Cooperative Diversity in Wireless Networks

Erasmus Project at the University of Edinburgh

Andreas Meier

Supervisor: Dr John Thompson

- March 2004 -

Abstract

An ad-hoc network with a sender, a destination and a third station acting as a relay is analysed. The channels are modelled containing thermal noise, Rayleigh fading and path loss. Different combining methods and diversity protocols are compared. The amplify and forward protocol shows a better performance than the decode and forward protocol, unless an error correcting code is simulated. To combine the incoming signals the channel quality should be estimated as well as possible. Information about the average quality shows nice benefits, and a rough approximation about the variation of the channel quality increases the performance even more. Whatever combination of diversity protocol and combining method is used second level diversity is observed. The relative distances between the relay and the stations has a large effect on the performance.

Index Terms– wireless networks, cooperative diversity, relay, diversity protocols, combining methods, fading, path loss

Mission Statement

Visiting Student Project Mission Statement
Cooperative Diversity in Wireless Networks

Student: Andreas Meier

Supervisor: John Thompson

Subject Areas: Wireless Network

Project Definition

In a basic arrangement with one sender, one relay-station and one receiver, figure out the advantages and disadvantages of a system which use 'Cooperative Diversity' compared with a system without a relay station.
In particular research the following characteristics:

- Bit/Block Error Ratio
- Signal to Noise Ratio
- Throughput
- Interference

Preparatory Tasks

- Familiarisation with Matlab and LaTeX.
- Read papers about 'Cooperative Diversity' to get familiar with the topic.

Main Tasks

- Implement and test a 'Cooperative Diversity Network' in Matlab.

Scope for Extension

- Enhance the basic arrangement to a system with several relay- and receiver-stations.
- Use different types of retransmission at the relay station:
  - Amplify and forward
  - Decode and forward
  - Protocols with feedback

Background Knowledge

- Communication systems
- Signal processing

Resources

- Matlab

Location

- TLC/EE4 Project Lab
- Library
- at home

Declaration of Originality

I declare that this thesis is my

original work except where stated.

....

Contents

List of Figures

Symbols and Abbreviations

Symbol	Description
$h_{i,j}$	Attenuation in the channel$_{i,j}$
$a_{i,j}$	Attenuation caused by fading (channel$_{i,j}$)
$d_{i,j}$	Attenuation caused by path loss (channel$_{i,j}$)
$z_{i,j}[n]$	Noise added in channel$_{i,j}$
$x_i[n]$	Symbol sent by station$_i$
$y_j[n]$	Symbol received at station$_j$
$\hat{y}_j[n]$	Symbol estimated at station$_j$
$\bar{\gamma}_b$	Average signal-to-noise ratio
σ^2	Variance
N_0	Noise Power
S	Signal Power
ξ	Power of the transmitted signal
β	Gain of the amplifying relay
AAF	Amplify and Forward
BER	Bit Error Ratio
BPSK	Binary Phase Shift Keying
DAF	Decode and Forward
ERC	Equal Ratio Combining
ESNRC	Enhanced Signal-to-Noise Ratio Combining
FRC	Fixed Ratio Combining
MRC	Maximum Ratio Combining
QPSK	Quadrature Phase Shift Keying
SNR	Signal-to-Noise Ratio
SNRC	Signal-to-Noise Ratio Combining

Chapter 1

Introduction

1.1 Overview

In a wireless transmission the signal quality suffers occasionally severely from a bad channel quality due to effects like fading caused by multi-path propagation. To reduce such effects diversity can be used to transfer the different samples of the same signal over essentially independent channels. In this project diversity is realized by using a third station as a relay.

In such a system combinations of several relaying protocols and different combining methods are examined to see their effects on the performance. The transmission protocols used in this thesis are *Amplify and Forward* and *Decode and Forward.* In the simulation these can both be seen to achieve full diversity as was proved in [2]. Basically three different types of combining methods are examined which differs in the knowledge of the channel quality they need to work.

One combination that achieves a good performance is then used to see the effect on the performance depending on the location of the relay. This information is crucial to decide the worth of a mobile relay.

1.2 Structure of this Thesis

The heart of this thesis is in the following three chapters:

Chapter 2 explains the model of a single link channel. Two different modulation types are introduced (BPSK, QPSK) and the channel model (fading, path loss, noise) is explained.

Chapter 3 explains the arrangement of the diversity system used in this thesis. Two relay protocols are described and various combining methods are introduced.

Chapter 4 presents the results of the simulation. In a first part the performance of different combinations of diversity protocols and combination methods are shown. In a second part the effect of the location of the relay station are presented.

In chapter five the main results of this projects are summarised. The Matlab code used for the simulations can be found in the appendix.

Chapter 2

Single Link Transmission

In this chapter the system model for a single link transmission as illustrated in Fig. 2.1 is presented. This thesis considers the modulator, channel and demodulator block which are described below.

2.1 Signal Model and Modulation

The transfered data is a random bipolar bit sequence which is either modulated with *Binary Phase Shift Keying* (BPSK) or *Quadrature Phase Shift Keying* (QPSK). As illustrated in Fig. 2.2, QPSK in fact consists of two independent (orthogonal) BPSK systems and therefore has double bandwidth compared to BPSK. Without any loss of generality the simulations are done in the baseband.

2.2 Channel Model

In a wireless network, the data which is transferred from a sender to a receiver has to propagate through the air. During propagation several phenomena will distort the signal. Within this thesis, thermal noise, path loss and Rayleigh fading are considered, as illustrated in Fig. 2.3. Path loss and fading are multiplicative, noise is additive.

$$y_d[n] = \underbrace{h_{s,d}[n]}_{attenuation} \cdot x_s[n] + z_{s,d}[n] = \underbrace{d_{s,d}}_{path\ loss} \cdot \underbrace{a_{s,d}[n]}_{fading} \cdot x_s[n] + \underbrace{z_{s,d}[n]}_{noise} \tag{2.1}$$

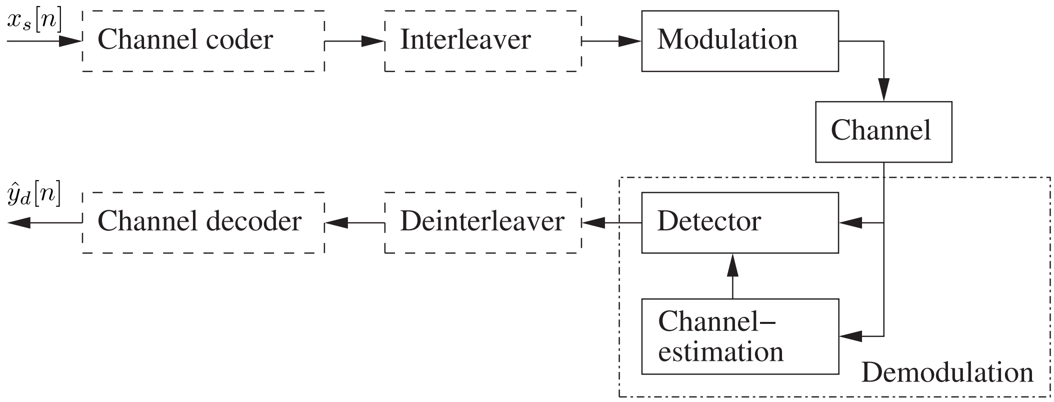

Figure 2.1: This thesis considers only the modulation, channel and demodulation block.

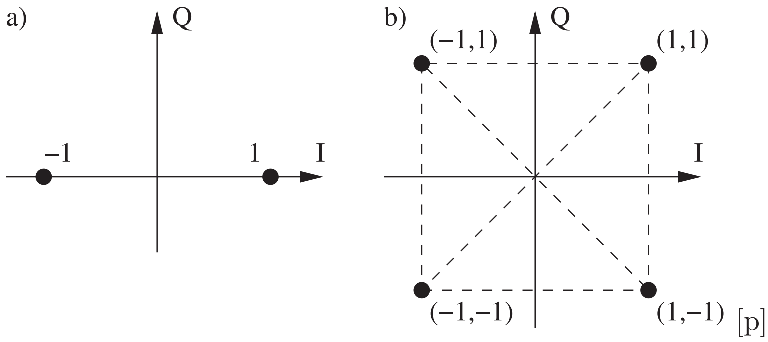

Figure 2.2: a) BPSK, b) QPSK, I denotes the in phase channel, and Q the quadrature channel.

In (2.1) s, d denote the sender respective the destination, $x_s[n]$ is the transmitted symbol and $y_d[n]$ the received symbol.

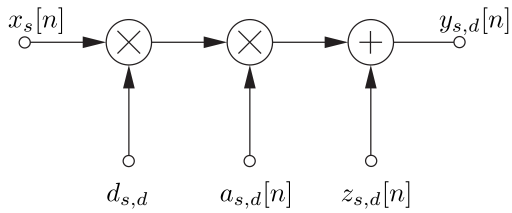

Figure 2.3: Channel model: path loss $d_{s,d}$, fading $a_{s,d}[n]$ and noise $z_{s,d}[n]$.

2.2.1 Noise

The main sources of noise in a wireless network are interference and electronic components like amplifiers. If the latter dominates, thermal noise can be assumed, which can be characterised as additive complex Gaussian noise. The scalar $z_{s,d}[n]$ can then be simulated as the sum of a real and a imaginary noise vector, both Gaussian distributed, mutually independent and zero mean with variance σ_n^2. The total noise power will be $N_0 = 2\sigma_n^2$.

2.2.2 Signal to Noise Ratio

The *signal-to-noise ratio* (SNR) is a widely used value to indicate the signal quality at the destination.

$$SNR = \left(\frac{S}{N_0}\right) = \frac{|h_{s,r}|^2 \cdot \xi}{N_0} \tag{2.2}$$

In (2.2) $\xi = E[|x_s|^2]$ denotes the energy of the transmitted signal and N_0 the total power of the noise.

2.2.3 Path Loss and Fading

The signal is attenuated mainly by the effects of free-space path loss and fading, both included in $h_{s,d} = d_{s,d} \cdot a_{s,d}$.

