University of Oslo
Department of Informatics

Simulation and design of MIMO algorithms for correlated wireless channels

Hilde Skjevling

Cand Scient Thesis

25th July 2003

Preface

This thesis is submitted to the Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo, for the Candidata Scientiarum (Cand.Scient.) degree.

Within the Department of Informatics, I have been working under the supervision of the Digital Signal Processing and Image Analysis Group.

I would like to thank my magnificent supervisors, Prof. David Gesbert and Prof. Nils Christophersen, for invaluable help and motivation.

Hilde Skjevling
25th July 2003

Contents

List of Figures

Abstract

Spatial multiplexing and space-time codes are competing ways of extracting capacity from MIMO wireless systems. We present examples of both approaches and study their performance, in particular in the case of correlated channels. Next, we show that multiplexing and diversity oriented schemes (like STC) react differently to the correlation structure of MIMO arrays.

We address the problem of finding an optimal combination of multiplexing and diversity in a MIMO system. We develop a combining approach in the form of an optimal spatial assignment of antennas, in order to multiplex space-time coded symbol blocks. We call this scheme SMAL, and develop it in two versions, first for the case when instantaneous channel information is available at the transmitter, and second when only long-term correlation statistics are known. We investigate the performance of both versions in the practical case when correlation is not uniform across all antenna pairs (e.g. in linear arrays).

The SMAL scheme is tested for two levels of correlated fading between neighbouring antenna elements, $r = 0.29$ and $r = 0.90$. A comparison between the two versions shows that the instantaneous SMAL is especially useful at low levels of correlated fading ($r = 0.29$). Under conditions of heavily correlated fading ($r = 0.90$) the correlation-based approach performs just as well and is preferred because of its lower complexity.

We also vary the number of transmit and receive (N, M) antennas, and test the SMAL scheme for the two MIMO systems $N = M = 4$ and $N = M = 6$.

For the instantaneous version of SMAL and $r = 0.29$, we show improvements of over 2 dB over the case of random pattern selection , at a target bit-error rate of 10^{-4}, both for $N = M = 4$ and $N = M = 6$. With the statistical SMAL for $r = 0.90$, a performance gain of almost 5 dB is shown in the case of $N = M = 4$.

The development and results of the SMAL scheme have also resulted in the submission of a conference article to NORSIG 2003 [17].

Chapter 1

Introduction

A wireless communication system transfers information through space, from one point to another. In its simplest form, it consists of a single antenna on each side of the wireless communication channel. This is what we call a SISO (Single-Input Single-Output) or a 1-1 system.

To offer high-quality communication services, the reliability and speed of the transmission are of key importance. Improved electrical components, along with advances in coding and modulation techniques are some causes for the high reliability and speed we enjoy today, but the demand for further increase seems to be without limit.

MIMO systems

One approach towards additional gain in quality or rate is to use multiple antennas, yielding SIMO, MISO and MIMO systems.

In SIMO (Single-Input Multiple-Output) and MISO (Multiple-Input Single-Output) systems multiple antennas are employed at the receive or the transmit side, respectively.

When arrays of multiple antennas are used in both ends of the communication channel, a MIMO (Multiple-Input Multiple-Output) system is formed, (see e.g. [8] for a recent tutorial). A MIMO system with N transmit and M receive antennas gives rise to NM wireless channel propagation coefficients in the frequency non-selective case.

The maximum achievable order of diversity is the number of such channel coefficients that are statistically independent. SIMO, MISO and MIMO systems can be used to improve the reliability of transmission through this property of spatial diversity.

The antenna arrays may be configured in a number of different patterns, for example along a line or in a circle. In our simulations, we will assume uniform, linear arrays (ULA) [21] on both sides, although the algorithms developed here are transparent to the array configuration.

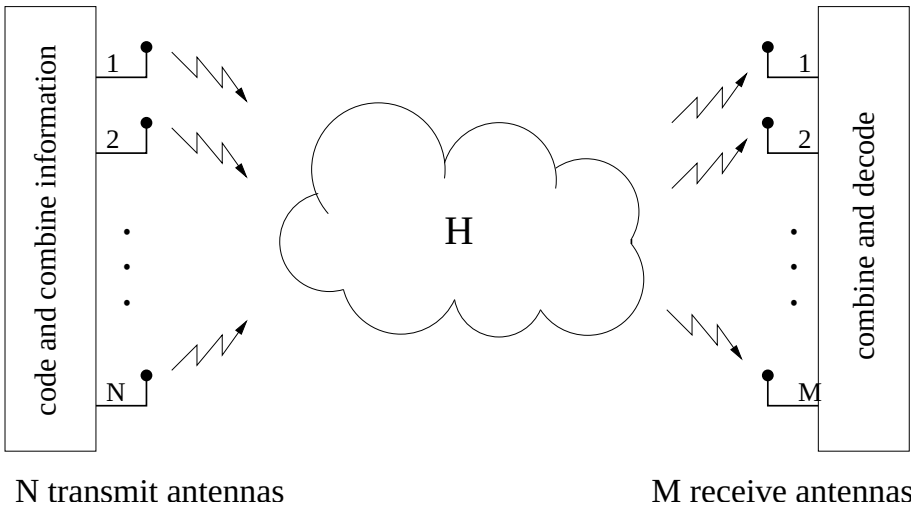

Figure 1.1: A general MIMO model.

A MIMO system with N transmit and M receive antennas is illustrated in figure 1.1. The propagation environment is drawn as a cloud, labelled H.

MIMO algorithms

By appropriate "combining" of the signals before transmission and after reception, we may improve the quality and/or speed of the communication process, compared to traditional 1-1 systems. The quality of a wireless communication process is measured by the bit-error rate (BER) between the original bit sequence and the detected bits at the receiver. The speed metric is given by the bit rate efficiency (in bits/sec/Hz).

A multitude of algorithms provide different forms of combining suitable for MIMO systems. The algorithms describe how to do coding in time and space before transmission along with the detection process after reception. The schemes may be divided into two categories; diversity-oriented (MD for MIMO diversity) and spatial multiplexing (SM) schemes.

In diversity-oriented transmission the information symbols are spread over multiple antennas, to mitigate the fading effects of the individual channels. With a simple SISO system, when the only observable channel goes into a deep fade, it is not possible to recover the signal. Multiple antenna systems using MD transmit schemes rely on the likely assumption that all the transmit channels are not heavily faded at the same time. The obtained diversity is used to ensure a more constant signal strength at the receiver. The effect of diversity transmission manifests itself in a

lower BER.

The low error rate provided by MD schemes may in turn be exploited to increase the capacity [14]. This is done by using higher order modulation.

Some examples of MD transmission schemes are transmit and receive MRC (Maximum Ratio Combining) [14], for MISO and SIMO systems, respectively. For MIMO systems, an extension of MRC has been developed, based on singular value decomposition. We refer to it as the maximum singular vector approach (MSVA), and it requires channel knowledge at both sides of the channel. It uses the top singular vectors of the communication channel as transmit and receive weights [2].

Yet another way to do diversity transmission is by Space-Time Coding (STC) [20]. One simple, but powerful STC algorithm is the Alamouti scheme [1]. By building orthogonal blocks of space-time coded information symbols the receiver may retrieve the information symbols with a simple linear receiver. This leads to a significant decrease in the number of errors is achievable.

For the second category of MIMO schemes, spatial multiplexing schemes use the multiple antennas of MIMO systems to increase the data rate in a more direct way than in MD. This is done by sending independent substreams of information symbols from each transmit antenna. In general, the capacity of the system is limited by the rank of the channel. Hence, for a full rank channel, the theoretical capacity is linear to $\min(N, M)$. In order to retrieve the transmitted symbols, the scheme implies that $M \geq N$. One famous spatial multiplexing scheme is Diagonal BLAST (Bell Labs Layered Space-time), also called D-BLAST [6]. A simpler version is the Vertical BLAST (V-BLAST) [22] [4].

Effects of correlation

The performance of many MIMO algorithms depends on the order of diversity and the rank of the channel matrix, both of which are affected by correlation between antennas. The maximum number of independent coefficients in a MIMO system is the product NM, setting the upper limit for the diversity order. The rank is at most $\min(N, M)$. In the likely case when the coefficients are spatially correlated to some degree the effective diversity order and the rank may not reach their upper limits. As the correlation levels on both sides of the channel rise, the individual channels of neighbouring antennas become increasingly dependent on each other. In fact, with fully correlated channels, the MIMO system has but a single independent channel, and both the diversity order and rank equals 1.

One key point of this thesis is that the performance of the algorithms suffers when the channels are correlated, but to different degrees. For

some MIMO algorithms, such as SM approaches, correlation is more destructive than for others. The impact of a decrease in the diversity order-/channel rank will be examined through studies of error rates for both independent (uncorrelated) and correlated channel coefficients.

Combining spatial multiplexing and diversity schemes

MIMO diversity and spatial multiplexing have so far been considered as competing approaches to exploiting the spatial dimension offered by MIMO systems. The trade-offs between the two are only beginning to be understood [8].

First, it is clear that diversity schemes yield diminishing returns when increasing the number of antennas [3]. This suggests that using all antennas on one array for diversity through for example STC is not a sensible approach.

It is also known that an SM scheme with a simple (e.g. linear) receiver lags in performance because of a lack of diversity [9], implying that using SM only on a transmit array is not a good idea either. Therefore, one of the remaining important problems in the field of MIMO algorithms is the question:

- How do we design schemes that offer the benefits of both MD and SM approaches, i.e. both diversity *and* a direct increase in the data rate?

Goals of the thesis

We address the above question in two ways. First, we compare the performance of known MIMO algorithms for different levels of correlated fading among the channels, and then we attempt to combine the MD and SM approaches in one scheme.

Regarding the combination, previous work has been carried out in the case where the desired combination is to switch between MD and SM *over time* [12]. Either MD or SM is chosen as the transmit algorithm, a choice based on instantaneous channel state information. In [12], the proposed scheme exploits the fact that MD is sensitive to total channel matrix energy, while SM performance depends on the channel eigenvalue spread.

Our approach will be different, as we focus on the problem of switching between SM and MD *in space*, i.e. both schemes are used simultaneously on the same array. To our knowledge, this problem has not been addressed before. Our attempt is a generalization of the work in [12], and the idea is to spatially multiplex several blocks of Alamouti coded symbols. We refer to this scheme as SMAL (Spatial Multiplexing of

ALamouti), and develop it in two versions for use under different levels channel knowledge.

To summarise, the contributions of this thesis are

- a comparison of known MIMO algorithms for different levels of correlated fading among the channels
- the proposed SMAL scheme, combining MD and SM, developed for the case when instantaneous and full channel feedback is available at the transmit side.
- the proposed SMAL scheme, combining MD and SM, developed for the case when only long-term correlation statistics are known to the transmitter.

We show the performance gains of both the instantaneous and the statistical version of SMAL, and evaluate the loss in using statistical information alone. These gains are illustrated with bit-error rate simulations using a fading channel model. The latter results are published in a submission to NORSIG 2003 [17].

The organisation of this thesis

The text is organised in 5 chapters, starting with this introduction. In addition, there are some appended sections at the end. Chapter 2 presents background on wireless communication and builds a channel model for MIMO systems. This is followed by a review of some existing MIMO algorithms in chapter 3, along with the study of their compared performance in the case of correlated channels.

In chapter 4, the SMAL combination of algorithms is proposed and evaluated, in search of further improvement. This is the main contribution of the thesis. The conclusions in chapter 5 summarise the results from the preceding chapter's analysis.

Appendix A holds some larger figures and extended mathematical developments that did not fit in the main text.

Common abbreviations and acronyms are explained when they appear in the text for the first time, but also in appendix B, along with a list of the mathematical symbols used in the text. The last chapter found as supplementary material is the Matlab code written to perform the simulations, in appendix C.

Chapter 2

Background on wireless communication and antenna systems

This chapter presents background and necessary theory, so that later algorithms and results can be described using this foundation. We examine the characteristics of digital wireless transmission, and describe what happens to a signal when it is propagating in space. From this knowledge we develop a suitable mathematical model, first for a SISO system, and later extended to the more general MIMO case.

2.1 Digital wireless transmission

Wireless communication systems transmit information from one point to another, with air as the propagation medium. Because electromagnetic waves are used to carry the information symbols, rather than the traditional wire, these systems are called 'wireless'. The transmission procedure requires at least one antenna at each end-point, in combination with signal processors and amplifiers. At the transmit side the system must be able to obtain and digitise information, transform it into transmittable form and send it out over the wireless channel. At the other end of the channel, the receive part of the system registers the wave-forms and extracts the original information from them.

We want to develop a conceptual model for a wireless digital communication system with multiple antennas, and in particular a model that depicts the parts important to this text. The next section starts out with a general model and later adjusts it to the topic of this thesis.

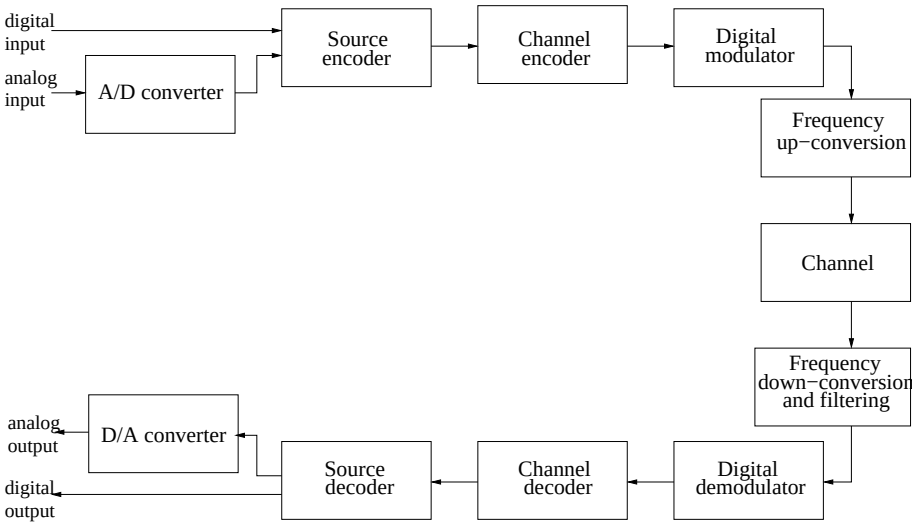

Figure 2.1: Basic elements of a digital communication system

2.1.1 Modelling a digital communication system

A general digital communication system, based on the one presented in [14], is shown in figure 2.1. First, we briefly explain the elements of such a general system, with reference to the boxes in figure 2.1.

A general digital communication system accepts digital input, and may convert possible analog input into digital form, using an A/D converter. The source encoder accepts a sequence of binary digits and codes them in an efficient way, with fewer bits and lower redundancy. The coded bit stream is passed on to the channel encoder, which inserts extra bits that will be used to detect and correct errors after reception. The extra bits introduce redundancy in a controlled manner. One example of channel encoding is to map a k-bit sequence into a unique n-bit sequence, where $n > k$.

Next, the digital modulator accepts the output from the channel encoder, and transforms this stream of bits into a stream of analog waveforms. In this text, M-ary modulation schemes will be used, where b coded bits can be transmitted at once, using one of $M = 2^b$ different waveforms, described later in this section (2.1.1). The waveforms are then frequency up-converted by impressing them on a high-frequency carrier wave.

Now, the signal wave is ready to be sent out over the wireless channel. This channel is assumed to be bandpass, with a limited bandwidth B_c around a centre frequency f_c. Such channels will only pass signals with frequency contents within the channels own bandwidth. When the

signals and channels have bandwidths much smaller than the carrier frequency, they are referred to as *narrow-band band-pass signals and systems*. Such systems will be used later on.

At the receiving end, a signal affected by the channel is registered. Imperfect electrical components will invariably add noise, so this is also assumed when modelling a communication system, although not depicted in the figure. The signal wave is frequency down-converted, and the demodulation extracts bit-values from the signal waveforms using some form of decision method.

The channel decoder removes and checks the redundant information introduced by the channel encoder. If the check-sums are not correct, an error is detected, and may be corrected. The source decoder reverses the encoding performed by the source encoder, attempting to restore the original data. The performance of the channel and source decoders depend on the amount of distortion introduced by the propagation and the added noise.

This text is mainly concerned with MIMO systems and algorithms, and for our purpose the model in figure 2.1 is not the optimal. Some of the model's components are of little importance to our theme, while other parts need further elaborations. We build a model that more accurately describes the focus of this thesis, based on the one in figure 2.1.

The first change we make is to ignore the frequency up-conversion of the low-pass information signal to a transmittable bandpass signal, along with the down-conversion after reception. This reduces the complexity of our model, allowing us to represent the transmission in baseband, independent of the carrier frequency.

The background for this representation is explained in the next section, after which we return to adjusting the communication system model.

Baseband representation of bandpass signals

Assume we have an information signal that we wish to transmit over a bandpass channel. To ensure passage, the signal wave is frequency up-converted, by impressing it on a cosine carrier wave with frequency f_c. This produces a bandpass signal $m(t)$, with frequency f_c, the centre frequency of the bandpass channel.

Depending on the method of modulation, both the amplitude and the phase of the carrier may be modulated according to the low-pass information signal. We show this by representing these values as time-dependent in the modulated signal, following the changing information signal. The bandpass signal can be expressed as

$$m(t) = a(t)cos\big(2\pi f_c t + \theta(t)\big) \tag{2.1}$$

This real signal $m(t)$ is what we transmit over the channel. We remark that the above expression may be rewritten as [14]

$$m(t) = Re\{a(t)e^{j\theta(t)}e^{j2\pi f_c t}\}, \tag{2.2}$$

where j is the complex number satisfying $j = \sqrt{-1}$. We label the low-pass, complex term

$$s(t) = a(t)e^{j\theta(t)} \tag{2.3}$$

the complex envelope of $m(t)$, and use it as a short-hand representation for the signal. That is; the real bandpass signal $m(t)$ is modelled as a complex low-pass signal $s(t)$. The main benefit of this is not having to involve the carrier frequency f_c. The relation between $m(t)$ and $s(t)$ is given by

$$m(t) = Re\{s(t)e^{j2\pi f_c t}\} \tag{2.4}$$

In nature, there is of course no room for complex numbers, everything is real. We understand what happens in reality by observing that (2.4) may be rewritten as

$$\begin{aligned} m(t) &= Re\{s(t)(cos(2\pi f_c t) + jsin(2\pi f_c t))\} \\ &= Re\{s(t)\}cos(2\pi f_c t) - Im\{s(t)\}sin(2\pi f_c t) \end{aligned} \tag{2.5}$$

In other words, we simply send two signals at the same time, but on orthogonal carriers, which enables us to separate the streams at the receiver. The real part is sent on the cosine carrier, while the imaginary part is sent on the sine carrier, together adding up to a real wave. The sine and cosine components are in phase quadrature, and are often referred to as the in-phase and quadrature components.

The baseband signal $s(t)$, will be used throughout this text, disregarding the frequency up- and down-conversion, as these operations do not help to clarify the subject of MIMO systems. The choice of baseband representation applies to the whole transmission model, including the representation of the propagation environment, as we see later.

The signal $s(t)$ is obtained by using a digital linear modulation method, which is discussed in the next section. Two examples of such modulation are also presented; namely QPSK and 16QAM.

Digital linear modulation and pulse shaping

The signal $s(t)$ is obtained in two steps. First we use a digital linear modulation constellation to map a bit stream b_k into complex symbols s_k and then we apply a pulse-shaping filter to these symbols.

The latter results in the analog signal $s(t)$ and is represented by the convolution

$$s(t) = \sum_{k=-\infty}^{\infty} s_k p(t - kT_s), \tag{2.6}$$

where $p(t)$ is an analog pulse-shaping filter and T_s is the symbol period, the time between two consecutive transmissions.

In most wireless communication systems, the traditional analog modulation techniques are replaced by digital schemes, representing the signals as a sequence of pulses. This choice has several advantages; for example that digital modulation methods provide better noise immunity and they are easy to implement due to compatibility with digital signal processing methods [16].

In this thesis, we focus on simulation using two digital linear modulation techniques; namely Quadrature Phase Shift Keying (QPSK) and 16 Quadrature Amplitude Modulation (16QAM), whose individual characteristics are described in the following sections. However, it is important to note that the choice of modulation scheme is not dictated by the MIMO algorithms.

Both QPSK and 16QAM generate complex symbols s_k by applying an M-ary complex, digital modulation method to a stream of bits. In this text, the information bits are modelled as random, to ensure complete generality. M-ary modulation schemes have a symbol constellation of size M, where each symbol corresponds to a unique sequence of $\log_2(M)$ bits. Hence, the bit rate is M times the baud rate, the latter representing the rate of change in the signal.

For both QPSK and 16QAM, the distance between neighbouring signal points in the constellation is an important characteristic. It is referred to as the minimum euclidean distance and denoted $d_{min}^{(e)}$. A large distance between the points reduces the probability of erroneous detection at the receiver because a transmitted constellation with a large $d_{min}^{(e)}$ is less likely to be distorted so severely that different signals are mistaken for each other. However, to increase $d_{min}^{(e)}$ uncritically consumes too much transmit power, so the average power of the symbols is limited to one; $E(|s_n|^2) = \sigma_s^2 = 1$.

Finally, we note that Gray encoding is used for both methods. This implies that the symbols that are neighbours in the constellation diagrams correspond to bit-permutations that differ by only one bit. The reason for this choice is that neighbouring symbols are most easily mistaken for each other, in which case only one bit error will occur [14].

Quadrature Phase Shift Keying

In Quadrature Phase Shift Keying (QPSK) modulation, a cosine carrier

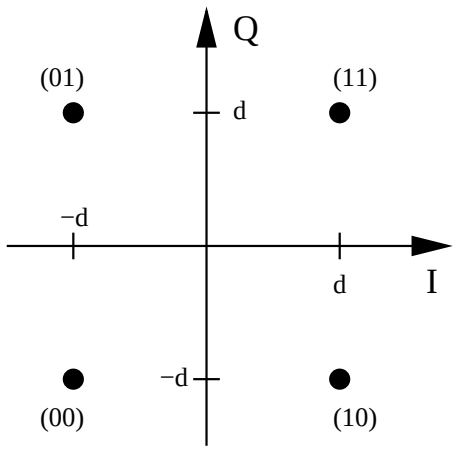

Figure 2.2: QPSK constellation.

is varied in phase while keeping a constant amplitude and frequency. Information is conveyed using the phase variations. The term "Quadrature" implies that there are 4 phases, i.e. 4 different states. That means each state can contain $R = 2$ bits of information, as $M = 2^2 = 4$ are the number of possible variations with 2 bits. The original QPSK symbols are placed in the 4 phases $\{0, \pi/2, \pi, 3\pi/2\}$, but in this text a variation is used. Our constellation is simply rotated by $\pi/4$ (counter-clockwise), as can be seen from the illustration in figure 2.2.

