channel delay spread and fading. Although suboptimal, it demonstrated excellent performance when evaluated in a channel with delay spread and Doppler fading. The algorithm utilizes the OFDM cyclic extended guard interval and a guard interval of twice the maximum delay spread is shown to be sufficient. Further evaluation is performed in Section 5.4 for a receiver that is frequency offset.

For evaluating higher speeds, for example travelling on a highway away from built-up areas the environment is expected to have less multipath and the angles of arrival can no longer be assumed to be uniformly distributed as assumed when deriving the channel model in Section 4.5. Rician fading may be expected because of the presence of a direct line of sight path. Such a scenario would require a modified version of the channel model described in Section 4.5. A direct line of sight would provide a dominant frequency component which is offset depending on the radial motion of the receiver relative to the base station. The frequency offset estimation could estimate this dominant component and provide improvement and is subject to further investigation.

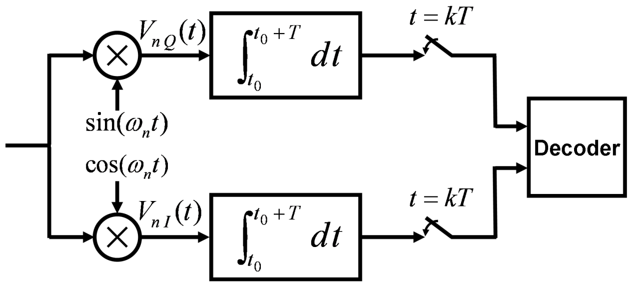


Figure 5.28: Conceptual quadrature correlation receiver for the n'th OFDM subcarrier.

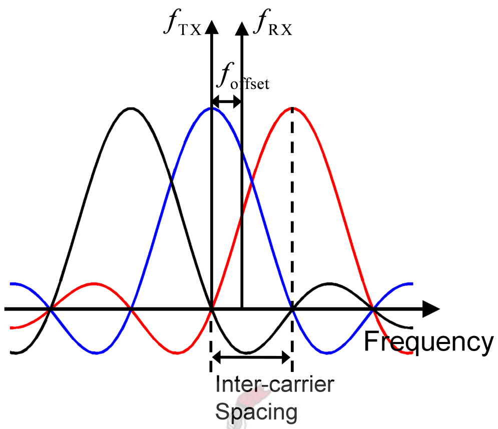


Figure 5.29: OFDM spectrum envelope illustrating the receiver local oscillator offset in frequency from the transmitter.

5.4 Frequency Offset

A frequency offset occurs when the transmit and/or receive local oscillators do not operate at the exact desired frequency. This is typically caused by the temperature and aging instability of the reference oscillators. An OFDM signal consists of independently modulated orthogonal subcarriers spaced close together in the frequency domain and inter-carrier interference (ICI) results due to the loss of orthogonality between subcarriers as they no longer have an integer amount of cycles in the demodulating IFFT interval.

The discrete Fourier transform, used in OFDM demodulation, effectively acts as a matched filter or correlator at the receiver in order to demodulate the different subcarriers. Figure 5.28 represents the conceptual coherent demodulator for each OFDM subcarrier at the receiver. The discrete Fourier transform may be conceptualised as a set of parallel correlators, one for each subcarrier.

When a local oscillator (LO) is frequency offset, as demonstrated in Figure 5.29 the correlation between a desired subcarrier and its corresponding demodulating reference signal decreases. Thus the desired output signal amplitude decreases. The offset also causes loss of subcarrier orthogonality and results in ICI as the correlation between a demodulating reference signal and an unwanted adjacent subcarrier increases.

The effect of frequency offset on SISO- and Alamouti-OFDM communication systems is investigated as well as the simultaneous time and frequency estimation algorithm developed by Van de Beek et alii [14].

The systems operate in a delay spread Doppler fading environment and the effects of a frequency offset without attempting to correct for it are initially assessed. It is revealed that the systems making use of Alamouti's STBC are far more sensitive to frequency offset and thus accurate estimation thereof is crucial. Finally the offset estimation algorithm is shown to improve system performance significantly.

5.4.1 Frequency Offset Implementation

In the synchronous data flow simulation environment a band limited RF signal is represented by the sampled complex envelope and its characterization frequency as discussed in Section 4.1.1.

$$v(t) = \Re\left\{\tilde{v}(t)e^{j2\pi f_c t}\right\} \tag{5.4.1}$$

with

$v(t)$	the real RF bandpass signal in the time domain,
$\tilde{v}(t) = \tilde{v}_I(t) + j\tilde{v}_Q(t)$	the modulating complex envelope and
f_c	the RF characterisation frequency.

When mixing the signal in (5.4.1) to a chosen frequency f_{c1}, and ignoring any image frequencies and spurii the signal is represented by

$$v_1(t) \;=\; \Re\left\{\tilde{v}(t)e^{j2\pi(f_c \pm f_{LO})t}e^{j2\pi f_{c1}t}\right\} \tag{5.4.2}$$

where the addition or subtraction sign depends on whether the signal is up- or down-converted respectively. If the LO is exact $f_{LO} = f_{c1} \mp f_c$ and the resulting signal is given by:

$$v_1(t) \;=\; \Re\left\{\tilde{v}(t)e^{j2\pi f_{c1}t}\right\} \tag{5.4.3}$$

producing a modulating envelope which is identical to that of (5.4.1) with the only difference that it now modulates a different carrier. On the contrary, when f_{LO} is offset by some value unknown to the system $f_{LO} = f_{c1} \mp f_c + f_{\text{offset}}$. The resulting signal is represented by

$$\begin{aligned} v_1(t) &= \Re\left\{\left(\tilde{v}(t)e^{j2\pi f_{\text{offset}}t}\right)\right\} && (5.4.4)\\ &= \Re\left\{\tilde{v}_1(t)e^{j2\pi f_{c1}t}\right\} && (5.4.5)\end{aligned}$$

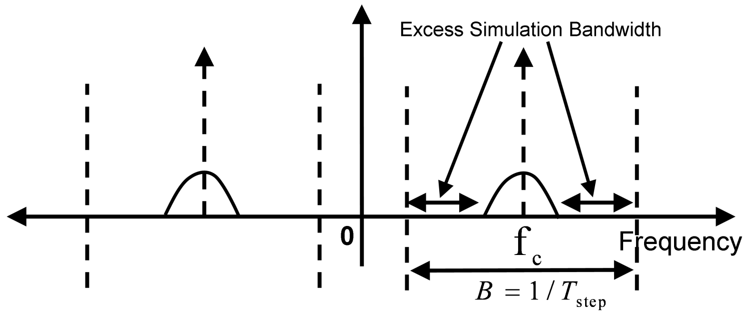


Figure 5.30: Excess simulation bandwidth with over-sampled envelope modulating a carrier in the simulation domain.

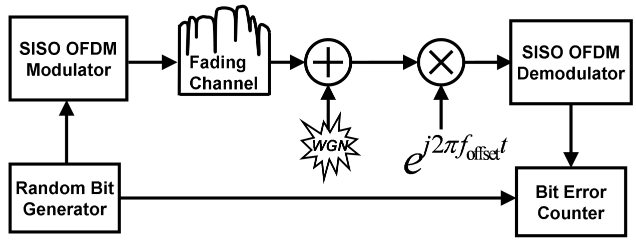


Figure 5.31: SISO frequency offset simulation setup.

and the complex envelope modulating the carrier at f_{c1} now becomes

$$\tilde{v}_1(t) = \tilde{v}(t)e^{j2\pi f_{\text{offset}}t} \tag{5.4.6}$$

where f_{offset} may be a result of multiple inaccurate frequency conversion stages. The frequency offset is therefore created in the simulation domain by multiplying the sampled modulating envelope by the factor $e^{j2\pi f_{\text{offset}}t}$. It should be noted that in order to avoid ambiguity and aliasing in the simulation domain when representing the frequency offset in (5.4.6), the sampled modulating envelope ($\tilde{v}(t)$) must be adequately sampled and contain "excess simulation bandwidth". This concept is presented graphically in Figure 5.30 and illustrates that $|f_{\text{offset}}| < \frac{\text{excess bandwidth}}{2}$. For simulation purposes frequency offsets of up to a maximum of 50% of the OFDM subcarrier spacing, which is the limit of the frequency synchronisation algorithm, are considered. A 2048 complex point IFFT with 1705 subcarriers is used to generate the OFDM signal. The remaining 343 elements in the IFFT are filled with zeros and this creates the excess simulation bandwidth. No aliasing occurs in the simulation domain due to offsetting the signal frequency by up to 50% of the subcarrier spacing.

5.4.2 Simulation Setups

The effect of frequency offset and the compensation technique discussed in Section 2.4.1 are evaluated in a multi-path delay spread Doppler fading environment, described in Section 4.5, with the receiver estimating the channel response by means of pilots as discussed in Section 2.4.3. It is as-

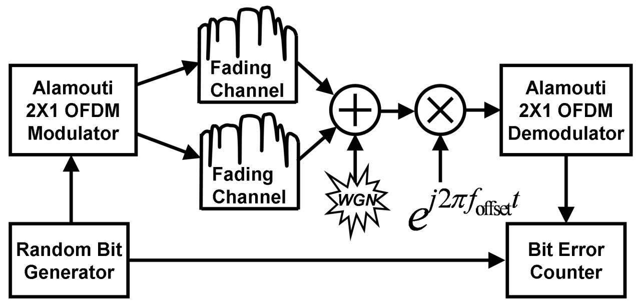


Figure 5.32: Alamouti2X1 frequency offset simulation setup.

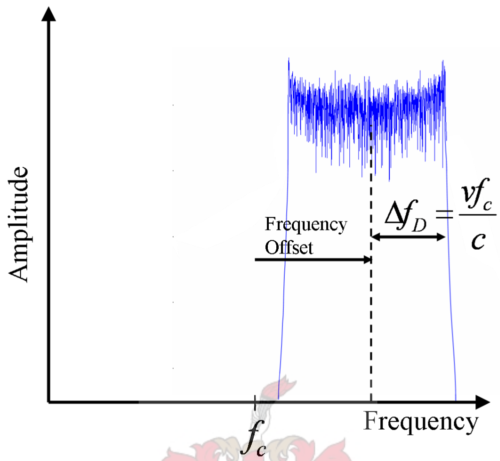


Figure 5.33: Power spectral density of a single tone which has passed though a multi-path Doppler spread channel with frequency offset.

sumed that the same reference oscillators are used for driving corresponding conversion stages of the multiple analogue front-ends. This implies that every receiver front-end would have the same frequency offset with respect to every transmitting front-end in a communication system.

The simulation setups are depicted by Figures 5.31 and 5.32 where the channel together with the frequency offset effectively convolves the signal in the frequency domain with a random process which has an offset Jakes power spectral density shown in Figure 5.33.

5.4.3 Simulation Results

Simulations are performed with and without frequency offset estimation to clearly illustrate the improvement. Figures 5.34 and 5.35 demonstrate the deterioration in bit error rate as the frequency offset increases in SISO and Alamouti2X1 systems with no frequency offset correction. As shown in Figure 5.34, a frequency offset relative to an inter-carrier spacing of 10% for a SISO system may be tolerable as it results in an implementation loss of roughly 2dB at a BER of 10^{-3} compared to the case with no frequency offset. On the other hand a 10% offset in the Alamouti2X1 STBC system, in Figure 5.35, causes a considerably larger BER deterioration which is so severe that more than two bit errors occur for every hundred bits sent even for a completely noiseless receiver.

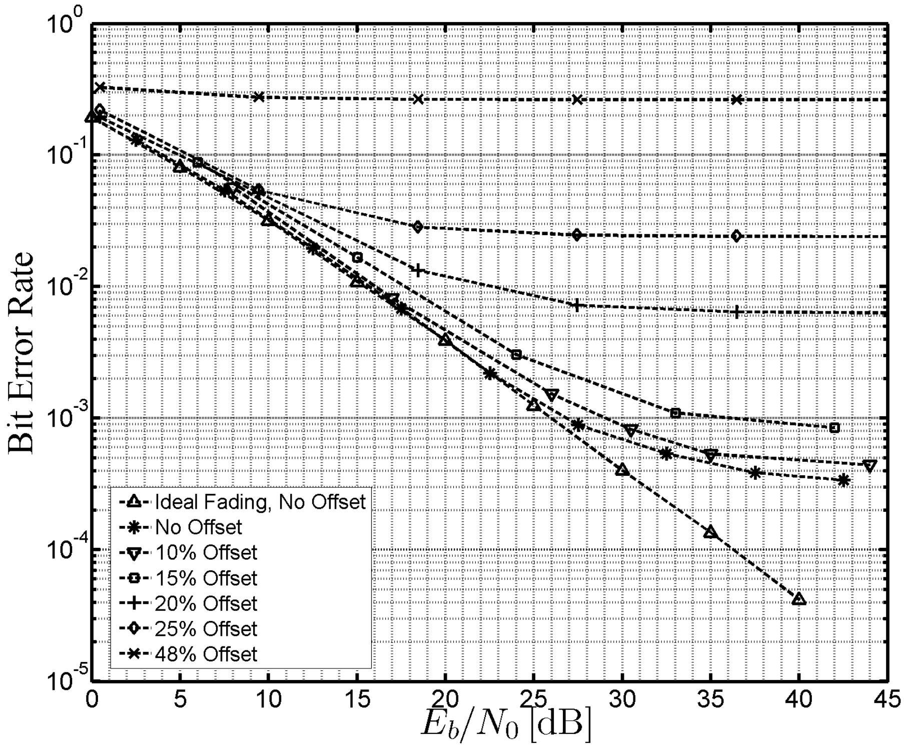


Figure 5.34: SISO BER simulation for different frequency offsets relative to the subcarrier spacing without frequency offset estimation in a multi-path urban fading channel with maximum Doppler shift of 0.37% of the subcarrier spacing.

Figure 5.36 illustrates the improvement provided by the frequency offset estimator described in Section 2.4.1. The frequency offset estimation algorithm is limited to estimating an offset of up to $\pm 50\%$ the subcarrier spacing and simulations with offsets up to 48% for both SISO and Alamouti2X1 systems result in practically identical performance compared to a system with no frequency offset.

The derivation of the frequency offset estimation algorithm in [14] does not take into account that the channel may be fading nor that it is delay spread, nevertheless the results in Figure 5.36 verify that the algorithm operates well under these conditions.

5.4.4 Conclusions

Frequency offset, which may be caused by unstable local oscillators, is shown to be of great concern in OFDM systems as it may significantly deteriorate the performance if it is not compensated for.

Simulations results are presented for constant frequency offsets at the receiver in uncorrelated fading channels with a delay spread and a maximum Doppler spread of 0.37% of the subcarrier spacing. It was illustrated that the Alamouti 2X1 system is considerably more sensitive to frequency offset than the SISO case while the frequency offset estimation provides practically perfect

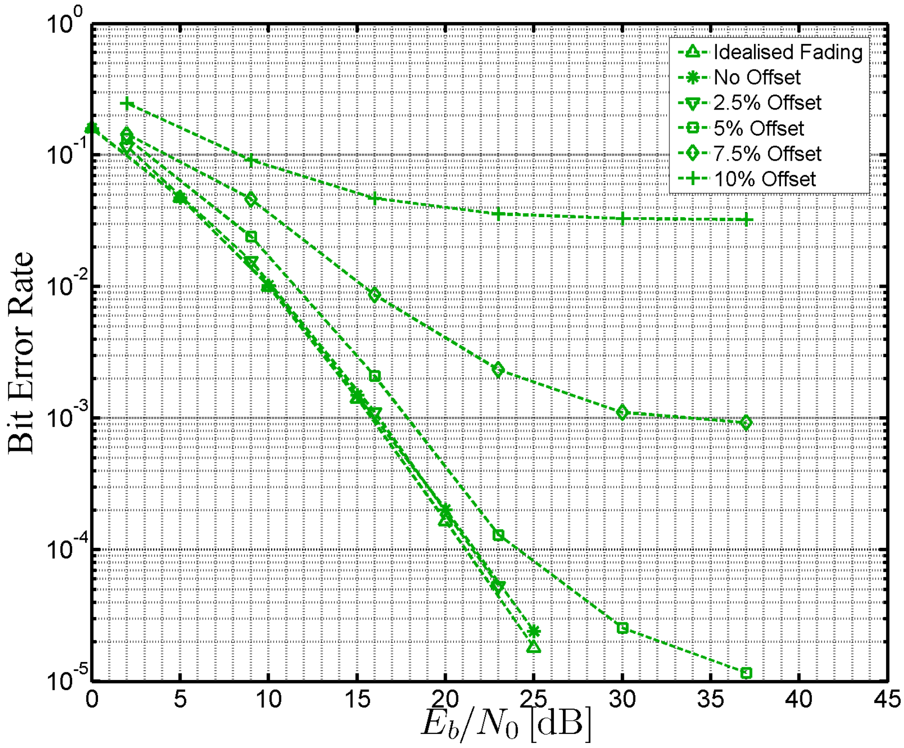


Figure 5.35: Alamouti2X1 BER simulation for different frequency offsets relative to the subcarrier spacing without frequency offset estimation in a multi-path urban fading channel with maximum Doppler shift of 0.37% of the subcarrier spacing.

correction for both systems under these channel conditions.