The path loss $d_{s,d}$ (assuming a plane-earth model) is proportional to $\frac{1}{R^2}$. As long as the distance between the sender and receiver does not change too much, it can be assumed to be constant for the whole transmission. The power of the received signal is attenuated proportional to $\frac{1}{R^4}$.

In a wireless network it occurs quite often that the line-of-sight link is blocked. Instead of this direct connection, the signal will propagate to the sender on many different ways. This occurs especially in an urban environment, where buildings prevent a line-of-sight link but enable various different ways for indirect connection by reflecting the propagating signal. This effect is referred to as *multi-path* propagation.

Only small changes in the whole system might change the characteristic of the channel and therefore the signal quality considerably. This effect, known as fading, will alter the signal by attenuating it and adding a phase shift to it. The fading coefficient $a_{s,d}$ can be modelled as a zero mean, complex Gaussian random variable with variances $\sigma^2_{s,d}$. This means that the angle $\angle a_{s,d}$ is uniformly distributed on $[0, 2\pi)$ and the magnitude $|a_{s,d}|$ is Rayleigh distributed [2]. This Rayleigh distributed magnitude can have a bad effect on the signal quality at the receiver, as illustrated in Fig. 2.4. Even a system with a high SNR might experience significant errors due to fading[1].

Block Fading

In a fast fading channel, the channel characteristic changes within one burst of data. The block fading channel model pays attention to this effect. The burst is broken up into smaller pieces, blocks, which can then be assumed to have a constant channel characteristic. The block length has to be long enough, to allow the channel characteristic to be estimated perfectly. The magnitude and the angle of the fading coefficient $a_{s,d}$ of the block is known by the receiver.

In a block fading channel, there is a high possibility that burst errors occur, i.e. that there are a lot of errors within one block. Such bursts of

[1] There is a chance of about one percent of the signal being 20dB below the mean level.

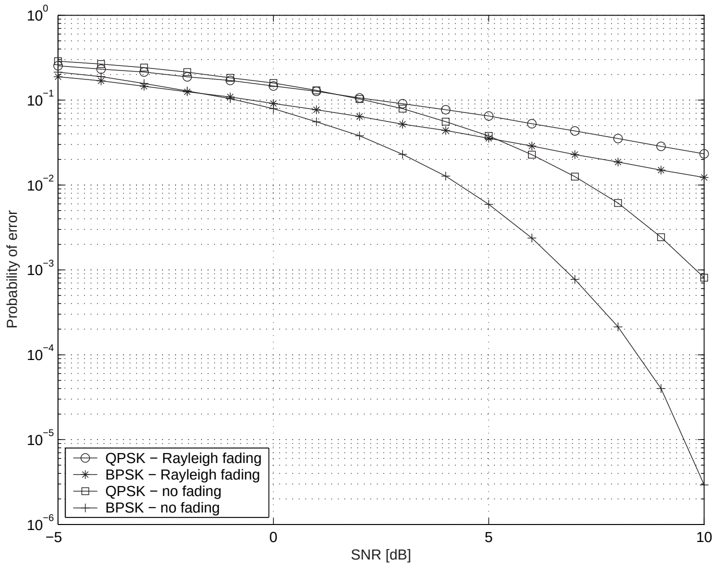

Figure 2.4: The severe effect of a Rayleigh faded channel, compared to the non faded channel.

errors are very difficult to correct with an error correcting code. To prevent them occurring, the signal can be interleaved to get the errors distributed uniformly over the whole signal, as illustrated in Fig. 2.1. The interleaving and the coding block are not simulated but it is assumed that they exist[2]. Therefore, if you simulate such a transmission, it does not matter how the errors are distributed over the whole signal. The only thing that is of interest is the average *bit error ratio* (BER). To get an accurate result the signal should be transferred over as many different channel characteristics as possible. Without loss of generality the block length within the simulation can be assumed to be one[3]. This significantly reduces the computing time.

[2]To enhance the model to simulate the packet error rate using error correction codes, these blocks need to be simulated as well.

[3]In contrast to a real transmission, in a simulation the channel characteristic is fully known by the receiver although the characteristic changes after every transmitted symbol.

2.3 Receiver Model

The receiver detects the received signal symbol by symbol. In the case of a BPSK modulated signal the symbol/bit is detected as

$$\hat{y}_d[n] = \begin{cases} +1 & (\mathrm{Re}\{y_d[n]\} \geq 0) \\ -1 & (\mathrm{Re}\{y_d[n]\} < 0) \end{cases} . \tag{2.3}$$

For a QPSK modulated signal there are two bits transfered per symbol, which are detected as

$$\hat{y}_d[n] = \begin{cases} [+1,+1] & (0° \leq \angle y_d[n] < 90°) \\ [-1,+1] & (90° \leq \angle y_d[n] < 180°) \\ [+1,-1] & (-90° \leq \angle y_d[n] < 0°) \\ [-1,-1] & (-180° \leq \angle y_d[n] < -90°) \end{cases} . \tag{2.4}$$

2.4 BER of a Single Link Transmission

The signal quality received at the destination depends on the SNR of the channel and the way the signal is modulated. The theoretical probability of a bit error is derived in [1] and is summarised in Tab. 2.1.

Modulation Type	no Fading	Rayleigh Fading
BPSK	$P_b = Q\left(\sqrt{\frac{\xi}{\sigma^2}}\right)$	$P_b = \frac{1}{2}\left(1 - \sqrt{\frac{\bar{\gamma}_b}{1+\bar{\gamma}_b}}\right)$
QPSK	$P_b = Q\left(\sqrt{\frac{\xi}{2\sigma^2}}\right)$	$P_b = \frac{1}{2}\left(1 - \sqrt{\frac{\bar{\gamma}_b}{2+\bar{\gamma}_b}}\right)$

Table 2.1: Theoretical BER for a single link transmission. $\bar{\gamma}_b$ denotes the average signal-to-noise ratio, defined as $\bar{\gamma}_b = \frac{\xi}{2\sigma^2}E(a^2)$, where $E(a^2) = a^2$.

The same result can be obtained by simulating the transmission using (2.1) and (2.3), (2.4). This simulation, which is illustrated in Fig. 2.4 shows the negative effect on the signal quality due to fading. The figure also shows that the performance of the BPSK modulated signal is in general 3dB better than the one modulated with QPSK.

2.5 Conclusions

In this section the model of a single link transmission has been presented. The signal is modulated using *Binary Phase Shift Keying* (BPSK) or *Quadrature Phase Shift Keying* (QPSK). The channel consists of path loss and Rayleigh fading which are multiplicative components and thermal noise which is additive.

In the next chapter this simple single link model is enhanced to a diversity arrangement using one direct link and one multi-hop link.

Chapter 3

Multi hop

There are several approaches to implement *diversity* in a wireless transmission. Multiple antennas can be used to achieve space and/or frequency diversity. But multiple antennas are not always available or the destination is just too far away to get good signal quality. To get diversity, an interesting approach might be to build an ad-hoc network using another mobile station as a relay. The model of such a system is illustrated in Fig. 3.1. The sender S, sends the data to the destination D, while the relay station R is listening to this transmission. The relay sends this received data burst after processing to the destination as well, where the two received signals are combined. As proposed in [2], orthogonal channels are used for the two transmissions. Without loss of generality, this can be achieved using time divided channels, which is done in all the simulations in this thesis.

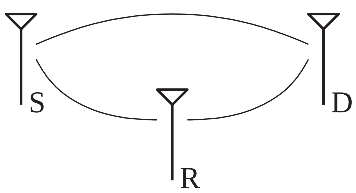

Figure 3.1: The data is transmitted on one hand directly to the destination, and on the other hand the data is sent to the receiver via the relay.

3.1 Cooperative Transmission Protocols

The cooperative transmission protocols used in the relay station are either *Amplify and Forward* (AAF) or *Decode and Forward* (DAF). These protocols describe how the received data is processed at the relay station before the data is sent to the destination.

3.1.1 Amplify and Forward

This method is often used when the relay has only limited computing time/power available or the time delay, caused by the relay to de- and encode the message, has to be minimised. Of course when an analogue signal is transmitted a DAF protocol can not be used.

The idea behind the AAF protocol is simple. The signal received by the relay was attenuated and needs to be amplified before it can be sent again. In doing so the noise in the signal is amplified as well, which is the main downfall of this protocol.

The incoming signal is amplified block wise. Assuming that the channel characteristic can be estimated perfectly, the gain for the amplification can be calculated as follows.

The power of the incoming signal (2.1) is given by

$$E[|y_r^2|] = E[|h_{s,r}|^2]E[|x_s|^2] + E[|z_{s,r}|^2] = |h_{s,r}|^2\xi + 2\sigma_{s,r}^2,$$

where s denotes the sender and r the relay. To send the data with the same power the sender did, the relay has to use a gain of

$$\beta = \sqrt{\frac{\xi}{|h_{s,r}|^2\xi + 2\sigma_{s,r}^2}} \tag{3.1}$$

This term has to be calculated for every block and therefore the channel characteristic of every single block needs to be estimated.

3.1.2 Decode and Forward

Nowadays a wireless transmission is very seldom analogue and the relay has enough computing power, so DAF is most often the preferred method to process the data in the relay. The received signal is first decoded and then re-encoded. So there is no amplified noise in the sent signal, as is the case using a AAF protocol. There are two main implementations of such a system.

The relay can decode the original message completely. This requires a lot of computing time, but has numerous advantages. If the source message contains an error correcting code, received bit errors might be corrected at the relay station. Or if there is no such code implemented a checksum allows the relay to detect if the received signal contains errors. Depending on the implementation an erroneous message might not be sent to the destination[1].

But it is not always possible to fully decode the source message. The additional delay caused to fully decode and process the message is not acceptable, the relay might not have enough computing capacity or the source message could be coded to protect sensitive data. In such a case, the incoming signal is just decoded and re-encoded symbol by symbol. So neither an error correction can be performed nor a checksum calculated.