The rate of change (baud) in this signal determines the signal bandwidth, but the throughput or bit rate for QPSK is twice the baud rate, because $R = 2$.

QPSK is also called 4PSK, and we can define this technique for other number of states too, such as 8PSK. All the forms of PSK modulation maintain equal power in all the states, unlike other methods.

Referring to the constellation figure, we see that all QPSK symbols have equal power. The constellation also shows that for a complex QPSK symbol, we have

$$s = s_r + s_i i, \quad \text{where } |s_r| = |s_i| = d \tag{2.7}$$

The Pythagorean theorem then yields the following for the power of the symbol s:

$$|s|^2 = s_r^2 + s_i^2 = 2d^2 \tag{2.8}$$

By demanding unit power, $|s|^2 = 1$, we get that

$$d = \frac{1}{\sqrt{2}} \tag{2.9}$$

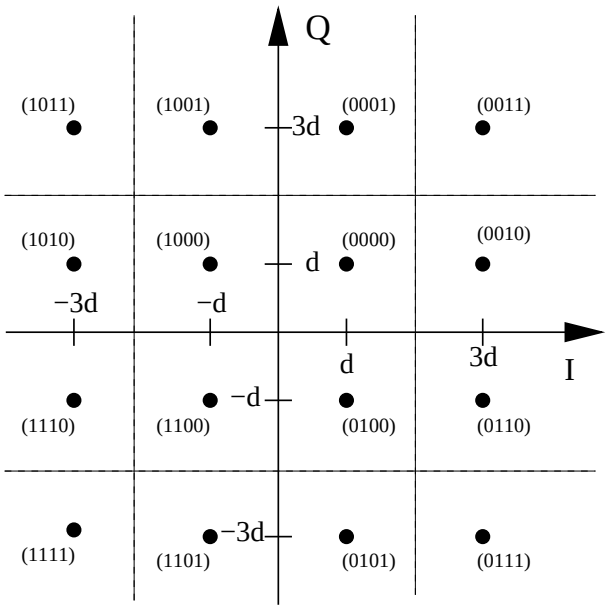

Figure 2.3: 16QAM constellation.

We may then express the minimum euclidean distance between two symbols as

$$d_{min}^{(e)} = 2d = 2/\sqrt{2} = \sqrt{2} \tag{2.10}$$

16 Quadrature Amplitude Modulation

In the family of QAM-techniques, both the amplitude and the phase of the carrier wave is modulated. Some forms are 4QAM, 8QAM, 16QAM and 64QAM, where the 4QAM has a constellation identical to that of QPSK. Each symbol is generated from $R = 4$ bits, yielding $M = 2^R = 16$ states. The constellation of states and their corresponding Gray-encoded bit-pairs are pictured in figure 2.3.

Compared to QPSK, 16QAM encodes twice as many bits into one complex symbol, increasing the bit rate. The drawback, however, is that the rate of error will increase too. The reason is that the symbols in 16QAM are closer together than those in the QPSK constellation, because a maximum level of power must be considered.

As seen from the figure, the symbols does not have equal power. Because the bits required to generate the symbols are assumed to be random, we stipulate that all the symbols in the constellation are equally likely to occur for a given sequence of $R = 4$ bits. In that case, to maintain unit power over time, we must only ensure that average symbol

power is equal to 1. With reference to the constellation, we see that there are three levels of power, depending on how far from origin a given symbol is. These levels are

$$\begin{aligned} |s_1|^2 &= d^2 + d^2 = 2d^2, \\ |s_2|^2 &= d^2 + (3d)^2 = 10d^2 \quad \text{and} \\ |s_3|^2 &= (3d)^2 + (3d)^2 = 18d^2. \end{aligned} \tag{2.11}$$

Given that all the symbols are equally likely to occur, the average power is given by

$$\begin{aligned} |s_{avg}|^2 &= \frac{4s_1^2 + 8s_2^2 + 4s_3^2}{16} \\ &= 10d^2 \end{aligned} \tag{2.12}$$

With the demand that $|s_{avg}|^2 = 1$, we see that

$$\begin{aligned} 10d^2 &= 1 \\ \implies d &= \frac{1}{\sqrt{10}}, \end{aligned} \tag{2.13}$$

where d is the same as in the constellation figure. Recall that the euclidean distance d and the minimum inter-point distance $d_{min}^{(e)}$ are related as $d_{min}^{(e)} = 2d$, so the latter is given by

$$d_{min}^{(e)} = \frac{2}{\sqrt{10}}, \tag{2.14}$$

which we note is a smaller inter-point distance than for QPSK, as expected.

Summing up, we note that transmission with 16QAM modulation achieves a higher bit rate than with QPSK. However, the decrease in $d_{min}^{(e)}$ means that symbols modulated with 16QAM are also more likely to be erroneously determined at reception.

Pulse shaping

Assume we have used a certain modulation method (e.g. QPSK or 16QAM) to map bits into complex symbols s_k. Next, we generate the analog signal $s(t)$ by convolving the symbols with an analog pulse-shaping filter $p(t)$, as given in (2.6). This signal is transmitted over the baseband channel.

To avoid interference between consecutive symbols, only pulse-shaping filters that satisfies the Nyquist criterion are used [14].

With a clear model of the transformation from bit sequences via complex symbols to a transmittable analog baseband signal, we now return to the task of customising the system model in figure 2.1 to our needs.

A customised model for the digital transmission

The first adaption made was to use a complex baseband model for the real bandpass signals, a model that is now established. Next, we make the simplification that only digital input is assumed in this text, making the A/D and D/A converters superfluous.

Furthermore, we ignore the components for source encoding and decoding. In an real-world wireless communication system they are certainly implemented, as data compression is an important part of efficient transmission. However, including them here will not contribute to the understanding of MIMO systems and algorithms, so we choose to ignore them. We also disregard the channel encoding and decoding parts.

One aspect that needs to be included in the model is the choice of transmit algorithm, as there are many possibilities in MIMO systems. On the transmit side, they provide descriptions on how the information symbols are mapped to the transmit antennas, both in time and space. We refer to this component as the space-time (ST) encoder. On the receive side, the ST decoder reverses the encoding performed by the transmitter.

When all the simplifications and adaptations to the communication system model in figure 2.1 are considered, the result is the customised model in figure 2.4.

In short, what happens is that the input bit stream b_k is sent to the digital modulator, which forms a stream of complex symbols, s_k. The ST encoder maps the symbols to the N transmit antennas and transmits them over time, possibly with some space-time coding.

The receiver registers an incoming signal on each antenna, affected by the wireless channel and additive noise. The ST decoder uses knowledge of the channel to equalise the fading effects of the propagation, and decodes the matrix into a stream of approximated symbols $\hat{s}_k$. The digital demodulator slices these symbols to bits $\hat{b}_k$, after which the original and the approximated bit-sequences may be compared to count the number of errors.

The channel-component in the model is actually composed of several parts, consisting of both a transmit pulse shaping filter, a channel propagation coefficient and a time-reversed, receive filter, matched to the one on the transmit side. The total channel component will be denoted h, and it is presented in detail by section 2.1.2.

In this section, we have described two models of a digital communication system; one general and the other more adjusted to the topics of this text. We have introduced the concept of representing a real, bandpass signal by a complex, low-pass signal $s(t)$. We have also established

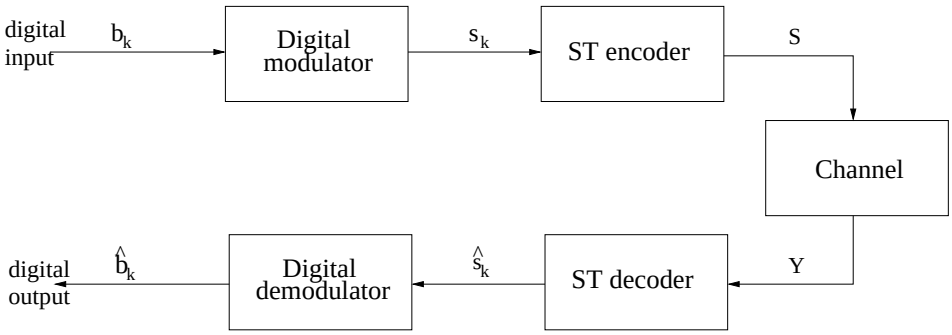

Figure 2.4: A simplified digital communication system

the relationship between this signal and the complex symbols s_k that are results of applying digital linear modulation methods to a stream of information bits.

From this background, we are ready to move on to study how the wireless channel and its characteristics affect the signal wave.

2.1.2 Signal propagation in multi-path environments

We assume that the analog signal $s(t)$, see (2.6) is transmitted over the baseband channel. The received signal $r(t)$ is given by the convolution of $s(t)$ with a channel propagation filter $h_p(t)$, and distorted by additive noise $\nu(t)$. All signals are represented in baseband. The noise is modelled as being added at the receive side, and $r(t)$ is expressed as [14]

$$r(t) = s(t) * h_p(t) + \nu(t), \tag{2.15}$$

In order to improve wireless transmission it is important to know what affects a signal on its way to the receiver. This makes the characteristics of the wireless propagation channel an interesting study.

The multi-path propagation environment

We characterise a propagation environment depending on the amount of obstructing bodies found in the area. These obstructions can be large objects like tall buildings and skyscrapers, or hills and mountains. Smaller obstructions such as street signs and vegetation will also influence the propagating wave, and must be taken into account.

Urban and suburban areas are assumed to have numerous and large objects (buildings), while flat, rural areas have few or no obstructions.

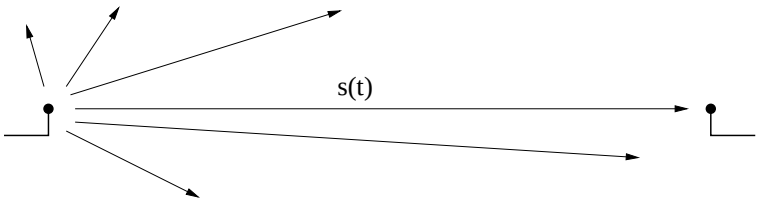

Figure 2.5: Propagation in free space (only LOS)

If an area naturally provides an unobstructed signal path between the transmitter and the receiver, it is called a LOS (Line-of-Sight) environment, usually found in the rural and flat areas. If there are no obstacles in the area at all, there is only one possible path from transmitter to receiver; the straight-lined LOS path. We refer to this as a free space environment. An illustration of propagation in free space is shown in figure 2.5.

Typically, free space propagation conditions are unachievable; flat areas without any large structures are rare. In fact, in urban and sub-urban surroundings, it is common to assume N-LOS (Non-Line-of-Sight) conditions, i.e. no line of sight path at all. Throughout this text, we assume that the signals propagate in N-LOS surroundings.

The large objects and structures in N-LOS environments cause changes in the direction of propagation of a signal wave. This can happen through reflection, when a propagating wave hits a very large object and is sent out in another direction. Another way to change the direction of wave is by diffraction, when the wave hits a large object and secondary waves are formed behind the object. A third effect of multi-path propagation is scattering, which occurs when the propagation medium contains a large number of objects smaller than the signal's wavelength, for example vegetation, clouds and street signs. These objects scatter the signal wave in all directions. The reflection, diffraction and scattering mechanisms are illustrated together in figure 2.6.

The N-LOS propagation effects of reflection, diffraction and scattering result in what we call *multi-path propagation*, and the mechanisms are collectively termed multi-path propagation mechanisms. The result of multi-path propagation is that the transmitted signal reaches the receiver on numerous paths, and from different directions. Multi-path propagation under N-LOS conditions is illustrated in figure 2.7.

In this section, we have described the propagation environment, in particular under N-LOS multi-path conditions. The next step is to study what is the result of propagation under such conditions.

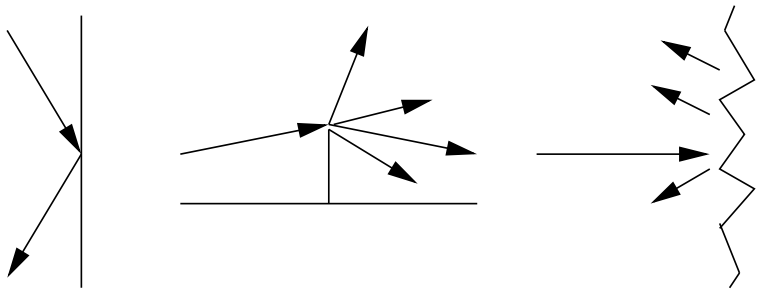

Figure 2.6: Three basic multi-path propagation mechanisms.

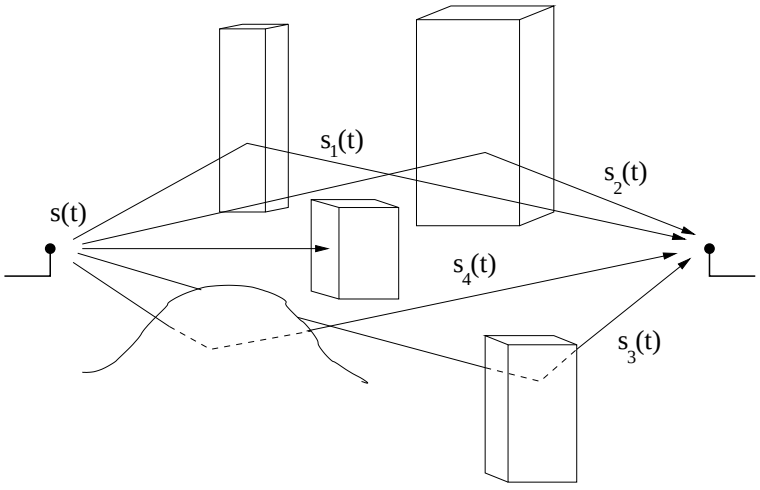

Figure 2.7: Propagation in multi-path environments (N-LOS)

Multi-path fading

A wave transmitted over wireless channels, in any kind of environment, has lost power when it reaches the receiver. In the pure LOS case, the received signal power depends only on the distance d from transmitter to receiver. This power loss is called attenuation or path loss.

However, in a multi-path environment, this free-space model can not explain all the effects the channel propagation is observed to have on a transmitted signal. In such environments, we have seen that all received signals have experienced reflection, diffraction or scattering on their way, as there is no direct path available. What is registered at the receiver is a sum of several versions of the same signal, coming in from different directions.

We assume a finite number, L, of possible paths the signal can reach the receiver over. In figure 2.7, we have that $L = 4$. With sufficient multi-path it is likely, and will be assumed, that the various paths experience independent attenuation. Also, the paths are naturally of different lengths, so the different multi-path components of the transmitted signal wave will not arrive simultaneously. Hence, a pulse transmitted in a multi-path environment is received as a train of individually delayed and attenuated pulses [14].

We recall the transmission model given in (2.15), where $h_p(t)$ is the time-variant impulse response of the channel. In our assumed case of multi-path propagation we give this channel propagation coefficient as

$$h_p(t) = \sum_{l=1}^{L} \alpha_l(t) e^{j\phi_l(t)} \delta[t - \tau_l(t)]. \tag{2.16}$$

Here, $\alpha_l(t)$ and $\phi_l(t)$ are the attenuation factor and phase rotation induced by path l, respectively. The value $\tau_l(t)$ represents the propagation delay of path l. All three are represented as time-dependent, due to changes in the structure of the medium.

Mainly due to the time-variances in the phases $\phi_l(t)$, we experience a variation in the received signal's amplitude. We refer to this effect as small-scale or signal fading, and say we have a fading channel.

Small-scale fading can lead to to dramatic changes in signal amplitude and phase by changing the distance between transmitter and receiver by as little as half a λ_{f_c}, where λ_{f_c} is the wave-length of the carrier wave [18].

The small-scale fading manifests itself in two effects; time-spreading of the transmitted signal and a time-variant behaviour of the channel due to the relative motion of the transmit and receive antennas [18].

Delay spread

The first effect is a result of the individual path delays $\tau_l(t)$ not being equal. The maximum excess delay T_m, is defined to be the time between the arrival of the first and the last component of a transmitted signal.

It is clear that if the maximum excess delay is larger the than symbol period of (2.6), i.e. if $T_m > T_s$, consecutively transmitted symbols are summed together and interfere at the receiver. This destructive mechanism is called inter-symbol interference (ISI). In the frequency domain, ISI results in frequency selectivity, the fact that different frequencies in the transmitted signal are subject to different attenuation and phase shifting.

Throughout this text and in the results from the simulations, we assume the multi-path channel to be free of ISI. In the frequency domain, we say the channel is frequency non-selective or flat fading. The assumption of flat fading reduces (2.15) to

$$r(t) = h_p(t)s(t) + \nu(t), \tag{2.17}$$

In practise, ISI is very frequent in urban environments, but it is still common to model the channel without it.

This choice facilitates the analysis and is also justified by the following:

- Short-range wireless communication (such as in a wireless LAN) is a hot topic these days, and over short distances ISI may be avoided because the travelling distance is very short compared to current symbol periods
- Multi-carrier systems, such as the OFDM (Orthogonal Frequency Division Multiplexing), obtain ISI-free channels by transmitting independent data on multiple carriers, each satisfying a narrow-band criterion. In this case we may apply transmit algorithms on each sub-carrier independently.

Doppler spread

Fading was presented above as a time-variant behaviour of the channel. It is caused by a non-zero relative motion of the transmit and receive antennas, or motion of the structures that causes the multi-path propagation. These movements are expected to make the values of (2.16) change over time. In practise, this includes changes in the number of paths, along with the attenuation and the phase rotation experienced on the individual paths. This relative motion is what causes the so-called Doppler spread effect.

The coherence time, $(\Delta t)_c$, of the channel is the period for which the channel may be assumed to be constant. We wish to transmit a certain number of symbols. The time it takes to transmit this symbol block is denoted T_{block}. If $(\Delta t)_c$ and T_{block} are related so that $T_{block} \ll (\Delta t)_c$ the channel attenuation and phase shift are essentially constant during the transmission of the block of symbols. In this case, we say the channel is slow fading [14], and (2.17) is further simplified to

$$s(t) = h_p s(t) + \nu(t), \quad \text{for } t \in [0, T_{block}] \tag{2.18}$$

for which time the channel propagation coefficient h_p may be considered time-invariant and given as

$$h_p = \sum_{l=1}^{L} \alpha_l e^{j\phi_l}. \tag{2.19}$$

A slowly changing channel allows us to model the it as quasi-static, constant for during a period T_{block}. After this time, h_p is assumed to change in a burst, to a new and independent value. In all, such a channel is called *bursty and quasi-static*.

The quasi-static model is appropriate because perfect channel knowledge is demanded for all the later described transmit algorithms. Also, none of the algorithms that will be tested depend on any preceding channel values. Thus, the smoothness of the change in the channel coefficient is irrelevant to the transmit schemes.

When there are a large number of paths, the structures in the environment are randomly placed and no LOS-path is available, we say that there is a lot of multi-path in the channel. When assuming the paths to experience independent fading, the central limit theorem tells us that the impulse response of such a channel may be modelled as a complex, Gaussian random process [14].

When h_p is a zero-mean, complex Gaussian process, its envelope follows a Rayleigh distribution and we refer to the channel as *Rayleigh fading*. With a dominant non-fading component, such as a LOS component, the envelope of $h_p(t)$ can be described by a Ricean probability density function. Rayleigh fading channel coefficients will be assumed throughout this thesis.

2.1.3 A SISO signal model.

As a seen in the previous section, the propagation channel is now modelled as flat fading and quasi-static. Next, we develop this to a full model of how a signal is affected from transmitter to receiver. Although the main focus in this text is on MIMO systems, we allow for a slow start

by first developing the model for a Single-Input Single-Output (SISO) system.

We have seen that the N-LOS multi-path conditions affects the transmitted signal as a complex-valued, multiplicative factor of attenuation and fading, denoted h_p. Another distorting effect is that of noise $\nu(t)$, which may arise from imperfections in the electrical components or as interference noise from the channel.

For convenience, we repeat the model developed in last section, in (2.18). The received signal $r(t)$ is

$$r(t) = h_p s(t) + \nu(t). \tag{2.20}$$

in which an information signal $s(t)$ is transmitted over the baseband channel h_p with additive noise represented by $\nu(t)$.

We recall the relation between the modulated complex symbols s_k and the analog signal $s(t)$ from (2.6). At the receiver the signal $r(t)$ is filtered through a time-reversed pulse-shaping filter $p(-t)$, matched to the one applied before transmission. This is done in order to maximise the energy at sampling intervals of T_s, the symbol period.

$$\begin{aligned} y(t) &= r(t) * p(-t) \\ &= \big(h_p s(t) + \nu(t)\big) * p(-t) \\ &= h_p s(t) * p(-t) + \nu'(t) \end{aligned} \tag{2.21}$$

By substituting (2.6) for the signal $s(t)$, we get

$$y(t) = h_p\Big(\sum_{k=-\infty}^{\infty} s_k p(t - kT_s)\Big) * p(-t) + \nu'(t) \tag{2.22}$$

We define the effective channel h, which takes into account both the transmit and receive pulse shaping filters and the channel propagation coefficient h_p, such that

$$h(t) = p(t) * h_p * p(-t), \tag{2.23}$$

which means that (2.21) may be rewritten as

$$y(t) = \sum_{k=-\infty}^{\infty} s_k h(t - kT_s) + \nu'(t) \tag{2.24}$$

When $y(t)$ is sampled once every symbol period T_s, the filtered received signal can be expressed as

$$y(n) = \sum_{k=-\infty}^{\infty} s_k h(n - k) + \nu'(n) \tag{2.25}$$

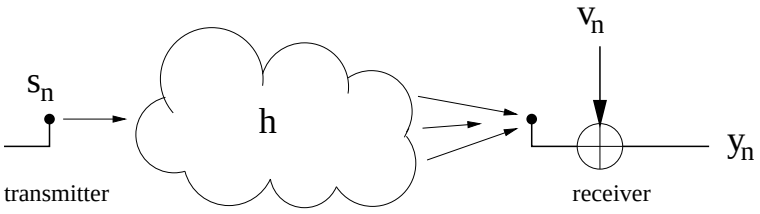

Figure 2.8: A wireless model.