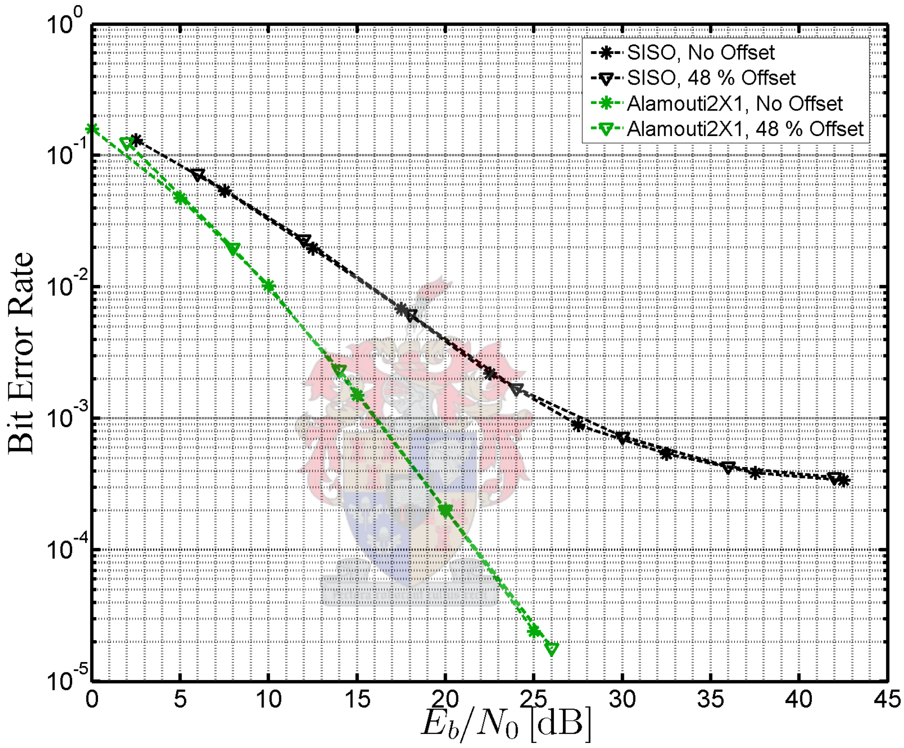


Figure 5.36: BER simulation for different frequency offsets relative to the subcarrier spacing with frequency offset estimation in a multi-path urban fading channel with maximum Doppler shift of 0.37% of the subcarrier spacing.

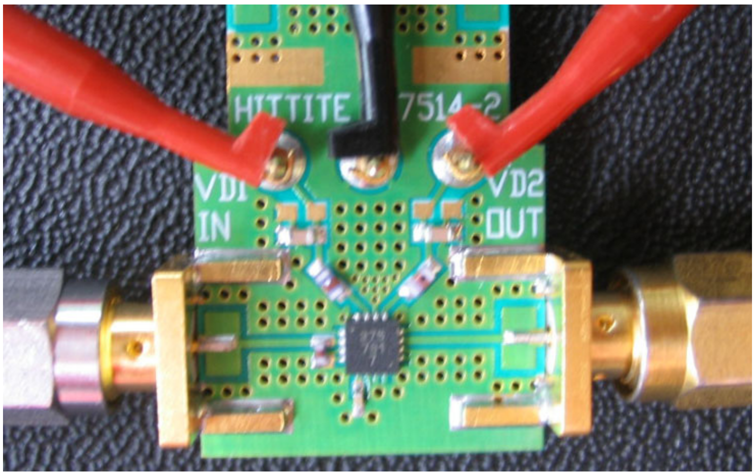


Figure 5.37: Hittite amplifier.

5.5 Nonlinear Amplifier

Amplifiers play an important role in both the transmit and receive parts of a communication system. While noise generally places a limit on the smallest decodable signal, the nonlinearity of components at high input signal power dictates the upper limit of a system's dynamic range. This section addresses the effects of memoryless nonlinear behaviour and provides guidelines to the degradation in system performance.

As a practical example, measurements were performed on the Hittite HMC375LP3 GaAs PHEMT MMIC amplifier in Figure 5.37 to demonstrate the general technique of creating a nonlinear amplifier behavioural model in ADS. The necessary parameters are extracted from single- and 2-tone measurements described in Sections 5.5.1 and 5.5.2 while the model response is compared to measurements in Section 5.5.3. The model is further utilised in BER simulations in Section 5.5.4 to examine the effects of a nonlinear component in the communication systems.

5.5.1 Single-tone Measurements

Figure 5.38 displays the amplifier's fundamental output to a single tone excitation at 2.14 GHz measured on a spectrum analyser over a wide input power range. A typical fundamental output versus single tone input power curve can be described in terms of a linear, transition and saturation area. Gain compression occurs at high input power and is defined as the reduction in the fundamental gain relative to extrapolated small signal gain as illustrated in Figure 5.38.

In order to accurately obtain the 1dB and 5dB output compression points ($P_{1\text{dB}}$and$P_{5\text{dB}}$), required as parameters for the behavioural model, the amplifier gain is plotted versus the the fundamental output power in Figure 5.39. With an average gain of 14.85dB in the linear region (below 6dB$_\text{m}$ output), the 1dB and 5dB compression points referred to the output are deduced from the graph as 17.95dB$_\text{m}$ and 19dB$_\text{m}$ respectively.

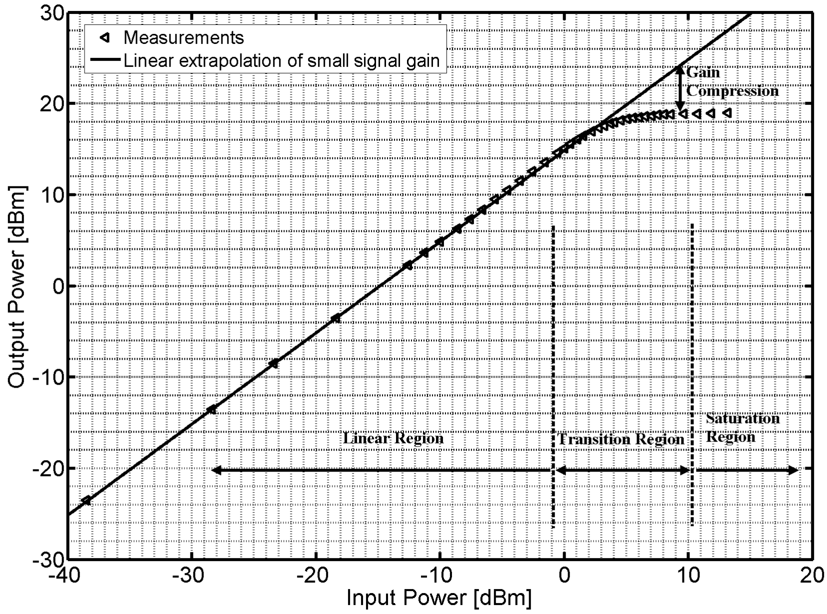


Figure 5.38: Amplifier fundamental output with single tone input at 2.14 GHz.

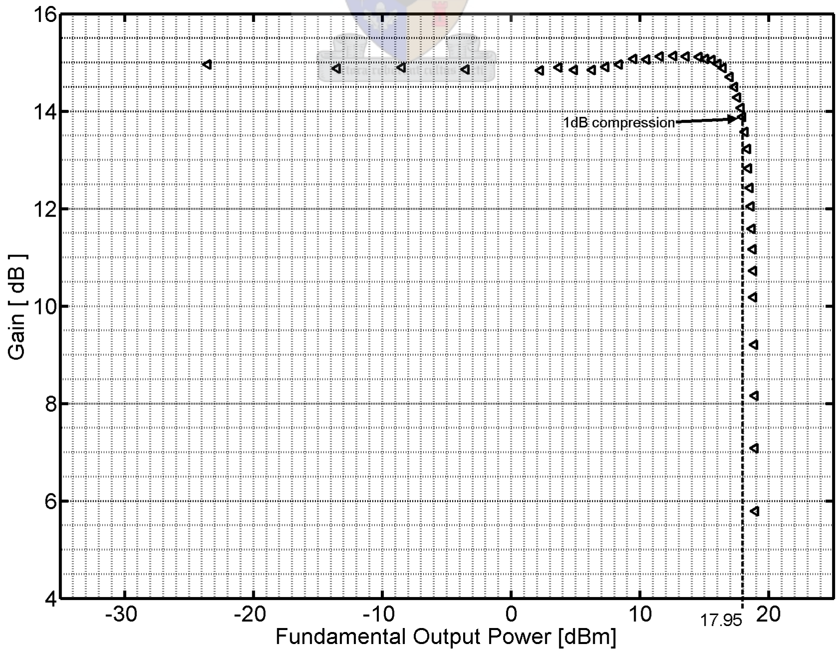


Figure 5.39: Amplifier fundamental gain versus fundamental output for single tone excitation at 2.14 GHz.

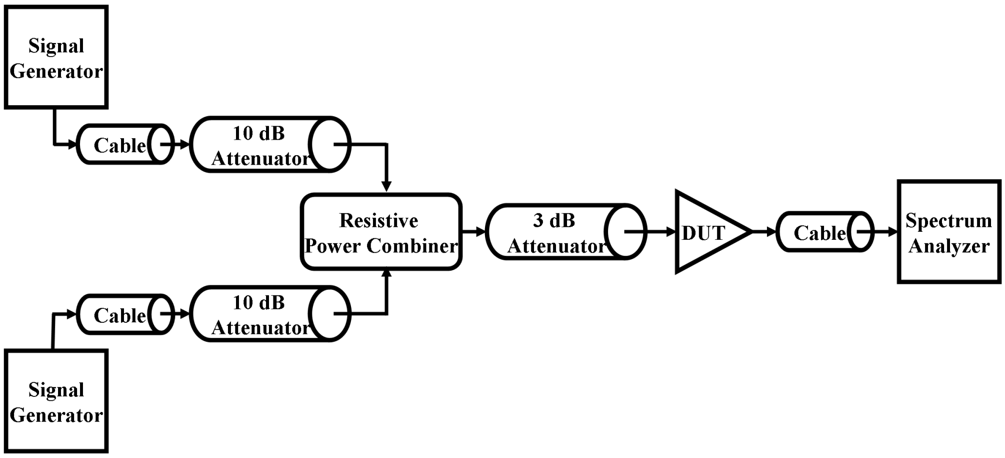


Figure 5.40: Intermodulation distortion measurement setup.

5.5.2 Two-tone Measurements

The odd-order intermodulation distortion is of particular interest as it causes in-band signal distortion which cannot be removed by filtering. It is measured by means of a two-tone measurement setup with equal power tones spaced 100kHz apart as illustrated in Figure 5.40 and is similar to that used by [56]. The two 10dB attenuators serve to isolate the signal generators from one another in order to avoid any possible intermodulation distortion caused by an incident signal on the generators' output stage. A wide-band resistive power combiner is used as it is matched at all ports and the 3dB attenuator, placed after the combiner, serves to reduce the reflected signal due to any mismatch at the amplifier input. The device under test's (DUT) input signal is measured with a spectrum analyser to check for unwanted spurii from the signal generating network and to compensate for losses in the combining network and inaccuracies of the signal generators for every power setting.

Care is taken that the spectrum analyser's input circuitry is not driven into saturation and that its spurious response is negligible compared to that of the amplifier's. This is done by measuring the amplifier's response while the signal power increases and increasing the spectrum analyser's internal attenuator until the measured spurii levels no longer change. In order to maintain relative and absolute measurement accuracy the spectrum analyser's resolution bandwidth is kept at a constant 1kHz which eliminates resolution bandwidth switching uncertainty. The reference level is optimally adjusted for every measurement so that the measurements are made in the upper most part of the logarithmic dynamic range to improve accuracy and reduce scale fidelity uncertainty. Peak sample detector mode is selected to ensure that the spectrum analyser's limited display resolution does not effect the power measurements.

The measured fundamental, third and fifth order intermodulation products are plotted in Figure 5.41. At low input signal power the third and fifth order intermodulation products have a 30dB and 50dB per decade gradient respectively. The small signal responses are extrapolated by minimising

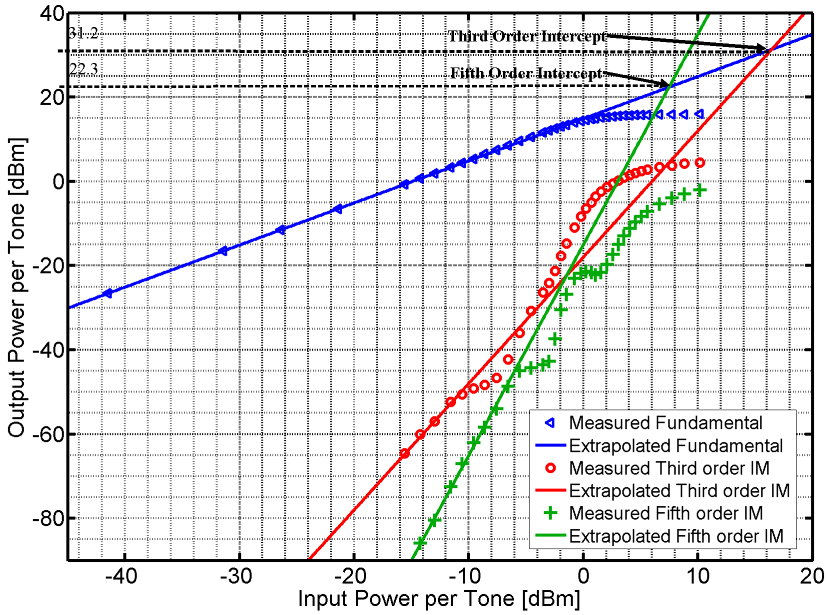


Figure 5.41: Hittite amplifier intercept points with two-tone excitation at 2.14 GHz spaced 100kHz apart.

Table 5.3: Measured amplifier nonlinear parameters.

G_s	14.85dB
P_{1dB} (referred to output)	17.95dB$_m$
P_{1dB} (referred to input)	4.1dB$_m$
P_{5dB} (referred to output)	19dB$_m$
TOI (referred to output)	31.2dB$_m$
FOI (referred to output)	22.3dB$_m$

the mean square perpendicular distance between the low power measurements and the corresponding lines displayed in Figure 5.41. The third and fifth order intercept points (TOI and FOI) are where these lines intercept the fundamental extrapolate and are given in Table 5.3.

5.5.3 Amplifier Behavioural Modelling

The simulation of bit error rates for complex communication systems requires device models which are computationally efficient especially for low error rates which require the simulation of a large number of bits to accomplish accurate results as discussed in Section 4.4. Circuit level simulations generally require impractically large simulation time and a trade-off has to be made between model accuracy and feasible simulation time.

Behavioural modelling is a technique by which the operation of a device is described as a whole. It is aimed at modelling without containing detailed circuit level or device physics knowledge. In this study the ninth order polynomial model, implemented by ADS's *GainRF* and *Ampli-*

Table 5.4: ADS's *RFGain* or *Amplifier2* component parameters.

Parameter	**Value**
S_{21}	$14.85dB$
TOI	31.75dB_m
Psat	19dB_m
GainCompSat	5.0dB_m
GainCompPower	17dB_m

fier2 components, is employed. The paper by Maas and Pedro [57] provides an excellent review of more advanced behavioural modelling techniques.

The ADS components approximate the amplifier's nonlinear response by performing a polynomial fitting of the form

$$y = a_1x + a_3x^3 + a_5x^5 + ... \tag{5.5.1}$$

while its frequency response is considered flat over the signal bandwidth. A ninth order fitting is chosen as it was found to be the most accurate [56]. The polynomial coefficients in (5.5.1) are calculated in order to match the amplifier's figures of merit namely the 1dB and 5dB compression points, the third order intercept point, output saturation power and small signal gain as described in [58]. The measured parameters for the HMC375LP3 amplifier are given in Table 5.4. The model attempts to match these points exactly and has no further knowledge of the amplifier's response. It is however evident from Figures 5.42 to 5.44 that the polynomial approximation is fairly accurate. In order to prevent (5.5.1) from exploding the output is hard limited at very high input power levels which occur beyond the first zero derivative of the polynomial.

Single and two-tone simulations are performed with ADS's harmonic balance simulator in order to compare the behavioural model to measured values. Figure 5.42 demonstrates that the model is in good agreement with the measured fundamental output for single tone excitation. The fundamental gain modelling is more apparent from Figure 5.43 which illustrates that the linear and saturation regions are modelled very accurately but the model is unable to anticipate the slight increased gain in the nonlinear transition area. The model comparison for two-tone excitation is given by Figure 5.44. The general trend of the third and fifth order intermodulation products is followed by the behavioural model and is considered satisfactory for studying the general effect of amplifier saturation.

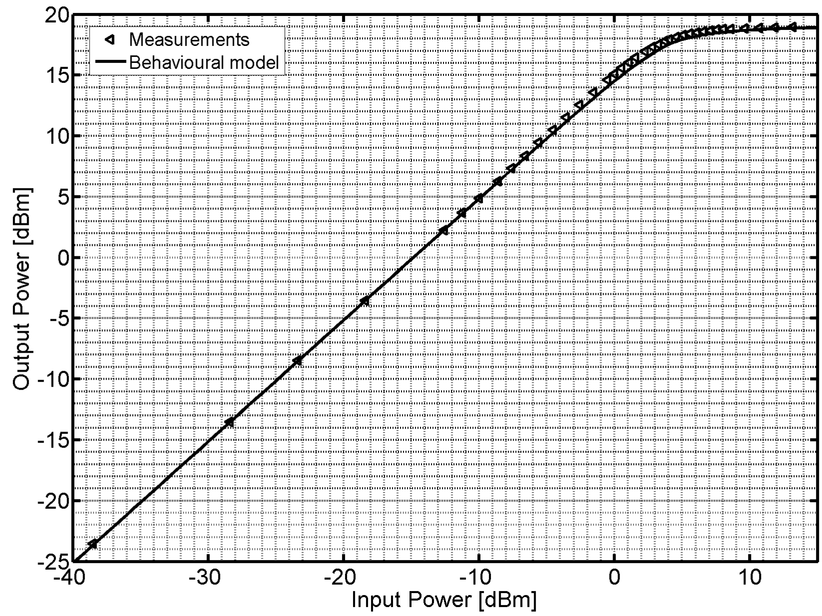


Figure 5.42: Amplifier behavioural model approximation of the fundamental tone output with single tone excitation.