Magic Genie

In this thesis, no error correcting code has been implemented. So it is not possible to correct the signal received by the relay. To simulate this scenario, an all knowing *magic genie* is used. The genie, sitting on the relay station, checks every decoded symbol and allows this symbol to be re-encoded and sent if and only if it was correctly detected. This is a much more powerful approach than deciding block wise (up to some hundred symbols) if all symbols in it are correct. The overall performance of a system supported by

[1]Normally it does not make sense to send an incorrect data packet.

a magic genie is similar to one using error correction and therefore an error correcting code can be simulated in this way.

3.2 Combining Type

As soon as there is more than one incoming transmission with the same burst of data, the incoming signals have to be combined before they will be compared as indicated in (2.3) and (2.4).

3.2.1 Equal Ratio Combining (ERC)

If computing time is a crucial point, or the channel quality could not be estimated, all the received signals can just be added up. This is the easiest way to combine the signals, but the performance will not be that good in return.

$$y_d[n] = \sum_{i=1}^{k} y_{i,d}[n]$$

Within this thesis one relay station is used, so the equation simplifies to

$$y_d[n] = y_{s,d}[n] + y_{r,d}[n], \tag{3.2}$$

where $y_{s,d}$ denotes the received signal from the sender and $y_{r,d}$ the one from the relay.

3.2.2 Fixed Ratio Combining (FRC)

A much better performance can be achieved, when fixed ratio combining is used. Instead of just adding up the incoming signals, they are weighted with a constant ratio, which will not change a lot during the whole communication. The ratio should represent the average channel quality and therefore should not take account of temporary influences on the channel due to fading or other effects. But influences on the channel, which change the average channel quality, such as the distance between the different stations, should be considered. The ratio will change only gently and therefore needs only a little amount computing time. The FRC can be expressed as

$$y_d[n] = \sum_{i=1}^{k} d_{i,d} \cdot y_{i,d}[n],$$

where $d_{i,d}$ denotes weighting of the incoming signal $y_{i,d}$. Using one relay station, the equation simplifies to

$$y_d[n] = d_{s,d} \cdot y_{s,d}[n] + d_{s,r,d} \cdot y_{r,d}[n]. \tag{3.3}$$

where $d_{s,d}$ denotes the weight of the direct link and $d_{s,r,d}$ the one of the multi-hop link. Within this thesis only the best achievable performance of a FRC system is of interest. So the best ratio is approximated[2] by comparing different possible values. This ratio is then used to compare with the other combining methods.

3.2.3 Signal to Noise Ratio Combining (SNRC)

A much better performance can be achieved, if the incoming signals are weighted on an intelligent way. An often used value to characterise the quality of a link is the SNR, which can be used to weight the received signals.

$$y_d[n] = \sum_{i=1}^{k} \mathrm{SNR}_i \cdot y_{i,d}[n]$$

Using one relay, the equation can be written as

$$y_d[n] = \mathrm{SNR}_{s,d} \cdot y_{s,d}[n] + \mathrm{SNR}_{s,r,d} \cdot y_{r,d}[n], \tag{3.4}$$

where $\mathrm{SNR}_{s,d}$ denotes the SNR of the direct link and $\mathrm{SNR}_{s,r,d}$ the one over the whole multi-hop channel.

The estimation of the SNR of a multi-hop link using AAF or a direct link can be performed by sending a known symbol sequence in every block[3]. If the

[2] To figure out an intelligent algorithm to determine the best ratio is not a part of thesis.

[3] The sequence is used to estimate the phase shift as well.

multi-hop link is using a DAF protocol the receiver can only see the channel quality of the last hop. It is assumed that the relay sends some additional informations about the quality of the unseen hops to the destination, so the SNR of the multi-hop link can be estimated as well. Whatever protocol is used, an additional sequence needs to be sent to estimate the channel quality. This results in a certain loss of bandwidth.

Estimate SNR using AAF

Using AAF, the received signal from the relay is

$$y_{r,d} = h_{r,d}x_r + z_{r,d} = h_{r,d}\beta(h_{s,r}x_s + z_{s,r}).$$

The received power will then be

$$E[|y_{r,d}|^2] = \beta^2|h_{r,d}|^2(|h_{s,r}|^2\xi + 2\sigma_{s,r}^2) + 2\sigma_{r,d}^2,$$

so the SNR of the one relay multi-hop link can be estimated as

$$\text{SNR} = \frac{\beta^2|h_{s,r}|^2|h_{r,d}|^2\xi}{\beta^2|h_{r,d}|^2 2\sigma_{s,r}^2 + 2\sigma_{r,d}^2}. \tag{3.5}$$

Estimate SNR using DAF

To calculate the SNR of a multi-hop link using DAF, first the BER of the link is calculated which can then be translated to an equivalent SNR.

The BER of a single link is given in Tab. 2.1. The BER over a one relay multi-hop link can then be calculated as

$$\text{BER}_{s,r,d} = \text{BER}_{s,r}(1 - \text{BER}_{r,d}) + (1 - \text{BER}_{s,r})\text{BER}_{r,d}.$$

To calculate the SNR, the inverse functions of those in Tab. 2.1 are used. For a BSPK modulated Rayleigh faded signal this will be

$$\text{SNR} = \frac{1}{2}\left[Q^{-1}(\text{BER})\right]^2. \tag{3.6}$$

For a QPSK modulated signal this will change to

$$\text{SNR} = \left[Q^{-1}(\text{BER})\right]^2. \tag{3.7}$$

3.2.4 Maximum Ratio Combining (MRC)

The *Maximum Ratio Combiner* (MRC) achieves the best possible performance by multiplying each input signal with its corresponding conjugated channel gain. This assumes that the channels' phase shift and attenuation is perfectly known by the receiver.

$$y_d[n] = \sum_{i=1}^{k} h_{i,d}^*[n] \cdot y_{i,d}[n]$$

Using a one relay system, this equation can be rewritten as

$$y_d[n] = h_{s,d}^*[n] y_{s,d}[n] + h_{r,d}^*[n] y_{r,d}[n]. \tag{3.8}$$

By looking at this equation a little bit closer, the big disadvantage of this combining method in a multi-hop environment can be seen. The MRC only considers the last hop (i.e. the last channel) of a multi-hop link. So in this thesis the MRC should only be used in combination with a DAF protocol. There is still the problem that the relay might send incorrectly detected symbols, which will have severe effects on the performance. So the use of MRC is only recommended if an error correcting code is used. This can be simulated by using a magic genie as described in 3.1.2.

3.2.5 Enhanced Signal to Noise Combining (ESNRC)

Another plausible combining method is to ignore an incoming signal when the data from the other incoming channels have a much better quality. If the channels have more or less the same channel quality the incoming signals are rationed equally. In the system used in this thesis this can be expressed as

$$y_d[n] = \begin{cases} y_{s,d}[n] & (\mathrm{SNR}_{s,d}/\mathrm{SNR}_{s,r,d} > 10) \\ y_{s,d}[n] + y_{s,r,d}[n] & (0.1 \leq \mathrm{SNR}_{s,d}/\mathrm{SNR}_{s,r,d} \leq 10) \\ y_{s,r,d}[n] & (\mathrm{SNR}_{s,d}/\mathrm{SNR}_{s,r,d} < 0.1) \end{cases} . \tag{3.9}$$

Using this combining method, the receiver does not have to know the channel characteristic exactly. An approximation of the channel quality

Abbreviation	Meaning	Reference
Modulation Types		
BPSK	Binary Phase Shift Keying	p. 3
QPSK	Quadrature Phase Shift Keying	
Combining Types		
ERC	Equal Ratio Combining	eq. 3.2
FRC x:y	Fixed Ratio Combining x: Weight of the direct link y: Weight of the multi-hop link	eq. 3.3
MRC	Maximum Ratio Combining	eq. 3.8
SNRC	SNR Combining	eq. 3.4
ESNRC	Enhanced SNR Combining	eq. 3.9
Amplifying Types		
AAF	Amplify and Forward	p. 11
DAF	Decode and Forward	p. 12
Special		
Magic Genie	Magic Genie is used	p. 12
Distance x:y:z	x: Distance between sender and destination y: Distance between sender and relay z: Distance between relay and destination	

Table 3.1: In the legends of all the performance figures, following abbreviation to describe the curves are used.

is enough to combine the signals[4]. As a further benefit, the equal ratio combining does not need a lot of computing power.

3.3 Simulation

All the figures presented in the next chapter are labelled using the same abbreviations, which are described in Tab. 3.1.

There are two popular implementations to transmit over a wireless network. One is the simple direct link which sends the data only once. The other is the two sender arrangement which sends the data twice over different antennas. These two standard implementations put the performance of

[4]The phase shift has still to be estimated as precisely as possible, but the attenuation, which is much more difficult to estimate, can be approximated.

the arrangements used in this thesis into perspective.

The diversity arrangement has to send the data twice and therefore requires twice the bandwidth of the single link transmission. To compensate for this effect, the single link channel is modulated using BPSK and the diversity arrangement uses QPSK. As QPSK has twice the bandwidth of BPSK both arrangements have the same overall bandwidth. Notice that the relay causes a certain time delay for the diversity arrangement.

The performance of a two sender transmission with MRC at the receiver can be expressed [1] as

$$P_b = \frac{1}{4}(1-\mu)^2(2+\mu) \qquad \mu = \sqrt{\frac{\bar{\gamma}_b}{1+\bar{\gamma}_b}},$$

where $\bar{\gamma}_b$ denotes the average signal-to-noise ratio, defined as $\bar{\gamma}_b = \frac{\xi}{2\sigma^2}E(a^2)$, where $E(a^2) = a^2$.

3.4 Conclusions

In this chapter the different aspects of a multi-hop and a diversity arrangement have been presented. First two different transmission protocols *Amplify and Forward* (AAF) and *Decode and Forward* (DAF) have been described. For the latter protocol, a *Magic Genie* can be used to simulate an error correcting code. When the destination receives different samples of the same data, these samples need to be combined. The *Equal Ratio Combining* (ERC) just adds up the different received signal while the *Fixed Ratio Combining* (FRC) is weighting the incoming signals with a fixed ratio. When the channel quality is estimated precisely, more powerful combining methods as *Maximum Ratio Combining* (MRC), *Signal-to-Noise Ratio Combining* (SNRC) or *Enhanced Signal-to-Noise Combining* (ESNRC) can be used.