Setting $T_s = 1$ does not loose generality. For a flat fading channel, we have that $h(n-k) = h\delta(n-k)$, so (2.25) is further simplified to

$$y(n) = hs_n + v'(n), \tag{2.26}$$

or equivalently, only more clearly expressed as complex symbols

$$y_n = hs_n + v_n \tag{2.27}$$

Given our quasi-static channel model, we may also build a vector version of (2.27). Let us assume the channel may be considered constant for the duration of K symbol periods. We collect K information symbols in the $1 \times K$ vector $\mathbf{s}$, and independent noise values in $\mathbf{v}$, of equal size. The received symbols are placed in the vector $\mathbf{y}$, also with dimensions $1 \times K$, and we have

$$\mathbf{y} = h\mathbf{s} + \mathbf{v}. \tag{2.28}$$

The transmission models of (2.27) and (2.28) is employed throughout the text. A graphical view of a SISO system that fits the above model is shown in figure 2.8, with one transmit and one receive antenna.

The channel is represented as a cloud, and is the same channel as the one depicted in figure 2.4. The complex-valued s_n, h, y_n and v_n represents the complex symbol we wish to transmit, a channel coefficient, the noise and a received noisy symbol, respectively.

In section 2.1.1, the symbol s_n was defined in the discussion on digital, linear modulation. Now, we also take a brief look at how the noise symbol v_n and channel coefficient h are modelled.

Distributions of noise sample and channel

The noise sample v_n is modelled as complex-valued random, following a normal distribution. We may express v_n as

$$v_n = (u + w \cdot j) \tag{2.29}$$

where u and w are real and normally distributed numbers; $(u, w) \sim \mathcal{N}(0, \sigma_\nu/\sqrt{2})$. Here, σ_ν is the standard deviation of the complex noise, so that

$$\nu_n \sim C\mathcal{N}(0, \sigma_\nu) \tag{2.30}$$

The variances of the noise and the signal coefficients form the relation known as the signal-to-noise ratio (SNR). In linear scale it is expressed as

$$SNR_{lin} = \frac{E(|s_n|^2)}{E(|\nu_n|^2)} = \frac{\sigma_s^2}{\sigma_\nu^2}, \tag{2.31}$$

We fix the variance of the information symbols to unity, so $\sigma_s^2 = 1$. The SNR is also often expressed in dB, as

$$SNR_{dB} = 10\log_{10}\left(\frac{1}{\sigma_\nu^2}\right), \tag{2.32}$$

and σ_ν^2 is given by

$$\sigma_\nu^2 = 10^{-\left(\frac{SNR_{dB}}{10}\right)}. \tag{2.33}$$

The channel coefficient is assumed to have sufficient multi-path to be modelled as a zero-mean Gaussian, see section 2.1.2. Each random realisation of h follows

$$h \sim C\mathcal{N}(0, 1), \tag{2.34}$$

that is; h is complex, with zero mean μ and unit variance σ^2. We recall that this is also referred to as a Rayleigh fading channel, because the envelope of h follows a Rayleigh distribution. The channel real and imaginary parts of h are modelled as $(u, w) \sim \mathcal{N}(0, 1/\sqrt{2})$.

So far, we have established a suitable signal model for transmission over a baseband SISO channel. The next step is to extend this to MIMO systems, yielding the final multiple antenna transmission model.

2.2 Multiple antenna systems (SIMO, MISO and MIMO)

This section is devoted to introducing and building a signal model for multiple antenna systems. The work will build on the model for the a SISO system, from (2.20). We begin with a short section to motivate the use of multiple antennas, then section 2.2.2 presents SIMO, MISO and MIMO systems. Finally, the MIMO signal model is given in section 2.2.3.

2.2.1 Motivations for MIMO

We present two major motivations for the introduction of multiple antenna systems; to increase the quality (lower bit-error rate (BER)) and to increase the data rate of a communication process.

Starting with the increased quality, we note that if the channel of a SISO system is in a fading dip, the original information is hard to recover at the receiver, which makes the transmission less reliable.

Now, consider having several transmit antennas, each with an independent channel coefficient. With knowledge of the coefficients, we may ensure a steady level of tolerable quality by transmitting on the channel with the highest field strength at any time. This approach is called switched or selection diversity and exploits the fact that it is unlikely that all the channels (antennas) are in fading dips at the same time. It will help decrease the BER, as long as the channel coefficients are not all equal.

The method of selection diversity, and is only one of several possible BER-reducing diversity schemes for MIMO systems.

Our second motivation concerns increasing the data rate, as more capacity is always in question. The concept of multiple antennas opens the possibility of exploiting the spatial dimension. With more than one antenna on the sending and/or the receiving side, we obtain spatial diversity. This diversity can be used to reduce the BER, as described above, but also to send data faster through higher order modulation. In MIMO systems, the data rate may also be directly increased by transmitting independent data streams on different antennas.

Before going any further, we define the three different types of gain the use of multiple antennas and appropriate transmit algorithms may produce. These are array gain, diversity gain and multiplexing gain.

Array gain is the improvement we get from coherent combining of signals arriving on the separate elements of an antenna array. This gain is seen in increased signal power and quality, i.e. the SNR of the combined array is better than that of the individual elements.

Diversity gain is the result of transmitting or receiving over more than one channel, but relies on the channels being sufficiently independent of each other, so that all are not severely faded at the same time. Transmission over independent channels offers diversity gain in the form of fading mitigation, the result of which is seen in a lowered BER.

Multiplexing gain is simply the gain of increasing the data rate, as done in spatial multiplexing schemes, where each antenna transmits a separate sub-stream of data.

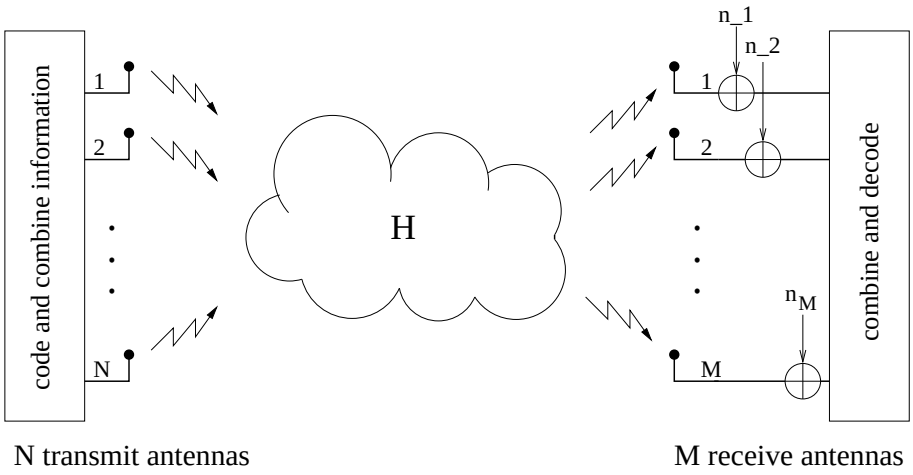

Figure 2.9: A general MIMO model.

Having presented the simple motivations above, we move on to describe different multi-antenna systems in the next section.

2.2.2 SIMO, MISO and MIMO systems

We define three cases of systems with multiple antennas; Single-Input Multiple-Output (SIMO), Multiple-Input Single-Output (MISO) and Multiple-Input Multiple-Output (MIMO). They are named after where the multiple antennas are placed; on the sender side, the receiver side or both places.

A SIMO system has 1 transmit antenna and M receive antennas, while a MISO system has N transmit and 1 receive antennas. A MIMO system is a generalisation of the two, with N transmit and M receive antennas. An important measure is the *order of diversity*, defined here as the number of independently fading propagation paths.

With several antennas, there are a number of possible geometries to arrange the antennas in. The elements may be distributed linearly, in a circle or a square pattern, just to give some examples. In our simulations, the simple uniform linear array (ULA) is assumed, that is; the elements lie on a straight line with a common inter-element distance d [21]. However, we note that the MIMO algorithms are transparent to the choice of array geometry.

In figure 2.9 a MIMO system is shown, and can be seen as a generalisation of the SISO figure in figure 2.8. The system has N transmit and M receive antennas and the multi-path propagation environment is represented as a cloud. The additive noise is independent at each of the receive antennas.

A SIMO, MISO or MIMO system with knowledge of the channel can adapt to the current channel conditions in order to optimise transmission and reception of information signals. Depending on the system and the choice of transmit algorithm, the channel information is used to exploit the spatial dimension to achieve different forms of gain, for example a decrease in the BER or a higher transmit data rate.

2.2.3 A MIMO signal model

We are now ready to extend the channel model in equation (2.28) to the general MIMO case.

In a MIMO system, $M \cdot N$ flat fading and independent channel coefficients are collected in an $M \times N$ channel matrix $\mathbf{H}$. Each entry h_{mn} follows the same Gaussian distribution as h in (2.34).

An entry h_{mn} denotes the multi-path channel from transmit antenna n to receive antenna m. The channels are assumed to be independent in space (uncorrelated), constant over a block of symbols and independent from block to block. This gives the so-called independent, identically distributed (i.i.d.) block fading Rayleigh matrix model.

Later, spatially correlated channels will also be considered, but for now, they are spatially independent.

By sending out symbols on all N transmit antennas at a given time, we can exploit the spatial diversity these multiple antennas provide. We build a general MIMO signal model, by defining the $N \times K$ sized complex symbol matrix $\mathbf{S}$. K is the number symbol periods T_s we consider at a time. For simplicity, we set $K = 1$ whenever possible. A symbol s_{nk} in $\mathbf{S}$ is transmitted from antenna n in time slot k.

For each transmission, the M receive antennas take in M symbols and add independent noise to each. These random complex noise symbols are arranged in the $M \times K$ matrix $\mathbf{V}$. Each noise entry v_{mk} is a random complex Gaussian, distributed as described in (2.30).

The receive matrix $\mathbf{Y}$ has dimensions $M \times K$. Now, an MIMO-version of the signal model of (2.20) is

$$\mathbf{Y} = \mathbf{HS} + \mathbf{V}, \tag{2.35}$$

We note that if $K = 1$, the above equation is simplified to a vector form, so that

$$\mathbf{y} = \mathbf{Hs} + \mathbf{v}, \tag{2.36}$$

where $\mathbf{y}$ and $\mathbf{v}$ are vectors of dimension $M \times 1$, and $\mathbf{s}$ has the size $N \times 1$. This form will be used in some of the transmit algorithms in chapter 3.

The channel matrix **H** in (2.36) is so far assumed to contain all independent fading coefficients. But in practise, proximity naturally implies similarity, and we present a model for correlated fading among the antennas.

2.2.4 A model for channel correlation

Spatial independence of the propagation channels is important for many schemes. This dependence makes it interesting to investigate how these algorithms behave when spatially correlated channels are introduced, thereby degrading the spatial independence. For simplicity in the correlation model, we assume a uniform linear antenna array geometry, but this choice is not important to the MIMO algorithms.

Finding a good model for the correlation is done by studying the normalised signal correlation coefficient between antenna elements i and j, r_{ij}. This value shows in what degree the fading coefficients of antennas i and j are correlated. With $r_{ij} = 1$, they are completely correlated, i.e. their corresponding fading coefficients are equal. Letting $r_{ij} < 1$ implies greater independence, and for $r_{ij} = 0$, the fading levels of the two antennas are completely independent.

From [11] and [10], the r_{ij} for a uniform linear array is given by

$$r_{ij} = \frac{1}{2\Delta}\int_{\phi-\Delta}^{\phi-\Delta} e^{jz(i-k)\sin\beta} d\beta, \quad z = 2\pi\frac{d}{\lambda}, \tag{2.37}$$

where 2Δ is the angle spread of the incoming multi-paths, ϕ is the average angle of arrival, d is the distance between the antenna array elements and λ is the carrier wavelength. It is possible to assume $\lambda = 1$, without loosing generality [11]. When $\Delta = \pi$, equation (2.37) is reduced to

$$\begin{aligned} r_{ij} =& J_0(z(i-j)) \\ =& J_0(\frac{2\pi(i-j)d}{\lambda}), \end{aligned} \tag{2.38}$$

where J_0 is the zero-order Bessel function of the first kind. Bessel functions are solutions to a differential equation called *Bessel's equation*. A plot of a few Bessel functions of the first kind, but of different order, are shown in figure 2.10.

From the figure we see that the odd-ordered Bessel functions are symmetric about $z = 0$, so that

$$r_{ij} = r_{ji}.$$

From the figure and (2.38) we are also able to verify that the model is correct when $i = j$; any antenna element is fully correlated with itself, $r_{ii} = J_0(z(i-i)) = J_0(0) = 1$. We also see that

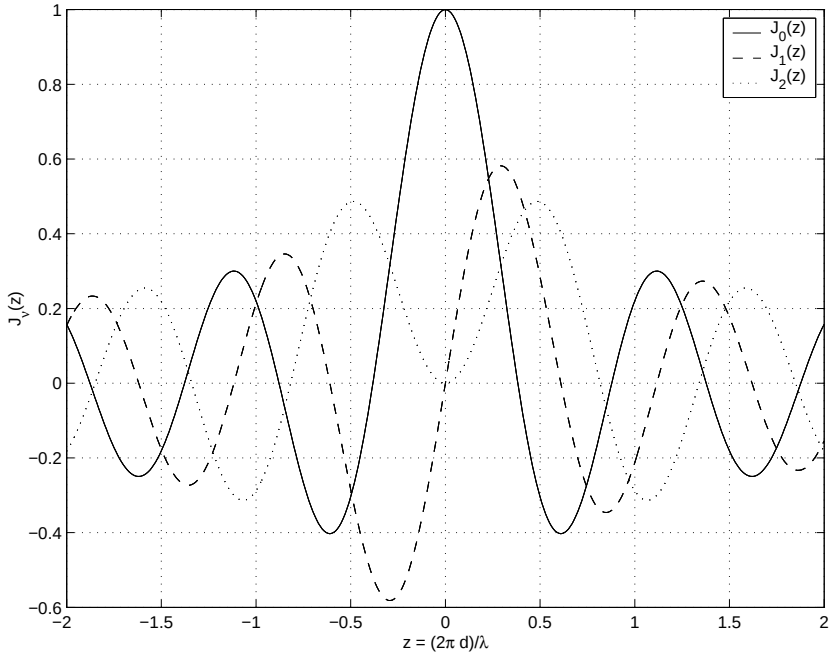

Figure 2.10: Bessel function of the first kind

$$r_{ij} = r_{kl}, \text{ when } |i-j| = |k-l|$$

i.e. the correlation between two antennas is only dependent on the distance between them, resulting in a symmetric correlation matrix. For later use, we denote the correlation between two neighbouring antenna elements by r, so that

$$r = J_0(z), \quad z = 2\pi\frac{d}{\lambda} \tag{2.39}$$

which is the case when $|i-j| = 1$

Another illustration is figure 2.11, where a zero-order Bessel function is plotted as a function of wave-lengths λ. The inter-element distance d is measured in meters.

As an example, let us look at the level of correlated fading for a few different distances d:

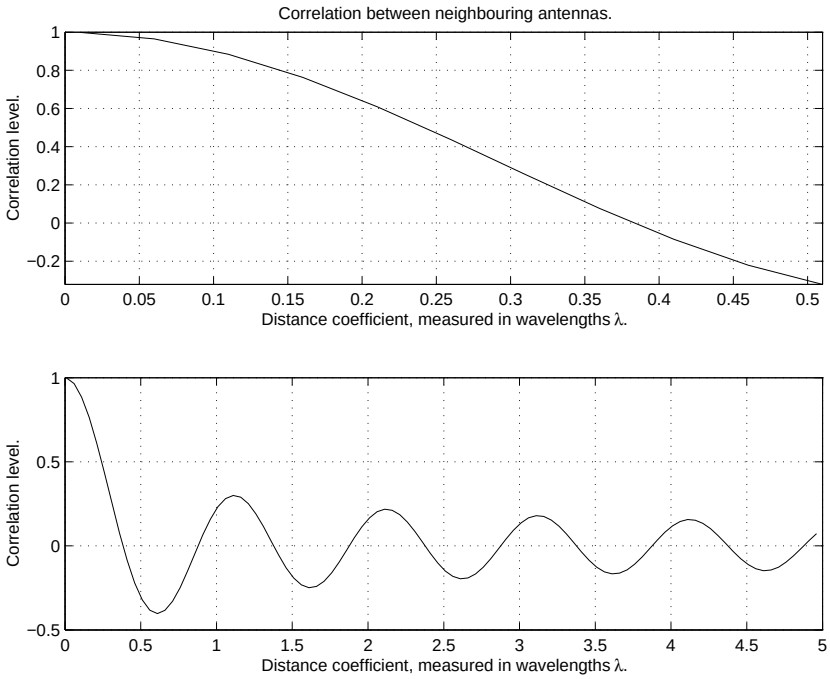

Figure 2.11: Correlation between neighbouring antennas, $\lambda = 0.15$ m

$$\begin{aligned} d &= 10\lambda \implies r = 7.10 \cdot 10^{-2} \\ d &= 1\lambda \implies r = 2.20 \cdot 10^{-1} \\ d &= 0.3\lambda \implies r = 2.91 \cdot 10^{-1} \\ d &= 0.2\lambda \implies r = 6.43 \cdot 10^{-1} \\ d &= 0.1\lambda \implies r = 9.04 \cdot 10^{-1} \end{aligned} \tag{2.40}$$

We see that when varying the inter-element distance from 10λ to 0.1λ, the correlation changes from a negligible level to almost 1. For $d > 0.3\lambda$, the correlation r is not so severe, one may still expect some diversity in the channel. It is also evident that the increase in r is quite slow from $d = 10\lambda$ to $d = 0.3\lambda$. This may be confirmed by the figure, 2.11.

Given the above discussion, we are able to build individual correlation matrices for the transmit and receive antennas. It is worth noting that the following simulations and results all employ only equi-distant antenna arrays, with the same distance on both sides of the transmission, although varying the inter-element distances is also possible.

Correlated MIMO model

The correlation matrices are called $\mathbf{R}_t$ on the transmit side and $\mathbf{R}_r$ on the receive side. The entry in the i-th row and j-th column corresponds to r_{ij} in (2.38), the correlation coefficient between antenna elements i and j.

The Bessel functions are deterministic, so if $N = M$ and d is the same on both sides of the wireless channel, then we also have that $\mathbf{R}_t = \mathbf{R}_r$. This text assumes d to be equal in both transmit and receive antenna arrays, so the neighbouring element correlation level r is always the same on both sides. An example of a transmit correlation matrix, $\mathbf{R}_t$, for a system of 3 transmit antennas, is

$$\mathbf{R}_t = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}.$$

Using the fact that $r_{ij} = r_{ji} = r_{|i-j|}$, and that the elements on the main diagonal all have the value 1, we may simplify the matrix to

$$\mathbf{R}_t = \begin{bmatrix} 1 & r_1 & r_2 \\ r_1 & 1 & r_1 \\ r_2 & r_1 & 1 \end{bmatrix}.$$

When building the correlated channel matrix $\mathbf{H}$, an uncorrelated Rayleigh fading channel matrix $\mathbf{H}_0$ is used, from which $\mathbf{H}$ is computed as follows [7]

$$\mathbf{H} = \sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}, \tag{2.41}$$

where the square root sign implies a matrix square root operation, so that $\sqrt{\mathbf{R}_t} \cdot \sqrt{\mathbf{R}_t} = \mathbf{R}_t$ and $\sqrt{\mathbf{R}_r} \cdot \sqrt{\mathbf{R}_r} = \mathbf{R}_r$.

Recall that the Bessel function is unity when the input variable is zero; $J_0(0) = 1$. Hence, if d is close to zero, then all the entries in the correlation matrix will be close to 1. The result is that the coefficients in $\mathbf{H}$ are approximately equal and rank($\mathbf{H}$) = 1, which means that the channel matrix is not invertible. This $\mathbf{H}$ is sufficiently ill-conditioned to consist of only one effective channel, not a set of NM independent ones.

On the other hand, a low level of correlation is obtainable when the antenna spacing d is kept above a certain level. This is a requirement that can be hard to meet, e.g. if we wish to place more than one antenna on a small unit, such as a cell-phone.

As seen from (2.38), the correlation depends both on the inter-element spacing of the antenna array and on the carrier wavelength of the signal wave. In this text we use $\lambda = 0.15$ meters, corresponding to a carrier frequency

$$f_c = \frac{c}{\lambda} = \frac{3.0 \cdot 10^8 \, m/s}{1.5 \cdot 10^{-1} \, m} = \; 2.0 \, GHz, \tag{2.42}$$

which is used for the third generation of mobile communication systems, also known as 3G.

We now move on to presenting transmit algorithms for MIMO systems using the model in (2.35), possibly with correlated fading. The next chapter presents known schemes and evaluates their performance under different conditions.

Chapter 3

Performance of MIMO algorithms in correlated channels

This chapter presents some well-known techniques for wireless transmission in MIMO systems. Each has a set of advantages and limitations, and which algorithm is best may depend on the channel conditions. In section 3.2, the performance of the various transmit algorithms are compared, for both uncorrelated and correlated channels.

3.1 MIMO schemes

SIMO, MISO and MIMO systems can be used with a variety of transmit algorithms, providing improvement in different forms. The algorithms essentially exploit the spatial diversity offered by MIMO systems in one of two ways: By reducing the number of errors through diversity-oriented transmission, or by increasing the data rate through multiplexing of independent symbol streams. We refer to these different approaches as MIMO diversity (MD) and spatial multiplexing (SM) [12], respectively.

Some MD schemes, aimed at reducing the bit-error rate (BER), are

- Switched or selection diversity
  For SIMO/MISO systems, see section 2.2
- Maximum Ratio Combining (MRC)
  For SIMO/MISO systems, see section 3.1.2
- The maximum singular vector approach (MSVA)
  For MIMO systems, see section 3.1.2

- Space-time coding schemes (e.g. Alamouti STC)
  For $2 \times M$ systems, see section 3.1.4

An SM scheme, which maximises the data rate, is

- Spatial multiplexing (SM) with zero-forcing
  For MIMO systems, see section 3.1.5.

As mentioned in the introduction, chapter 1, the SM and MD schemes, have so far been competing approaches to exploit the spatial dimension offered by MIMO systems. Spatial multiplexing uses the spatial degrees of freedom to transmit information faster, by sending independent symbol streams simultaneously.

MD algorithms, on the other hand, use the multiple antennas to mitigate channel fading, effectively lowering the BER. This is done through transmission of redundant information, for example by using space-time coding (STC) techniques[20].

However, despite different approaches, the ultimate goal of both approaches is increased capacity. With MD, this is achieved through increasing the order of the modulation when the BER goes down, while maintaining a certain target BER. In this way, the data rate is increased by transmitting more bits with a single symbol. SM schemes increase the capacity more directly, through multiplexing independent symbol streams on the transmit antennas.