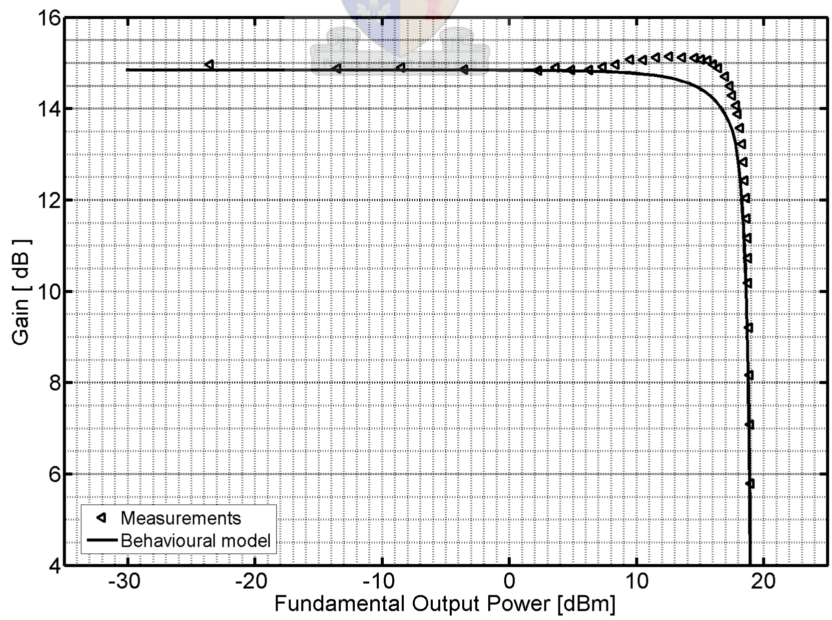


Figure 5.43: Amplifier behavioural model approximation of the fundamental gain with single tone excitation.

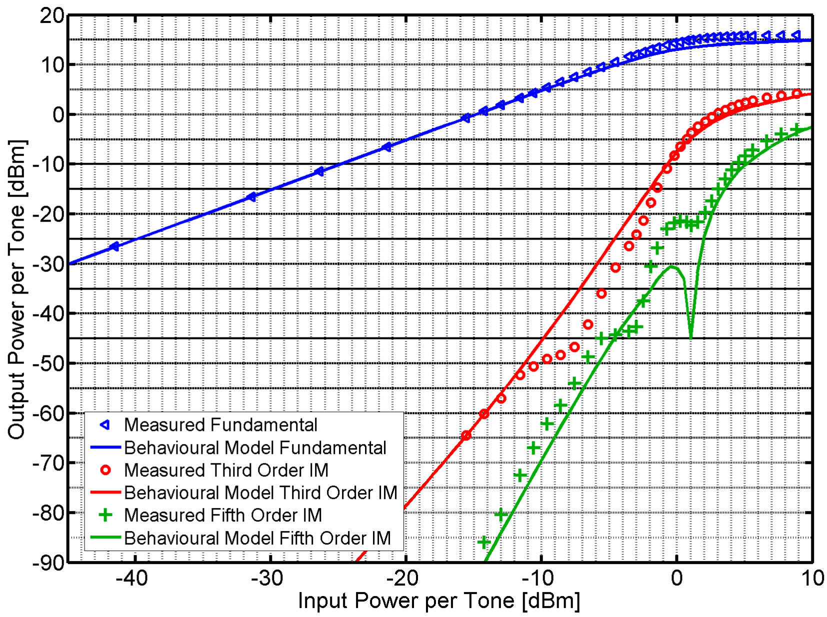


Figure 5.44: Amplifier intermodulation behavioural model approximation for two-tone excitation.

5.5.4 Bit Error Rate Simulations

An OFDM signal has a large peak-to-average power ratio as discussed in [9] and it is therefore difficult to determine the amount of distortion energy in the signal by simply considering the level of the intermodulation products as given by Figure 5.44.

The polynomial amplifier model is used in BER simulations in order to establish the influence of amplifier saturation on OFDM communication systems.

A band-limited RF signal envelope is subjected to intermodulation distortion as a result of nonlinear amplifier behaviour. Intermodulation products are produced at frequencies of any linear combination of the input signal frequencies. Most of these intermodulation products fall outside of the desired signal bandwidth and when the signal is reasonably narrow-band the only intermodulation products that may fall within the signal bandwidth are of odd-order [59].

In the simulation domain only the signal complex envelope is adequately sampled and not the actual RF signal. The amplifier model acts on the sampled modulating envelope and only predicts the effect of the nonlinearity in the signal bandwidth around the carrier frequency. The amplifier was approximated by an odd-order polynomial (5.5.1) because even-order intermodulation products typically fall outside the RF signal bandwidth and would be suppressed by filtering.

Simulation techniques do exist by which the signals located at frequencies around harmonics of the carrier frequency may be accurately computed. This can simply be done by over-sampling the RF signal (and not only the modulating envelope) to an extent that harmonics of the carrier frequency can be represented. Another and more efficient technique is using a mixed harmonic

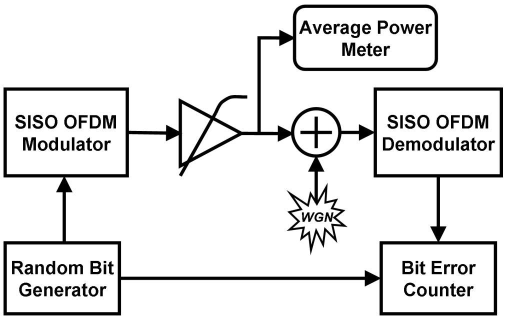


Figure 5.45: Simulation setup for nonlinear transmit amplifier in a SISO system.

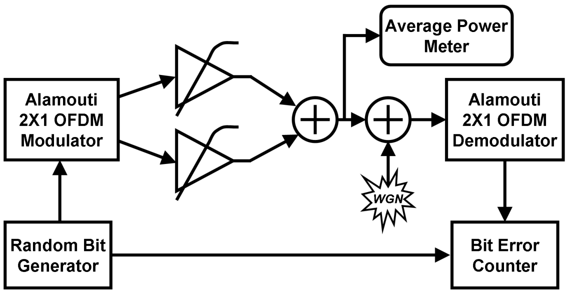


Figure 5.46: Simulation setup for nonlinear transmit amplifiers in a Alamouti STBC 2X1 system.

balance and time domain simulation which is implemented by commercial packages such as Agilent's ADS *Circuit Envelope*® simulator.

5.5.4.1 Transmitter Saturation

The effect of a nonlinear amplifier in the transmitter chain(s) is examined by the simulation setups described in Figures 5.45 and 5.46. The transmitter is assumed noiseless and white Gaussian noise is added at the receiver. It is important to note that half the power is applied to each of the transmit amplifiers of the Alamouti 2X1 system for a fair comparison to the SISO case.

In order to prevent simulation domain aliasing the signal is up-sampled by 9.6 times the minimum sampling rate required for the input signal as discussed in Section 5.5.4.3

Figures 5.47 and 5.48 present the BER simulations for different total average input power levels applied to the transmit amplifier(s). The implementation loss as a function of the total input power to the transmit amplifier(s), at a BER of 10^{-3} is given by Table 5.5. The loss is similar for both systems if the transmit amplifiers were to be driven at the same input power but for the same total transmitted power, the Alamouti transmitters each operate at half power providing superior immunity to transmit amplifier saturation.

The SISO system encounters a 1dB insertion loss when the amplifier is driven at 1dB below $P_{1\text{dB}}$. In the Alamouti system the power is divided between two amplifiers with each driven at 4dB below $P_{1\text{dB}}$ and a deterioration of 0.3dB is observed. The reduced input power at each transmitting element more the compensates for the fact the Alamouti scheme is more sensitive to channel

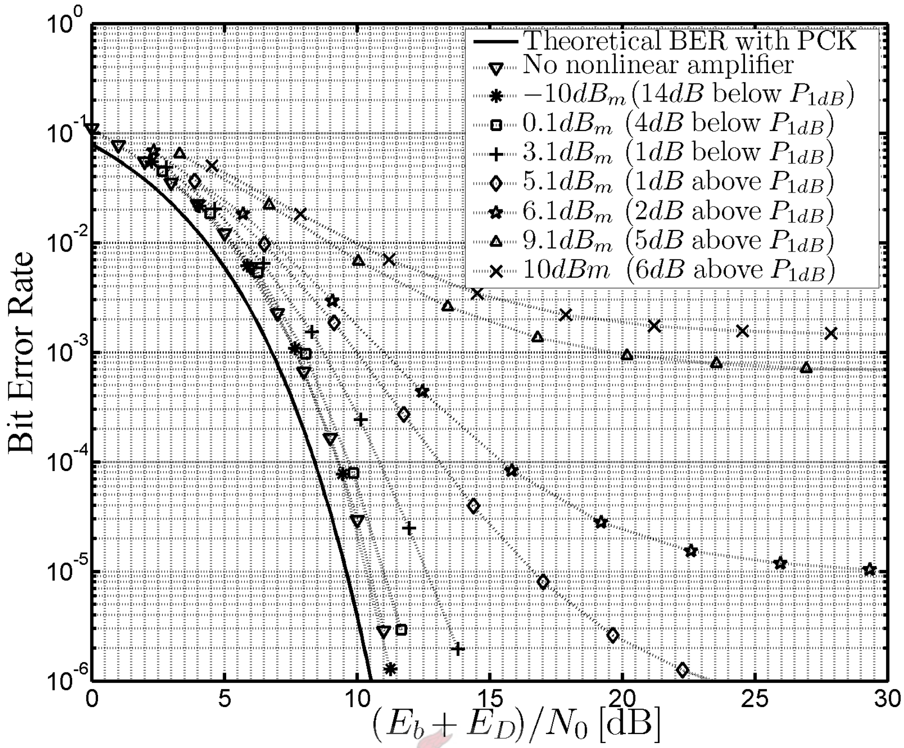


Figure 5.47: SISO BER simulation with nonlinear transmit amplifier for swept input power.

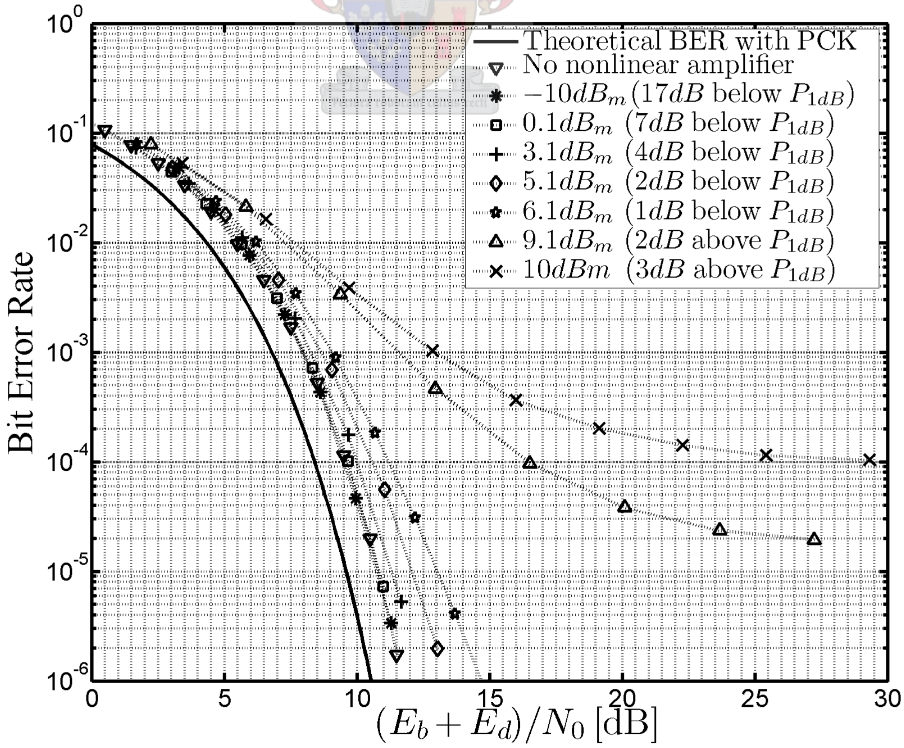


Figure 5.48: Alamouti 2X1 BER simulation with nonlinear transmit amplifier for swept total input power.

Table 5.5: Loss in E_b/N_0 at a BER of 10^{-3} due to amplifier saturation.

Total input power		**-10dB$_\mathrm{m}$**	**0.1dB$_\mathrm{m}$**	**3.1dB$_\mathrm{m}$**	**5.1dB$_\mathrm{m}$**	**6.1dB$_\mathrm{m}$**	**9.1dB$_\mathrm{m}$**	**10dB$_\mathrm{m}$**
SISO	**$\mathrm{P_{in}/P_{1dB}}$**	**-14dB**	**-4dB**	**-1dB**	**1dB**	**2dB**	**5dB**	**6dB**
	IL	$\approx$ 0dB	0.36dB	1.1dB	1.3dB	3.3dB	12dB	-
Alamouti 2X1	**$\mathrm{P_{in}/P_{1dB}}$**	**-17dB**	**-7dB**	**-4dB**	**-2dB**	**-1dB**	**2dB**	**3dB**
	IL	$\approx$ 0dB	$\approx$ 0dB	0.31dB	0.72dB	1.1dB	3.7dB	5.0dB

estimation errors, as demonstrated in Section 5.1.2, which is aggravated by distortion.

With the transmit amplifier(s) driven into saturation, there is no longer analytic control in the simulation environment over the signal power at the receiver and therefore the average received power is measured numerically as portrayed by the average power meters in Figures 5.45 and 5.46. The signal power measured in the receiver is the sum of the useful signal and distortion and thus the abscissa of Figures 5.47 and 5.48, is $(E_b + E_D)/N_0$ with E_b the useful bit energy and E_D the total distortion energy per bit present in the FFT period of one OFDM symbol.

5.5.4.2 Relating Distortion to Equivalent Additive White Gaussian Noise

An OFDM signal consists of independently modulated subcarriers spaced closely in the frequency domain with overlapping spectra. It is therefore postulated that the distortion caused by the nonlinear amplifier may be approximated as white Gaussian noise relative to the narrow bandwidth of an OFDM subcarrier. Hence, the signal emerging from the nonlinear amplifier is regarded as the sum of useful signal energy in the presence of white Gaussian distributed distortion with a power spectral density of D_0. The effective useful bit energy to distortion power spectral density ($\frac{E_b}{D_0}$) is determined by performing a BER simulation of a noiseless system containing the nonlinear amplifier and comparing it to the known linear AWGN bit error rate behaviour as illustrated by the red lines in Figure 5.49. The function for the simulated BER with AWGN at the receiver is referred to as $r_n()$ while $r_{n\&d}()$ refers to the simulated results with AWGN at the receiver together with distortion at the transmitter.

As an example, when the SISO transmit amplifier is driven at $10.1\mathrm{dB_m}$ (6dB above the 1dB compression point) a BER of 1.42×10^{-3} results for a noiseless system. This BER is found on the $r_n\left(\frac{E_b}{N_0+D_0}\right)$ curve at 7.28dB as illustrated by Figure 5.49. Thus the transmitted signal emerging from the nonlinear amplifier is modelled as a useful signal with added noise in the ratio $E_b/D_0 = 7.28\mathrm{dB}$ where D_0 is the distortion spectral density while assuming the distortion is white relative to each subcarrier.

The noiseless bit error rates and corresponding bit energy to distortion spectral densities for a range of input power levels relative to the 1dB compression point is provided by Table 5.6.