In the next chapter these combining methods and transmissions protocols are compared with each other to determine, which combinations results in a good performance. In a second part, the effect of the position of the relay station is presented.

Chapter 4

Key Results

In this chapter the performance of different combinations of the methods described in the last chapter are analysed to illustrate their potential benefits. In the first part, it is assumed that the three stations (sender, relay and destination) have an equal distance from each other and therefore the same path loss and average signal-to-noise ratio is assumed. With this equidistant arrangement the different combining and amplifying types are compared to see their advantages and disadvantages. In the second part, the location of the relay station is varied to see the effect on the performance for different locations of the relay.

4.1 Equidistant Arrangement

In this section it is assumed, that the three stations are arranged at the edges of a triangle with a length of one. This means that all the channels will have the same path loss and therefore the same average signal-to-noise ratio.

4.1.1 Amplify and Forward

To compare the benefits of the different combining method, the optimal ratio for the FRC needs to be evaluated first. Fig. 4.1 illustrates the effects of the different weighting. As seen, a much better performance is achieved using

FRC instead of ERC simply by assuming that the direct link has in general a better quality than the multi-hop link. This is obvious in an equidistant arrangement, where the signal over the multi-hop has to propagate over twice the distance than over the direct link. The result of the simulation illustrated in Fig. 4.1 shows that the best performance using FRC is achieved with a ratio of 2:1. FRC with this ratio is now used to compare performances with one of the other combining types.

In Fig. 4.2 the effect on the performance of the different combining types using a AAF protocol can be seen. The BPSK single link transmission should demonstrate if there is any benefit at all using diversity, while the QPSK two senders link indicates a lower bound for the transmission. Using the equidistant arrangement, the aim is to get as close to the latter curve as possible or to get an even better performance.

The first pleasant result is that whatever combining type is used, the AAF diversity protocol achieves a benefit compared to the direct link. Even the equal ratio combining shows advantages. But compared to the fixed ratio combining, the performance looks quite poor. Otherwise you should call to mind that the equal ratio combining does not need any channel information, except the phase shift, to perform the combining. The fixed ratio combining on the other hand, needs some channel information to calculate the appropriate weighting.

The signal-to-noise ratio combining (SNRC) and the enhanced signal-to-noise ratio combining (ESNRC) show roughly the same performance, which is much better then the one using FRC/ERC. This is not surprising considering that the former two combining methods are using much more detailed channel information than the latter two. Actually the big surprise is that the performance of the combining methods, which have precise information about every single block, is just about one decibel better then the one using FRC which has just average knowledge of the channel quality.

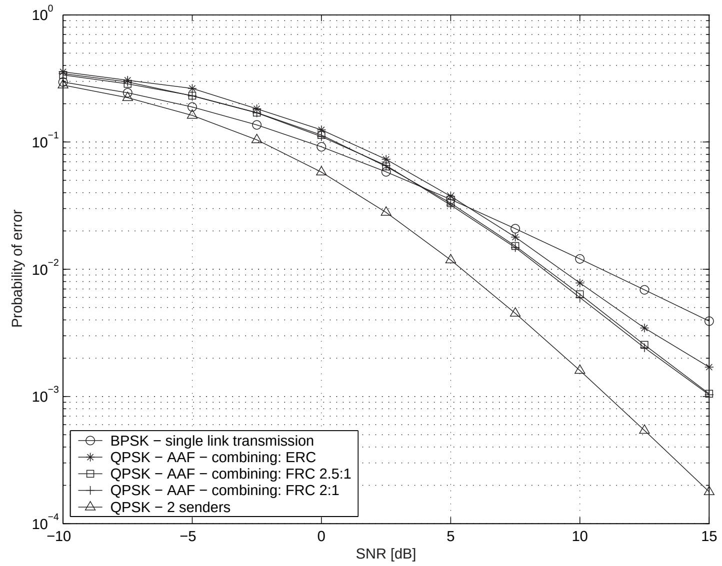

Figure 4.1: To estimate the best ratio for FRC different ratios are plotted. The ratio 2:1 gives a good result.

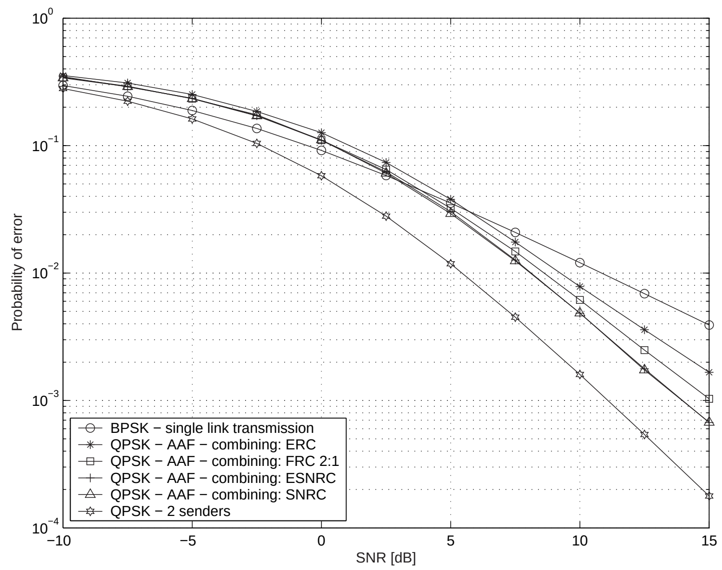

Figure 4.2: The different combining types are compared with each other. The best performance results when using SNRC/ESNRC.

The other unexpected thing is that the SNRC shows approximately the same performance than the ESNRC. Remember that for the ESNRC a roughly estimated channel quality for every single block is sufficient. This is in contrast to the SNRC, which needs exact information of the channel quality for every single block. This means that the transferred signal in an AAF system contains some information that allows correcting of a small difference in the channel quality.

Using the AAF protocol, there is no point in wasting a lot of computing power and bandwidth to get some exact channel information. And even if the channel quality could not be estimated at all (and therefore ERC is used), there is still a benefit using diversity.

4.1.2 Decode and Forward

To compare the benefits of the different combining method, the optimal ratio for the FRC needs to be evaluated first, which is done exactly in the same way as before. The FRC is simulated with different weighting to estimate the ratio that results in the best performance. The simulations, illustrated in Fig. 4.3, show the best performance when a ratio of 3:1 is used. It is quite surprising that this ratio differs that much from the ratio using AAF. The reason for that is discussed in Sec. 4.1.3. The FRC with a ratio of 3:1 can now be used to compare with the other combining methods.

The different combining methods using the DAF protocol are illustrated in Fig. 4.4. The first thing that attracts attention is the bad performance of the equal ratio combining. Especially for a small SNR the performance is significantly worse than the one of the BPSK single link transmission and therefore should not be used at all.

The fixed ratio combing shows obviously a much better performance than the BPSK single link transmission. To achieve a BER of about 10^{-2} the required SNR for the FRC is about 2.5 dB less than the one for the

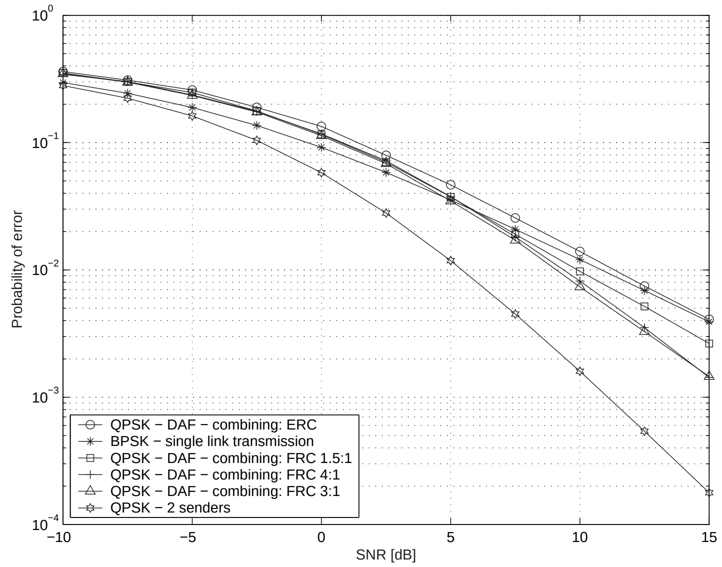

Figure 4.3: To estimate the best ratio for FRC different ratios are plotted. The ratio 3:1 results in the best performance.

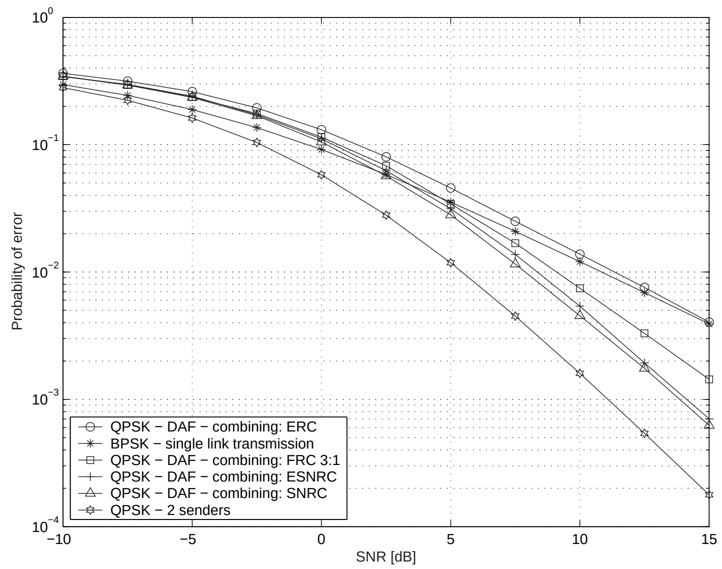

Figure 4.4: The different combining types are compared with each other. The best performance results when using SNRC.

single link transmission. That is quite a remarkable benefit.