3.1.1 Channel knowledge

All the above algorithms require some knowledge about the channel matrix $\mathbf{H}$, on either one or both sides. One way to distinguish between the algorithms is to study if and where this knowledge is required. As we will see later in this chapter, the algorithms presented are different in this respect, demanding knowledge either on the transmit side (transmit MRC), the receive side (receive MRC, SM and Alamouti STC) or on both sides (The MSVA approach).

In a cell phone system, it is common to see the base station as the transmitter and the personal cell phone as the receiver. The receiver constantly accepts incoming data, containing both the desired information symbols, the channel coefficients and the additive noise, as in equation (2.35).

By use of some form of training sequence, i.e. transmitting data known to the receiver a priori, the receiver can extract the channel coefficients, only distorted by the noise. The channel matrix estimated from these training data is denoted $\hat{\mathbf{H}}$. This estimated channel matrix may then be used to retrieve information symbols later in the transmission.

Depending on how fast the channel coefficients change, the training sequence must be retransmitted at intervals, to update $\hat{\mathbf{H}}$.

When only the receiver needs knowledge of the channel, the use of training sequences is a sensible way for it to acquire that knowledge. Because these training sequences do not need to be very long, the overhead is limited.

The transmitter has no natural way to learn about the channel, unless if the receiver sends feedback data, to let it know what it has learnt itself. This introduces overhead and increased complexity in the transmitter.

For the sake of the derivation of the algorithms we assume to have perfect channel knowledge.

In the following sections, we present some multiple antennas transmit schemes; namely MRC, MSVA, Alamouti STC and SM. The MRC is not suitable for MIMO systems, but is presented to explain the concept of array gain and because the MSVA can be thought of as a MIMO extension/generalisation of this scheme.

3.1.2 Maximum Ratio Combining

Maximum Ratio Combining (MRC) [14] is an optimal combining technique for both transmit and receive diversity. The scheme uses weighting and combining of the information symbols in order to increase the link reliability, i.e. to reduce the number of errors between original and received information symbols. The weights are chosen adaptively, using channel knowledge.

In a $1 \times M$ SIMO system, the transmitted signal reaches the M receive antennas as M independently faded signals, at which point they are weighted and combined. With a $N \times 1$ MISO system, the MRC technique is employed before transmission. N copies of a signal are weighted with different coefficients, and transmitted from the antennas. The single receive antenna registers a sum of independently faded and weighted versions of the same signal.

Both with receive and transmit diversity, we see that the same choice of weight vector is optimal, selected to mitigate the fading effects of the channel. This weight vector is optimal in the sense that it equalises the phase rotations induced by the channels.

Depending on which side of the communication channel the multiple antennas are placed on, either the transmitter or the receiver that is set to handle the computational complexity of the MRC scheme. Because the weights are chosen adaptively, using knowledge of the channel, this side is also required to know the true channel coefficients or an estimate of them.

MRC for MIMO?
The MRC scheme can be used with SIMO or MISO systems; offering receive or transmit diversity. For MIMO systems, pure MRC is not achievable. This is because the phase rotations induced by a MIMO channel matrix of $N \cdot M$ coefficients can not be completely mitigated by applying weight vectors containing $(M + N)$ coefficients at the transmit and the receive sides. A MIMO approximation to MRC, henceforth referred to as The Maximum Singular Vector Approach (MSVA), will be defined later.

A signal model for transmit and receive MRC

Based on the model presented in (2.36), this section presents signal models for both receive and transmit MRC.

Starting with receive MRC, we have 1 transmit antenna and M receive antennas, offering receive diversity. The channels coefficients are collected in a $M \times 1$ sized vector, $\mathbf{h}$, each entry representing one independent channel. Channel knowledge is required at the receiver.

We consider only one symbol period, over which one symbol s is transmitted. The noise vector $\mathbf{v}$ is $M \times 1$, and the received symbols are placed in $\mathbf{y}$, of dimensions $M \times 1$.

We weigh and combine all the received symbols, in such a way as to mitigate fading and increase the likelihood of recovering the correct symbol. We call the weight vector $\mathbf{w}$, with dimensions $M \times 1$. After combining, the received symbol $\hat{s}$is

$$\hat{s} = \mathbf{w}^T\mathbf{y} = \mathbf{w}^T\mathbf{h}s + \mathbf{w}^T\mathbf{v}, \tag{3.1}$$

The MRC uses the inverted and normalised channel as its weight vector,

$$\mathbf{w} = \mathbf{h}^* / \|\mathbf{h}\|_2, \quad \text{ensuring that} \quad \|\mathbf{w}\|_2 = 1 \tag{3.2}$$

For receive MRC, $\|\mathbf{h}\|_2 = \sqrt{\mathbf{h}^H\mathbf{h}}$, the length of the complex vector $\mathbf{h}$. We note that the phase rotations caused by the channel coefficients are cancelled out and the recovered symbol $\hat{s}$ can be written as

$$\hat{s} = \frac{\mathbf{h}^H}{\|\mathbf{h}\|_2}\mathbf{y} = \|\mathbf{h}\|_2 s + \frac{\mathbf{h}^H}{\|\mathbf{h}\|_2}\mathbf{v}, \tag{3.3}$$

At the end of this section, we will see that the factor $\|\mathbf{h}\|_2$ in front of s yields the improvement we refer to as array gain.

In transmit MRC, the weights are applied before transmission, and we send an $N \times 1$-sized vector of pre-weighted versions of the symbol s over the channel. We denote this vector $\mathbf{s}_w$, and the relation is

$$\mathbf{s}_w = \mathbf{w}^T s. \tag{3.4}$$

where the weight vector distributes the limited transmit power over the N transmit antennas, fixed to unity by setting $\|w\|_2 = 1$.

In MISO systems, the propagation channels are represented in a channel vector $\mathbf{h}$, of size $1 \times N$. There is no processing at the receiver with transmit diversity, so the recovered information symbol is given as

$$\hat{s} = y = \mathbf{h}\mathbf{s}_w + \nu = \mathbf{h}\mathbf{w}^T s + \nu. \tag{3.5}$$

where both y, $\hat{s}$ and the noise ν are scalars.

The weight vector in transmit MRC is chosen in the same way as for receive MRC, and (3.5) is simplified to

$$\hat{s} = \|\mathbf{h}\|_2 s + n, \tag{3.6}$$

now with $\|\mathbf{h}\|_2 = \sqrt{\mathbf{h}\mathbf{h}^H}$, in order to produce a scalar.

We see that this result is the same as for the receive diversity case, except from a difference in the noise-term. However, given that both the channel coefficients and the noise are modelled as independent zero-mean processes, the expectation of $\hat{s}$ is the same in (3.3) and (3.6).

For a certain target BER, the SNR of the combined array when using MRC in a SIMO system is M times better than the SNR-levels of the separate channels, in linear scale. For MISO systems, the order of improvement is N. The advantage is that the output has an acceptable SNR even if none of the individual channels are acceptable on their own [15].

In figure 3.1 a system with receive diversity is depicted, including the estimation of the MRC weights. A mirrored illustration for the case of transmit diversity is easily derived.

Array gain with MRC

The signal-to-noise ratio of a MISO or SIMO system with MRC is better than that of a single-channel system, an improvement we denote array gain. First, assume a symbol s is transmitted over the single channel h. The received, noisy signal is expressed as

$$y = hs + \nu \tag{3.7}$$

We recall that the channel follows the distribution in (2.34). The SNR is the variance of the signal part divided by the variance of the noise. Here the signal part is hs. We refer to this single-channel SNR as input SNR SNR_i.

$$\text{SNR}_i = \frac{E(|hs|^2)}{E(|\nu|^2)} = \frac{E(|s|^2)}{E(|\nu|^2)} \tag{3.8}$$

Now, in a MISO system, the symbol s is weighted and transmitted from N antennas simultaneously. All the channels in $\mathbf{h}$ are assumed to

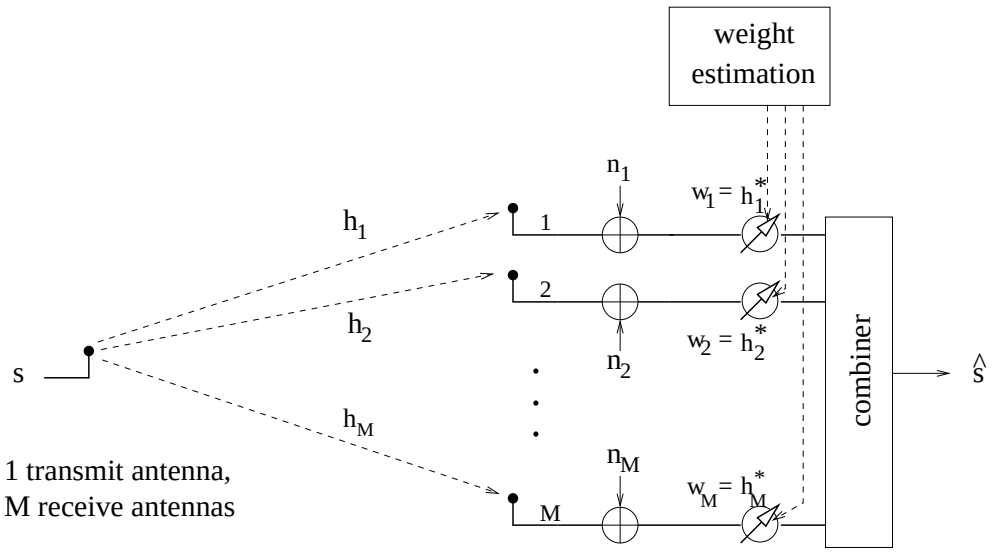

Figure 3.1: SIMO system with MRC

have the same SNR-level, SNR_i. The received signal in a MISO system is presented in (3.6), and the signal part is $\|\mathbf{h}\|_2 s$. We refer to the SNR of the combined signal at the receiver as the output SNR; SNR_o.

$$\begin{aligned}\text{SNR}_o &= \frac{E(|(\|\mathbf{h}\|_2)s|^2)}{E(|\nu|^2)} = \frac{E(|\mathbf{h}|^2|s|^2)}{E(|\nu|^2)} \\ &= N\frac{E(|s|^2)}{E(|\nu|^2)}\end{aligned} \tag{3.9}$$

The SNR of the MISO system with MRC is improved by a scale factor N, when compared to the SNR of a single channel, and this is what we refer to as array gain.

$$\text{SNR}_o = N\text{SNR}_i \tag{3.10}$$

In dB-scale, this means that

$$\text{SNR}_{\text{o,dB}} = 10\log_{10}(N)\ \text{SNR}_{\text{i,dB}} \tag{3.11}$$

This result also applies when receive diversity with MRC is employed, it is easily shown that array gain of the order M is obtained.

3.1.3 MRC for MIMO: The Maximum Singular Vector Approach (MSVA)

For the general case of multiple antennas on both sides, we have a MIMO system with an $M \times N$-sized channel matrix $\mathbf{H}$. We want to generalize the transmit- and receive-only diversity schemes of section 3.1.2 to a combined transmit/receive diversity scheme.

There is no way to use pure Maximum Ratio Combining for this case, as two weight arrays with a total of $N + M$ coefficients can not equalise the phase rotations induced by all the MN channel coefficients.

Instead, we will define an extension here, through use of a singular value decomposition (SVD). For short notation, the MRC approximation technique with SVD will be referred to as The Maximum Singular Vector Approach (MSVA) in this text.

Due to coherent combining on both sides of the channel this approach benefits from array gain, although of a weaker kind than MRC.

To enable adaptive weighting of the information symbols both before transmission and after reception, the MSVA scheme requires knowledge of the channel matrix in both ends of the channel. In practise, this demand results in the need for both training sequences and feedback information, making this scheme rather complex. However, for simplicity we disregard that here, simply assuming that both the transmitter and the receiver have perfect knowledge of the channel.

Signal model for The Maximum Singular Vector Approach

We consider one symbol period, the time it takes to transmit the symbol s. As in transmit MRC, what is transmitted over the channel is not N copies of the symbol s, but a vector of weighted symbols, $\mathbf{s}_w$. It is related to s as follows

$$\mathbf{s}_w = \mathbf{w}_t s, \tag{3.12}$$

where the transmit weight array $\mathbf{w}_t$ has the dimensions $N \times 1$, Substitution of this $\mathbf{s}_w$ into a vector version of the transmission model of (2.36) yields

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\mathbf{s}_w + \mathbf{v} \\ &= \mathbf{H}\mathbf{w}_t s + \mathbf{v} \end{aligned} \tag{3.13}$$

When the receiver applies an $M \times 1$-sized receive weight array $\mathbf{w}_r$, the retrieved symbol $\hat{s}$ is written as follows

$$\hat{s} = \mathbf{w}_r^T \mathbf{y} = \mathbf{w}_r^T \mathbf{H}\mathbf{w}_t s + \mathbf{w}_r^T \mathbf{v} \tag{3.14}$$

An illustration of MIMO diversity is shown in figure 3.2. The applications of the weight vectors are depicted both before transmission and after reception. Also, the figure indicates that the receiver estimates the weights, possibly from known training sequences. The channel information learnt by the receiver is transmitted to the sender as feedback, so the appropriate weights may be applied there too.

To find the optimal weight vectors, we would like to maximise the SNR of this expression; i.e. maximise the gain factor $\mathbf{w}_r^T \mathbf{H} \mathbf{w}_t$. A criterion is that both $\mathbf{w}_t$ and $\mathbf{w}_r$ have unit norm, in order to avoid boosting the transmit power with $\mathbf{w}_t$ and amplifying the noise with $\mathbf{w}_r$.

For a given channel matrix, the maximum gain is found when $\mathbf{w}_t$ and $\mathbf{w}_r$ correspond to the principal conjugate left and principal right singular vectors, respectively [2].

These optimal weight vectors, $\mathbf{w}_t$ and $\mathbf{w}_r$, are found by performing a Singular Value Decomposition (SVD) on the channel matrix $\mathbf{H}$. A singular value σ and its corresponding singular vectors $\mathbf{u}$ and $\mathbf{v}$ are related as

$$\begin{aligned} \mathbf{H}\mathbf{v} &= \sigma \mathbf{u} \quad \text{and} \\ \mathbf{H}^H \mathbf{u} &= \sigma \mathbf{v}. \end{aligned} \tag{3.15}$$

With the singular values on the diagonal of a diagonal matrix Σ and the corresponding singular vectors forming the columns of two unitary matrices $\mathbf{U}$ and $\mathbf{V}$, we have

$$\begin{aligned} \mathbf{H}\mathbf{V} &= \mathbf{U}\Sigma \quad \text{and} \\ \mathbf{H}^H \mathbf{U} &= \mathbf{V}\Sigma. \end{aligned} \tag{3.16}$$

The fact that $\mathbf{U}$ and $\mathbf{V}$ are unitary means that $\mathbf{V}\mathbf{V}^H = \mathbf{U}\mathbf{U}^H = \mathbf{I}$, the identity matrix, and we can combine (3.16) to

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H \tag{3.17}$$

We define

$$\begin{aligned} \mathbf{w}_t &= \mathbf{V}_1 \quad \text{and} \\ \mathbf{w}_r &= \mathbf{U}_1^*, \end{aligned} \tag{3.18}$$

where $\mathbf{V}_1$ and $\mathbf{U}_1$ are the top column vectors of $\mathbf{V}$ and $\mathbf{U}$. Now, (3.14) is rewritten into

$$\begin{aligned} \hat{s} &= \mathbf{U}_1^H \mathbf{H} \mathbf{V}_1 s + \mathbf{U}_1^H \mathbf{v} \\ &= \mathbf{U}_1^H \mathbf{U} \Sigma \mathbf{V}^H \mathbf{V}_1 s + \mathbf{U}_1^H \mathbf{v} \end{aligned} \tag{3.19}$$

Recall that $\mathbf{U}$ and $\mathbf{V}$ are unitary. Then, the expression is simplified to

$$\hat{s} = \sigma_1 s + \mathbf{U}_1^H \mathbf{v} \tag{3.20}$$

Figure 3.2: MIMO system with diversity

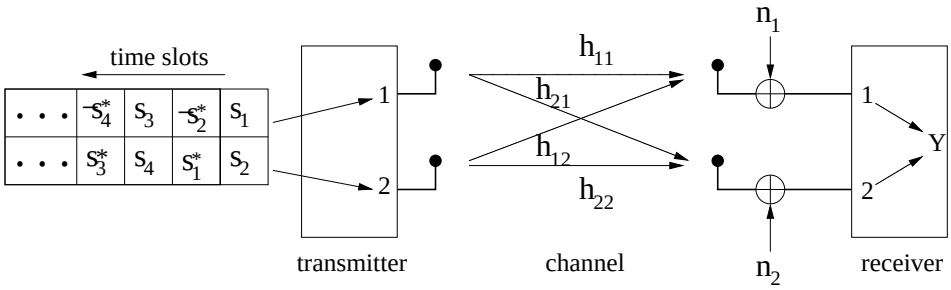

Figure 3.3: A 2-by-2 Alamouti system .

where σ_1 is the largest singular value of **H**.

Now we have used only the top singular vectors to weight, transmit and combine information symbols. This scheme optimises the BER for diversity transmission with a rate of one symbol per period. However, it is also possible to increase the data rate directly by spatial multiplexing of separate symbol streams, through using more than one singular vector. In fact, one may use the SVD to create $\min(N, M)$ independent spatial channels and transmit one sub-stream on each ([2], [13]).

3.1.4 Alamouti space-time coding

Space-time coding (STC) techniques, as described in [20], code the symbols in both space and time, before transmission. A common idea is to build a $N \times K$ matrix of symbols by spreading out the information symbols over space (the N transmit antennas) and time (the K columns may be thought of as time slots). With multiple antennas, there are naturally many ways to code a transmit pattern, but all are not equally effective.

The last of the diversity algorithms to be presented, and also a vital part of the combination scheme described in chapter 4, is the Alamouti space-time block code (STBC) [1], a particular STC scheme. This is a simple, full rate MIMO algorithm, developed for a system of 2 transmit antennas and M receive antennas, providing a diversity order of $2M$. It is full rate in the sense that it transmits one symbol per symbol period, regardless of the multiple antennas.

The Alamouti scheme rearranges the symbol array to an orthogonal space-time coded symbol matrix, targeting BER improvement. Principles of STBC from orthogonal design are discussed in [19]. A graphical model of transmission with Alamouti coding is shown in figure 3.3, where $M = 2$ for simplicity. The matrix to the left shows how the symbols are coded in space and time.

For a 2×1 system, the Alamouti scheme offers only diversity gain. However, for $M > 1$, there is also combining on the receive side, which

gives the additional benefits of array gain.

The Alamouti STBC is part of the group of *complex orthogonal designs*, as defined and discussed in [19]. The same text also proves that such designs exist only for $N = 2$ transmit antennas, so it is not possible to extend the very favourable qualities of orthogonality to $N > 2$. Hence, the Alamouti STBC is in some sense a unique scheme.

Signal model for Alamouti STC

A vital assumption for the Alamouti algorithm is that the channel coefficients can be regarded as constant over two time slots. This allows transmitting the whole pattern over the same channel.

The Alamouti-coded symbol matrix for time interval $(2k, 2k+1)$ is given by $\mathbf{S} = \mathbf{S}_k$, where

$$\mathbf{S}_k = \begin{bmatrix} s_{2k} & -s_{2k+1}^* \\ s_{2k+1} & s_{2k}^* \end{bmatrix} \tag{3.21}$$

With this $\mathbf{S}$, transmission with the Alamouti algorithm may be expressed using the model in (2.35), rewritten here as

$$\mathbf{Y} = \mathbf{HS} + \mathbf{V} \tag{3.22}$$

where the noise matrix $\mathbf{V}$ has dimensions $M \times 2$. To limit the total transmitted power from the two transmit antennas to 1, we normalize each entry in the channel matrix $\mathbf{H}$ so that

$$h_{ij} \sim C(0, \frac{1}{\sqrt{N}}) \quad i \in [1, M],\ j \in [1, N], \tag{3.23}$$

which gives mean $\mu = 0$ and a standard deviation of $\sigma = \frac{1}{\sqrt{N}}$.

A modified version of this representation is obtained if we move the space-time coding from $\mathbf{S}$ to $\mathbf{H}$, yielding a coded channel matrix $\tilde{\mathbf{H}}$, of dimensions $2M \times 2$. With this model, the received signal is given by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{s} + \mathbf{n}, \tag{3.24}$$

where the signal vector $\mathbf{s}$ is

$$\mathbf{s} = \mathbf{s}_k = \begin{bmatrix} s_{2k} \\ s_{2k+1} \end{bmatrix}, \tag{3.25}$$

and the receive vector $\tilde{\mathbf{y}}$ has dimensions $2M \times 1$. The top M entries are the received symbols in the first time slot, while the bottom half represents the second symbol period.

We observe that the uncoded $M \times 2$ channel matrix $\mathbf{H}$ can be seen as a collection of 2 transmit vectors, $\mathbf{h}_n, n \in [1, 2]$.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 \end{bmatrix}, \tag{3.26}$$

where each $M \times 1$-sized $\mathbf{h}_n$ is given by

$$\mathbf{h}_n = \begin{bmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{Mn} \end{bmatrix}, \quad n \in [1, 2] \tag{3.27}$$

Using the notation from (3.26), the modified channel matrix $\tilde{\mathbf{H}}$ is written as

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 \\ \mathbf{h}_2^* & -\mathbf{h}_1^* \end{bmatrix}, \tag{3.28}$$

and we observe that the relation between the matrix $\mathbf{Y}$ in (3.22) and the vector $\tilde{\mathbf{y}}$ in (3.24) is

$$\tilde{\mathbf{y}} = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{12}^* \\ y_{22}^* \end{bmatrix}, \quad \text{given} \quad \mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \tag{3.29}$$

To detect the original information symbols, the Alamouti algorithm exploits the orthogonal structure of $\tilde{\mathbf{H}}$, which may be expressed as

$$\begin{aligned} \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} &= (|\mathbf{h}_1|^2 + |\mathbf{h}_2|^2) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \|\mathbf{H}\|_F^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.30}$$

Now, the retrieved symbols may be expressed as

$$\begin{aligned} \hat{\mathbf{s}} &= \frac{1}{\|\mathbf{H}\|_F^2} \tilde{\mathbf{H}}^H \tilde{\mathbf{y}} \\ &= \mathbf{s} + \frac{1}{\|\mathbf{H}\|_F^2} \tilde{\mathbf{H}}^H \mathbf{n} \end{aligned} \tag{3.31}$$

This operation requires that the receiver knows the channel matrix $\mathbf{H}$.