As explained previously, the BER simulation results for the nonlinear amplifier were presented as a function of $\frac{E_b+E_d}{N_0}$ as $r_{n\&d}\left(\frac{E_b+E_d}{N_0}\right)$. In order to verify that the distortion has the same effect

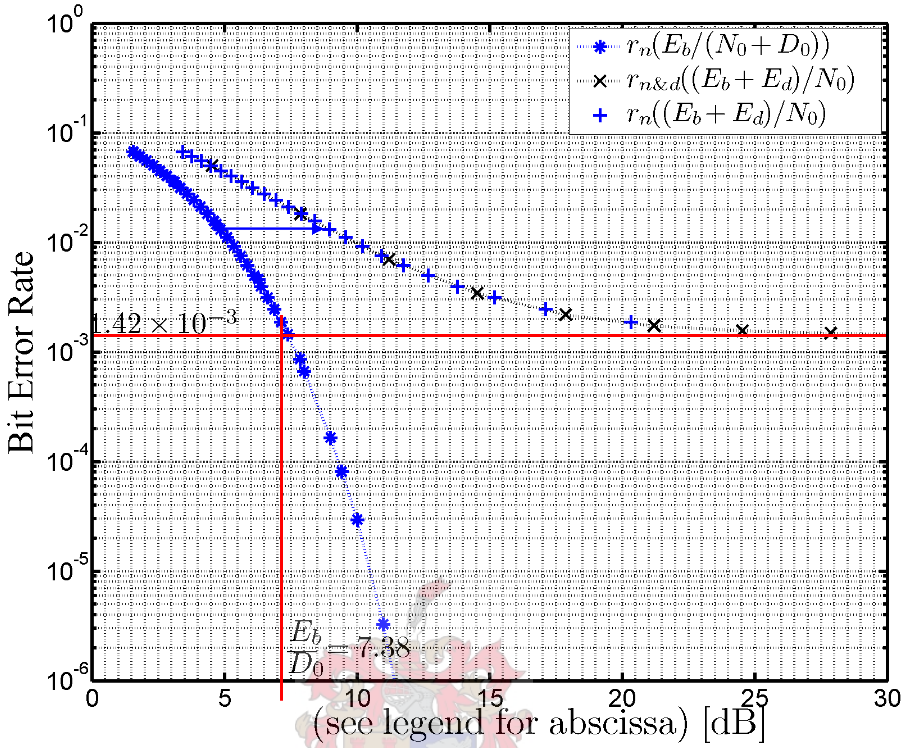


Figure 5.49: Effective bit energy to distortion spectral density for SISO system with non-linear transmit amplifier. $P_T = P_{in} = 10.1\text{dB}_\text{m}$ (6dB above $P_{1\text{dB}}$). The channel response is estimated by means of pilots with $R_p = \frac{16}{9}$. Estimation of timing and frequency offset is performed.

Table 5.6: Loss in E_b/N_0 at a BER of 10^{-3} due to amplifier saturation.

Total input power		**5.1dB$_\text{m}$**	**6.1dB$_\text{m}$**	**9.1dB$_\text{m}$**	**10dB$_\text{m}$**
SISO	$\mathbf{P_{in}/P_{1dB}}$	**1dB**	**2dB**	**5dB**	**6dB**
	BER($N_0 = 0$)	4.48×10^{-7}	9.00×10^{-6}	6.38×10^{-4}	1.42×10^{-3}
	$\mathbf{\frac{E_b}{D_0}}$	11.7dB	10.5dB	8.0dB	7.4dB
Alamouti 2X1	$\mathbf{P_{in}/P_{1dB}}$	**-2dB**	**-1dB**	**2dB**	**3dB**
	BER($N_0 = 0$)	-	-	1.574×10^{-5}	1.014×10^{-4}
	$\mathbf{\frac{E_b}{D_0}}$	-	-	9.54dB	10.5dB

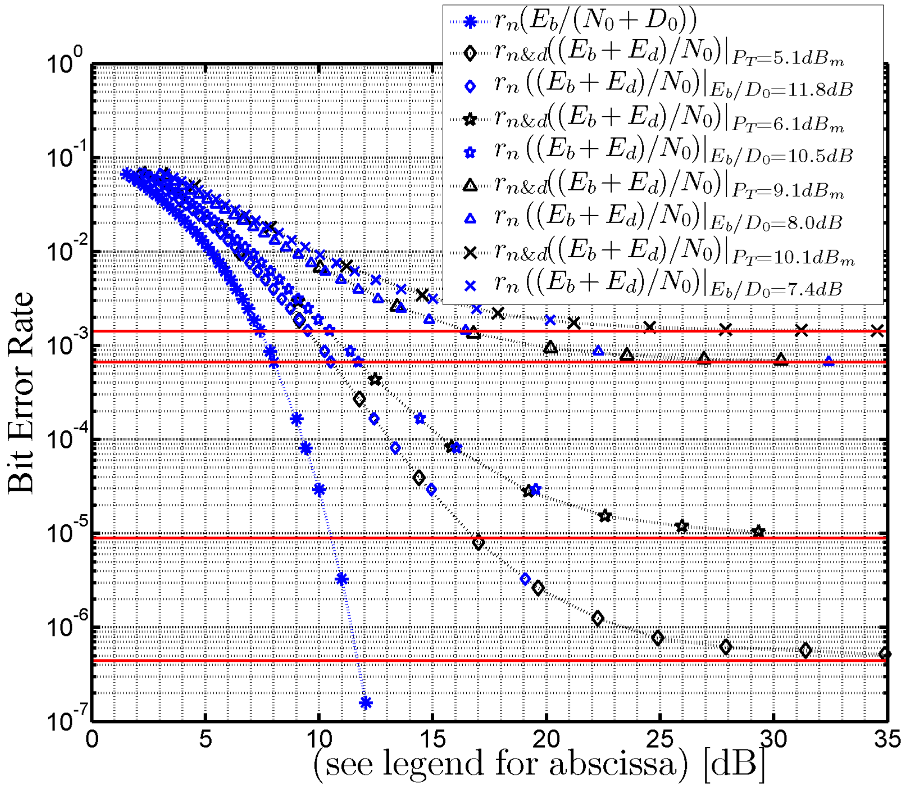


Figure 5.50: Comparison of SISO simulation results with AWGN at the receiver to a system with nonlinear amplifier(s) at the transmitter and AWGN at the receiver. The channel response is estimated by means of pilots with $R_p = \frac{16}{9}$. Timing and frequency offset estimation is performed.

as additive white Gaussian noise at the transmitter, the BER without distortion $r_n\left(\frac{E_b}{N_0+D_0}\right)$ is expressed in terms of $\frac{E_b+E_d}{N_0}$:

$$r_n\left(\frac{E_b+E_d}{N_0}\right) = r_n\left(\frac{\frac{E_b}{N_0+D_0}\left(\frac{E_b}{D_0}+\frac{1}{\log_2(m)}\right)}{\frac{E_b}{D_0}-\frac{E_b}{N_0+D_0}}\right) \tag{5.5.2}$$

where m is the constellation size and for QPSK $m=4$ and $\frac{E_d}{D_0}=\log_2(m)$ is the distortion energy per bit to distortion power spectral density ratio. By comparing $r_n\left(\frac{E_b+E_d}{N_0}\right)$ to $r_{n\&d}(\frac{E_b+E_d}{N_0})$ in Figures 5.50 and 5.51, it is verified that representing the distortion as AWGN at the transmitter is an accurate approximation for both SISO and Alamouti systems. The significance of this finding is that the effect that a saturating transmit amplifier has on the system performance can be characterised by performing a single noiseless simulation. The simulation results for AWGN at the receiver can then simply be transformed to include the saturating amplifier.

5.5.4.3 Aliasing

In the performance evaluations preceding that of the nonlinear amplifier, 1705 carriers were modulated by means of a 2048 point IFFT giving a sampling ratio of 1.2 times the RF bandwidth (or

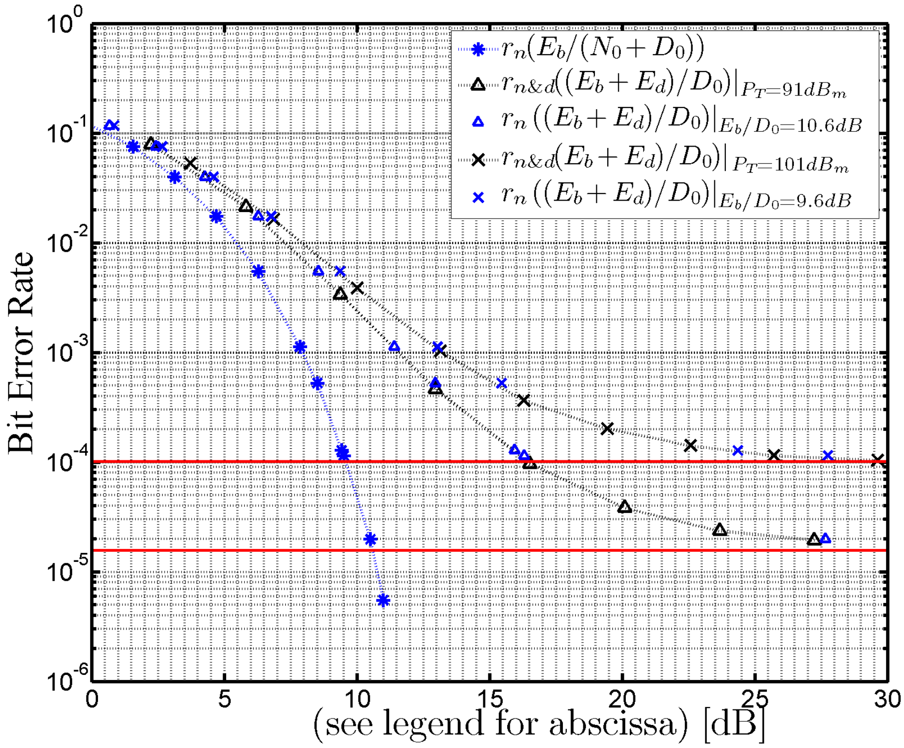


Figure 5.51: Comparison of Alamouti 2X1 simulation results with AWGN at the receiver to a system with nonlinear amplifier(s) at the transmitter and AWGN at the receiver. The channel response is estimated by means of pilots with $R_p = \frac{16}{9}$. Timing and frequency offset estimation is performed.

equivalently 2.4 times the low-pass modulating envelope bandwidth). This sampling rate is sufficient as long as the operations on the signal is linear or the nonlinear effects investigated do not spread the signal spectrum extensively.

A higher sampling rate is required for the simulations with the amplifier component. The ninth order polynomial model, used to approximate the amplifier's nonlinear response, results in a smearing effect in the frequency domain as the signal spectrum is effectively convolved with itself. The signal envelope emerging from the amplifier model is spread over a spectral band of up to nine times the input signal bandwidth as the highest order of the polynomial model is nine. Adequate simulation bandwidth (a sufficiently high sampling rate) is required in order to represent the effects of the nonlinear amplifier without aliasing.

A typical system would employ filtering before sampling the signal in order to prevent aliasing. The designer of a practical system may be free to choose the sampling rate and could implement any one of a large range of anti-aliasing filter responses. The system performance would depend on the specific filter response and chosen sampling rate. In order to produce the "best case" reference, it is desired to perform the simulations so that the effects of aliasing and of the filter response are eliminated. The signal is therefore oversampled in the simulations by inserting zeros into the OFDM-generating IFFT vector. With the signal adequately sampled, the distortion does not

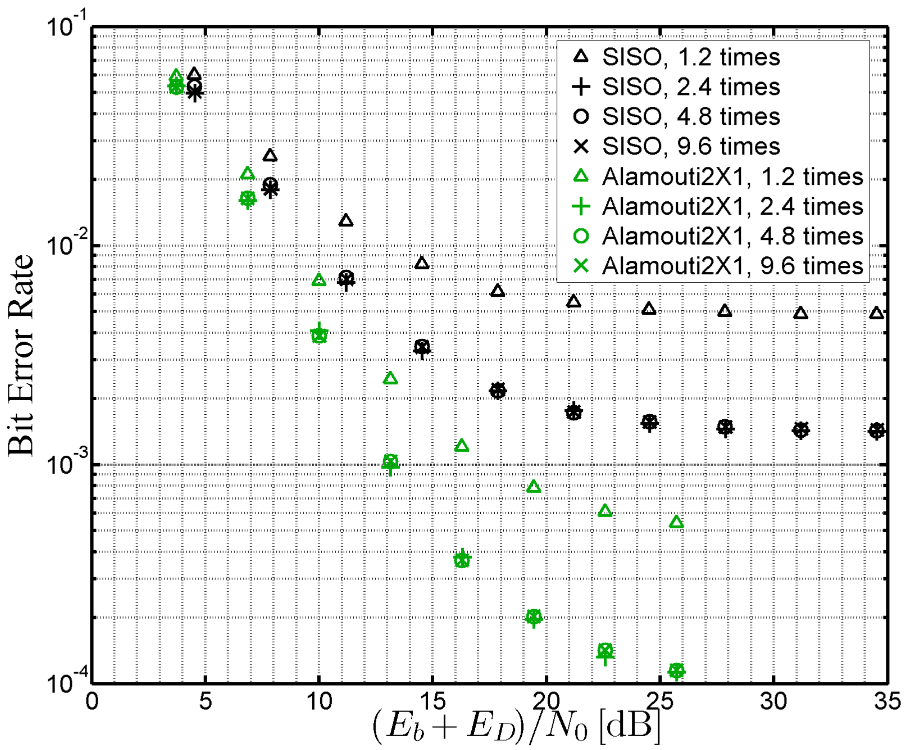


Figure 5.52: BER simulation results for different sampling ratios for the nonlinear amplifier driven at $10.1\mathrm{dB_m}$ (6dB above the 1dB compression point)

cause simulation domain aliasing. Filtering is also not required in the simulation domain when demodulating the signal with an extended FFT (with the same length as the zero-padded OFDM-generating IFFT).

Figure 5.52 illustrates the effect of aliasing due to insufficient sampling when a total signal power of $10.1\mathrm{dB_m}$ is passed through the nonlinear amplifier(s). It is evident that a sampling rate of 1.2 times the minimum required rate for sampling the input signal is insufficient. Insignificant aliasing occurs at a sample ratio of 2.4 times as the results compares well to the sampling ratio of 9.6 times (2^{14} point IFFT with 1705 subcarriers).

5.5.5 Conclusions

A memoryless behavioural model of a nonlinear amplifier was created from measurements with reasonable accuracy.

Bit error rate simulations were performed for an amplifier in the transmit module(s) and the deterioration due to a memoryless saturating amplifier for SISO and Alamouti OFDM systems was demonstrated. Despite its increased sensitivity to channel estimation errors, the deterioration was found to be the least for Alamouti systems due to the transmitted power being split among the two amplifiers.

There were concerns about how far the transmit amplifiers can be driven into saturation by a typical OFDM signal due to the large average to peak power ratio. It was found that the performance deterioration is negligible as long as the average input power is less than 4dB below $P_{1\mathrm{dB}}$ or roughly below where the 3rd order intermodulation product is 30dB below the fundamental.

The hypothesis that the nonlinear distortion of an OFDM signal could be considered as white Gaussian noise which is added at the transmitter was shown to be valid by transforming the simulation results of AWGN at the receiver.

Most nonlinear components contain some form of memory which results in amplitude modulation to phase modulation (AM-to-PM) conversion [60]. Detailed knowledge of the device physics or more sophisticated measuring equipment which can measure the phase of harmonics is required in order to create an accurate model and may be considered for future research.

5.6 Summary

This chapter highlighted some important considerations when designing an OFDM system together with the use of a space-time block code.

SISO, MRC and Alamouti OFDM schemes were evaluated by means of simulation which commenced with the simplified case where the systems have prefect channel knowledge in the presence of AWGN. The evaluation was systematically evolved into a complete system that includes channel response estimation and synchronisation.

The simulation results provide guidelines for choosing suitable OFDM parameters. For example, the minimum possible subcarrier spacing is determined by the maximum anticipated Doppler spread in the channel as well as oscillator stability. On the other hand the channel delay spread, which causes frequency selectivity, dictates the maximum subcarrier spacing. The guard time must be chosen adequately in order to absorb all the delay spread while still providing adequate redundancy for frequency offset estimation.

A static channel with AWGN was considered in Section 5.1. The channel response estimation was evaluated for different pilot power levels and it was illustrated that an optimal pilot to data subcarrier power ratio exists. The BER degradation due to estimating the channel in the presence of noise was revealed and it was found that the Alamouti systems are the most sensitive to channel estimation errors.

A joint frequency and timing offset estimation technique was assessed in AWGN and it was shown to bring about no noticeable deterioration.

An idealised Rayleigh fading channel was introduced in section 5.2 and the simulation results compare well to theoretical predictions. The deterioration due to channel response estimation was revealed and the timing and frequency offset estimation again exhibited no observable implementation loss.

A delay spread fading channel with Doppler shift was evaluated in Section 5.3. The maximum Doppler shift relative to the subcarrier spacing was used as a normalised measure of receiver speed. Similar results were obtained for a receiver with low mobility as for the idealised fading channel and the consequences of travelling at higher speeds were revealed. The Alamouti systems were found to be superior to systems with the same amount of receivers but only one transmitter provided the mobile receiver speed remains low enough so that the maximum Doppler shift does not exceed 4.5% of the subcarrier spacing.

A frequency offset between the transmitting and receiving front-ends was investigated in Section 5.4. The Alamouti system displayed significantly greater sensitivity to a frequency offset than the

SISO case. The operation of frequency offset estimation was established as practically ideal for correcting offsets of up to 50% of the subcarrier spacing.

A nonlinear amplifier was modelled and the consequence of driving it into saturation at the transmitter was examined in Section 5.5. It was revealed by simulation that the distortion has an effect equivalent to white Gaussian noise which is added at the transmitter. This effect was anticipated due to the nature of an OFDM signal which has closely spaced subcarriers that are independently modulated.

Despite its increased sensitivity to channel estimation errors, the Alamouti scheme experiences the least deterioration as a result of the transmit power being distributed among two amplifiers.

Chapter 6

Conclusions and Recommendations

6.1 Summary

The modelling of OFDM communication systems with comprehensive channel estimation and synchronisation algorithms was introduced in Chapter 2.

An overview of MIMO was given in Chapter 3 and the information theoretical advantages of using multiple transmit and receive arrays were illustrated. The integration of maximum ratio combining and Alamouti's orthogonal space-time block code with OFDM together with channel estimation and synchronisation was performed. Numerical simulation techniques that express signals and systems as equivalent low-pass complex envelopes were discussed in Chapter 4. The statistical properties of estimating bit error rates by means of simulation and the measure of confidence in the simulation was introduced. A radio channel for a mobile receiver in a delay spread channel was derived with the capability of generating multiple uncorrelated fading channels.