In contrast to the AAF protocol, a big benefit results using one of the block analysing combining methods (SNRC/ESNRC). Using the DAF protocol shows now the benefit estimating every single block separately and hence using more computing power.

There is now an additional benefit, to estimate every block precisely when using SNRC, instead of just approximating the channel quality combining the signals with ESNRC. But considering that the achievable benefit is about half a decibel it might not be worth wasting the additional computing power and bandwidth which is required to get a precise channel estimation. If AAF is used, there is no benefit at all, using the SNRC instead of ESNRC. From now on, the focus will be laid on ESNRC, FRC and ERC.

4.1.3 Amplify and Forward versus Decode and Forward

Fig. 4.5 illustrates the performance of the AAF diversity protocol compared with the DAF protocol. The surprising result is that the AAF diversity protocol always results in a better performance than the DAF protocol whatever combining type is used.

Using equal ratio combining results in a big difference between the two protocols. While the one using AAF shows a quite good performance already, the one using DAF shows no improvement at all. The reason is that a wrong detected symbol at the relay station is really difficult to correct at the destination, where the two incoming signals are combined. The incorrectly detected symbol is sent by the relay with the same power as the correct symbol over the direct link. This means that, when the two signals are combined at the destination, it is equally likely that the symbol is both correctly and incorrectly detected. So an incorrectly detected symbol at the relay station will have a fifty percent probability of also being incorrectly

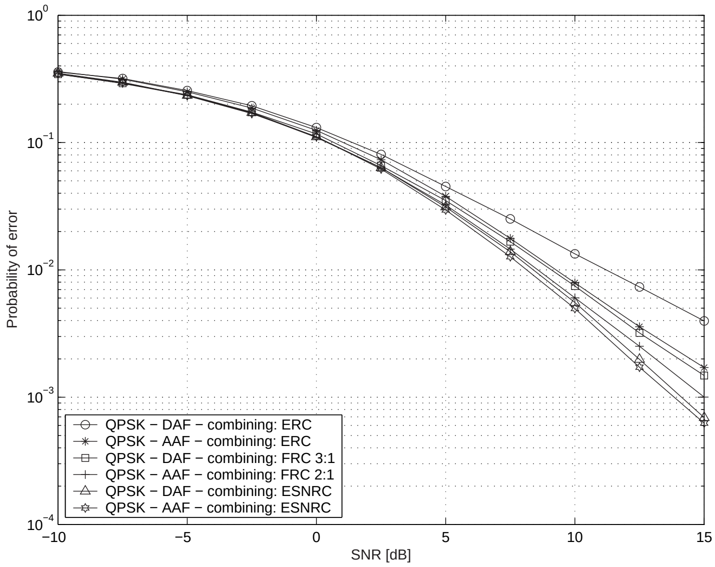

Figure 4.5: The two diversity protocols (AAF and DAF) are compared with each other. Independent of the combining type, the AAF always results in the better performance.

detected at the destination.

This stands in contrast with the equal ratio combining in a system using AAF. Instead of detecting the symbol at the relay, it is amplified and transmitted to the sender. Normally a symbol that would have been detected wrongly is just 'a little bit' wrong. When this symbol is amplified before sent to the destination, it has on average much less energy than the correct symbol coming directly from the sender. There is now a high probability that the incorrect symbol will be corrected by the signal from the direct link, when combined at the destination. This is of course only the case when the symbol over the direct link did not suffer too much from a bad channel.

It is obvious now, why the fixed ratio combining shows such a good performance. The direct link has on average the better quality than the multi-hop link, so it is sensible to weigh the direct link more by assuming that the multi-hop link is more susceptible to errors than the direct link. It also explains why the optimal ratio in the system using DAF is higher than the one using AAF. The DAF relay sends the wrongly detected symbols with the full power, so it takes much more to correct this wrong powerful symbol.

The ESNRC shows roughly the same performance in a AAF or DAF system. The DAF using system benefits a lot by analysing every single block. Using this combining method the big disadvantage of the wrongly detected symbol at the relay can be reduced. In the majority of the cases, when a symbol is wrongly detected by the relay, the multi-hop has a much poorer channel quality than the direct link, and therefore will not be considered at all.

It might be sensible to ask now, what the purpose is of making the effort at the relay station to decode and re-encode the data, when there is no benefit at all doing that. As mentioned in Sec. 3.1.2 there are mainly two different types of how a DAF system can be implemented. Within this thesis there is no error correcting code added to the data, so there is no chance to correct any wrongly detected bits at the receiver. This is, as seen before, crucial to get a good performance in a DAF system. To estimate the effects of an error correcting code, a magic genie, as suggested in Sec. 3.1.2 can be used.

4.1.4 Magic Genie

The effect on the channel quality using the magic genie is illustrated in Fig. 4.6. The DAF system with the magic genie shows a much better performance than the AAF system, which of course can not use one.

As seen in Fig. 4.6, the combining method does not make a big difference in the bit error ratio when a DAF system with a magic genie is used. The performance of the maximum ratio combining is less than half a decibel better than the one using equal ratio combining. Remember, that the former one is the optimal combining method, yet the easy to implement equal ratio

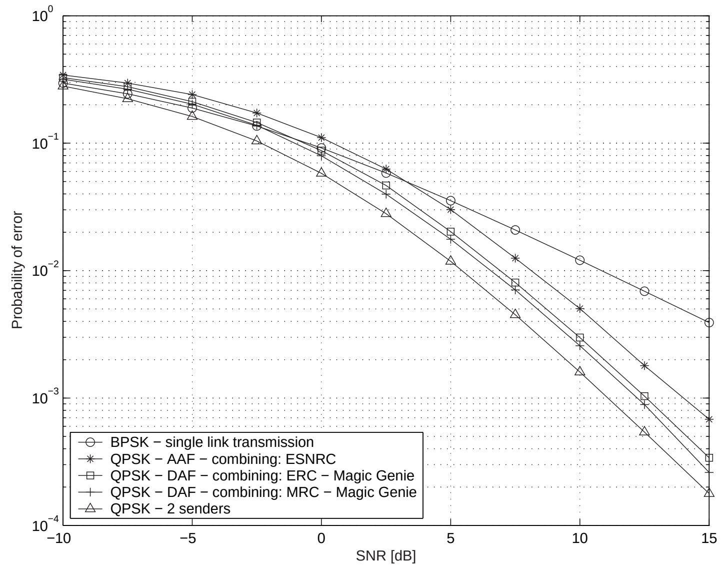

Figure 4.6: The effect of an error correcting code is simulated by using a magic genie.

combining shows a very good performance as well.

The system using a magic genie gets close to the performance of the two sender system, which is a quite nice benefit. It should be noticed as well, that all the simulations illustrated in Fig. 4.6 have approximately the same slope as the two sender system and therefore show full second level diversity.

Using a magic genie shows a really nice benefit, but it should be kept in mind, that the genie is assumed to have information about the behaviour of the noise in the channel, to be able to detect wrongly transmitted symbols. This is a contradiction in terms. So by analysing these results, it should be considered that the magic genie is just an approximation to estimate the effects of an error correcting code. So caution is advisable here, interpreting the results.

4.2 Moving the Relay

So far, the three stations were positioned equidistantly and therefore the three channels had all the same average signal-to-noise ratio. In this section the effect is shown when the relay station is moved. For the following simulations the AAF diversity protocol is used and the incoming signals at the destination are combined using ESNRC. As seen in the Sec. 4.1.3 this is the combination that results in the best possible performance.

The x-axis in the figures shows the average signal-to-noise ratio, for a channel of length one. This was the case for all three channels in the equidistant arrangement in the last section. In this section, the relay is moved, so the distance from the relay to the sender/destination will change. But in all the simulations, the distance between the sender and the destination is set to one, and therefore the signal-to-noise ratio shown in the x-axis is only valid for the direct link.

4.2.1 Relay between Sender and Destination

The propagation over the multi-hop link does not need to make any detour, when the relay is situated between the sender and the destination. This is the optimal scenario and should result in the best possible performance.

If the relay is situated very close to the sender, the whole arrangement corresponds approximately to a two sender system. The effect on the signal quality when moving the relay between the two other stations is shown in Fig. 4.7. With this optimal configuration, the resulting benefit is huge and much better than the one for the two sender system. The best performance is achieved, when the relay is situated in the middle between the sender and the destination, or slightly closer to the sending station.

The resulting performance is not symmetrical at all. The preferred position of the relay is in the middle between the sender and the destination. When this is not possible the relay should be closer to the sender than to the

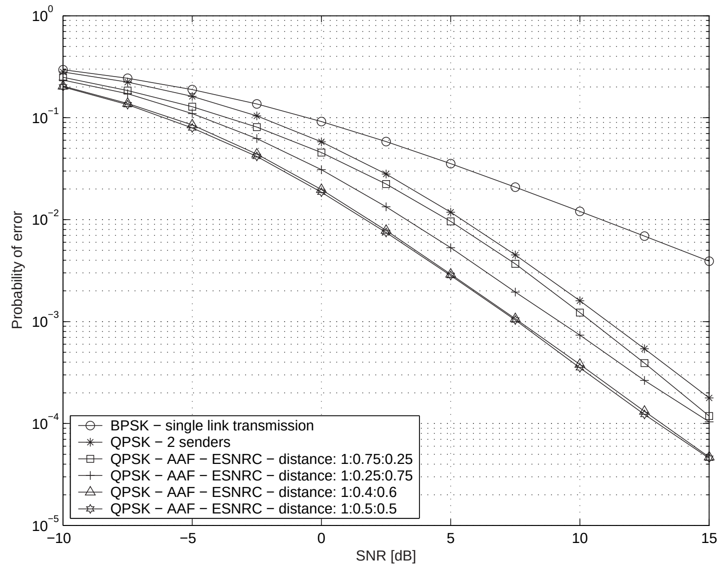

Figure 4.7: A big benefit results when the relay is located between the sender and the destination.

destination. Recall to mind how the AAF protocol works, this is obvious. The noise received in the relay station is amplified with the signal. So on one hand it is desirable, that the received noise at the relay station does not has much energy. On the other hand, the closer the relay comes to the sender, the further away is the destination and therefore the worse is the channel quality of the second hop. The quality of the first hop is more important for the overall channel quality than the second hop, so the performance is not symmetrical[1].