3.1.5 Spatial multiplexing with zero-forcing

Theoretic research in information theory, such as in [6] and [5], has shown that multi-path channels have enormous capacities regarding the transmission data rate. The demand is that the multi-path scattering

must be sufficient and properly exploited. One approach that ensures the latter is a diagonally layered space-time architecture by Foschini; known as diagonal BLAST (Bell Labs Layered Space-Time) or D-BLAST [6]. Associated with this algorithm is a simplified version; vertical BLAST or V-BLAST, see [22] and [4]. Both approaches are spatial multiplexing (SM) schemes.

The effects of multi-path propagation have traditionally been considered to be destructive, but SM uses them constructively to increase the data rate. This is done by demultiplexing an incoming symbol stream into N independent sub-streams, N being the number of transmit antennas. Next, these sub-streams are transmitted simultaneously from separate antennas.

Each of the receive antennas sees a differently faded sum of all the sub-streams, with noise added. Demultiplexing and detection is done according to some chosen criterion, such as minimum mean-squared error (MMSE) or zero-forcing (ZF). In this text, the symbol recovery is shown using simple ZF, which means the inverse or the pseudo-inverse of $\mathbf{H}$, denoted $\mathbf{H}^{\#}$, must satisfy

$$\mathbf{H}^{\#}\mathbf{H} = \mathbf{I}_N, \tag{3.32}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. One requirement is that the number of receive antennas must at least equal the number of transmit antennas, $M \geq N$, or else the system of equations has more unknowns than equations. Figure 3.4 shows a square MIMO system with $N = M = 3$ antennas on each side. The incoming symbol stream is demultiplexed and $1/3$ of the symbols are transmitted from each antenna. At the receiver, demultiplexing is performed to retrieve the original symbols.

With sufficient multi-path, $\mathbf{H}$ is a full rank channel and the data rate increases linearly with $\min(N, M)$. Cities and other densely populated areas are well suited for SM transmission, but it does not work as well with line-of-sight (LOS) paths. This is because the channel coefficients in a pure LOS environment are highly correlated in space, rendering spatial multiplexing useless.

A high level of correlation due to closely spaced antenna elements at the transmit or receive side (or both) has the same effect, the BER increases dramatically.

The BER-performance of a 2×2 MIMO system with spatial multiplexing is similar to that of a SISO system, hiding the fact that the latter case has a higher data rate. When adding more receive antennas, so that $M > N$, the extra elements will help bring the BER down for SM[6].

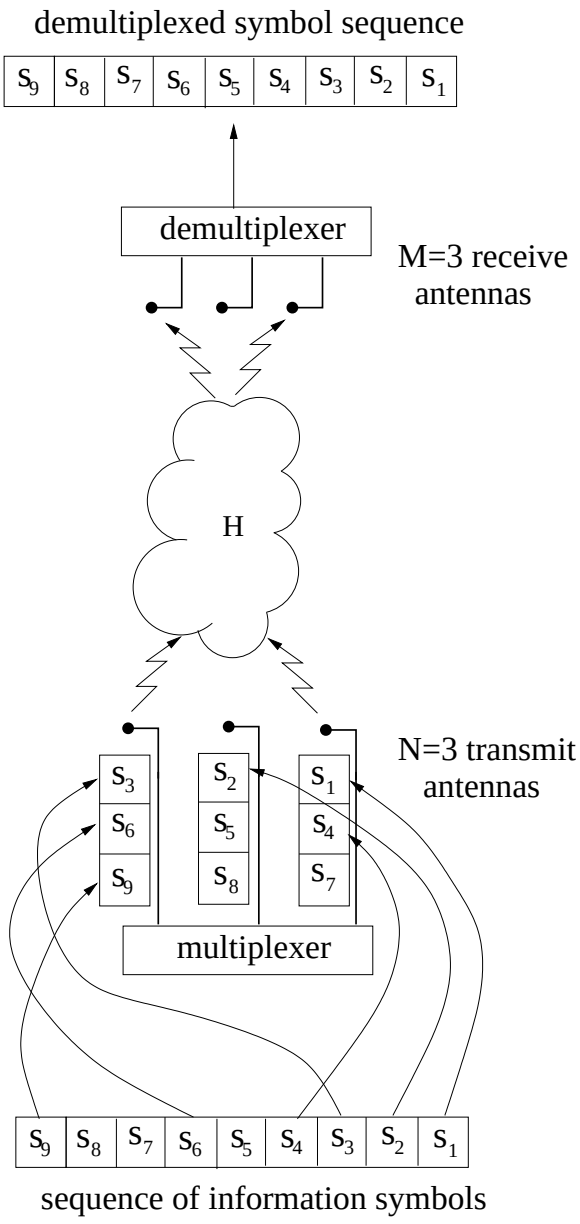

Figure 3.4: A square MIMO system with SM.

Signal model for SM with zero-forcing

We study transmission over one symbol period, and build a symbol vector $\mathbf{s}$ as

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix}.$$

This vector fits the model in equation (2.36), repeated here for convenience:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v}, \tag{3.33}$$

where the dimensions of $\mathbf{H}$, $\mathbf{s}$ and $\mathbf{v}$ are $M \times N$, $N \times 1$ and $M \times 1$. We ensure that the total transmit power is limited to 1 by normalising the channel coefficients so that any h_{ij} in $\mathbf{H}$ is described as in (3.23).

We invert the system by using $\mathbf{H}^{\#}$, the pseudo-inverse of $\mathbf{H}$, and recover the symbols by

$$\begin{aligned} \hat{\mathbf{s}} =& \mathbf{H}^{\#}\mathbf{y} \\ =& \mathbf{H}^{\#}\mathbf{H}\mathbf{s} + \mathbf{H}^{\#}\mathbf{v} \\ =& \mathbf{s} + \mathbf{H}^{\#}\mathbf{v} \end{aligned} \tag{3.34}$$

Impact of channel rank

From studying the above equation it is clear why spatial multiplexing does not perform well with low rank channels. If $\mathbf{H}$ does not have full rank, the relation in (3.32) does not hold, and the symbols will not be properly resolved. In practise, the rank of real life MIMO channels is almost always full. However, the matrix may still be ill-conditioned, in which case inverting the system leads to amplification of the noise and an increase in the BER.

Summary

This section has presented ideas and mathematical models for some well-known multiple antenna transmit schemes. Both MD and SM approaches are represented, differing in the way they achieve higher capacity by use of antenna arrays. The next section is devoted to comparing the BER performance of the respective algorithms, under assumptions of both uncorrelated and correlated fading.

Before moving on, we note that we expect the BERs of both SM and MD approaches to rise with increasing levels of correlated fading among the antennas. However, SM with zero-forcing depends critically on the

rank of the channel matrix, and can break down completely if it is lower than expected. The MD schemes are expected to suffer from correlation in a more smooth way, as they are not directly dependent the channel being invertible.

3.2 Performance comparisons of MD and SM schemes

First, we present some of the conditions for the comparison, in order to secure equal terms. We show and discuss the performance results of transmission in both uncorrelated and correlated channels, and for different number of antennas. We will see that, for given MIMO system with (N, M), the compared schemes are very different when it comes to how badly correlated channels they can tolerate and still perform at a reasonable level.

We wish to compare and discuss performance results for the algorithms previously presented, with varying levels of correlated fading. We repeat the correlation model in (2.41) here. The channel matrix $\mathbf{H}$ contains correlated fading coefficients, generated from the matrix of independent entries $\mathbf{H}_0$ and the correlation matrices $\mathbf{R}_t$ $\mathbf{R}_r$, so that

$$\mathbf{H} = \sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t} \tag{3.35}$$

We assume the same level of correlated fading on both sides of the channel, so a given correlation level between neighbouring antennas r, defined in (2.39) applies to both the transmit and receive arrays.

We choose to study the following schemes: MSVA, SM with zero-forcing detection and the Alamouti STC. In order to perform a suitable comparison of these transmit methods, we need to ensure identical premises for all. This is done by keeping an equal number of antennas on both sides of the channel and transmitting at the same bit rate for all the algorithms. In addition, the order of diversity is kept constant by ensuring that the level of correlated fading does not change between the compared schemes.

With N transmit antennas the spatial multiplexing approach will transmit at N times the symbol rate of the diversity-oriented schemes (MSVA and Alamouti STC). To keep the the bit rates equal we use modulation schemes of different order.

If an M-ary modulation method is used for SM, each symbol represents $R_{SM} = \log 2(M)$ bits. The MD schemes transmit one symbol per period, so they achieve the same bit rate as SM by using a scheme with a rate of $R_{MD} = N \log 2(M)$. N is the number of transmit antennas used for the SM.

We choose to study the performance of the various algorithms for systems of sizes 2×2 and 2×4. The choices of modulation methods in

this text are $\pi/4$ QPSK for SM and 16QAM for MSVA and Alamouti STC, for details on modulation see section 2.1.1. Hence, all the schemes have a common bit rate of 4 bits transmitted per symbol period.

We compare the schemes for the following settings:

- $N = 2,\ M = 2$
  - no correlation, see figure 3.5
  - correlation $r = 0.29$, see figure 3.6
  - correlation $r = 0.90$, see figure 3.7
- $N = 2,\ M = 4$
  - no correlation, see figure 3.8
  - correlation $r = 0.29$, see figure 3.9
  - correlation $r = 0.90$, see figure 3.10

These parameters are also found in the legends of the figures, for easy reference.

With reference to (2.40), we see that the correlation levels $r = 0.29$ and $r = 0.90$ correspond to a distance between neighbouring antenna elements of $d = 0.3\lambda$ and $d = 0.1\lambda$, respectively.

3.2.1 Results

We observe that the BER performance of the schemes vary for different number of antennas and varying levels of correlation.

First, we concentrate on the system $N = M = 2$. The case of no correlation is shown in figure 3.5 while the results with correlated channel fading of $r = 0.29$ and $r = 0.90$, between neighbouring antenna elements, are shown in figures 3.6 and 3.7.

The MSVA out-performs the Alamouti STBC, a result that is expected because MSVA uses instantaneous and full channel state information both at the transmit and receive sides of the communication channel. Both schemes experience full diversity gain in the uncorrelated case, and use channel information at the receiver to achieve receive array gain. But, because MSVA has channel knowledge at the receiver too, it combines to achieve transmit array gain as well.

Even in the case of no correlation, SM does not reach low BER-values, and we see that it suffers from a lack of diversity in a square MIMO system. It is also apparent from the plots that SM is heavily affected by increasing correlation. For $r = 0.90$ the BER is very high, even for relatively good SNR-levels, the performance is poor. At an SNR of about 23 dB, 1 out of 10 bits is erroneously determined at detection.

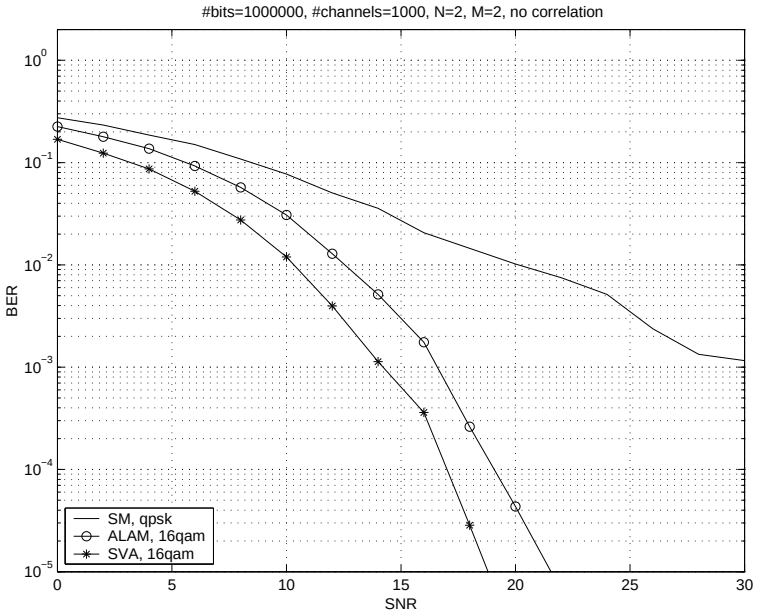

Figure 3.5: Comparison of MIMO algorithms, no correlation, $N = M = 2$

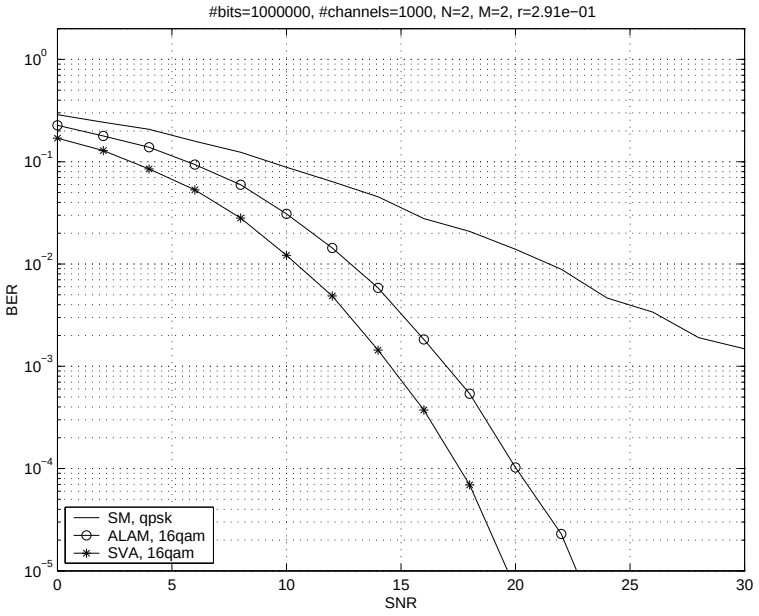

Figure 3.6: Comparison of MIMO algorithms, correlation $r = 0.29$, $N = M = 2$

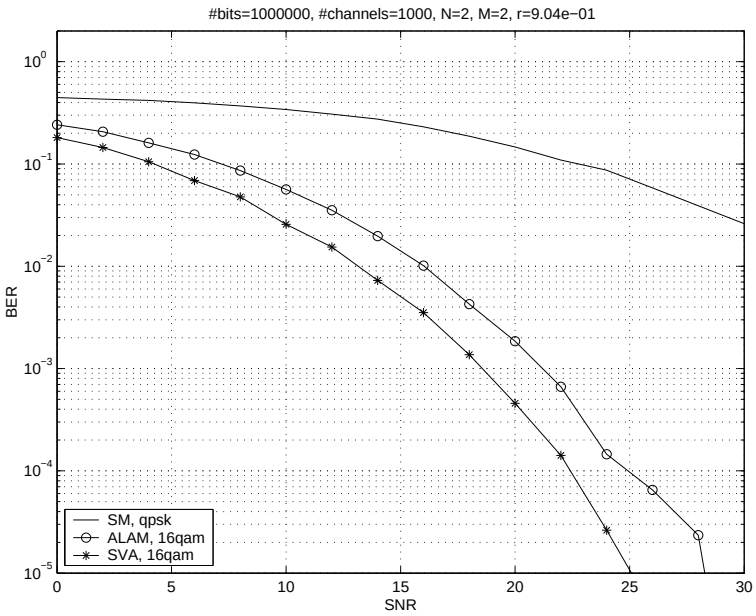

Figure 3.7: Comparison of MIMO algorithms, correlation $r = 0.90$, $N = M = 2$

This is consistent with the SM model in section 3.1.5, when $N = M$ there is no diversity in SM, as all the degrees of freedom are used to transmit independent data. So, in the case of a square H, SM is very dependent on the channel matrix having full rank.

The MD schemes in figures 3.6 and 3.7 are also affected by the correlation, but not as destructively, thanks to the diversity advantage and the absence of any matrix inversions. In fact, there is not much of a difference in performance between the cases of no and moderate correlation, figures 3.5 3.6. Both schemes enjoy the benefits of array gain.

If we move our attention to the figures 3.8, 3.9 and 3.10, where the number of antennas are $N = 2$ and $M = 4$, we see that things are different. For one, none of the transmit schemes are affected nearly as much by a high level of correlation as they were when $M = N$. The extra receive antennas help in bringing the BER down, even when the fading is correlated.

For SM with zero-forcing, a system where $M > N$ is advantageous under assumptions of correlated fading. More receive than transmit antennas translates to having a system with more equations than unknowns, which means that there is now some diversity to be exploited, even for SM.

For the uncorrelated case and the case of low correlation, we see that

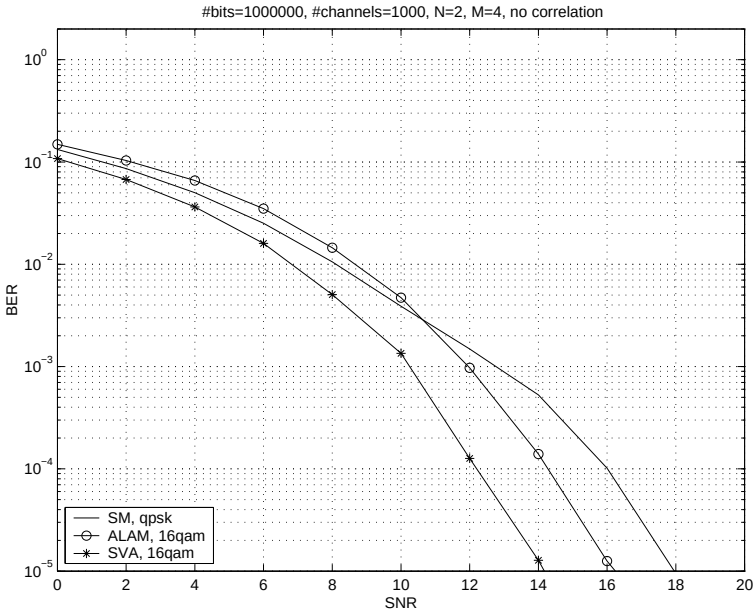

Figure 3.8: Comparison of transmit algorithms, no correlation, $N = 2$, $M = 4$

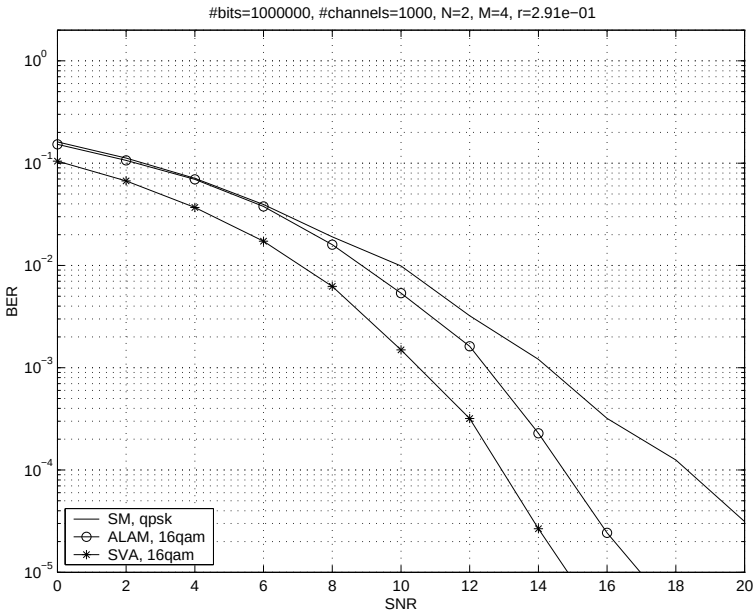

Figure 3.9: Comparison of transmit algorithms, correlation $r = 0.29$, $N = 2$, $M = 4$

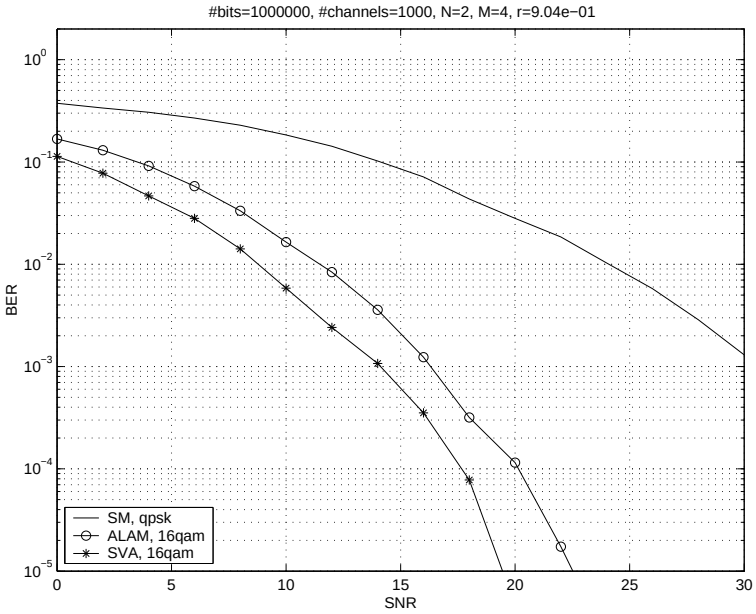

Figure 3.10: Comparison of transmit algorithms, correlation $r = 0.90$, $N = 2$, $M = 4$

the SM approach actually out-performs the Alamouti STC at low SNR-values, because it contains both diversity and multiplexing gain.

Summing up, it is clear that the performance comparison between SM and MD schemes depends on the number of antennas. For MD schemes, a higher number of antennas yield diminishing returns, while SM has trouble dealing with a square MIMO system due to a lack of diversity. This calls for using both SM and MD schemes on the same array instead of just one of the schemes.

Another point is that the SM and MD approaches react differently to antenna correlation, SM performance breaks down more abruptly than that of MD schemes. In a practical array, the level of correlated fading is not equal for all antenna pairs. Again, this calls for mixing the two approaches over the same array, along with a clever mapping of SM and MD to particular groups of antennas.

On this background, we attempt to build a spatial combination of SM and MD schemes in the next chapter.

Chapter 4

Combining SM and MD algorithms via the SMAL scheme

4.1 Motivation

In the previous chapter we studied examples of schemes following one of two standard approaches; spatial multiplexing (SM) or MIMO diversity (MD). SM schemes use the spatial diversity to maximize the data rate, while MD algorithms aim at reducing the bit-error-rate, in order to make the transmission more reliable.

However, both sets of algorithms have their limitations. Spatial multiplexing is very sensitive to severely faded channels or systems with a high degree of correlation. Such conditions will increase the BER dramatically, as seen in section 3.2.1. For MIMO diversity the increased reliability tends to saturate when we add more antennas [3].

To avoid allocating all degrees of freedom to any particular scheme (SM or MD), which results in either an excess of diversity or a lack of it, we consider the problem of combining both techniques on the same array via allocation of groups of antennas to one scheme or the other. The idea is that this spatial combination should give some combined gain from both approaches; i.e. robustness against fading and correlation as in MD schemes, while at the same time directly increasing the transmit data rate, as in SM.