The communication systems were evaluated in Chapter 5 and simulations were initially performed under simplified conditions which produced results that coincide with theoretical predictions. Effects of channel estimation were investigated for numerous channel states and the simultaneous orthogonal pilot scheme for Alamouti systems was tested and shown to perform adequately. The synchronisation algorithm, which exploits the redundancy of the guard interval provided outstanding frequency offset correction and exhibited excellent performance under all channel conditions investigated. Rayleigh fading was investigated for the ideal fading case and eventually a delay- and Doppler spread was added. The consequence of increasing the mobile receiver's speed was examined and it was established that a practical Alamouti system could outperform a system with only one transmitter provided the maximum Doppler shift remains below approximately 4.5% of the OFDM subcarrier spacing.

A nonlinear amplifier model was created from measurements and evaluated at the transmit side while driven into saturation. The Alamouti scheme experienced less deterioration than the SISO case due to the fact that the transmitted power is divided among two amplifiers and the resulting distortion to the OFDM signal was shown to be equivalent to AWGN at the transmitter for both systems.

6.2 Future Research Recommendations

The communication systems were evaluated for the specific OFDM parameters used in DVB-T (2k mode) [1]. A multitude of possible parameter combinations exist and other applications may require an OFDM signal which is structured differently depending on the environment in which it operates. Performance differences may result due to, among others differently spaced pilot signals, a change in the guard time and in the amount of subcarriers. A change in one or more of these parameters can readily be implemented by the simulation domain models to determining the sensitivity of the system to such changes. For example, the synchronisation algorithm performance for different guard lengths and number of OFDM subcarriers would be valuable.

Uncorrelated Rayleigh channel phenomena for omnidirectional antenna elements with no coupling were investigated. When the elements are directional, the fading power spectral density changes and is no longer the classical U-shaped Jakes spectrum [25]. Different antenna patterns could be considered in an attempt to improve system performance. The channel links no longer fade independently when the environment has less multi-path. Fading correlation is also introduced by the mutual coupling between antenna array elements and this may significantly degrade system performance. An investigation into the impact of semi-correlated fading would have practical significance.

The frequency offset estimation that was implemented is capable of correcting for frequency offsets of up to half the OFDM subcarrier spacing. A coarse frequency offset estimation technique which can extend the estimation range without introducing training symbols or added complexity at the transmitter would make for interesting research. Such a system was suggested in Section 2.4.1.2. The idea is to exploits the fact that the demodulating FFT produces “zero” elements and a coarse estimate could be made after the fine frequency offset estimation and requires further investigated.

A comprehensive investigation of MIMO communication systems that rely on the orthogonality of the links in the channel in order to regain the transmitted vector symbol such as the BLAST [32, 33] algorithms is another interesting field for further research.

Appendices

Appendix A

Simultaneous Orthogonal Channel Estimation

The linear processing described in Section 3.7 and given by (A.1) and (A.2) is used for extracting the channel estimates $h_{0\ est}$ and $h_{1\ est}$ while employing simultaneous orthogonal pilots.

$$\begin{aligned} h_{0\ est} &= \frac{S_{p0}^{*} r_{p0} - S_{p1} r_{p1}}{|S_{p0}|^2 + |S_{p1}|^2} \\ &= h_0 + \frac{S_{p0}^{*} n_0 - S_{p1} n_1}{|S_{p0}|^2 + |S_{p1}|^2} \end{aligned} \tag{A.1}$$

$$\begin{aligned} h_{1\ est} &= \frac{S_{p1}^{*} r_{p0} + S_{p0} r_{p0}}{|S_{p0}|^2 + |S_{p1}|^2} \\ &= h_1 + \frac{S_{p1}^{*} n_0 + S_{p0} n_1}{|S_{p0}|^2 + |S_{p1}|^2} \end{aligned} \tag{A.2}$$

It is shown that the noise terms of the two estimates are uncorrelated for arbitrary complex pilot symbols meaning that

$$\begin{gathered} E[\Re(S_{p0}^{*} n_0 - S_{p1} n_1)\Re(S_{p1}^{*} n_0 + S_{p0} n_1)] = 0 \\ \cup \\ E[\Im(S_{p0}^{*} n_0 - S_{p1} n_1)\Im(S_{p1}^{*} n_0 + S_{p0} n_1)] = 0 \end{gathered} \tag{A.3}$$

while $n_0 \in \mathcal{CN}(0, \sigma^2)$ and $n_1 \in \mathcal{CN}(0, \sigma^2)$ are independent zero mean circular symmetric complex Gaussian random variables with variance σ^2.

In order to show that (A.3) holds, let the pilot symbols

$$S_{p0} = a + jb$$

$$S_{p1} = c + jd$$

where a, b, c and d are arbitrary constants and

$$n_0 = e + jf$$
$$n_1 = g + jh$$

with $e, f, g,$ and f independent zero mean Gaussian random variables with variance $\sigma^2/2$.

$$\begin{aligned}
E[\Re(S_{p0}^* n_0 - S_{p1} n_1)\Re(S_{p1}^* n_0 + S_{p0} n_1)] &= E[(ae + bf - cg + dh)(ce + df + ag - bh)] \\
&= E[ace^2 + bdf^2 - acg^2 - bdh^2] \\
&\quad + E[(ad + cb)ef + (a^2 - c^2)eg + (ab + dc)eh \\
&\quad + (ab - dc)fg + (-b^2 + d^2)fh + (cd + ad)gh] \\
&= \tfrac{1}{2}[ac\sigma^2 + bd\sigma^2 - ac\sigma^2 - bd\sigma^2] + 0 \\
&= 0
\end{aligned} \tag{A.4}$$

similarly

$$\begin{aligned}
E[\Im(S_{p0}^* n_0 - S_{p1} n_1)\Im(S_{p1}^* n_0 + S_{p0} n_1)] &= E[(ae + bf - cg + dh)(ce + df + ag - bh)] \\
&= E[acf^2 + bde^2 - ach^2 - bdg^2] \\
&\quad + E[\text{weighted sum of cross-products} \\
&\quad \text{between uncorrelated noise terms}] \\
&= 0
\end{aligned} \tag{A.5}$$

Appendix B

Pilot to Data Subcarrier Power Distribution

An optimal power distribution between the data and pilot subcarriers is expected and is derived in an approximate manner for a SISO OFDM system as follows:

A received pilot symbol from any pilot subcarrier is described as

$$r_p = hs_p + n_p \tag{B.1}$$

resulting in the channel estimation

$$\begin{aligned} \tilde{h} &= \frac{r_p}{s_p} \\ &= h + \frac{n_p}{s_p} \end{aligned} \tag{B.2}$$

where n_p is AWGN and s_p the transmitted pilot which is known to the receiver. A received data symbol from any subcarrier is given by

$$r_d = hs_d + n_d. \tag{B.3}$$

The transmitted signal is recovered by multiplying the received signal by the complex conjugate of the channel estimation and delivers

$$\begin{aligned} \tilde{s}_d &= \tilde{h}^* r_d \\ &= (h + \frac{n_p}{s_p})^*(hs_d + n_d) \\ &= |h|^2 s_d + h^* n_d + \frac{hs_d n_p^*}{s_p^*} + \frac{n_p n_d}{s_p} \end{aligned} \tag{B.4}$$

The pilot symbols are transmitted at an increased power relative to the data symbols with $|s_p| = \sqrt{R_P}|s_d|$ and because the noise in (B.4) is complex circular symmetric, the recovered symbol can be written as

$$\tilde{s}_d = |h|^2 s_d + h n_d + \frac{h n_p}{\sqrt{R_P}} + \frac{n_p n_d}{\sqrt{R_P} s_d}. \tag{B.5}$$

For large SNR the contribution of the last term of (B.5) is ignored and the signal-to-noise power ratio is approximated as

$$\text{SNR} \approx \frac{|h|^4 P_d}{|h|^2 \sigma^2 + \frac{|h|^2 \sigma^2}{R_p}} \tag{B.6}$$

where P_d is the transmitted data symbol power and σ^2 the variance of each noise term. The aim is to find an R_P which minimises the SNR. By differentiating (B.6) and finding the roots with respect to R_P, the SNR is found to be a minimum when

Appendix C

Source Code of Simulation Components

C.1 QPSK Encoder (*QPSKCoder*)

```
defstar {
    name {QPSKCoder}
    domain {SDF}
    desc {QPSK Encoder}
    version    {@(#) $Source: <DIR>/OFDM/QPSKCoder.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Encode data bits to QPSK symbols.}

    input{
        name{InData}
        type{int}
    }
    output{
        name{OutData}
        type{complex}
    }
    private {
        double scale;
        double OutReal;
        double OutImag;
    }
    protected {
    }
    constructor{
    }
    setup {
        scale=1/sqrt(2);
        InData.setSDFParams(2,1);
        OutData.setSDFParams(1,0);
    }
    go{
        if(int(InData%1)==0){
            OutReal=scale;
        }
        else{
            OutReal=-scale;
        }
        if(int(InData%0)==0){
            OutImag=scale;
        }
        else{
            OutImag=-scale;
        }
        OutData%0 << Complex(OutReal,OutImag);
    }
    destructor {
    }
}
```

C.2 OFDM Multiplexer (*OFDMMux*)

```
defstar {
    name {OFDMMux}
    domain {SDF}
    desc {OFDM Multiplexer}
    version    {@(#) $Source: <DIR>/OFDM/OFDMMux.pl $
        $Revision: 0.4 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Prepares the OFDM symbol with data, pilots and TPS.
    Note:    The data symbols must be normalised so that E[c*conj(c)]=1
        to ensure the correct pilot to data power ratio.}

    input{
        name{InData}
        type{complex}
    }
    output{
        name{OutData}
        type{complex}
    }
    defstate {
        name {Carriers}
        type {int}
        default{1705}
        units {UNITLESS_UNIT}
        desc {Number of OFDM carriers}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Data}
        type {int}
        default{1512+17}
        units {UNITLESS_UNIT}
        desc{Number of data carriers (not including TPS)
            in one OFDM symbol}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate{
        name{PilotAmplitude}
        type{float}
        default{4/3}
        units{UNITLESS_UNIT}
        desc{Pilot signal amplitude}
        attributes{A_SETTABLE|A_CONSTANT}
    }
    private {
        int numCP;    //number of continuous pilot
        //const int numTPS;    //number of TPS
    }
    protected {
        int* CP;
        //int* TPSPos;
        bool* pilotPos;

        int OFDMSymbolCount;
        bool* PRBS;
        int regLength;
    }
    constructor{
        CP=0;
        //TPSPos=0;
        pilotPos=0;
        numCP=45;
    }
    setup {
        int i;
        //numTPS=17;
```

```
    //int TPS[]={34,50,209,346,413,
    ///             569,595,688,790,901,
    //             1073,1219,1262,1286,1469,
    //             1594,1687};
    LOG_DEL; delete [] CP;
    LOG_NEW; CP=new int[numCP];
    int tmp[]={0,48,54,87,141,156,192,
        201,255,279,282,333,432,450,
        483,525,531,618,636,714,759,
        765,780,804,873,888,918,939,
        942,969,984,1050,1101,1107,1110,
        1137,1140,1146,1206,1269,1323,1377,
        1491,1683,1704};
    for(i=0;i<numCP;i++){
        CP[i]=tmp[i];
    }
    //LOG_DEL; delete [] TPSPos;
    //LOG_NEW; TPSPos=new int[int(Carriers)];
    LOG_DEL; delete [] pilotPos;
    LOG_NEW; pilotPos=new bool[int(Carriers)*68];

    //initialise position vectors
    //TPSPos contains a 1 if the position corresponds to TPS
    //for(i=0;i<int(Carriers);i++){
    //    TPSPos[i]=0;
    //}
    //for(i=0;i<numTPS;i++){
    //    TPSPos[TPS[i]]=1;
    //}

    //setup PRBS registry
    regLength=11;
    LOG_NEW; PRBS=new bool[int(Carriers)];
    //Generate the Pseudo Random Bit Sequence
    getPRBS(PRBS,regLength,int(Carriers));

    //Initialise pilot position matrix for 68 symbol frame
    OFDMSymbolCount=0;
    getPilotPos(pilotPos);

    //setup port data block sizes
    InData.setSDFParams(int(Data),int(Data)-1);
    OutData.setSDFParams(int(Carriers),int(Carriers)-1);
}
go {
    int i;
    int j;
    int dataCount=int(Data)-1;

    // note: particle at maximum delay is the first one
    for(i=0;i<int(Carriers);i++){
        //check if carrier is a pilot
        //if(pilotPos[OFDMSymbolCount*int(Carriers)+i]!=0){
        j=OFDMSymbolCount*int(Carriers)+i;
        if(pilotPos[j]!=0){
            //assign pilot at augmented power ratio
            OutData%(int(Carriers)-1-i) << double(PilotAmplitude)*2*(0.5-PRBS[i]);
        }
        //check if carrier is a TPS
        //else if(TPSPos[i]==1){
        //    OutData%(int(Carriers)-1-i) <<5; //2*(0.5-PRBS[i]);
        //}
        //insert data
        else{
            OutData%(int(Carriers)-1-i) << Complex(InData%dataCount);
            dataCount--;
        }
    }
  if(OFDMSymbolCount>=67){
    OFDMSymbolCount=0;
  }
  else{
    OFDMSymbolCount++;
  }
}
```

```
  destructor {
   //LOG_DEL; delete [] TPSPos;
   LOG_DEL; delete [] pilotPos;
   LOG_DEL; delete [] PRBS;
   LOG_DEL; delete [] CP;
  }
  method{
    //Determines the pilot positions and stores the values in pilotList[]
    name {getPilotPos}
    access {private}
    arglist {"(bool* const pilotList)"}
    type {void}
    code {
      int p;
      int i;
    int k;
    int symNum;
      int mod;
    for(symNum=0;symNum<68;symNum++){
      mod=symNum%4;
      p=0;
      for(i=0;i<int(Carriers);i++){
        pilotList[symNum*int(Carriers)+i]=0;
      }
       while(1){//indecate position of Scattered Pilots
        k=3*mod+12*p;
        if (k>=int(Carriers)){//check if all pilots found
          break;
        }
        pilotList[symNum*int(Carriers)+k]=1;
        p++;
      }
      for(i=0;i<numCP;i++){//indicate Continuous Pilot positions
        pilotList[symNum*int(Carriers)+CP[i]]=1;
      }
    }
    return;
   }
  }
  method{
    name{getPRBS}
     access{private}
        arglist{"(bool* const PRBS,const int regLength,int PRBSLength)"}
     type{void}
     code{
     int i;
     int k;
    int rOn;
      int rOff;
    LOG_NEW; int * reg=new int[regLength];
    for(i=0;i<regLength;i++){
         reg[i]=1;
      }
      for(i=0;i<PRBSLength;i++){
         rOff=reg[regLength-1];
         PRBS[i]=rOff;
         rOn=(rOff^reg[9-1])&int(1);
         //shift the register
         for(k=regLength-1;k>0;k--){
        reg[k]=reg[k-1];
         }
         reg[0]=rOn;
       }
    LOG_DEL; delete [] reg;
      return;
    }
  }
}
```

C.3 Zero Padding (*OFDMZeroPad*)

```
defstar {
  name {OFDMZeroPad}
```

```
domain {SDF}
desc {OFDM zero padder}
version    {@(#) $Source: <DIR>/OFDM/OFDMZeroPad.pl $
    $Revision: 0.1 $}
author {Shaun Schulze}
location {OFDM} // Name of Library used in the Schematic
explanation {Zero-pads OFDM array}

input{
    name{InData}
    type{complex}
}
output{
    name{OutData}
    type{complex}
}
defstate {
    name {Carriers}
    type {int}
    default{1705}
    units {UNITLESS_UNIT}
    desc {Number of OFDM carriers}
    attributes {A_SETTABLE | A_CONSTANT}
}
defstate {
    name {FFTSize}
    type {int}
    default{2048}
    units {UNITLESS_UNIT}
    desc {Size of the FFT}
    attributes {A_SETTABLE|A_CONSTANT}
}
private {
    int M; //FFTSize
    int N; //Number of Carriers
    int MpL; //array position of the last positive symbol in FFT vector
    int MzF; //array position of first zero in FFT vector
    int MzL; //array position of last zero in FFT vector
}
protected {
}
constructor{
}
setup {
    M=int(FFTSize);
    N=int(Carriers);
    if(M<N) {
        FFTSize.rangeError(">= Carriers");
    }
    if(N%2==0){
        Carriers.rangeError("odd.");
    }
    MpL=(N-1)/2;
    MzF=(N+1)/2;
    MzL=(M-MzF);
    //setup port data block sizes
    InData.setSDFParams(N,N-1);
    OutData.setSDFParams(M,M-1);
}
go {
    int i;

    // note: particle at maximum delay is the first one
    for(i=0;i<M;i++){
        if(i<=MpL){
            //place symbols on positive Frequencies
            OutData%(M-1-i) << Complex(InData%(N-1-(MpL+i)));
        }
        else if((MzF<=i)&&(i<=MzL)){
            //insert zeros;
            OutData%(M-1-i) << Complex(0);
        }
        else{
            //insert symbols on negative frequencies
            OutData%(M-1-i) << Complex(InData%(N-1-(i-M+MpL)));
```

```
            }
        }
    }
    destructor {
    }
}
```