Another point that should be paid attention to is the huge benefit compared to the BSPK direct link. To achieve a BER of about 10^{-2} the SNR is up to eight decibels less than using only a direct link transmission.

4.2.2 Equidistant Position of the Relay

Normally there is no relay station available just between the sender and the destination. To see the effect the length of the multi-hop link has on the system performance, the relay is moved away gently from the optimum position between the sender and the destination. This is illustrated in Fig. 4.8.

The first thing that attracts attention is how fast the performance gets worse when the distance of the relay increases. By increasing the distance by fifty percent, the resulting performance is roughly the same as the one for a two sender system, which is about three decibels less than the one of the optimal position. The position of the relay, where all three stations are equidistant, results in another 2.5 decibel loss in the system performance. This equidistant arrangement still shows an advantage compared to the BPSK single link transmission.

This changes pretty fast, when the distance of the relay is increased further. Another fifty percent, result in a situation, where there is no useful advantage anymore using the relay link. But the higher diversity level can still be recognised.

When the relay is situated in the double distance of the equidistant arrangement, there is no benefit at all using the relay link. The resulting performance is roughly the same as the one of the QPSK single link transmission. This means, that the relay link, does not contain any useful information anymore. There is now just too much noise in the signal to get any benefit.

4.2.3 Moving the Relay Close to the Sender/Destination

In Fig. 4.9 the arrangement is illustrated where the relay is much closer to either the sender or the destination. In contrast to the situation where

[1]Using a DAF protocol the asymmetry is even more dominant. This is for the same reasons, why the optimal ratio of the fixed ratio combining in a system using DAF is higher than in one using AAF.

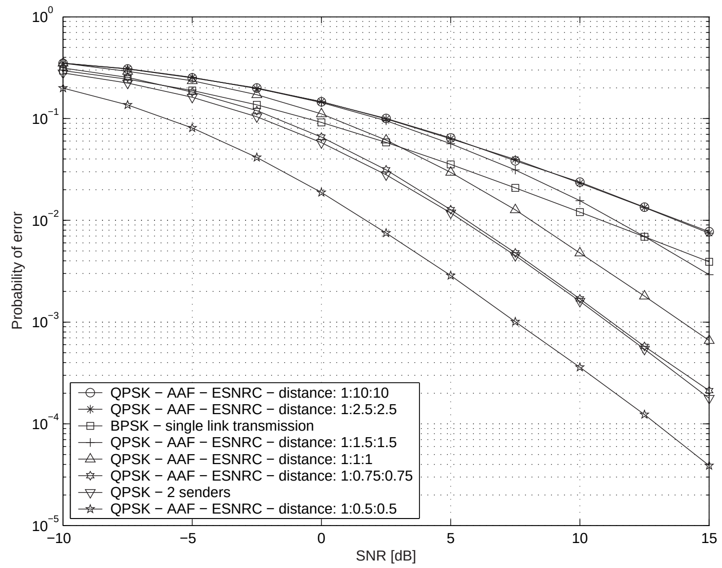

Figure 4.8: Shows the effect of increasing the distance of the relay to the sender and the destination.

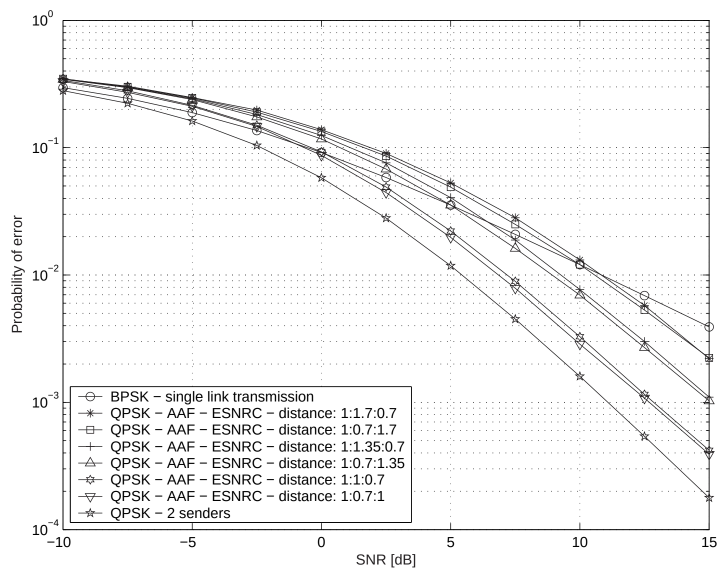

Figure 4.9: The relay is moved close to the sender/destination.

the relay was situated between the two other stations, the arrangement shows now much more symmetry. The reason for that is that the direct link contains the better signal quality and therefore is mainly responsible for the performance.

The main interest is now to determine where a mobile station can be located so that there is some benefit from using it as a relay station. Looking at Fig. 4.8 and Fig. 4.9 you can get the basic idea. If the relay is located close to the sender or the destination, the distance to the other station can be about forty percent longer then the one to the direct link. When the relay is roughly the same distance from both stations, this distance should not be much longer than the direct link to get a benefit. This results roughly in an elliptical region between base and mobile, where a second mobile station has to be situated to make it an attractive candidate as a relay.

Chapter 5

Conclusions

5.1 Summary

This thesis has shown the possible benefits of a wireless transmission using cooperative diversity to increase the performance. The diversity is realized by building an ad-hoc network using a third station as a relay. The data is sent directly from the base to the mobile or via the relay station. Such a system has been simulated to see the performance of different diversity protocols and various combining methods.

The AAF protocol has shown a better performance than the DAF protocol whatever combining method was used at the receiver. But it must be considered that no error correcting code was added to the transferred signal. Therefore it was not possible to take full advantage of the DAF protocol. To get an idea of the potential of the DAF protocol the *magic genie* was introduced to simulate an error correcting code. The performance of a system using the DAF protocol in combination with a magic genie was much better than one using the AAF protocol.

The choice of combining method has a big effect on the error rate at the receiver. When AAF is used at the relay station the easy to implement *Equal Ratio Combining* (ERC) shows some benefits compared to the single link transmission. If possible the *Fixed Ratio Combining* (FRC) should be used. This only need knowledge of the average channel quality, and shows a much better performance than the ERC. If knowledge of the current state of the channel quality is available more sophisticated combining methods can be used. The *Enhanced Signal-to-Noise Ratio Combining* (ESNRC) has shown a very good performance considering that a rough approximation of the channel quality is sufficient.

The location of the relay is crucial to the performance. The best performance was achieved when the relay is at equal distance from the sender and the destination or slightly closer to the former. In general the relay should not be to far from the line between the two stations.

5.2 Further Work

There are many ways to take this project further:

The current arrangement can be enhanced to get more detailed results. An error correcting code in combination with a checksum could be added to the signal to show the potential of fully decode and re-encode the data at the relay. The relay could then correct wrong detected symbols and send the message to the destination only if the data was corrected. This means that if the relay sends a burst of data, the whole sequence is correct.

Another approach would be to enhance the diversity protocol with some feedback in combination with the error correcting code as described above. The destination could try to decode the data received from the sending station and send a short message to relay and sender that they know if the transmission was successful. If this was not the case the relay can send the data as usual. But otherwise it is useless wasting bandwidth by sending the message again. Instead the base can send the next message.

During a wireless communication the involved stations might moving around. Sometimes there is a well placed mobile station available that can be used as a relay. But most of the time the mobile station is not located optimally or is too far away to be useful as a relay at all. It would be very interesting to see the overall performance of this more complicated system.

Another way to enhance this project would be to use more than one relay. Such a system should show higher levels of diversity and might have a lot of potential.