Our combining approach

We present a combination where the allocation of antenna groups to transmit schemes may be done in several spatial patterns, shown to yield significantly different performance when using an antenna array where

the level of correlation is uneven across the antenna pairs (such as in a linear array). Our goal is to choose the best of these antenna allocation patterns.

The combination attempted in this thesis is inspired by the ideas in an article by Heath and Paulraj [12], where instantaneous channel information is used to switch between multiplexing and MIMO diversity schemes *over time*.

Our approach is different, we study the problem of switching between multiplexing and diversity *over space*, using both schemes simultaneously. We propose a simple algorithm allowing us to generalise the work of [12], and develop it both for the case where instantaneous full channel feedback is available to the transmitter and for the case where only long-term correlation statistics are known.

In this text, MD schemes are represented by the Alamouti space-time coding. The spatial combination is essentially a spatial multiplexing of blocks of Alamouti-coded symbol blocks. This is done over a transmit array with a multiple of two antennas. For easy reference, we call this scheme SMAL (Spatial Multiplexing of ALamouti). Instantaneous SMAL and statistical SMAL refers to two versions of the scheme, suitable for the cases where full channel information is available and when we only know the correlation statistics, respectively. We show the performance gains by plotting the bit-error rate (BER) against the SNR for both cases.

This work has also resulted in an article submission to NORSIG 2003 [17].

Although the SMAL scheme differs greatly from the time-switching presented in [12], parts of the work and results still apply , and we take some time to present them in section 4.2.

The organisation of this chapter

First, section 4.2 presents results from the article by Heath and Paulraj [12].

In section 4.3 we develop the model for the new SMAL scheme. It illustrates how the combination of the two algorithms is performed, in particular by developing a mathematical model.

A primary aim is to find a good decision metric in order to select the best allocation pattern available, and some optimisation principles are presented in section 4.3.2. They are based on the wish to maximise the distance between the received symbols, in order to decrease the probability of making the wrong decision. This metric is used in the two following sections, but for different scenarios of channel knowledge.

In 4.3.4 the criterion is used directly by using instantaneous channel information. BER-performance results from simulations are also given.

Section 4.3.6, is devoted to finding a metric that is not dependent on instantaneous channel matrices. This is accomplished using the statistical characteristics of the channel and the correlation matrices. The performance measures of this version of SMAL is discussed and compared to the instantaneous SMAL, which is shown to yield better results but at the cost of increased complexity.

4.2 Combining SM and MD in time

In [12], the idea is to compare SM and MD schemes are compared to determine which is best suited under the current conditions. One uses different modulation schemes for MD and SM, to ensure the same bit rate and a fair comparison.

The article presents and justifies a criterion for switching between SM and MD schemes. The choice scheme is based on a desire to minimise the BER and requires instantaneous channel state information.

A large distance between the symbols in the receive constellation is one way to lower this measure; wrong decisions are less likely if the symbols are spaced further apart.

Hence, at any given time, we choose the approach which offers the largest *minimum, squared, Euclidean distance* of the receive constellations [12], denoted $d^2_{min,MD}(\mathbf{H})$ for MD and $d^2_{min,SM}(\mathbf{H})$ for SM methods. Lower and upper bounds on these distances are derived as [12]

$$d^2_{min,SM}(\mathbf{H}) \geq \sigma^2_{min}(\mathbf{H})\frac{d^2_{min,sm}}{N} \tag{4.1}$$

$$d^2_{min,MD}(\mathbf{H}) \leq \frac{1}{N}\|\mathbf{H}\|^2_F d^2_{min,md} \tag{4.2}$$

where $d^2_{min,sm}$ and $d^2_{min,md}$ are the minimum, squared, Euclidean distances of the transmit constellations, $\sigma_{min}(\mathbf{H})$ is the minimum singular value of $\mathbf{H}$, and $\|\mathbf{H}\|_F$ is the Frobenius norm of the channel matrix $\mathbf{H}$.

For the spatial multiplexing approach, we see that if the minimum singular value of the channel matrix $\mathbf{H}$ is high, we know that even the most heavily faded channel has tolerable quality. Then the chances of a successful, simultaneous transmission of N independent data streams are good.

When, on the other hand, $\sigma^2_{min}(\mathbf{H})$ is small, the probability of error is high. In extreme cases, the matrix may not have full rank, in which case it is not invertible and the symbols will not be correctly determined upon arrival.

The Frobenius norm of a matrix gives a measure of the magnitude of its elements. For a channel matrix, these magnitudes may be seen

as power measures. For MIMO diversity methods, higher total transmit power means a lower probability of error.

When combining the two bounds, (4.2) and (4.1), and using a conservative approach, we can derive a simple rule in which spatial multiplexing is only used when

$$\sigma^2_{min}(\mathbf{H})d^2_{min,sm} \geq \|\mathbf{H}\|^2_F d^2_{min,md} \tag{4.3}$$

For a given channel matrix **H**, a large minimum eigenvalue implies SM transmission, while a large Frobenius norm means diversity is preferable.

With periodic evaluation of instantaneous channel information, this time-switching scheme is able to adapt to the current conditions. Given that the decision metric is reasonable, this adaptivity may greatly improve transmission reliability.

In the next section, a new scheme is presented, based on switching between various spatial patterns of SM- and MD-combinations rather than switching between SM and MD over time.

4.3 Combining SM and MD in space; the SMAL scheme

Inspired by the work in [12], this section presents an attempt to combine SM and MD methods in space through mapping SM and MD-based blocks of data to particular groups of antennas. In order to simplify notation and exposition, the MD scheme considered is the Alamouti STC algorithm. However, extensions to other space-time block codes are possible.

We wish to spatially multiplex several Alamouti ST-coded blocks containing independent groups of 2 symbols each. We assume an even number of transmit antennas N

$$N = 2k,\ k \geq 2 \quad \text{and} \quad M \geq N/2, \tag{4.4}$$

in which case there are room for $N/2$ Alamouti-coded blocks over the transmit antenna array.

The symbol rate of a SMAL system with N transmit antennas is $R_{SMAL} = N/2$ per symbol period. This is obvious as there are $N/2$ Alamouti ST-coded blocks that each transmits independent data at a symbol rate of $R_{AL} = 1$. In this thesis we do not optimise the rate of transmission, but rather the antenna mapping for a fixed symbol rate of $N/2$.

Given (4.4), we know that $N > 4$ and the $N/2$ Alamouti-coded blocks may be mapped to the transmit antennas in more than one spatial pattern. Depending on the level of correlation, some of these antenna assignment patterns may yield better performance than others, i.e. a lower

BER. With negligible correlation, switching between different spatial pattern is without effect on the performance, as all the channel coefficients are equally independent.

The SMAL scheme attempts to choose the pattern that offers the best performance, with a decision metric inspired by the ones in (4.1) and (4.2).

4.3.1 The SMAL channel model

We make the SMAL channel model fit the previously presented model in equation (2.35), rewritten here as

$$\mathbf{Y} = \mathbf{HS} + \mathbf{V} \tag{4.5}$$

The channel matrix $\mathbf{H}$ is correlated, and defined as in (2.41). Assume we wish to transmit the block

$$\mathbf{s} = \begin{bmatrix} s_0 & s_1 & \dots & s_{N-1} \end{bmatrix}^T,$$

over two symbol durations. To do that, we build an $N \times 2$-sized symbol matrix $\mathbf{S}$, from $N/2$ Alamouti ST-coded matrices $\mathbf{S}_k$. This notation was presented in (3.21), but the expression is repeated here.

$$\mathbf{S}_k = \begin{bmatrix} s_{2k} & -s_{2k+1}^* \\ s_{2k+1} & s_{2k}^* \end{bmatrix} \tag{4.6}$$

The most trivial way to assign the Alamouti blocks to all N antennas is by selecting $\mathbf{S}$ in (4.5) such that

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_0 \\ \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_{N/2-1} \end{bmatrix}, \tag{4.7}$$

in which case one Alamouti-coded block is mapped to the first two antennas, the second block is mapped to the next antenna pair and so on through the array. For $N = 4$, this is illustrated as pattern p_1, to the left in figure 4.1.

Alternatively, the combination may also be expressed by moving the space-time coding structure from $\mathbf{S}$ to the channel matrix, as seen in the section on Alamouti STC 3.1.4. In that case, we express the model as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{s} + \mathbf{n}, \tag{4.8}$$

where $\tilde{\mathbf{H}}$ has dimensions $2M \times N$, because it represents "the channel" over two time slots. The noise vector $\mathbf{n}$ is $2M \times 1$ and $\tilde{\mathbf{y}}$ is the $2M \times 1$

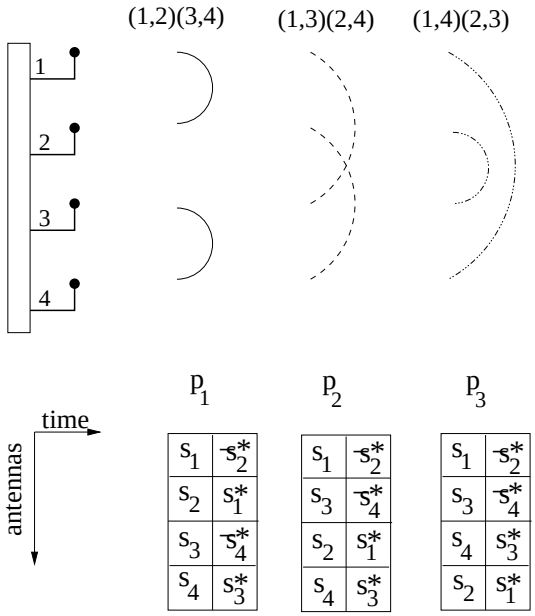

Figure 4.1: Antenna assignment patterns for $N = 4$.

vector of received signals. The symbols are now represented by the $N\times 1$-sized vector $\mathbf{s}$, in which each symbol appears only once. It is built from concatenating $N/2$ vectors of length 2, one for each ST-coded Alamouti block, so that

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_{N/2-1} \end{bmatrix}, \quad \text{where} \quad \mathbf{s}_k = \begin{bmatrix} s_{2k} \\ s_{2k+1} \end{bmatrix} \tag{4.9}$$

More generally, the Alamouti-coded blocks may be assigned to any antenna subgroups. Let us define P_N, the number of non-trivially equivalent antenna assignment patterns. These patterns are labelled p_k, $k \in [1, P_N]$.

By trivially equivalent patterns, we mean patterns that result in the same correlation information, as seen by the algorithms. We give an example for the case of 6 transmit antennas. Pattern p_2 in figure 4.2 maps the 3 Alamouti-coded blocks to the following antenna pairs: $(1,2)$, $(3,5)$ and $(4,6)$. This pattern is trivially equivalent to a mirrored combination p_2', in which antenna pairs $(1,3)$, $(2,4)$ and $(5,6)$ are coupled for transmission of ST-coded Alamouti blocks.

For each possible pattern p_k, the space-time coded matrix $\tilde{\mathbf{H}}$ has a unique structure. The different $\tilde{\mathbf{H}}$ are denoted $\tilde{\mathbf{H}}(p_k)$, and for pattern

p_k, the general input/output model is now

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}(p_k)\mathbf{s} + \mathbf{n}, \tag{4.10}$$

The goal is to find the best pattern p_k, evaluated by a criterion used on the space-time coded channel matrix $\tilde{\mathbf{H}}(p_k)$. We later explain how to express $\tilde{\mathbf{H}}(p_k)$ mathematically and we develop a criterion, based on the one in [12].

For the case of $N = 4$, the different patterns are shown in figure 4.1. There are $P_4 = 3$ unique arrangements, and the system transmits $N/2 = 2$ Alamouti-coded blocks simultaneously. The lines between the transmit antennas and the labels show which antennas are paired to transmit one ST-coded Alamouti block in cooperation. The figure also shows the space-time coded symbol matrix $\mathbf{S}$ of the original model in (4.5) over two symbol periods, for each of the patterns.

Pattern p_1 is the most obvious arrangement, and equivalent to the one in (4.7). In p_1, the two Alamouti blocks are placed next to each other. The two remaining patterns are p_2; where antennas 1 and 3 transmit one block, and antennas 2 and 4 the other, and p_3; where the first transmission pair consists of antennas 1 and 4, while antennas 2 and 3 make up the second.

From the space-time coded symbol matrices $\mathbf{S}$ in figure 4.1 we read that, for p_1, antennas 1, 2, 3 and 4 transmit the symbols s_0, s_1, s_2, and s_3 in the first time slot, respectively. In the second time slot, the same antennas transmit the symbols $-s_1^*, s_0^*, -s_3^*$, and s_2^*. So, over two time slots, antennas 1 and 2 have transmitted one Alamouti block together, while the other block has been transmitted by antennas 3 and 4.

Next, we study the structure of the coded channel matrix $\tilde{\mathbf{H}}(p_k)$. Using the notation from section 3.1.4, equation (3.26), we see that any channel matrix may be rewritten as a collection of N transmit vectors $\mathbf{h}_n, n \in [1, N]$.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \cdots & \mathbf{h}_{N-1} & \mathbf{h}_N \end{bmatrix}, \tag{4.11}$$

where $\mathbf{H}$ is the uncoded, correlated MIMO channel matrix. Each $M \times 1$-sized $\mathbf{h}_n$ is given by

$$\mathbf{h}_n = \begin{bmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{Mn} \end{bmatrix}, \quad n \in [1, 2] \tag{4.12}$$

We continue to use $N = 4$ and pattern p_1 as an example, and extend the $\tilde{\mathbf{H}}$ from the section on Alamouti STC, see (3.28), to our more general

case. The SMAL space-time coded channel matrix for pattern p_1 is given as

$$\tilde{\mathbf{H}}(p_1) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 \\ \mathbf{h}_2^* & -\mathbf{h}_1^* & \mathbf{h}_4^* & -\mathbf{h}_3^* \end{bmatrix}, \tag{4.13}$$

where the top row represent the spatial coding of the channel coefficients in the first time period, while the bottom row is rearranged for the second time period.

For the other patterns, p_2 and p_3, the vector-versions of the space-time coded channel matrices are given as

$$\begin{aligned} \tilde{\mathbf{H}}(p_2) &= \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_3 & \mathbf{h}_2 & \mathbf{h}_4 \\ \mathbf{h}_3^* & -\mathbf{h}_1^* & \mathbf{h}_4^* & -\mathbf{h}_2^* \end{bmatrix} \\ \tilde{\mathbf{H}}(p_3) &= \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_4 & \mathbf{h}_2 & \mathbf{h}_3 \\ \mathbf{h}_4^* & -\mathbf{h}_1^* & \mathbf{h}_3^* & -\mathbf{h}_2^* \end{bmatrix} \end{aligned} \tag{4.14}$$

Other examples, $N = 6$

A more complex case arises when $N = 6$, because more antennas generate more patterns. There are $P_6 = 11$ non-trivially equivalent patterns for 6 transmit antennas, as shown in figure 4.2.

To avoid mirrored patterns, we always start with antenna 1, which may be involved in 5 different pairs when $N = 6$. Given that a certain pair has been chosen, there are only 4 unused antennas left, which we know generates 3 patterns. The total number of patterns are the given by $5 \cdot 3 = 15$. However, some of these are mirrored versions of each other, which means they yield equivalent results because of the symmetric nature of the correlation pattern.

For pattern p_5, we show $\tilde{\mathbf{H}}(p_5)$ as the following

$$\tilde{\mathbf{H}}(p_5) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_3 & \mathbf{h}_2 & \mathbf{h}_6 & \mathbf{h}_4 & \mathbf{h}_5 \\ \mathbf{h}_3^* & -\mathbf{h}_1^* & \mathbf{h}_6^* & -\mathbf{h}_2^* & \mathbf{h}_5^* & -\mathbf{h}_4^* \end{bmatrix} \tag{4.15}$$

The full list of the $P_6 = 11$ unique patterns is found in A.1.

As the number of transmit antennas, N, increases, so does the number of possible patterns. To keep the latter at a reasonable level, $N = 4$ and $N = 6$ are the values used for the SMAL scheme, in this text.

In contrast to the pure Alamouti STC, the space-time coded channel matrices $\tilde{\mathbf{H}}(p_k)$ is not orthogonal, so the SMAL detection can not be as simple as matched filter based decoding. The reason is that several Alamouti-coded blocks are only internally orthogonal, and not to each other.

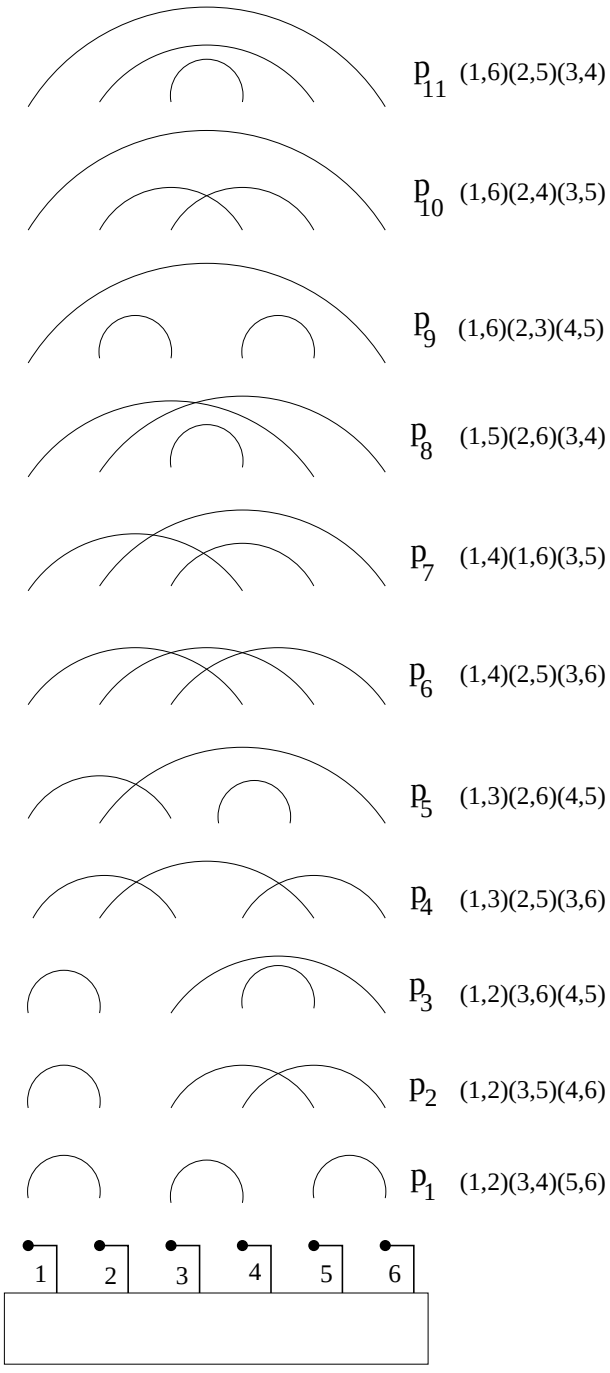

Figure 4.2: Antenna assignment patterns for $N = 6$.

To retrieve the symbols, we rather use zero-forcing, in the same manner as described in section 3.1.5. This is done by computing a pseudo-inverse of $\tilde{\mathbf{H}}(p_k)$, denoted $\tilde{\mathbf{H}}^{\#}(p_k)$, and applying that to the received vector $\tilde{\mathbf{y}}$ of (4.10).

$$\begin{aligned}\hat{\mathbf{s}} &= \tilde{\mathbf{H}}^{\#}(p_k)\tilde{\mathbf{y}} \\ &= \tilde{\mathbf{H}}^{\#}(p_k)\left(\tilde{\mathbf{H}}(p_k)\mathbf{s} + \mathbf{v}\right) \\ &= \mathbf{s} + \tilde{\mathbf{H}}^{\#}(p_k)\mathbf{v}\end{aligned} \tag{4.16}$$

So far, we have established a suitable model for the spatial combination of spatial multiplexing and Alamouti STC, where $\tilde{\mathbf{H}}(p_k)$ depends on the uncoded channel matrix $\mathbf{H}$ and the chosen spatial pattern p_k.

Our goal is to choose the best pattern p_k, in a way inspired by the work of Heath and Paulraj [12]. In order to do this evaluate the patterns against each other, an expression for the space-time coded $\tilde{\mathbf{H}}(p_k)$ is required, and this is the subject of the next section.

Expressing $\tilde{\mathbf{H}}(p_k)$ as function of H and p_k

We wish to write the space-time coded channel matrix $\tilde{\mathbf{H}}(p_k)$ as a function of the correlated channel matrix $\mathbf{H}$ and the pattern p_k.

As seen in the examples (4.13) to (4.15), the top half of $\tilde{\mathbf{H}}(p_k)$ represents the space-time coded coefficients in the first time slot of an Alamouti STC transmission, while the bottom half are transmit vectors belonging to the second time slot.

We postulate that the space-time coded channel matrix may be expressed as

$$\tilde{\mathbf{H}}(p_k) = \begin{bmatrix} \mathbf{H}\,\mathbf{I}(p_k^1) \\ \\ \mathbf{H}^*\,\mathbf{I}(p_k^2)\,\mathbf{G} \end{bmatrix} \tag{4.17}$$

where $\mathbf{I}(p_k^1)$ and $\mathbf{I}(p_k^2)$ are $N \times N$ column permutation matrices for the first and second symbol period, respectively. They rearrange the column order of $\mathbf{H}$ according to the chosen pattern. To enable this, these matrices are identity matrices, but with the columns arranged in the same, desired order.

The matrix $\mathbf{G}$ ensures the even-numbered transmit vectors in the second time slot are negated, and is given as the $N \times N$

$$\mathbf{G} = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & d_N \end{bmatrix}, \quad \text{with } \vec{d} = [1, -1, 1, -1, \dots, 1, -1] \tag{4.18}$$

With this, we have enabled correct rearrangement of the columns of H to generate $\tilde{\mathbf{H}}(p_k)$ in (4.17). To clarify, we continue with some examples.