C.4 Guard Insertion (*OFDMInsertGuard*)

```
defstar {
    name {OFDMInsertGuard}
    domain {SDF}
    desc {OFDM guard interval insertor}
    version    {@(#) $Source: <DIR>/OFDM/OFDMInsertGuard.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Insert the OFDM cyclic prefix.}

    input{
        name{InData}//first block of length Data must equal r0
                    //while the second block equals r1
        type{Complex}
    }
    output{
        name{OutData}
        type{Complex}
    }
    defstate {
        name {InputLength}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {Length of input block}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate {
        name {GuardLength}
        type {int}
        default{128}
        units {UNITLESS_UNIT}
        desc {Length of guard interval}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        int G;
        int L;
    }
    protected {
    }
    constructor{
    }
    setup {
        L=int(InputLength);
        G=int(GuardLength);
        InData.setSDFParams(L,L-1);
        OutData.setSDFParams(G+L,G+L-1);
    }
    go{
        int i;
        for(i=0;i<(int(InputLength)+int(GuardLength));i++){
            if(i<G){
                OutData%(L+G-1-i)<<Complex(InData%(L-1-(L-G+i)));
            }
            else
            {
                OutData%(L+G-1-i)<<Complex(InData%(L-1-(i-G)));
            }
        }
    }
    destructor {
    }
}
```

C.5 Maximum Likelihood Timing and Frequency Offset Estimation

C.5.1 Correlator (*OFDMMLSync*)

```
defstar {
    name {OFDMMLSync}
    domain {SDF}
    desc {OFDM maximum likelihood timing and frequency offset estimator.}
    explanation {OFDM maximum likelihood timing and frequency offset estimator.
                 note: Output is delayed by N+L samples}
    version    {@(#) $Source: <DIR>/OFDM/OFDMMLSync.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InData}
        type{Complex}
    }
    output{
        name{OutTheta}
        type{float}
    }
    output{
        name{OutEps}
        type{float}
    }
    defstate {
        name {FFTSize}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {FFT length}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Guard}
        type {int}
        default{128}
        units {UNITLESS_UNIT}
        desc {Guard length}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate {
        name {SNR}
        type {float}
        default{19}
        units {UNITLESS_UNIT}
        desc {Signal-to-noise ratio}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate {
        name {EstimateSNR}
        type {int}
        default{0}
        range{{0,1}}
        units {UNITLESS_UNIT}
        desc {Activate signal-to-noise ratio estimation}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        Complex gamma;
        double phi_2;
        Complex sigPowEst;
        double sigNoisePowEst;
        Complex * corrVec;
        double * energyVec;
        int blockCount;
        double energyFirst;
        Complex corrFirst;
    }
    protected {
```

```
    int L;
    int N;
    double p;
    double pi;
    Complex * dataIn;
}
constructor{
    dataIn=0;
    corrVec=0;
    energyVec=0;
}
setup {
    if (int(Guard) < 1) {
        Guard.rangeError(">= 1");
    }
    if (int(FFTSize) < 2) {
        FFTSize.rangeError(">= 2");
    }
    pi=asin(1)*2;
    L=int(Guard);
    N=int(FFTSize);
    p=double(SNR)/(double(SNR)+1.0);
    LOG_DEL; delete [] dataIn;
    LOG_NEW; dataIn=new Complex[2*(N+L)];
    LOG_DEL; delete [] corrVec;
    LOG_NEW; corrVec=new Complex[N+2*L];
    LOG_DEL; delete [] energyVec;
    LOG_NEW; energyVec=new double[2*(N+L)];
    gamma=0;
    phi_2=0;
    int m;
    for(m=0;m<N+L;m++){
        energyVec[m]=0;
        corrVec[m]=0;
    }
    blockCount=0;
    energyFirst=0;
    //setup port data block sizes
    InData.setSDFParams(N+L,2*(N+L)-1);
    OutTheta.setSDFParams(N+L,N+L-1);
    OutEps.setSDFParams(N+L,N+L-1);
}
go{
    int i;
    int k;
    for(i=0;i<2*(N+L);i++){
        dataIn[i]=Complex(InData%(2*(N+L)-1-i));
    }
    sigPowEst=0;
    sigNoisePowEst=0;
    if(blockCount!=0){
        for(k=0;k<N+2*L;k++){
            corrVec[k]=dataIn[k]*conj(dataIn[k+N]);
            energyVec[k]=abs(dataIn[k])*abs(dataIn[k]);
        }
        for(k=N+2*L;k<2*(N+L);k++){
            energyVec[k]=abs(dataIn[k])*abs(dataIn[k]);
        }
        if(int(EstimateSNR)!=0){
            for(k=0;k<N+L;k++){
                sigPowEst=sigPowEst+corrVec[k];
                sigNoisePowEst=sigNoisePowEst+energyVec[k];
            }
            p=double(N+L)*abs(sigPowEst)/(double(L)*sigNoisePowEst+double(L)*abs(sigPowEst));
        }
        for(k=0;k<N+L;k++){
            if(k==0&&blockCount==1){
                gamma=0;phi_2=0;
                for(i=0;i<L;i++){
                    gamma=gamma+corrVec[k+i];
                    phi_2=phi_2+energyVec[k+i]+energyVec[k+i+N];
                }
                blockCount++;
            }
            else{
```

```
                gamma=gamma-corrFirst+corrVec[k+L-1];
                phi_2=phi_2-energyFirst+energyVec[k+L-1]+energyVec[k+L-1+N];
            }
            corrFirst=corrVec[k];
            energyFirst=energyVec[k]+energyVec[k+N];
            OutTheta%(N+L-1-k)<<real(abs(gamma)-p*0.5*phi_2);
            OutEps%(N+L-1-k)<<-real(arg(gamma))/2.0/pi;
        }
    }
    else{
        blockCount++;
        for(k=0;k<N+L;k++){
            OutTheta%(N+L-1-k)<<0;
            OutEps%(N+L-1-k)<<0;
        }
    }
}
destructor {

    LOG_DEL; delete [] dataIn;
    LOG_DEL; delete [] corrVec;
    LOG_DEL; delete [] energyVec;
}
}
```

C.5.2 Peak Search Algorithm (*OFDMMLPeakSearch*)

```
defstar {
    name {OFDMMLPeakSearch}
    domain {SDF}
    desc {Search for the peaks of ML estimator.}
    explanation {Search for the peaks of ML estimator.
                note: Input must be delayed by N+L samples.
                    Output is delayed by a total of 2*(N+L) samples}
    version    {@(#) $Source: <DIR>/OFDM/OFDMPeakSearch.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InTheta}
        type{float}
    }
    input{
        name{InEps}
        type{float}
    }
    output{
        name{OutThetaML}
        type{int}
    }
    output{
        name{OutEpsML}
        type{float}
    }
    defstate {
        name {FFTSize}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {FFT length}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Guard}
        type {int}
        default{128}
        units {UNITLESS_UNIT}
        desc {Guard length}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        int blockCount;
        int thisPeak;
```

```
        int nextPeak;
        double thisEpsML;
        double nextEpsML;
        bool searchCentre;
    }
    protected {
        int L;
        int N;
        int middleStart;
        double * thetaInFirst;
        double * thetaInLast;
        double * thetaInMiddle;
        double * epsIn;
    }
    constructor{
        thetaInFirst=0;
        thetaInLast=0;
        thetaInMiddle=0;
        epsIn=0;
    }
    setup {
        if (int(Guard) < 1) {
            Guard.rangeError(">= 1");
        }
        if (int(FFTSize) < 2*int(Guard)) {
            FFTSize.rangeError(">= 2*Guard");
        }
        L=int(Guard);
        N=int(FFTSize);
        middleStart=(N+L)/2-1;
        LOG_DEL; delete [] thetaInFirst;
        LOG_DEL; delete [] thetaInLast;
        LOG_DEL; delete [] thetaInMiddle;
        LOG_DEL; delete [] epsIn;
        LOG_NEW; thetaInFirst=new double[N+L];
        LOG_NEW; thetaInLast=new double[N+L];
        LOG_NEW; thetaInMiddle=new double[N+L];
        LOG_NEW; epsIn=new double[2*(N+L)];

        //setup port data block sizes
        InTheta.setSDFParams(N+L,2*(N+L)-1);
        InEps.setSDFParams(N+L,2*(N+L)-1);
        OutThetaML.setSDFParams(1,0);
        blockCount=0;
        thisPeak=0;
        nextPeak=0;
        thisEpsML=0;
        nextEpsML=0;
        searchCentre=0;
    }
    go {
        int i;
        for(i=0;i<N+L;i++){
            thetaInFirst[i]=double(InTheta%(2*(N+L)-1-i));
            thetaInLast[i]=double(InTheta%(N+L-1-i));
            thetaInMiddle[i]=double(InTheta%(2*(N+L)-1-middleStart-i));
            epsIn[i]=double(InEps%(2*(N+L)-1-i));
        }
        for(i=N+L;i<2*(N+L);i++){
            epsIn[i]=double(InEps%(2*(N+L)-1-i));
        }

        if(blockCount==0||blockCount==1){
            blockCount++;
            OutThetaML%0<<0;
            OutEpsML%0<<0;
            return;
        }
        else if(blockCount==2){
            //very first symbol to be synchronised
            thisPeak=searchPeak(thetaInFirst,0,N+L-1);
            blockCount++;
            if(thisPeak<L||thisPeak>N){
                //peak occurs in first or last L samples
                //peak may be false!
```

```
                nextPeak=searchPeak(thetaInMiddle,0,N+L-1)+middleStart-N-L;
                searchCentre=1;
                nextEpsML=epsIn[nextPeak+N+L];
                // In case the first full guard inteval was not received entirely
                // rely on the next symbol peak for synchronization.
                thisPeak=nextPeak;
                // Issue warning that first symbol may not have been received
                // completely and its data may be in error.
                Error::warn(*this,"First symbol synchronization lies in first L samples
                        thus symbol may not have been received entirely");
            }
            OutThetaML%0<<thisPeak;
            OutEpsML%0<<epsIn[thisPeak];
            return;
        }
        else if(blockCount>2){
            if(searchCentre){
                thisPeak=nextPeak;
                nextPeak=searchPeak(thetaInMiddle,0,N+L-1)+middleStart-N-L;
                thisEpsML=nextEpsML;
                nextEpsML=epsIn[nextPeak+N+L];
            }
            else{
                thisPeak=searchPeak(thetaInFirst,0,N+L-1);
                thisEpsML=epsIn[thisPeak];
            }
            OutThetaML%0<<thisPeak;
            OutEpsML%0<<thisEpsML;
        }
    }
    destructor {
        LOG_DEL; delete [] thetaInFirst;
        LOG_DEL; delete [] thetaInLast;
        LOG_DEL; delete [] thetaInMiddle;
        LOG_DEL; delete [] epsIn;
    }
    method{
        name{searchPeak}
        access{private}
        arglist{"(double* const data,int start,int stop)"}
        type{int}
        code{
        int i;
        double max=data[start];
        int maxIndex=start;
        for(i=start;i<=stop;i++){
            if(data[i]>max){
                max=data[i];
                maxIndex=i;
            }
        }
        return maxIndex;
        }
    }
}
```

C.6 Guard Removal (*OFDMMLRemoveGuard*)

```
defstar {
    name {OFDMMLRemoveGuard}
    domain {SDF}
    desc {Remove the OFDM guard inteval.}
    explanation {Remove the OFDM guard inteval. To be used with OFDMPeakSearch.
                note: Input data has to be delay by 2*(N+L) samples.
                    Output data has total delay of 3N samples.}
    version    {@(#) $Source: <DIR>/OFDM/OFDMMLRemoveGuard.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InData}
        type{Complex}
```

```
    }
    input{
        name{InThetaML}
        type{int}
    }
    output{
        name{OutData}
        type{Complex}
    }
    defstate {
        name {FFTSize}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {FFT length}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Guard}
        type {int}
        default{128}
        units {UNITLESS_UNIT}
        desc {Guard length}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        int blockCount;
      }
    protected {
        int N;
        int L;
    }
    constructor{
    }
    setup {
        if (int(Guard) < 1) {
            Guard.rangeError(">= 1");
        }
        if (int(FFTSize) < 2*int(Guard)) {
            FFTSize.rangeError(">= 2*Guard");
        }
        L=int(Guard);
        N=int(FFTSize);
        blockCount=0;
        //setup port data block sizes
        InData.setSDFParams(N+L,3*(N+L)-1);
        InThetaML.setSDFParams(1,1);
        OutData.setSDFParams(N,N-1);
    }
    go{
        int i;
        if(blockCount<3){
            for(i=0;i<N-1;i++){
                OutData%(N-1-i)<<0;
            }
            blockCount++;
        }
        else{
            for(i=0;i<N-1;i++){
                OutData%(N-1-i)<<Complex(InData%(2*(N+L)-1-int(InThetaML%1)-L-i));
            }
        }
    }
    destructor{
    }
}
```

C.7 Frequency offset removal (*OFDMMLRemoveFreqOfsset*)

```
defstar {
    name {OFDMMLRemoveFreqOffset}
    domain {SDF}
    desc {Remove the OFDM guard inteval.}
```

```
    explanation {Remove the OFDM frequency offset. To be used with OFDMPeakSearch}
    version    {@(#) $Source: <DIR>/OFDM/OFDMMLRemoveFreqOffset.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InData}
        type{Complex}
    }
    input{
        name{InEpsML}
        type{float}
    }
    output{
        name{OutData}
        type{Complex}
    }
    defstate {
        name {FFTSize}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {FFT length}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Guard}
        type {int}
        default{128}
        units {UNITLESS_UNIT}
        desc {Guard length}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    private {
        int N;
        int L;
      }
    protected {
        double pi;
        double dt;
        double t;
    }
    constructor{
        pi=asin(1)*2;
    }
    setup {
        //setup port data block sizes
        N=int(FFTSize);
        L=int(Guard);
        InData.setSDFParams(N,N-1);
        InEpsML.setSDFParams(1,0);
        OutData.setSDFParams(N,N-1);
        t=0;
    }
    go{
        dt=double(InEpsML%0)*2*pi/double(N);
        t=t+(double(L)*dt);
        int i;
        for(i=0;i<N;i++){
            OutData%(N-1-i)<<Complex(InData%(N-1-i))*Complex(cos(t),-sin(t));
            t+=dt;
        }
        if(t>2*pi){
            t=t-2*pi;
        }
    }
    destructor{
    }
}
```

C.8 Zero Removal (*OFDMZeroRemove*)

```
defstar {
    name {OFDMZeroRemove}
    domain {SDF}
    desc {OFDM zero remover}
    version    {@(#) $Source: <DIR>/OFDM/OFDMZeroRemove.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Removes the zeros from a zero-padded OFDM array.}

    input{
        name{InData}
        type{complex}
    }
    output{
        name{OutData}
        type{complex}
    }
    defstate {
        name {Carriers}
        type {int}
        default{1705}
        units {UNITLESS_UNIT}
        desc {Number of OFDM carriers}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {FFTSize}
        type {int}
        default{2048}
        units {UNITLESS_UNIT}
        desc {Size of the FFT}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        int M; //FFTSize
        int N; //Number of Carriers
        int MpL; //array position of the last positive symbol in FFT vector
        int MzF; //array position of first zero in FFT vector
        int MzL; //array position of last zero in FFT vector
    }
    protected {
    }
    constructor{
    }
    setup {
        M=int(FFTSize);
        N=int(Carriers);
        if(M<N) {
            FFTSize.rangeError(">= Carriers");
        }
        if(N%2==0){
            Carriers.rangeError("odd.");
        }
        MpL=(N-1)/2;
        MzF=(N+1)/2;
        MzL=(M-MzF);
        //setup port data block sizes
        InData.setSDFParams(M,M-1);
        OutData.setSDFParams(N,N-1);
}
    go {
        int i;

        // note: particle at maximum delay is the first one
        for(i=0;i<M;i++){
            if(i<=MpL){
                //insert symbols from positive Frequencies
                OutData%(N-1-(MpL+i)) << Complex(InData%(M-1-i));
            }
            else if(i>MzL){
                //insert symbols from negative frequencies
                OutData%(N-1-(i-M+MpL)) << Complex(InData%(M-1-i));
```

```
            }
        }
    }
    destructor {
    }
}
```