Appendix A

Matlab Code of the Simulation

A.1 Main Sequence - main.m

```
%Cooperative Diversity - Main Sequence

tic

% --------------
% Set Parameters
nr_of_iterations = 10^3;
SNR = [-10:2.5:15];
use_direct_link = 1;
use_relay = 1;

global statistic;
%statistic = generate_statistic_structure;

global signal;
signal = generate_signal_structure;
signal(1).nr_of_bits = 2^10;
signal.modulation_type = 'QPSK'; % 'BPSK', 'QPSK'
calculate_signal_parameter;

channel = generate_channel_structure;
channel(1).attenuation(1).pattern = 'Rayleigh';% 'no','Rayleigh'
channel.attenuation.block_length = 1;
channel(2) = channel(1);
channel(3) = channel(1);
channel(1).attenuation.distance = 1;
```

```
channel(2).attenuation.distance = 1;
channel(3).attenuation.distance = 1;

rx = generate_rx_structure;
rx(1).combining_type = 'ESNRC'; %'ERC','FRC','SNRC','ESNRC','MRC'
rx(1).sd_weight = 3;

global relay;
relay = generate_relay_structure;
relay(1).mode = 'AAF'; %'AAF', 'DAF'
relay.magic_genie = 0;
relay(1).rx(1) = rx(1); % same beahaviour

% ----------------
% Start Simulation
BER = zeros(size(SNR));

for iSNR = 1:size(SNR,2)
 channel(1).noise(1).SNR = SNR(iSNR);
 channel(2).noise(1).SNR = SNR(iSNR);
 channel(3).noise(1).SNR = SNR(iSNR);

 disp(['progress: ',int2str(iSNR),'/',int2str(size(SNR,2))])

 for it = 1:nr_of_iterations;
  % --------------
  % Reset receiver
  rx = rx_reset(rx);
  relay.rx = rx_reset(relay.rx);

  % -----------
  % Direct link
  if (use_direct_link == 1)
   [channel(1), rx] = add_channel_effect(channel(1), rx,...
   signal.symbol_sequence);
   rx = rx_correct_phaseshift(rx, channel(1).attenuation.phi);
  end

  % ---------
  % Multi-hop
  if (use_relay == 1)
   % Sender to relay
   [channel(2), relay.rx] = add_channel_effect(channel(2),...
    relay.rx, signal.symbol_sequence);
```

```
   relay = prepare_relay2send(relay,channel(2));

   % Relay to destination
   [channel(3), rx] = add_channel_effect(channel(3), rx,...
    relay.signal2send);

   switch relay.mode
    % Correct phaseshift
    case 'AAF'
     rx = rx_correct_phaseshift(rx,...
      channel(3).attenuation.phi + channel(2).attenuation.phi);
    case 'DAF'
     rx = rx_correct_phaseshift(rx,channel(3).attenuation.phi);
   end
  end

  % Receiver
  [received_symbol, signal.received_bit_sequence] = ...
    rx_combine(rx, channel, use_relay);

  BER(iSNR) = BER(iSNR) + sum(not(...
   signal.received_bit_sequence == signal.bit_sequence));

  if (BER(iSNR) > 10000)
   % Stop iterate
   break;
  end

 end % Iteration

 if (BER(iSNR)<100)
  warning(['Result might not be precise when SNR equal ',...
    num2str(SNR(iSNR))])
 end
 BER(iSNR) = BER(iSNR) ./ it ./ signal.nr_of_bits;
end

% -----------------------------------
% Present the result of the simulation
txt_distance = [' - distance: ',...
  num2str(channel(1).attenuation.distance), ':',...
  num2str(channel(2).attenuation.distance), ':',...
  num2str(channel(3).attenuation.distance)];
```

```
txt_distance='';

if (use_relay == 1)
 if (relay.magic_genie == 1)
  txt_genie = ' - Magic Genie';
 else
  txt_genie = '';
 end

 txt_combining = [' - combining: ', rx(1).combining_type];
 switch rx(1).combining_type
  case 'FRC'
  txt_combining = [txt_combining, ' ',...
    num2str(rx(1).sd_weight),':1'];
 end
  add2statistic(SNR,BER,[signal.modulation_type, ' - ',...
    relay.mode, txt_combining, txt_distance, txt_genie])
else
 switch channel(1).attenuation.pattern
  case 'no'
   txt_fading = ' - no fading';
  otherwise
   txt_fading = ' - Rayleigh fading';
 end
 add2statistic(SNR,BER,[signal.modulation_type,txt_fading])
end

% % -----------------
% % Graphs to compare
SNR_linear = 10.^(SNR/10);
% add2statistic(SNR,ber(SNR_linear,'BPSK', 'Rayleigh'),'BPSK - single link transmis
% add2statistic(SNR,ber_2_senders(SNR_linear, 'QPSK'),'QPSK - 2 senders')
show_statistic;

toc
```

A.2 Initialise

A.2.1 Signal Parameter - calculate_signal_parameter.m

```
function calculate_signal_parameter
% Calculates some additional signal parameters

global signal;
```

```
% Bits per symbol
switch signal.modulation_type
 case 'BPSK'
  signal.bits_per_symbol = 1;

 case 'QPSK'
  signal.bits_per_symbol = 2;
  if (signal.nr_of_bits/2 ~= ceil(signal.nr_of_bits/2))
   error(['Using QPSK, number of bits must be a multiple of 2'])
  end

 otherwise
  error(['Modulation-type unknown: ', signal.modulation_type])
end

% Number of symbols to transfer
signal.nr_of_symbols = signal.nr_of_bits/signal.bits_per_symbol;

% Bit sequence (random sequence of -1 and 1)
signal.bit_sequence = floor(rand(1,signal.nr_of_bits)*2)*2-1;

% Symbol sequence
signal.symbol_sequence = bit2symbol(signal.bit_sequence);
```

A.2.2 Reset Receiver - rx_reset.m

```
function [rx] = rx_reset(rx);
% Reset the receiver

rx.signal2analyse = [];
```

A.3 Channel - add_channel_effect.m

```
function [channel, rx] = add_channel_effect(channel,rx,...
 signal_sequence)
% Add noise fading and path loss

global signal;

%---------------------
% Fading and path loss

channel.attenuation.d = 1 / (channel.attenuation.distance ^ 2);
```

```
% Path loss is constant for the whole transmission

switch channel.attenuation.pattern
 case 'no'
  % No fading at all (only path loss)
  channel.attenuation.phi = zeros(size(signal_sequence));
  channel.attenuation.h = ones(size(signal_sequence)) * ...
   channel.attenuation.d;
  channel.attenuation.h_mag = channel.attenuation.h;

 case 'Rayleigh'
  % Rayleigh fading and path loss
  nr_of_blocks = ceil(size(signal_sequence,2) /...
   channel.attenuation.block_length);
  h_block = (randn(nr_of_blocks,1) + j * randn(nr_of_blocks...
   ,1)) * channel.attenuation.d;

  h = reshape((h_block * ...
   ones(1, channel.attenuation.block_length))', 1,...
   channel.attenuation.block_length * nr_of_blocks);
  channel.attenuation.h = h(1:(size(signal_sequence,2)));

  [channel.attenuation.phi, channel.attenuation.h_mag] =...
   cart2pol(real(channel.attenuation.h),...
   imag(channel.attenuation.h));

  channel.attenuation.phi = -channel.attenuation.phi;

 otherwise
  error(['Fading-pattern unknown: ',...
    channel.attenuation.pattern])
end


% ------------
% Noise (AVGN)

S = mean(abs(signal_sequence).^2);
SNR_linear = 10^(channel.noise.SNR/10);

%SNR = a^2/(2*sigma^2)
channel.noise.sigma = sqrt(S / (2 * SNR_linear));
noise_vector = (randn(size(signal_sequence)) +...
 j * randn(size(signal_sequence))) * channel.noise.sigma;
```

```
% Add fading, path loss and noise to the signal
rx.received_signal = signal_sequence .* channel.attenuation.h...
 + noise_vector;
```

A.4 Receiver

A.4.1 Correct Phase Shift - rx_correct_phaseshift.m

```
function [rx] = rx_correct_phaseshift(rx, phi);
% Correct phaseshift of the received signal

switch rx.combining_type
 case 'MRC'
  % No phaseshift correction in MRC mode.
  % Phaseshift will be corrected when the received signal are
  % combined
  rx.signal2analyse = [rx.signal2analyse; rx.received_signal];

 otherwise
  % Assuming that perfect phaseshift estimation possible
  rx.signal2analyse = [rx.signal2analyse;...
    rx.received_signal .* exp(j * (phi))];
end
```

A.4.2 Combine Received Signals - rx_combine.m

```
function [symbol_sequence, bit_sequence] = rx_combine(...
 rx, channel, use_relay);
% Combine all received signals

global signal;
global relay;

values2analyse = rx.signal2analyse;

if (use_relay == 1) & (relay.magic_genie == 1)
 switch relay.mode
  case 'DAF'
   values2analyse(2,:) = (signal.symbol_sequence ==...
    relay.symbol_sequence) .* values2analyse(2,:);

  otherwise
   error(['Magic Genie works only with "DAF"'])
```

```
 end
end

switch rx.combining_type
 case 'MRC'
  switch relay.mode
   case 'DAF'
    if (use_relay == 0)
     h = conj(channel(1).attenuation.h);
    else
     h = conj([channel(1).attenuation.h; channel(3).attenuation.h]);
    end
    bit_sequence = (mean(symbol2bit(h .*...
     values2analyse),1)>=0)*2-1;

   otherwise
    error('Maximum ratio combining works only with DAF')
  end

 case {'ERC', 'FRC', 'SNRC', 'ESNRC'}
  % The received values are already in phase
  values2analyse = symbol2bit(values2analyse);

  switch rx.combining_type
   case 'ERC'
    % Equal Ratio Combining
    bit_sequence = (mean(values2analyse,1)>=0)*2-1;
   case 'FRC'
    % Fixed Ratio Combining
    if (use_relay == 0)
     bit_sequence = (mean(values2analyse,1)>=0)*2-1;
    else
     bit_sequence = (mean([rx.sd_weight;1] *...
      ones(1,size(values2analyse,2)) .*...
      values2analyse,1)>=0)*2-1;
    end

   case {'SNRC', 'ESNRC'}
    % Ratio depending on the SNR
    if (use_relay == 0)
     bit_sequence = (mean(values2analyse,1)>=0)*2-1;
    else
     SNR_direct = estimate_channel_SNR(channel(1), ...
      signal.modulation_type, relay.mode);
```

```
     SNR_via = estimate_channel_SNR([channel(2),...
       channel(3)], signal.modulation_type, relay.mode);

     if (signal.modulation_type == 'QPSK')
      SNR_via = [SNR_via, SNR_via];
      SNR_direct = [SNR_direct, SNR_direct];
     end

     switch rx.combining_type
      case 'SNRC'
       bit_sequence_ratio = (sum([SNR_direct; SNR_via] .* ...
        values2analyse,1)>=0)*2-1;
       bit_sequence_inf = (mean(values2analyse,1)>=0)*2-1;
       SNR_equal_inf = ((SNR_via == inf) &...
        (SNR_direct == inf));
       bit_sequence = SNR_equal_inf .* bit_sequence_inf +...
        not(SNR_equal_inf) .* bit_sequence_ratio;

      case 'ESNRC'
       % .1 < SNR_direct/SNR_via < 10 : the to channels are
       % weighted equally otherwise only the channel with the
       % higher SNRR is used.
       use_direct = (SNR_direct == inf) & (SNR_via ~= inf)...
        | ((SNR_direct ./ SNR_via) > 10);
       use_via = (SNR_via == inf) & (SNR_direct ~= inf) | ...
        ((SNR_via ./ SNR_direct) > 10);
       use_equal_ratio = not(use_direct + use_via);

       bit_sequence_equal_ratio =...
        (mean(values2analyse,1)>=0)*2-1;
       bit_sequence_direct = (values2analyse(1,:)>=0)*2-1;
       bit_sequence_via = (values2analyse(2,:)>=0)*2-1;
       bit_sequence = ...
        use_equal_ratio .* bit_sequence_equal_ratio + ...
        use_direct .* bit_sequence_direct + ...
        use_via .* bit_sequence_via;
     end
    end

   otherwise
    error(['Combining-type unknown: ',rx.combining_type])
  end

end
```

```
symbol_sequence = bit2symbol(bit_sequence);
```