Example 1, N=4 with pattern p_2
With a transmit array of $N = 4$ antennas and a choice of pattern p_2, the 4×4 matrices $\mathbf{I}(p_2^1)$, $\mathbf{I}(p_2^2)$ and $\mathbf{G}$ look like

$$\mathbf{I}(p_2^1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{I}(p_2^2) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Example 2, $N = 6$ with pattern p_5
Here, we use 6 transmit antennas and the matrices $\mathbf{I}(p_5^1)$, $\mathbf{I}(p_5^2)$ and $\mathbf{G}$ have dimensions 6×6. To generate the space-time coded channel matrix $\tilde{\mathbf{H}}(p_5)$, we build

$$\mathbf{I}(p_2^1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{I}(p_2^2) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Finally, we write out (4.17) in one expression. It is given as the sum of two sets of matrix multiplication, such that

$$\tilde{\mathbf{H}}(p_k) = \mathbf{I}_{r1}\, \mathbf{H}\, \mathbf{I}(p_k^1) + \mathbf{I}_{r2}\, \mathbf{H}^*\, \mathbf{I}(p_k^2)\, \mathbf{G}, \tag{4.19}$$

where the matrices $\mathbf{I}_{r1}$ and $\mathbf{I}_{r2}$ are two simple $2M \times M$ 'row permutation' matrices

$$\mathbf{I}_{r1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{I}_{r2} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & \dots & 1 \end{bmatrix} \tag{4.20}$$

each being a combination of two $M \times M$ matrices; a zero matrix and an identity matrix I_M.

$\mathbf{I}_{r1}$ and $\mathbf{I}_{r2}$ are not dependent on the choice of pattern p_k, their only purpose is to ensure that the top and bottom halves of $\tilde{\mathbf{H}}(p_k)$ represent the first and second time slots.

Expressing $\tilde{\mathbf{H}}(p_k)$ for correlated channels

We assume a non-negligible level of correlated fading. Then, the entries of $\tilde{\mathbf{H}}(p_k)$ are correlated channel matrix coefficients, and as seen in (2.41) and the correlated $\mathbf{H}$ is given by

$$\mathbf{H} = \sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}, \tag{4.21}$$

where $\mathbf{H}_0$ is the channel matrix of uncorrelated entries, while $\mathbf{R}_r$ and $\mathbf{R}_r$ are the correlation matrices on the transmit and receive side, respectively. We assume the same level of correlation on both sides of the channel, so for a square system the matrices will be equal. Writing out $\mathbf{H}$ in (4.19) yields

$$\tilde{\mathbf{H}}(p_k) = \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\,\mathbf{H}_0\,\sqrt{\mathbf{R}_t}\,\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\,\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\,\mathbf{I}(p_k^2)\,\mathbf{G} \tag{4.22}$$

After the development of the transmission model in (4.16) and the expression for $\tilde{\mathbf{H}}(p_k)$ in (4.19), the next step is to formulate a decision criterion for the choice of antenna pattern.

The following section deals with this subject, and the goal is to choose the best pattern in the sense of the BER-based bounds shown in (4.2) and (4.2). We develop a scheme for two cases, when instantaneous channel conditions are available or when we only have access to some long-term statistics. The latter reduces the feedback load and complexity.

4.3.2 Pattern optimisation: principles

One way to solve the problem of which pattern is the best, is to find the optimal antenna groups over which the independent Alamouti-coded blocks will be multiplexed. This can be done by picking the pattern p_{k_0} which maximises the performance in detecting $\mathbf{s}$ in (4.10), in the sense that the squared, minimum, Euclidean distance of the receive constellation, denoted $d^2_{min,SMAL}\left(\tilde{\mathbf{H}}(p_k)\right)$, is maximised.

Inspired by [12], we consider the bounds in (4.2) and (4.1) and use them as building blocks for a selection criterion for SMAL.

For our purpose we recognise that the model used in 4.10 is that of an SM scheme where $N/2$ independent symbols are multiplexed over a MIMO channel matrix $\tilde{\mathbf{H}}(p_k)$. This MIMO channel has its structure from the Alamouti STC and the choice of a specific pattern p_k.

Seeing SMAL as a spatial multiplexing scheme, we choose the pattern p_{k_0} in order to maximise the lower bound of the performance. According to (4.1) this is done by choosing the pattern p_{k_0} satisfying

$$p_{k_0} = \arg\left(\max_{p_k}\left[\, d^2_{min,SM}\left(\tilde{\mathbf{H}}(p_k)\right)\,\right]\right), \tag{4.23}$$

4.3.3 Instantaneous channel vs long-term statistics

In the following sections, we will find optimisation criteria for two scenarios; when we have access to instantaneous channel information and when only long-term correlation statistics are known. The expression for $d^2_{min,SM}\left(\tilde{\mathbf{H}}(p_k)\right)$ is different for the two cases, and in the latter it is only an approximation based on "average behaviour" of the channel.

In the common case of frequency division duplex (FDD) systems, the transmission from the transmitter to the receiver is carried out over one frequency band, while another band is used for communication in the other direction. When instantaneous channel state information is required at the transmit side, the use of feedback information from the receiver is necessary. This has the disadvantage of occupying channel bandwidth.

In time division duplex (TDD) systems, two-way communication is carried out over the same frequency band in both directions. This means the propagation channel matrices seen from the two sides are each others transpose. If the channel may be assumed constant for a long enough period, the channel estimated from incoming data may also be used when transmitting a reply. Hence, there is no need for feedback when using TDD.

As FDD systems are more commonly used than TDD systems, we assume that required channel information at the receiver implies the

need for feedback information, in which case it occupies valuable channel bandwidth.

In the case of the SMAL schemes, the the version based on instantaneous channel state information will require feedback information, while the correlation-based approach can do without it as we expect the long-term correlation statistics to change very slowly.

We expect the former to achieve a BER-performance at least as good as the latter, but must include the drawback of having to feedback channel information. We observe that there is a trade-off between performance and complexity that must be taken into account when evaluating the results.

4.3.4 Instantaneous pattern optimisation

Assume instantaneous channel information is available at the transmitter, that is; we know the channel $\mathbf{H}$ and may generate the space-time coded $\tilde{\mathbf{H}}(p_k)$. Then we may find p_{k_0} by substituting (4.1) into (4.23). We choose p_{k_0} such that

$$\begin{aligned} p_{k_0} &:= \arg\left(\max_{p_k}(\sigma^2_{min}(\tilde{\mathbf{H}}(p_k))\frac{d^2_{min}}{N})\right) \\ &:= \arg\left(\max_{p_k}(\sigma^2_{min}(\tilde{\mathbf{H}}(p_k)))\right), \end{aligned} \tag{4.24}$$

where $\sigma_{min}(\tilde{\mathbf{H}}(p_k))$ is the minimum singular value of the space-time coded channel $\tilde{\mathbf{H}}(p_k)$.

In other words, we choose the pattern p_{k_0} that offers the largest squared minimum singular value of all the possible $\tilde{\mathbf{H}}(p_k)$. That the minimum singular value is relatively large secures some quality even in the most heavily faded channel.

For each update in the channel state information, the apply the criterion in (4.24) to select a new pattern p_{k_0}. This is done by building the space-time coded channel matrix $\tilde{\mathbf{H}}(p_k)$ for all $k \in [1, P_N]$, performing a singular value decomposition (SVD) on it and then applying the criterion to find the best pattern p_{k_0}. We refer to this as the *instantaneously optimised* or *instantaneous* SMAL.

4.3.5 Performance evaluation of the instantaneous SMAL-version

Through simulations, we have obtained separate BER performance-results for all the P_N patterns, assumed fixed over time. For the same settings, we have also tested the instantaneously optimised SMAL and a non-optimised scheme based on a random pattern choice are given in the same way. The latter picks a random pattern for each new realisation of the channel matrix, yielding a performance equal to the average

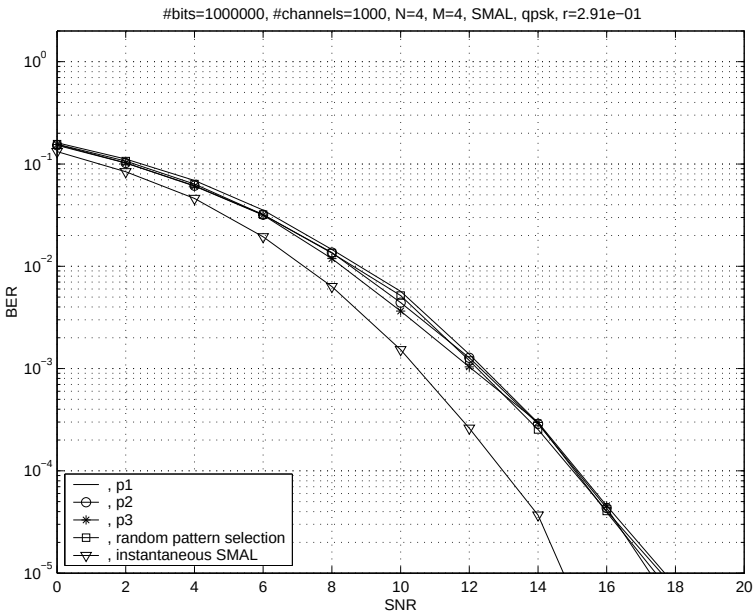

Figure 4.3: BER results for 3 patterns fixed over time, random pattern selection and the instantaneous SMAL, $N = M = 4$, $r = 0.29$

of all the patterns p_k, $k \in [1, P_N]$, where we remember that P_N is the number of non-trivially equivalent patterns for a given N. For example, given 4 transmit antennas, the random approach gives the same result as averaging the performance of patterns p_1, p_2 and p_3.

We consider two different levels of correlation, one low and one high. The neighbouring element correlation coefficients, see (2.39) for a definition, are given as $r = 0.29$ and $r = 0.90$. Also, we employ two sets of antenna arrays, both $N = M = 4$ and $N = M = 6$.

For the $N = M = 4$ case, there are only $P_4 = 3$ patterns, and we plot them together with the random and the instantaneous SMAL, all 5 curves in one figure.

On the other hand, $N = M = 6$ yields $P_6 = 11$ patterns. Hence, in that case we rather depict only the instantaneous SMAL and the random pattern selection. For reference, the respective BER results of the $P_6 = 11$ fixed patterns are moved to appendix A.3. The plot for low correlation is given in figure A.1, while the high correlation results is shown in figure A.2.

The results are generated from simulations using $N = M = 4$, a block size of 10^6 bits per channel iteration, averaged over 1000 channel realisations. The bits are modulated to complex symbols using the QPSK modulation scheme.

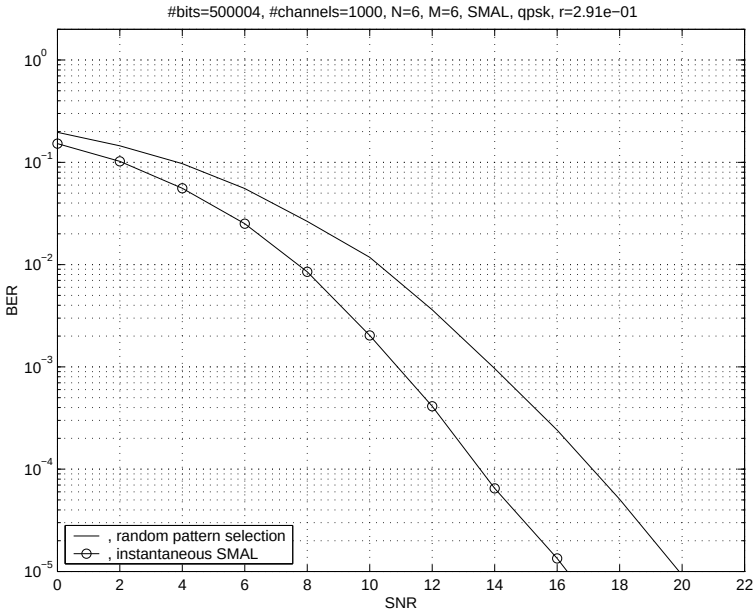

Figure 4.4: BER results for random pattern selection and the instantaneous SMAL, $N = M = 6$, $r = 0.29$

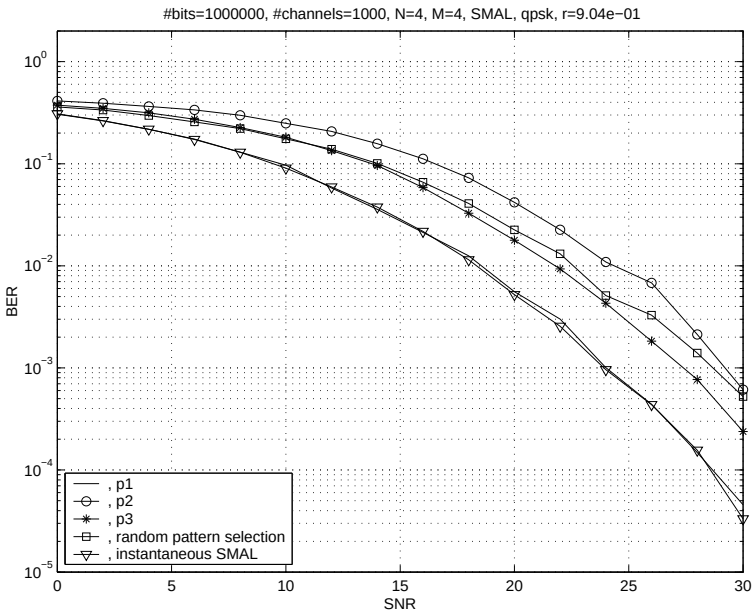

Figure 4.5: BER results for 3 patterns fixed over time, random pattern case and the instantaneous SMAL, $N = M = 4$, $r = 0.90$

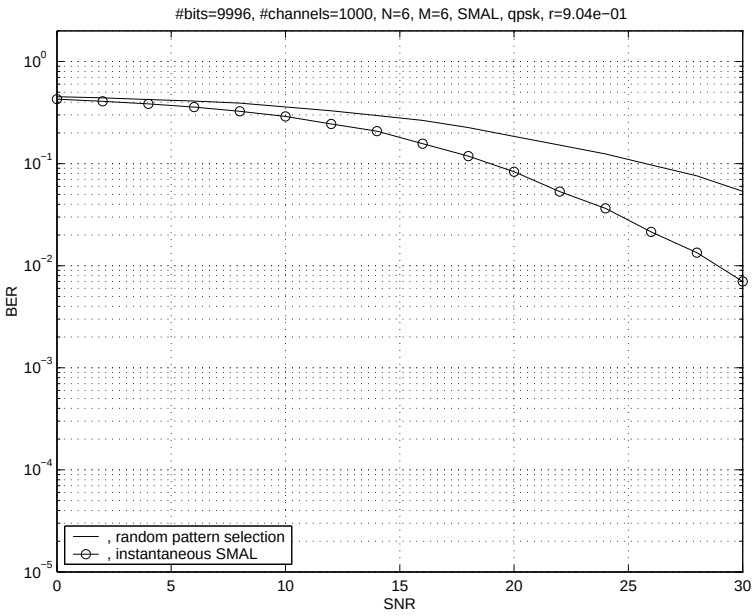

Figure 4.6: BER results for random pattern selection and the instantaneous SMAL, $N = M = 6$, $r = 0.90$

By referring to the legends in the figures we observe that the optimal approach achieves the lowest BER in both cases, as was expected. However, the degree of improvement over the performance curves of the fixed pattern differs:

Low-level correlation, r=0.29

BER-results for the case of low correlation $r = 0.29$ is plotted against the SNR in figures 4.3 and 4.4, for $N = M = 4$ and $N = M = 6$, respectively.

We observe that all three patterns for $N = M = 4$ yield approximately the same results, hiding the average curve between them. It is also possible to refer to the $N = M = 6$ settings, figure A.1 in the appendix, which confirms this observation.

For both antenna settings, the time optimised case is better than the random approach and all the patterns. We compare the former two; the time optimised SMAL and the random pattern selection. The improvement of the former over the latter at a target bit-error rate of 10^{-4} is about 2 dB with $N = M = 4$ and over 3 dB with $N = M = 6$.

This tells us that there is a significant potential for improvement using the instantaneous SMAL scheme for low correlation, especially as the gain offered by the separate patterns is much less, even for the best patterns.

High-level correlation, r=0.90

Figures 4.5 and 4.6 show results given the same two antenna settings, but for a high level of correlation $r = 0.90$.

In the case of heavily correlated fading, we observe that the bit-error rates for all the curves are higher than for low correlation, as expected. For $N = M = 4$, the instantaneous SMAL is about 5 dB better than the random SMAL at a BER of 10^{-3}, and we expect a similar result at the target BER of 10^{-4}.

However, we also see that the instantaneous SMAL has a performance curve almost equivalent to that of pattern p_1, telling us that this is almost always the pattern, for every new channel realisation. The reason is that long-term correlation structure takes over possible short-term variations and intantaneous correlation.

For $N = M = 6$, we observe that the instantaneous SMAL has a much steeper slope than the random pattern, leading it towards lower BER-values faster.

In figure 4.5 (and A.2 for reference), we observe a greater spread of BER curves for the P_N fixed patterns, the results are not similar as they are for low correlation. This means that some patterns yield much better BER performance than others, in some cases close to the instantaneous SMAL version.

We conclude that for both high- and low-level correlation, the instantaneous SMAL scheme performs better than using any of the patterns fixed over time. However, we observe that the instantaneous SMAL is especially useful at low correlation levels, while at high correlation levels choosing the best fixed pattern works just as well.

The question then becomes:

- **How do we decide which pattern is the best over time?**

The idea is that with such knowledge we may transmit with this best pattern fixed over time, regardless of instantaneous channel state information. This question is of highest importance when the average correlation is strong.

In fact, the SMAL pattern optimization based on instantaneous channel state information may not always be the most sensible approach, when both performance and complexity are considered. On this background, we move on to the task of developing a criterion for choosing the best pattern based on long-term statistics.

4.3.6 Pattern optimisation based on correlation

This section is devoted to finding the best pattern based only on long-term statistics, such as the average correlation. Our approach is to base

the optimisation on the 'average behaviour' of the channel. We will refer to this approach as the *correlation-based* SMAL.

We quickly note that to replace the exact expression for $\tilde{\mathbf{H}}$ with an average value is not rewarding. We remember that the channel coefficients of $\mathbf{H}_0$ are complex, random numbers with zero mean. The same goes for the coefficients of $\mathbf{H}$ and $\tilde{\mathbf{H}}(p_k)$, so the expected value of $\tilde{\mathbf{H}}(p_k)$ is a zero-matrix. So, in the following, we attempt other ways of finding which pattern can be expected to yield the maximum σ^2_{min}.

A singular value decomposition (SVD) allows a decomposition of $\tilde{\mathbf{H}}(p_k)$ into

$$\tilde{\mathbf{H}}(p_k) = \mathbf{U}\Sigma\mathbf{V}^H \tag{4.25}$$

Because $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, it is clear that

$$\begin{aligned} \tilde{\mathbf{H}}(p_k)^H\,\tilde{\mathbf{H}}(p_k) &= \mathbf{V}\Sigma^2\mathbf{V}^H \\ \tilde{\mathbf{H}}(p_k)\,\tilde{\mathbf{H}}(p_k)^H &= \mathbf{U}\Sigma^2\mathbf{U}^{\mathbf{H}}, \end{aligned} \tag{4.26}$$

which are two eigenvalue decompositions, rather than SVDs. The diagonal matrix Σ^2 contains the common eigenvalues of $\tilde{\mathbf{H}}(p_k)^H\tilde{\mathbf{H}}(p_k)$ and $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H$, which are identical to the squared singular values of $\tilde{\mathbf{H}}(p_k)$. Evidently, the squared minimum singular value we are interested in may also be found from the these eigenvalue decompositions, and the evaluation in (4.24) yields the same result as

$$\begin{aligned} p_{k_0} &:= \arg\max_{p_k}\left[\lambda_{min}\left(\tilde{\mathbf{H}}(p_k)^H\tilde{\mathbf{H}}(p_k)\right)\right] \\ &:= \arg\max_{p_k}\left[\lambda_{min}\left(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H\right)\right] \end{aligned} \tag{4.27}$$

that is; maximising the minimum eigenvalue of either $\tilde{\mathbf{H}}(p_k)^H\tilde{\mathbf{H}}(p_k)$ or $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H$.

However, the assumption is now that we do not have access to any instantaneous channel matrices, so we rather attempt to maximise the minimum eigenvalues of *the averages* of $\tilde{\mathbf{H}}(p_k)^H\tilde{\mathbf{H}}(p_k)$ or $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H$, as given by the expected value. These averages are not necessarily equal, and we may choose p_{k_0} such that

$$p_{k_0} := \arg\max_{p_k}\left[\lambda_{min}\left(E\left(\tilde{\mathbf{H}}(p_k)^H\tilde{\mathbf{H}}(p_k)\right)\right)\right] \tag{4.28}$$

or, a possibly different pattern by using

$$p_{k_0} := \arg \max_{p_k} \left[\lambda_{min} \left(E\left(\tilde{\mathbf{H}}(p_k) \tilde{\mathbf{H}}(p_k)^H \right) \right) \right]. \tag{4.29}$$

To evaluate the criteria in (4.28) and (4.29), we use the expression for the combined channel matrix $\tilde{\mathbf{H}}(p_k)$, given in (4.19).

Before continuing on to the evaluation of expectation, it is useful to determine the expected value of some products that will be needed later. Two independent, circular complex Gaussian random coefficients h_1 and h_2 from H_0, have mean μ and standard deviation σ given as

$$\begin{aligned}
(h_1, h_2) &\sim \mathcal{CN}(\mu, \sigma_h) = \mathcal{CN}(0,1) \\
\\
E(h_1 h_2) &= E(h_1)E(h_2) = 0 \\
E(h_1 h_1) &= E(\Re\{h\}^2 + i \cdot \Re\{h\}\Im\{h\} - \Im\{h\}^2) \\
&= \frac{1}{2}\sigma_h^2 + 0 - \frac{1}{2}\sigma_h^2 = 0 \\
E(h_1^* h_1) &= E(\,|h_{ij}|^2\,) = E(\Re\{h\}^2 + \Im\{h\}^2) \\
&= \sigma_h^2 = 1
\end{aligned} \tag{4.30}$$

The results above assumes that the coefficients are complex numbers, where the real and the imaginary parts are independent, but have the same variance. The conclusion is that the only non-zero contribution arises when we look at the expectation of the product of a coefficient with its own complex conjugate; the squared complex norm.

The expected values of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)$ and $\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)$

In this section, we study the expectations

$$\begin{aligned}
&E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)) \quad \text{and} \\
&E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)),
\end{aligned} \tag{4.31}$$

and express them as functions of the correlation, which is assumed to be known at the transmit side.

The idea is to perform an eigenvalue decomposition on the result. It is worth noting that even though the eigenvalues of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)$ and $\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)$ are the same, so that (4.27) holds, the eigenvalues of the two expressions in (4.31) are not necessarily equal.

In fact, we will find that the evaluation of the first expression

$$E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)) \tag{4.32}$$

is fruitless for the SMAL scheme. The reason is that (4.32) is found not to depend on the choice of pattern, which makes it useless as a selection criterion. The development of this result is moved to appendix (A.2).