C.9 Channel Response Estimation (*OFDMChEst*)

```
defstar {
    name {OFDMChEst}
    domain {SDF}
    desc {OFDM channel estimator}
    explanation {Linear 2D channel estimation and removes pilots from the OFDM symbol}
    version    {@(#) $Source: <DIR>/OFDM/OFDMChEst.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InData}
        type{complex}
    }
    output{
        name{OutData}
        type{complex}
    }
    output{
        name{OutEst}
        type{complex}
    }
    defstate {
        name {Carriers}
        type {int}
        default{1705}
        units {UNITLESS_UNIT}
        desc {Number of OFDM carriers}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Data}
        type {int}
        default{1512+17}
        units {UNITLESS_UNIT}
        desc {Number of data carriers in OFDM symbol}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate {
        name {PilotAmplitude}
        type {float}
        default{4/3}
        units {UNITLESS_UNIT}
        desc {Pilot to data voltage ratio}
        attributes {A_SETTABLE|A_CONSTANT}
    }

    private {
        int numCP;    //number of continuous pilot
        //const int numTPS;    //number of TPS
        int pilotFirstPos;
        int pilotLastPos;
        Complex pilotFirst; Complex pilotLast;
        double pilotXFirst; double pilotXLast; double pilotYFirst; double pilotYLast;
        double pilotAmpFirst; double pilotAmpLast;
        double pilotPhaseFirst; double pilotPhaseLast;
        double estAmpFirst; double estAmpLast;
        double estPhaseFirst; double estPhaseLast;
        double diffEstPhase;
        double gradPhase; double gradAmp;
        double estAmp; double estPhase;
        double gradEstAmp; double gradEstPhase;
        Complex estimate;
        double pi;
```

```
        double pilotSum;
      }
    protected {
        int* CP;
        //int* TPSPos;
        bool* pilotPos;
        int* pilotPosNum;
        int OFDMSymbolCount;
        bool* PRBS;
        int regLength;
    }
    constructor{
        CP=0;
        //TPSPos=0;
        pilotPos=0;
        pilotPosNum=0;
        numCP=45;
        pi=asin(1)*2;
    }

    setup {
        int i;
        int j;
        //numTPS=17;
        //int TPS[]={34,50,209,346,413,
        ///            569,595,688,790,901,
        //            1073,1219,1262,1286,1469,
        //            1594,1687};

        LOG_DEL; delete [] CP;
        LOG_NEW; CP=new int[numCP];
        int tmp[]={0,48,54,87,141,156,192,
                    201,255,279,282,333,432,450,
                    483,525,531,618,636,714,759,
                    765,780,804,873,888,918,939,
                    942,969,984,1050,1101,1107,1110,
                    1137,1140,1146,1206,1269,1323,1377,
                    1491,1683,1704};
        for(i=0;i<numCP;i++){
            CP[i]=tmp[i];
        }

        //LOG_DEL; delete [] TPSPos;
        //LOG_NEW; TPSPos=new int[int(Carriers)];
        LOG_DEL; delete [] pilotPos;
        LOG_NEW; pilotPos=new bool[int(Carriers)*68];
        LOG_DEL; delete [] pilotPosNum;
        LOG_NEW; pilotPosNum=new int[(int(Carriers)-int(Data))*68];

        //initialise position vectors
        //TPSPos contains a 1 if the position corresponds to TPS
        for(i=0;i<int(Carriers);i++){
        //    TPSPos[i]=0;
        }
        //for(i=0;i<numTPS;i++){
        //    TPSPos[TPS[i]]=1;
        //}

        //setup PRBS registry
        regLength=11;
        LOG_NEW; PRBS=new bool[int(Carriers)];
        //Generate the Pseudo Random Bit Sequence
        getPRBS(PRBS,regLength,int(Carriers));

        OFDMSymbolCount=0;
        //Initialise pilot position matrix for 68 symbol frame
        getPilotPos(pilotPos);//pilotPos contains 1 if carrier is a pilot else 0

        //generate a vector containing the pilot indeces
        int* tmpP=pilotPosNum;
        for(j=0;j<68;j++)
            for(i=0;i<int(Carriers);i++){
                if(pilotPos[i+j*int(Carriers)]==1){
                    tmpP[0]=i;
                    if(j!=67||i!=int(Carriers)){
```

```
                    tmpP++;
                }
            }
        }
        //setup port data block sizes
        InData.setSDFParams(int(Carriers),int(Carriers)-1);
        OutData.setSDFParams(int(Data),int(Data)-1);
        OutEst.setSDFParams(int(Data),int(Data)-1);
        pilotSum=0;
    }
    go {
        int i;
        //identify the first two pilot positions
        pilotFirstPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))];
        pilotLastPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))+1];

        //keep track of the last pilot taken from the pilotPosNum vector
        int pilotLastPosCount=1;

        int dataOutCount=0;
        while(1){
            pilotFirst=Complex(InData%(int(Carriers)-1-pilotFirstPos));
            pilotLast=Complex(InData%(int(Carriers)-1-pilotLastPos));
            pilotSum=pilotSum+abs(pilotFirst.real());
            pilotXFirst=pilotFirst.real();
            pilotXLast=pilotLast.real();
            pilotYFirst=pilotFirst.imag();
            pilotYLast=pilotLast.imag();
            pilotAmpFirst=sqrt(pilotXFirst*pilotXFirst+pilotYFirst*pilotYFirst);
            pilotAmpLast=sqrt(pilotXLast*pilotXLast+pilotYLast*pilotYLast);
            if(pilotAmpFirst==0){
                pilotPhaseFirst=0;
            }
            else{
                pilotPhaseFirst=atan2(pilotYFirst,pilotXFirst);
            }
            if(pilotAmpLast==0){
                pilotPhaseLast=0;
            }
            else{
                pilotPhaseLast=atan2(pilotYLast,pilotXLast);
            }

            estAmpFirst=pilotAmpFirst/double(PilotAmplitude);
            estAmpLast=pilotAmpLast/double(PilotAmplitude);
            estPhaseFirst=pilotPhaseFirst-pi*PRBS[pilotFirstPos];
            estPhaseLast=pilotPhaseLast-pi*PRBS[pilotLastPos];
            diffEstPhase=estPhaseLast-estPhaseFirst;
            // Make sure phase lies between (-pi,pi] radians.
            if(diffEstPhase<=-pi){
                diffEstPhase=diffEstPhase+2*pi;
            }
            else if(diffEstPhase>pi){
                diffEstPhase=diffEstPhase-2*pi;
            }
            gradEstAmp=(estAmpLast-estAmpFirst)/(pilotLastPos-pilotFirstPos);
            gradEstPhase=(diffEstPhase)/(pilotLastPos-pilotFirstPos);

            //Create linear interpolation of channel estimate.
            for(i=pilotFirstPos+1;i<pilotLastPos;i++){
                estAmp=gradEstAmp*(i-pilotFirstPos)+estAmpFirst;
                estPhase=gradEstPhase*(i-pilotFirstPos)+estPhaseFirst;
                // note: particle at maximum delay is the first one
                OutEst%(int(Data)-1-dataOutCount)<<Complex(estAmp*cos(estPhase),estAmp*sin(estPhase));
                OutData%(int(Data)-1-dataOutCount)<<Complex(InData%(int(Carriers)-1-i));
                dataOutCount++;
            }
            if(pilotLastPos==int(Carriers)-1){
                break;
            }
            else{
                pilotLastPosCount++;
                pilotFirstPos=pilotLastPos;
                pilotLastPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))+pilotLastPosCount];
            }
```

```
    }
    if(OFDMSymbolCount>=67||pilotSum==0){
        OFDMSymbolCount=0;
    }
    else{
        OFDMSymbolCount++;
    }
    pilotSum=0;
}
destructor {
    //LOG_DEL; delete [] TPSPos;
    LOG_DEL; delete [] pilotPos;
    LOG_DEL; delete [] PRBS;
    LOG_DEL; delete [] CP;
    LOG_DEL; delete [] pilotPosNum;
}
method {
    //Determines the pilot positions and stores the values in pilotList[]
    name {getPilotPos}
    access {private}
    arglist {"(bool* const pilotList)"}
    type {void}
    code {
        int p;
        int i;
        int k;
        int symNum;
        int mod;

        for(symNum=0;symNum<68;symNum++){
            mod=symNum%4;
            p=0;
            for(i=0;i<int(Carriers);i++){
            pilotList[symNum*int(Carriers)+i]=0;
            }
              while(1){//indecate position of Scattered Pilots
                k=3*mod+12*p;
                if (k>=int(Carriers)){//check if all pilots found
                    break;
                }
                pilotList[symNum*int(Carriers)+k]=1;
                p++;
            }
            for(i=0;i<numCP;i++){//indicate Continuous Pilot positions
                pilotList[symNum*int(Carriers)+CP[i]]=1;
            }
        }
        return;
    }
}
method{
    name{getPRBS}
    access{private}
    arglist{"(bool* const PRBS,const int regLength,int PRBSLength)"}
    type{void}
    code{
        int i;
        int k;
        bool rOn;
        bool rOff;
        LOG_NEW; bool * reg=new bool[regLength];
        for(i=0;i<regLength;i++){
            reg[i]=1;
        }
        for(i=0;i<PRBSLength;i++){
            rOff=reg[regLength-1];
            PRBS[i]=rOff;
            rOn=(rOff^reg[9-1])&1;
            //shift the register
            for(k=regLength-1;k>0;k--){
                reg[k]=reg[k-1];
            }
            reg[0]=rOn;
        }
        LOG_DEL; delete [] reg;
```

```
            return;
        }
    }
}
```

C.10 Alamouti Channel Response Estimation (*OFDMAlamoutiChEst*)

```
defstar {
    name {OFDMAlamoutiChEst}
    domain {SDF}
    desc {OFDM Alamouti channel estimator}
    explanation {Linear channel estimation with simulateous Alamouti encoded pilots
        and removes pilots from the OFDM symbol}
    version    {@(#) $Source: <DIR>/OFDM/OFDMAlamoutiChEst.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic

    input{
        name{InData}
        type{complex}
    }
    output{
        name{OutData}
        type{complex}
    }
    output{
        name{OutEst}
        type{complex}
    }
    defstate {
        name {Carriers}
        type {int}
        default{1705}
        units {UNITLESS_UNIT}
        desc {Number of OFDM carriers}
        attributes {A_SETTABLE | A_CONSTANT}
    }
    defstate {
        name {Data}
        type {int}
        default{1512+17}
        units {UNITLESS_UNIT}
        desc {Number of data carriers in OFDM symbol}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    defstate {
        name {PilotAmplitude}
        type {float}
        default{4/3}
        units {UNITLESS_UNIT}
        desc {Pilot to data voltage ratio}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        int numCP;    //number of continuous pilot
        //const int numTPS;    //number of TPS
        int pilotFirstPos;
        int pilotLastPos;
        double sp0First,sp1First, sp0Last, sp1Last;
        Complex rp0First, rp1First, rp0Last, rp1Last;
        Complex estH0First, estH1First, estH0Last, estH1Last;
        double estH0AmpFirst, estH1AmpFirst, estH0AmpLast, estH1AmpLast;
        double estH0PhaseFirst, estH1PhaseFirst, estH0PhaseLast, estH1PhaseLast;
        double diffEstH0Amp, diffEstH1Amp;
        double diffEstH0Phase, diffEstH1Phase;
        double gradEstH0Amp, gradEstH1Amp;
        double gradEstH0Phase, gradEstH1Phase;
        double estH0Amp, estH1Amp;
        double estH0Phase, estH1Phase;
```

```
    double pi;
    double PNorm;
    double pilotSum;
}
protected {
    int* CP;
    //int* TPSPos;
    bool* pilotPos;
    int* pilotPosNum;
    int OFDMSymbolCount;
    bool* PRBS;
    int regLength;
}
constructor{
    CP=0;
    //TPSPos=0;
    pilotPos=0;
    pilotPosNum=0;
    numCP=45;
    pi=asin(1)*2;
}
setup {
    int i;
    int j;
    //numTPS=17;
    //int TPS[]={34,50,209,346,413,
    ///            569,595,688,790,901,
    //            1073,1219,1262,1286,1469,
    //            1594,1687};
    LOG_DEL; delete [] CP;
    LOG_NEW; CP=new int[numCP];
    int tmp[]={0,48,54,87,141,156,192,
               201,255,279,282,333,432,450,
               483,525,531,618,636,714,759,
               765,780,804,873,888,918,939,
               942,969,984,1050,1101,1107,1110,
               1137,1140,1146,1206,1269,1323,1377,
               1491,1683,1704};
    for(i=0;i<numCP;i++){
        CP[i]=tmp[i];
    }
    //LOG_DEL; delete [] TPSPos;
    //LOG_NEW; TPSPos=new int[int(Carriers)];
    LOG_DEL; delete [] pilotPos;
    LOG_NEW; pilotPos=new bool[int(Carriers)*68];
    LOG_DEL; delete [] pilotPosNum;
    LOG_NEW; pilotPosNum=new int[(int(Carriers)-int(Data))*68];
    //initialise position vectors
    //TPSPos contains a 1 if the position corresponds to TPS
    for(i=0;i<int(Carriers);i++){
    //    TPSPos[i]=0;
    }
    //for(i=0;i<numTPS;i++){
    //    TPSPos[TPS[i]]=1;
    //}
    //setup PRBS registry
    regLength=11;
    LOG_NEW; PRBS=new bool[int(Carriers)];
    //Generate the Pseudo Random Bit Sequence
    getPRBS(PRBS,regLength,int(Carriers));

    OFDMSymbolCount=0;
    //Initialise pilot position matrix for 68 symbol frame
    getPilotPos(pilotPos);//pilotPos contains 1 if carrier is a pilot else 0

    //generate a vector containing the pilot indeces
    int* tmpP=pilotPosNum;
    for(j=0;j<68;j++){
        for(i=0;i<int(Carriers);i++){
            if(pilotPos[i+j*int(Carriers)]==1){
                tmpP[0]=i;
                if(j!=67||i!=int(Carriers)){
                    tmpP++;
                }
            }
```

```
        }
    }
    //setup port data block sizes
    InData.setSDFParams(2*int(Carriers),2*int(Carriers)-1);
    OutData.setSDFParams(2*int(Data),2*int(Data)-1);
    OutEst.setSDFParams(2*int(Data),2*int(Data)-1);
    pilotSum=0;
    PNorm=2*double(PilotAmplitude)*double(PilotAmplitude);
}
go {
    int i;
        //identify the first two pilot positions
        pilotFirstPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))];
        pilotLastPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))+1];

        //keep track of the last pilot taken from the pilotPosNum vector
        int pilotLastPosCount=1;

        int dataOutCount=0;
        while(1){
            sp0First=double(PilotAmplitude)*(1-2*PRBS[pilotFirstPos]);
            sp0Last=double(PilotAmplitude)*(1-2*PRBS[pilotLastPos]);
            sp1First=sp0First;
            sp1Last=sp0Last;
            rp0First=Complex(InData%(int(Carriers)*2-1-pilotFirstPos));
            rp0Last=Complex(InData%(int(Carriers)*2-1-pilotLastPos));
            rp1First=Complex(InData%(int(Carriers)-1-pilotFirstPos));
            rp1Last=Complex(InData%(int(Carriers)-1-pilotLastPos));
            estH0First=(sp0First*rp0First-sp1First*rp1First)/PNorm;
            estH0Last=(sp0Last*rp0Last-sp1Last*rp1Last)/PNorm;
            estH1First=(sp1First*rp0First+sp0First*rp1First)/PNorm;
            estH1Last=(sp1Last*rp0Last+sp0Last*rp1Last)/PNorm;
            pilotSum=pilotSum+abs(rp0First);

            estH0AmpFirst=abs(estH0First);
            estH0AmpLast=abs(estH0Last);
            estH1AmpFirst=abs(estH1First);
            estH1AmpLast=abs(estH1Last);

            estH0PhaseFirst=atan2(estH0First.imag(),estH0First.real());
            estH0PhaseLast=atan2(estH0Last.imag(),estH0Last.real());
            estH1PhaseFirst=atan2(estH1First.imag(),estH1First.real());
            estH1PhaseLast=atan2(estH1Last.imag(),estH1Last.real());

            diffEstH0Phase=estH0PhaseLast-estH0PhaseFirst;
            // Make sure phase lies between (-pi,pi] radians.
            if(diffEstH0Phase<=-pi){
                diffEstH0Phase=diffEstH0Phase+2*pi;
            }
            else if(diffEstH0Phase>pi){
                diffEstH0Phase=diffEstH0Phase-2*pi;
            }
            diffEstH1Phase=estH1PhaseLast-estH1PhaseFirst;
            // Make sure phase lies between (-pi,pi] radians.
            if(diffEstH1Phase<=-pi){
                diffEstH1Phase=diffEstH1Phase+2*pi;
            }
            else if(diffEstH1Phase>pi){
                diffEstH1Phase=diffEstH1Phase-2*pi;
            }

            gradEstH0Amp=(estH0AmpLast-estH0AmpFirst)/(pilotLastPos-pilotFirstPos);
            gradEstH0Phase=(diffEstH0Phase)/(pilotLastPos-pilotFirstPos);
            gradEstH1Amp=(estH1AmpLast-estH1AmpFirst)/(pilotLastPos-pilotFirstPos);
            gradEstH1Phase=(diffEstH1Phase)/(pilotLastPos-pilotFirstPos);

            //Create linear interpolation of channel estimate.
            for(i=pilotFirstPos+1;i<pilotLastPos;i++){
                estH0Amp=gradEstH0Amp*(i-pilotFirstPos)+estH0AmpFirst;
                estH1Amp=gradEstH1Amp*(i-pilotFirstPos)+estH1AmpFirst;
                estH0Phase=gradEstH0Phase*(i-pilotFirstPos)+estH0PhaseFirst;
                estH1Phase=gradEstH1Phase*(i-pilotFirstPos)+estH1PhaseFirst;
                // note: particle at maximum delay is the first one
                OutEst%(int(Data)*2-1-dataOutCount)<<Complex(estH0Amp*cos(estH0Phase),
                                                estH0Amp*sin(estH0Phase));
```

```
                OutData%(int(Data)*2-1-dataOutCount)<<Complex(InData%(int(Carriers)*2-1-i));
                OutEst%(int(Data)-1-dataOutCount)<<Complex(estH1Amp*cos(estH1Phase),
                                                estH1Amp*sin(estH1Phase));
                OutData%(int(Data)-1-dataOutCount)<<Complex(InData%(int(Carriers)-1-i));
                dataOutCount++;
            }
            if(pilotLastPos==int(Carriers)-1){
                break;
            }
            else{
                pilotLastPosCount++;
                pilotFirstPos=pilotLastPos;
                pilotLastPos=pilotPosNum[OFDMSymbolCount*(int(Carriers)-int(Data))
                                            +pilotLastPosCount];
            }
        }
        if(OFDMSymbolCount>=67||pilotSum==0){
            OFDMSymbolCount=0;
        }
        else{
            OFDMSymbolCount++;
        }
        pilotSum=0;
}
destructor {
    //LOG_DEL; delete [] TPSPos;
    LOG_DEL; delete [] pilotPos;
    LOG_DEL; delete [] PRBS;
    LOG_DEL; delete [] CP;
    LOG_DEL; delete [] pilotPosNum;
}
method {
    //Determines the pilot positions and stores the values in pilotList[]
    name {getPilotPos}
    access {private}
    arglist {"(bool* const pilotList)"}
    type {void}
    code {
        int p;
        int i;
        int k;
        int symNum;
        int mod;

        for(symNum=0;symNum<68;symNum++){
            mod=symNum%4;
            p=0;
            for(i=0;i<int(Carriers);i++){
                pilotList[symNum*int(Carriers)+i]=0;
            }
              while(1){//indecate position of Scattered Pilots
                k=3*mod+12*p;
                if (k>=int(Carriers)){//check if all pilots found
                    break;
                }
                pilotList[symNum*int(Carriers)+k]=1;
                p++;
            }
            for(i=0;i<numCP;i++){//indicate Continuous Pilot positions
                pilotList[symNum*int(Carriers)+CP[i]]=1;
            }
        }
        return;
    }
}
method{
    name{getPRBS}
    access{private}
    arglist{"(bool* const PRBS,const int regLength,int PRBSLength)"}
    type{void}
    code{
        int i;
        int k;
        bool rOn;
        bool rOff;
```

```
            LOG_NEW; bool * reg=new bool[regLength];
            for(i=0;i<regLength;i++){
                reg[i]=1;
            }
            for(i=0;i<PRBSLength;i++){
                rOff=reg[regLength-1];
                PRBS[i]=rOff;
                rOn=(rOff^reg[9-1])&1;
                //shift the register
                for(k=regLength-1;k>0;k--){
                    reg[k]=reg[k-1];
                }
                reg[0]=rOn;
            }
            LOG_DEL; delete [] reg;
            return;
        }
    }
}
```