A.5 Relay - prepare_relay2send.m

```
function relay = prepare_relay2send(relay,channel);
% Relay: Prepare received signal to make it ready to send

global signal;

switch relay.mode
 case 'AAF'
  % Amplify and Forward
  % Normalise signal power to the power of the original signal

  xi = abs(signal.symbol_sequence(1))^2;
  relay.amplification = sqrt(xi ./ (xi .*...
   channel(1).attenuation.h_mag .^ 2 + 2 .*...
   channel(1).noise.sigma .^ 2));

  relay.signal2send = ...
   relay.rx.received_signal .* relay.amplification;

 case 'DAF'
  % Decode and Forward
  relay.rx = rx_correct_phaseshift(relay.rx,...
   channel.attenuation.phi);
  relay.symbol_sequence = rx_combine(relay.rx, channel, 0);
  relay.signal2send = relay.symbol_sequence;

 otherwise
  error(['Unknown relay-mode: ', relay.mode])
end
```

A.6 Structures

A.6.1 Signal - generate_signal_structure.m

```
function [signal_structure] = generate_signal_structure();
% Creates the structure for all signal parameters

signal_structure = struct(...
 'nr_of_bits',{},...         % nr of bits to transfer
```

```
 'nr_of_symbols',{},...       % nr of symbols to transfer
 'bits_per_symbol',{},...     % BPSK (1 bit/symbol)
 ...                          % QPSK (2 bits/symbol)
 'modulation_type',{},...     % 'BPSK', 'QPSK'
 'bit_sequence',{},...        % bit sequence of the signal
 'symbol_sequence',{},...     % symbol sequence of the signal
 'received_bit_sequence',{});% bit sequence after transmission
```

A.6.2 Channel - generate_channel_structure.m

```
 function [channel_structure] = generate_channel_structure();
% Creates the structure for all channel parameters

attenuation_structure = generate_attenuation_structure;
noise_structure = generate_noise_structure;

channel_structure = struct(...
 'attenuation', attenuation_structure,... % fading
 'noise', noise_structure);               % noise


function [fading_structure] = generate_attenuation_structure();
% Creates the structure for all fading parameters

fading_structure = struct(...
 'pattern',{},...    % 'no', 'Rayleigh'
 'distance', {},...  % distance
 'd', {},...         % path loss
 'h',{},...          % attenuation incl. phaseshift
 'h_mag',{},...      % magnitude
 'phi',{},...        % phaseshift
 'block_length',{}); % lenth of the block (bit/block)


function [noise_structure] = generate_noise_structure();
% Creates the structure for all noise parameters

noise_structure = struct(...
 'SNR',{},... % Signal to Noise Ratio (dB)
 'sigma',{}); % sigma of AVGN
```

A.6.3 Receiver - generate_rx_structure.m

```
function [rx_structure] = generate_rx_structure();
```

```
% Creates the structure for all receiver (Rx) parameters

rx_structure = struct(...
 'combining_type',{},... % 'ERC', 'SNRC', 'ESNRC', 'MRC'
 'sd_weight',{},...      % used for 'FRC'
 ...                     % relay link is weighted one
 'received_signal',{},...% signal originally received. after
 ...                       phaseshift is undone, saved in
 ...                       signal2analyse
 'signal2analyse',{});   % one row per incomming signal, which
                         % then are combined to estimate the
                         % bit-sequence
```

A.6.4 Relay - generate_relay_structure.m

```
function [relay_structure] = generate_relay_structure();
% Creates the structure for all relay parameters

rx_structure = generate_rx_structure;

relay_structure = struct(...
 'mode',{},...               % 'AAF' (Amplify and Forward)
 ...                           'DAF' (Decode and Forward)
 'magic_genie',{},...        % 'Magic Genie
 'amplification',{},...      % used in AAF mode
 'symbol_sequence',{},...    % used in DAF mode
 'signal2send',{},...        % Signal to be send
 'rx',struct(rx_structure)); % Receiver
```

A.6.5 Statistic - generate_statistic_structure.m

```
function [statistic_structure] = generate_statistic_structure();
% Creates the structure for all statistic parameters

statistic_structure = struct(...
  'xlabel','SNR [dB]',...            % label x-axis
  'ylabel','Probability of error',... % label y-axis
  'x',[],...                         % one graph per row x-axis
  'y',[],...                         %                   y-axis
  'legend','');                      %                   legend
```

A.7 Conversions

A.7.1 SNR to BER - ber2snr.m

```
function y = snr2ber(x)
% Calculates the BER of the channel

global signal;

switch signal.modulation_type
 case 'QPSK'
  y = q(sqrt(x));

 case 'BPSK'
  y = q(sqrt(2 * x));

 otherwise
  error(['Modulation-type unknown: ', signal.modulation_type])
end
```

A.7.2 BER to SNR - ber2snr.m

```
function y = ber2snr(x);
% Calculates the SNR of the channel
%
% The SNR of the channel can be estimated/calculated when the
% BER of the channel is known.

global signal;

switch signal.modulation_type
 case 'QPSK'
  y = qinv(x) .^ 2;

 case 'BPSK'
  y = qinv(x) .^ 2 / 2;

 otherwise
  error(['Modulation-type unknown: ', signal.modulation_type])
end
```

A.7.3 Symbol Sequence to Bit Sequence - symbol2bit.m

```
function [bit_sequence] = symbol2bit(symbol_sequence);
% Calculates bit_sequence from the symbol_sequence depending
```

```
% on the modulation type

global signal;

switch signal.modulation_type
 case 'BPSK'
  bit_sequence = symbol_sequence;

 case 'QPSK'
  bit_sequence = [real(symbol_sequence), imag(symbol_sequence)];

 otherwise
  error(['Modulation-type unknown: ', signal.modulation_type])
end
```

A.7.4 Bit Sequence to Symbol Sequence - bit2symbol.m

```
function [symbol_sequence] = bit2symbol(bit_sequence);
% Calculates symbol_sequence from the bit_sequence depending on
% the modulation type

global signal;

switch signal.modulation_type
 case 'BPSK'
  symbol_sequence = bit_sequence;

 case 'QPSK'
  symbol_sequence = bit_sequence(1:signal.nr_of_symbols) + j*...
   bit_sequence(signal.nr_of_symbols + 1 : signal.nr_of_bits);

 otherwise
  error(['Modulation-type unknown: ', signal.modulation_type])
end
```

A.8 Statistic

A.8.1 Add Statistic - add2statistic.m

```
function add2statistic(x,y,leg);
% Add graph to statistic

global statistic;
```

```
statistic.x = [statistic.x;x];
statistic.y = [statistic.y;y];
statistic.legend = strvcat(statistic.legend,leg);
```

A.8.2 Show Statistic - show_statistic.m

```
function [handle] = show_statistic(colour_bw, order);
% Shows the result in a plot

global statistic;

if (nargin<1), colour_bw = 0; end
if (nargin<2), order = 1:size(statistic.x,1); end

 if (colour_bw == 1)
  colours = ['k-o';'k-*';'k-s';'k-+';'k-^';'k-h';'k-v';'k-p'];
 else
  colours = ['b-o';'r-d';'g-s';'k-v';'m-^';'b-<';'r->';'g-p'];
 end

 legend_ordered = [];

handle = figure;
colour = 0;

for n = order
 colour = colour + 1;
 semilogy(statistic.x(n,:),statistic.y(n,:),colours(colour,:));
 legend_ordered = strvcat(legend_ordered,statistic.legend(n,:));
 hold on
end

grid on;
legend (legend_ordered,3)
xlabel (statistic.xlabel)
ylabel (statistic.ylabel)
```

A.9 Theoretical BER

A.9.1 Single Link Channel - ber.m

```
function [y] = ber(snr, modulation_type, fading_type);
% Calculates the BER(SNR) depending on the modulation-type and
% the fading-type
```

```
switch fading_type
 case 'Rayleigh'
  switch modulation_type
   case 'BPSK'
    y = (1 - sqrt(snr ./ (1 / 2 + snr))) / 2;

   case 'QPSK'
    y = (1 - sqrt(snr ./ (1  + snr))) / 2;

   otherwise
    error(['Modulation-type unknown: ', modulation_type])
  end

 case 'no'
  switch modulation_type
   case 'BPSK'
    y = q(sqrt(2 * snr));

   case 'QPSK'
    y = q(sqrt(snr));

   otherwise
    error(['Modulation-type unknown: ', modulation_type])
  end

 otherwise
  error(['Fading-type unknown: ', fading_type])
end
```

A.9.2 Two Independent Senders - ber_2_senders.m

```
function y = ber_2_senders(SNR_avg, modulation_type);
% BER(SNR) using two senders. The (average) SNR is assumed to be
% equal for both channel

switch modulation_type
 case 'BPSK'
  mu = sqrt(SNR_avg ./ (1 / 2 + SNR_avg));
 case 'QPSK'
  mu = sqrt(SNR_avg ./ (1 + SNR_avg));
 otherwise
  error(['Modulation-type unknown: ', modulation_type])
end
```

```
y = 1 / 4 * (1 - mu) .^ 2 .* (2 + mu);
```

A.10 Math functions

A.10.1 Q-function - q.m

```
function [y] = q(x);
% Q-probability function

y = erfc(x / sqrt(2)) / 2;
```

A.10.2 Inverse Q-function - invq.m

```
function [y] = qinv(x);
% Inverse Q-probability function

y = erfcinv(x*2) *sqrt(2);
```

Bibliography

[1] John G. Proakis, *Digital Communications*, McGraw-Hill, 4th edition (international), 2001.

[2] J. Nicholas Laneman, David N. C. Tse, and Gregory W. Wornell, *Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behaviour*, 2002.