C.11 Alamouti Decoder (*OFDMAlamoutiDecode*)

```
defstar {
    name {OFDMAlamoutiDeCode}
    domain {SDF}
    desc {Alamouti Decoder}
    version    {@(#) $Source: <DIR>/OFDM/OFDMAlamoutiDeCode.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Alamouti OFDM channel decoder for alamouti encoded data
                and pilot signal which alternate between transmit antennas
                Performs channel correction.}
    input{
        name{r} //first block of length Data must equal r0 while the second block equals r1
        type{Complex}
    }
    input{
        name{h}    //first block of length Data must equal h0 while the second block equals h1
        type{Complex}
    }
    output{
        name{OutData}
        type{Complex}
    }
    defstate {
        name {Data}
        type {int}
        default{1512+17}
        units {UNITLESS_UNIT}
        desc {Number of carriers in OFDM symbol}
        attributes {A_SETTABLE|A_CONSTANT}
    }
    private {
        Complex r0;
        Complex r1conj;
        Complex s0;
        Complex s1;
        Complex h0;
        Complex h1;

    }
    protected {
    }
    constructor{
    }
    setup {
        r.setSDFParams(int(Data)*2,int(Data)*2-1);
        h.setSDFParams(int(Data)*2,int(Data)*2-1);
        OutData.setSDFParams(int(Data)*2,int(Data)*2-1);
    }
    go{
        int i;
```

```
        for(i=0;i<int(Data);i++){
            h0=Complex(h%(int(Data)*2-1-i));
            h1=Complex(h%(int(Data)-1-i));
            r0=Complex(r%(int(Data)*2-1-i));
            r1conj=conj(Complex(r%(int(Data)-1-i)));
            s0=conj(h0)*r0+h1*r1conj;
            s1=conj(h1)*r0-h0*r1conj;
            OutData%(int(Data)*2-1-i)<<s0;
            OutData%(int(Data)-1-i)<<s1;
        }
    }
    destructor {
    }
}
```

C.12 QPSK Decoder (*QPSKDeCoder*)

```
defstar {
    name {QPSKDeCoder}
    domain {SDF}
    desc {QPSK Decoder}
    version    {@(#) $Source: <DIR>/OFDM/QPSKDeCoder.pl $
        $Revision: 0.1 $}
    author {Shaun Schulze}
    location {OFDM} // Name of Library used in the Schematic
    explanation {Encode decode QPSK symbols to data bits.}

    input{
        name{InData}
        type{Complex}
    }
    output{
        name{OutData}
        type{int}
    }
    private {
        double scale;
        int Out0;
        int Out1;
        double InReal;
        double InImag;
        Complex In;
    }
    protected {
    }
    constructor{
    }
    setup {
        scale=1/sqrt(2);
        InData.setSDFParams(1,0);
        OutData.setSDFParams(2,1);
    }
    go{
        In=Complex(InData%0);
        InReal=In.real();
        InImag=In.imag();
        if(InReal>=0){
            Out0=0;
        }
        else{
            Out0=1;
        }
        if(InImag>=0){
            Out1=0;
        }
        else{
            Out1=1;
        }
        OutData%1 << Out0;
        OutData%0 << Out1;
    }
    destructor {
```

```
    }
}
```

C.13 Bit Error Rate Counter (*BERSink*)

```
defstar{
    name{BERSink}
    domain{SDF}
    desc{BER sink}
    location{Custom Sinks}
    hinclude {"TargetTask.h"}
    ccinclude {"SimData.h" }

    defstate{
        name{Start}
        type {int}
        default{0}
    }
    defstate{
        name{Stop}
        type{int}
        default{1e8}
    }
    defstate{
        name{EstVar}
        type{float}
        default{0.001}
    }
    input{
        name{ref}
        type{int}
    }
    input{
        name{test}
        type{int}
    }
    protected{
        SimData *ber, *bitsProcessed;//, *BER_vs_Index;
        SinkControl sinkControl;
        int *bitErrorVec;

        int numBits, bitErrors, bitErrors1, bitErrors2, errorsToCount, blockCount;
    }
    begin {
        sinkControl.initialize( *this, Start, Stop, 1 );
        numBits=0;
        bitErrors=0;
        bitErrors1=0;
        bitErrors2=0;
        blockCount=0;
        errorsToCount=int(ceil(1.0/EstVar));
    }
    constructor{
        bitErrorVec=0;
    }
    setup{
        bitErrorVec=new int[int(ceil(double(int(Stop))/100000))+1];
    }
    go{
        if(sinkControl.collectData()){
            if(int(ref%0)!=int(test%0)){
                bitErrors++;
            }
            numBits++;

            if(numBits%100000==0){
                StringList msg;
                blockCount++;
                bitErrorVec[blockCount-1]=bitErrors;
                if(blockCount%2==0){
                    bitErrors2=bitErrors2+bitErrors;
                }
                else{
```

```
                    bitErrors1=bitErrors1+bitErrorVec[(blockCount-1)/2];
                    bitErrors2=bitErrors2-bitErrorVec[(blockCount-1)/2]+bitErrors;
                }
                bitErrors=0;
                msg<<"bits processed="<<numBits;
                msg<<", BER = "<<double(bitErrors1+bitErrors2)/double(numBits);
                msg<<", estimate variance = "<<1.0/double(bitErrors1+bitErrors2);
                msg<<", BER1="<<double(bitErrors1)/double(ceil(double(blockCount)/2.0)*100000);
                msg<<", BER2="<<double(bitErrors2)/double(floor(double(blockCount)/2.0)*100000);
                Error::message(*this, msg);
                if((bitErrors1+bitErrors2)>=errorsToCount){
                    sinkControl.stopControl();
                }
            }
        }
    }
    wrapup {
        ber=newSimData(this);
        ber->setIndepVar("Index",AgilentPtolemy::INT,State::UNITLESS_UNIT );
        ber->setDepVar(fullName(),"BER",AgilentPtolemy::FLOAT,State::UNITLESS_UNIT);
        ber->sendData(0,double(bitErrors1+bitErrors2+bitErrors)/double(numBits));
        bitsProcessed=newSimData(this);
        bitsProcessed->setIndepVar("Index",AgilentPtolemy::INT,State::UNITLESS_UNIT);
        bitsProcessed->setDepVar(fullName(),"BitsProcessed",AgilentPtolemy::INT,State::UNITLESS_UNIT );
        bitsProcessed->sendData(0,numBits);
        StringList msg;
        msg<<"total number of bits processed = "<<numBits;
        msg<<", number of bit errors = "<<bitErrors1+bitErrors2;
        msg<<", BER = "<<double(bitErrors1+bitErrors2+bitErrors)/double(numBits);
        msg<<", estimate variance = "<<1.0/double(bitErrors1+bitErrors2+bitErrors);
        Error::message(*this,msg);
    }
    destructor{
        delete [] bitErrorVec;
    }
}
```

Bibliography

[1] *ETSI EN 300 744 v1.5.1; Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, European Telecommunications Standards Institute, November 2004.

[2] *IEEE 802.16 Air Interface for Fixed Broadband Wireless Access Systems, IEEE Standard for Local and metropolitan area network*, October 2004.

[3] *IEEE 802.11a Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High-speed Physical Layer in the 5 GHz Band, IEEE Standard*, November 1999.

[4] *ETSI EN 300 401 v1.3.3; Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*, European Telecommunications Standards Institute, May 2001.

[5] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Feasibility Study for Orthogonal Frequency Division Multiplexing (OFDM) for UTRAN enhancement.*, 6th ed., 3GPP, June 2004.

[6] G. Fettweis and R. Irmer, "WIGWAM: system concept development for 1 Gbit/s air interface," 14th Wireless World Research Forum (WWRF 14), Tech. Rep. [Online]. Available: http://www.ifn.et.tu-dresden.de/MNS/veroeffentlichungen/2005/Fettweis_G_WWRF_05.pdf

[7] "IEEE 802.11n Wireless Local Area Networks." [Online]. Available: http://grouper.ieee.org/groups/802/11/

[8] R. E. Ziemer and W. H. Tranter, *Principles of Communications*, 5th ed. Wiley, 2002.

[9] R. v. Nee and R. Prasad, *OFDM For Wireless Multimedia Communications*. Artech House, 2000.

[10] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing. [Online]. Available: www.dspguide.com

[11] *ETSI EN 300 744 v1.5.1; Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, European Telecommunications Standards Institute, November 2004.

[12] M. D.Lacroix, N.Goudard, "OFDM with guard interval versus OFDM/OffsetQAM for high data rate UMTS downlink transmission," *Vehicular Technology Conference, IEEE*, vol. 4, no. 54, pp. 2682 – 2686, October 2001.

[13] P. H. Moose, "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction," *Communications, IEEE Transactions on*, vol. 42, no. 10, pp. 2908–2914, October 1994.

[14] J.-J. v. d. Beek, M. Sandell, and P. O. Börjesson, "ML Estimation of Time and Frequency Offset in OFDM Systems," *Signal Processing, IEEE Transactions on*, vol. 45, no. 7, pp. 1800–1805, July 1997.

[15] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*. John Wiley & Sons, 1987.

[16] *ADS Ptolemy Simulation*, Agilent Technologies, 395 Page Mill Road Palo Alto, CA 94304 U.S.A., September 2004. [Online]. Available: http://eesof.tm.agilent.com/docs/adsdoc2004A/manuals.htm

[17] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *Communication, IEEE Transactions on*, vol. 45, pp. 1613–1621, December 1997.

[18] M. Morelli and U. Mengali, "An Improved Frequency Offset Estimator for OFDM Applications," *Communications Letters, IEEE*, vol. 3, no. 3, pp. 75–77, March 1999.

[19] H. Nyquist, "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, p. 324, April 1924.

[20] R. V. L. Hartley, "Transmission of Information," *Bell System Technical Journal*, p. 535, July 1928.

[21] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.

[22] G. Foschini and M. Gans, "On Limits of Wireless Communications in a Fading Enviroment when Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, March 1998.

[23] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[24] I. E. Telatar, "Capacity of Multi-antenna Gaussian Channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, November 1999.

[25] W. C. Jakes, *Microwave Mobile Communications*. IEEE Press, 1994.

[26] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*, 4th ed. McGraw-Hill, 2001.

[27] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 1064–1070, May 1993.

[28] R. G. Gallager, "Low density parity check codes," *MIT Press, Number 21 in MIT Research monograph series*, 1963.

[29] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *Select Area In Communications, IEEE Journal On*, vol. 16, no. 8, pp. 1451–1458, October 1998.

[30] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-Time Block Codes from Orthogonal Designs," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, July 1999.

[31] L. Musavian, M. Dohler, R. Nakhai, and H. Aghvami, "Closed Form Capacity Expressions of Orthogonalised Correlated MIMO Channels," *Communications Letters, IEEE*, vol. 8, no. 6, pp. 365–367, June 2004.

[32] G. Foschini, "Layered Space-Time Architecture For Wireless Communication in a Fading Environment When Using Multi-Element Antennas," *Bell Labs Technical Journal*, pp. 41–59, Autumn 1996.

[33] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel," *Signals, Systems, and Electronics, Proceedings of the URSI I International Symposium on*, pp. 295–300, September 1998.

[34] J. G. Proakis, *Digital Communications*, 2nd ed. McGraw-Hill, 1989.

[35] H. Jafarkhani, "A quasi-othogonal space-time block code," *Communications, IEEE Transactions on*, vol. 46, no. 1, pp. 1–4, January 2001.

[36] A. van Zelst, "Space division multiplexing algorithms," *Electrotechnical Conference, 10th Mediterranean, IEEE*, vol. 3, pp. 1218–1221, May 2000.

[37] *Ptolemy II Heterogeneous Concurrent Modeling and Design in Java*, University of California at Berkeley, July 2005. [Online]. Available: http://ptolemy.eecs.berkeley.edu

[38] *Circuit Envelope Simulation*, Agilent Technologies, September 2004. [Online]. Available: http://eesof.tm.agilent.com

[39] H.-S. Yap, "Designing to digital wireless specifications using circuit envelope simulation," HP EEsof Division, Hewlett-Packard, Tech. Rep. [Online]. Available: http://eesof.tm.agilent.com/pdf/ckt_env.pdf

[40] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems: Modeling, Methodology and Techniques*. Plenum Press, 1994.

[41] W. J. Cody, "An Overview of Software Development for Special Functions," *Numerical Analysis Dundee, Conference Proceedings of, Lecture Notes in Mathematics*, vol. 506, 1975.

[42] M. C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems," *Selected Areas in Communication, IEEE Journal on*, vol. 2, no. 1, pp. 153–170, January 1984.

[43] *Recommendation ITU-R M.1225, Guidelines for the Evaluation of Radio Transmission Technologies for IMT-2000 (Question ITU-R 39/8)*, 1997.

[44] Y. R. Zheng and C. Xiao, "Improved Models for the Generation of Multiple Uncorrelated Rayleih Fading Waveforms," *Communications Letters, IEEE*, vol. 6, no. 6, pp. 256–258, June 2002.

[45] R. H. Clarke, "A statistical theory of mobile-radio reception," *The Bell System Technical Journal*, vol. 47, pp. 975–1000, Jul-Aug 1968.

[46] P. Dent, G. E. Bottomley, and T. Croft, "Jakes Fading Model Revisited," *Electronics Letters, IEEE*, vol. 29, no. 13, pp. 1162–1163, June 1993.

[47] Y. Li and X. Huang, "The Simulation of Independent Rayleigh Faders," *Communications, IEEE Transactions on*, vol. 50, no. 9, pp. 1503–1514, September 2002.

[48] M. Patzold and F. Laue, "Statistical properties of Jakes' fading channel simulator," *Vehicular Technology Conference, IEEE*, vol. 2, pp. 712 – 718, May 1998.

[49] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoid fading channel simulators," *Communications, IEEE Transactions on*, vol. 49, pp. 699–708, April 2001.

[50] *ETSI EN 300 910, Digital cellular telecommunications system (Phase 2+); Radio transmission and reception (GSM 05.05 version 8.5.1 Release 1999)*, V8.5.1 ed., European Telecommunications Standards Institute, 11 2000.

[51] *Technical Specification Group Radio Access Networks; UE Radio Transmission and Reception (FDD) (Release 1999)*, v3.10.0 ed., 3rd Generation Partnership Project, 03 2002.

[52] H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*. Artech House, 2002.

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

[54] M. Pätzold, *Mobile Fadding Channels*. John Wiley & Sons, LTD, 2002.

[55] C. Kuhnert, S. Schulteis, C. Waldschmidt, and W. Wiesbeck, "Front-End requirements for MIMO communication systems," *Proceedings of the European Microwave Association*, vol. 1, no. 2, pp. 140–148, June 2005.

[56] C. A. Nel, "The creation of nonlinear behavioral-level models for system level receiver simulation." Master's thesis, University of Stellenbosch, Department of Electrical and Electronic Engineering, December 2004.

[57] J. C. Pedro and S. A. Maas, "A Comparative Overview of Microwave and Wireless Power-Amplifier Behavioral Modeling Approaches," *Microwave Theory and techniques, IEEE Transactions on*, vol. 53, no. 4, pp. 1150–1163, April 2005.

[58] "System models," Agilent Technologies, Tech. Rep., September 2004. [Online]. Available: http://eesof.tm.agilent.com/docs/

[59] D. M. Pozar, *Microwave and RF Design of Wireless Systems*. John Wiley & Sons, INC., 2001.

[60] S. A. Maas, *Nonlinear Microwave and RF Circuits*, 2nd ed. Artech House, 1997.