However, $E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k))$ yields a useful result, given in lemma 1. Next, the derivation is shown. We then study the obtained criterion and compare it to the one based on instantaneous channel information.

Lemma 1 *The expectation* $E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k))$ *is given as*

$$E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)) = M\left(\mathbf{I}(p_k^1)^T\mathbf{R}_t\mathbf{I}(p_k^1) + \mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{R}_t\mathbf{I}(p_k^2)\mathbf{G}\right)$$

Derivation

Before evaluating the expected value of $\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)$, we simplify the expression. We recall (4.19), and use it to substitute for $\tilde{\mathbf{H}}(p_k)$, obtaining

$$\begin{aligned}\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k) &= \left(\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\right)^H \cdot \\ &\quad \left(\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\right) \\ &= \left(\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}^T\mathbf{H}_0^H\sqrt{\mathbf{R}_r}^T\mathbf{I}_{r1}^T + \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}^T\mathbf{H}_0^T\sqrt{\mathbf{R}_r}^T\mathbf{I}_{r2}^T\right) \cdot \\ &\quad \left(\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\right)\end{aligned} \tag{4.33}$$

From section 3.2, we recall that the correlation matrices $\mathbf{R}_t$ and $\mathbf{R}_r$ are symmetric. Simple linear algebra will show that this also apply to the square roots, so that

$$\sqrt{\mathbf{R}_t}^T = \sqrt{\mathbf{R}_t}, \quad \text{and} \quad \sqrt{\mathbf{R}_r}^T = \sqrt{\mathbf{R}_r}$$

By using this result to rewrite (4.33), we arrive at

$$\begin{aligned}\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k) = &\left(\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T\right) \\ &\left(\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\right)\end{aligned}$$

$$\begin{aligned}= &\ \mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \\ &\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T\mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G} + \\ &\mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1) + \\ &\mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T\mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\end{aligned} \tag{4.34}$$

The matrices $\mathbf{I}_{r1}$ and $\mathbf{I}_{r2}$ are not influenced by the choice of pattern, they are always as shown in equation (4.20). By studying these matrices, it is clear that the following identities hold:

$$\mathbf{I}_{r1}^T\mathbf{I}_{r1} = \mathbf{I}_{r2}^T\mathbf{I}_{r2} = \mathbf{I}, \quad \text{and} \quad \mathbf{I}_{r1}^T\mathbf{I}_{r2} = \mathbf{I}_{r2}^T\mathbf{I}_{r1} = 0$$

By incorporating this into the expression for $\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(\mathbf{p_k})$, the second and third parts of the sum equals 0, and the simplified result is given by

$$\begin{aligned}\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k) =& \mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)+ \\ & \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\mathbf{R}_r\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\end{aligned} \tag{4.35}$$

This approach seems to hold some promise, so we move on to the task of studying the expected value.

$$\begin{aligned}E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)) &= E\Big(\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)+ \\ &\qquad \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\mathbf{R}_r\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\Big) \\ &= \mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\,E(\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0)\,\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)+ \\ &\quad \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\,E(\mathbf{H}_0^T\mathbf{R}_r\mathbf{H}_0^*)\,\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G} \\ &= \mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\,E(\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0)\,\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)+ \\ &\quad \mathbf{G}^T\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\,E\Big((\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0)^T\Big)\,\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\end{aligned} \tag{4.36}$$

Now, $E(\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0)$ and $E\Big((\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0)^T\Big)$ are the only non-deterministic factors left in the expression. We evaluate the matrix product $\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0$, of dimensions $N\times N$. The entry in the kth column and the lth row of the correlation matrix $\mathbf{R}_r$ is denoted r_{kl}. With this notation, the coefficient in the mth row and nth column of $\mathbf{H}_0^H\mathbf{R}_r\mathbf{H}_0$ may be written as the sum

$$\sum_{k=1}^{M} h_{km}^*\Big(\sum_{l=1}^{M} r_{kl}h_{ln}\Big) = \sum_{k=1}^{M}\sum_{l=1}^{M} r_{kl}h_{km}^*h_{ln}.$$

From (4.30), we know that the expectation of two independent channel coefficients is zero, so the only cases for which the above expression has a non-zero expected value is when $k = l$ and $m = n$, i.e. along the main diagonal. Recalling that $E(h_{ij}^*h_{ij}) = 1$, we see that

$$E\Big(\sum_{k=1}^{M}\sum_{l=1}^{M} r_{kl}h_{km}^*h_{ln}\Big) = \sum_{k=1}^{M} r_{kk}E(h_{kk}^*h_{kk}) = M$$

The last equality follows from the fact that $\mathbf{R}_r$ is a symmetric matrix, and that the values the main diagonal are all equal to 1.

After this, the whole expression from (4.36) may be rewritten to the simpler form

$$E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k)) = M\left(\mathbf{I}(p_k^1)^T\mathbf{R}_t\mathbf{I}(p_k^1) + \mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{R}_t\mathbf{I}(p_k^2)\mathbf{G}\right), \quad (4.37)$$

where all factors are deterministic and may be evaluated without any channel realisations. This is the postulated result in Lemma 1, and the derivation is concluded. ■

Performance criterion of correlation-based SMAL

Now, the criterion of the correlation-based SMAL scheme, presented in (4.29), may be rewritten. Given the result in (4.37), we pick the pattern p_{k_0} that satisfies

$$p_{k_0} := \arg\max_{p_k}\left[\lambda_{min}\left(\mathbf{I}(p_k^1)^T\mathbf{R}_t\mathbf{I}(p_k^1) + \mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{R}_t\mathbf{I}(p_k^2)\mathbf{G}\right)\right] \quad (4.38)$$

We note that the choice of pattern depends on the uncorrelated channel matrix $\mathbf{H}_0$, the transmit correlation in $\mathbf{R}_t$ and the antenna assignment pattern p_k, the latter through the matrices $\mathbf{I}(p_k^1)$, $\mathbf{I}(p_k^2)$ and $\mathbf{G}$.

In (4.37), the contribution from the correlation on the receive side is only a constant factor M, and does not change the choice of pattern. Therefore, it is not included in (4.38).

That the receive side is unimportant in the choice of the best SMAL transmit pattern intuitively seems correct, as the assignments take place on the transmit side of the channel. The receiver registers a sum of all the transmitted signals, regardless of how they were mapped to the antennas before transmission.

4.3.7 Performance evaluation of the correlation-based SMAL

Selection of best pattern

Given the criterion in (4.38), we choose to space-time code our symbols according to the pattern with the largest minimum eigenvalue of $E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k))$ for $k \in [1, P_N]$.

This means the behaviour of the correlation-based SMAL follows the ordering of these minimum eigenvalues, which we plot against an increasing correlation level, one curve for each pattern. Figure 4.7 depicts the $N = 4$ case while the $P_6 = 11$ curves in figure 4.8 represent the

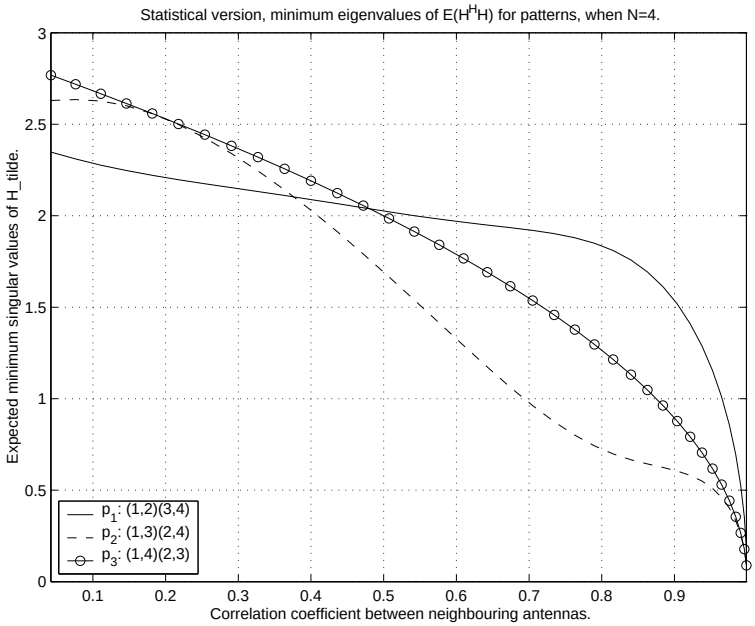

Figure 4.7: Minimum singular values of $E(\tilde{\mathbf{H}}^H(p_k)\mathbf{H}(p_k))$, for $k = 1, 2, 3$ and $N = 4$

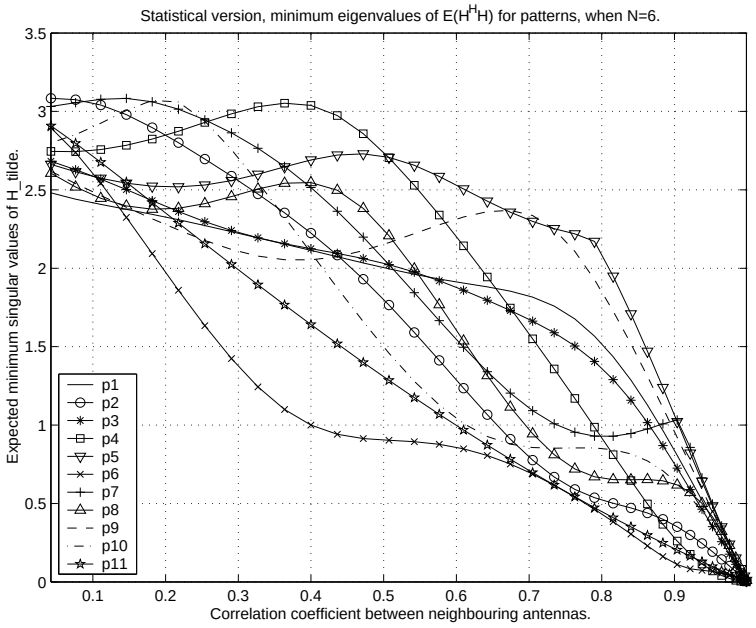

Figure 4.8: Minimum singular values of $E(\tilde{\mathbf{H}}^H(p_k)\mathbf{H}(p_k))$, for $k = 1, \ldots, 11$ and $N = 6$

patterns when $N = 6$. The correlation-based SMAL is, as noted earlier, independent of the number of receive antennas.

We consider the correlation-based SMAL at our chosen levels of correlation, namely $r = 0.29$ and $r = 0.90$, for the case of 4 transmit antennas. At $r = 0.29$, the chosen pattern is p_3, while p_1 is the best at $r = 0.90$.

The case of 6 transmit antennas yields a more crowded plot, because there are now 11 non-trivially equivalent patterns to choose from. All the same, some patterns still yield higher eigenvalues of $E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k))$ than others. At $r = 0.29$, pattern p_4 is chosen as the best. Although it is hard to see clearly, magnification of the figure shows that p_5 is the best at $r = 0.9$.[1]

We observe that for very high or low levels of correlated fading, the different patterns yield more similar results. At low correlation, there is little difference between the patterns because the entries of the uncoded channel matrix $\mathbf{H}$ are independent and spatial transmit patterns are irrelevant. The minimum eigenvalues under such conditions are relatively high. At a high level of correlation, the ill-conditioned $\mathbf{H}$ results in very low minimum eigenvalues of $E(\tilde{\mathbf{H}}^H(p_k)\tilde{\mathbf{H}}(p_k))$.

First, by referring to figures 4.3, 4.5, A.1 and A.2 we observe that the correlation-based SMAL really does pick the best pattern at the correlation levels $r = 0.29$ and $r = 0.90$, with respect the BER.

Performance gain of correlation-based SMAL

Next, we consider the gain achieved when using the instantaneous SMAL compared to the previously mentioned random pattern selection, performed for each channel realization.

Low-level correlation, r=0.29

For $r = 0.29$, a low level of correlated fading, we show the performance gains of correlation-based SMAL over random pattern selection in figure 4.9 for $N = M = 4$ and in figure 4.10 for $N = M = 6$.

We observe that in the former case, there is no difference between the correlation-based SMAL and the random approaches. This is reasonable because we recall from figure 4.3 that all three patterns (and their average) yield the same results.

When $N = M = 6$, the gain is larger, but no more than 2 dB at the target BER of 10^{-4}.

High-level correlation r=0.90

In the case of $r = 0.90$, we show the performance gains of correlation-based SMAL over random pattern selection in figure 4.11 for $N = M = 4$ and in figure 4.12 for $N = M = 6$.

[1] This is verified in a close-up shown in figure A.3.

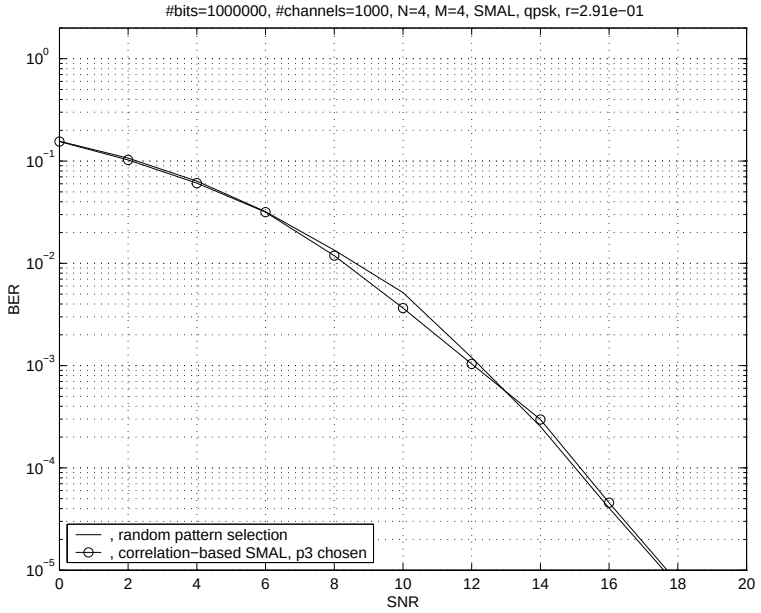

Figure 4.9: BER results for random pattern and correlation-based SMAL (pattern p_3 chosen), $N = M = 4$, $r = 0.29$

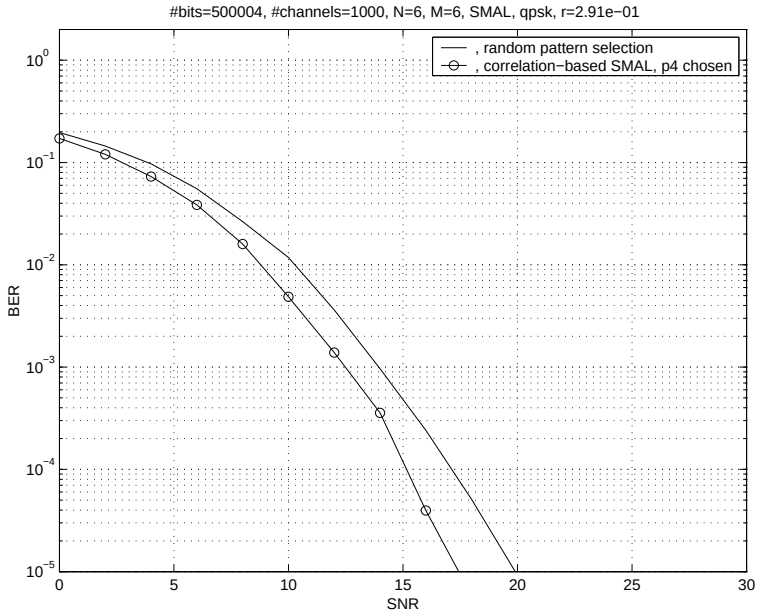

Figure 4.10: BER results for random pattern and correlation-based SMAL (pattern p_4 chosen), $N = M = 6$, $r = 0.29$

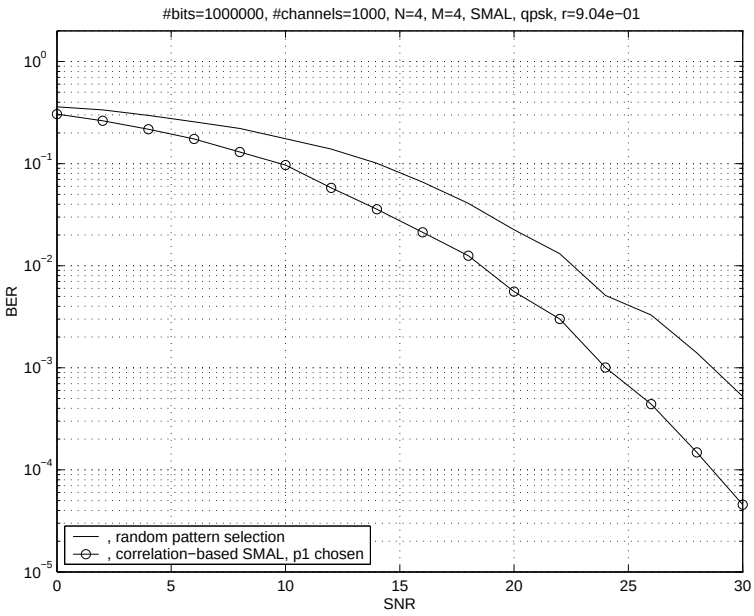

Figure 4.11: BER results for random pattern and correlation-based SMAL (pattern p_1 chosen), $N = M = 4$, $r = 0.90$

In general, we observe a greater gain here, although the overall performance has deteriorated due to increased correlation.

For $N = M = 4$, pattern p_1 is chosen and the gain at BER = 10^{-3} is between 4 and 5 dB. We expect similar results at the target BER of 10^{-4}.

When $N = M = 6$, the correlation-based SMAL transmits with the pattern p_5 and while it is hard to evaluate any gain properly at such high BER, we observe that the best pattern is a lot better than the average, and also with a steeper curve down towards lower BER-results.

4.3.8 Comparison between the instantaneous and the correlation-based SMAL

From the previous sections on instantaneous and correlation-based SMAL, we have developed models and studied the performance gains of two ways to decide which is the best transmit pattern. Now is the time to compare the schemes, and we recall the performance/complexity tradeoff discussed in section 4.3.3. We expect the instantaneous SMAL to outperform the correlation-based version, but at the cost of increased complexity and channel bandwidth being occupied to feedback channel state information to the transmitter.

Low-level correlation, r=0.29

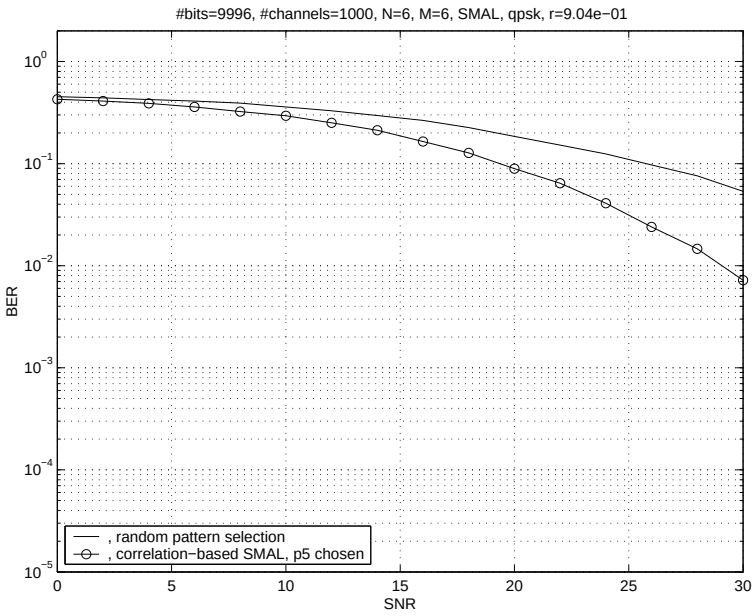

Figure 4.12: BER results for random pattern and correlation-based SMAL (pattern p_5 chosen), $N = M = 6$, $r = 0.90$

In figures 4.16 (for $N = M = 4$) and 4.16 (for $N = M = 6$), we observe that the instantaneous SMAL is clearly better than the correlation-based version for both sets of antennas. Both for $N = M = 4$ and $N = M = 6$ the gain of using the instantaneous approach is approximately 2 dB at the target BER 10^{-4}.

High-level correlation, r=0.90

When the level of correlated fading is high, there is not much gain to achieve from using the instantaneous SMAL. This is seen from figure 4.15 for $M = N = 4$ and figure 4.16 for $M = N = 6$.

In fact, the performance curves for $r = 0.90$ are identical; the best pattern over time (chosen by the correlation-based SMAL) has the same BER-development as the instantaneous approach. For $M = N = 4$, we recall from figure 4.5 that the three patterns have very different performance.

For the case when $M = N = 6$, the reference curves in figure A.2 show that it is not only the chosen pattern p_5 that has a low BER. A few other curves lie quite close to the one for p_5, and using the corresponding patterns will also yield good performance, if not as good as with pattern p_5.

We conclude that there is no gain in using the instantaneous SMAL

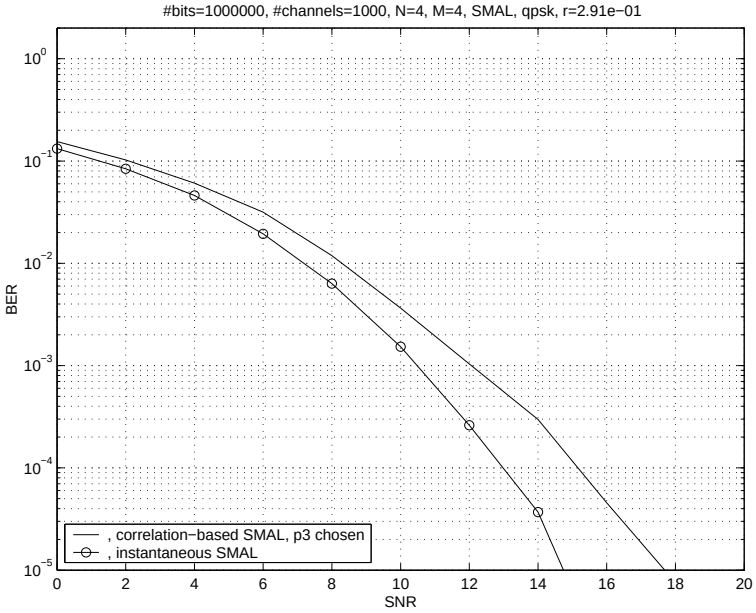

Figure 4.13: BER results for instantaneous SMAL and correlation-based SMAL (pattern p_3 chosen), $N = M = 4$, $r = 0.29$

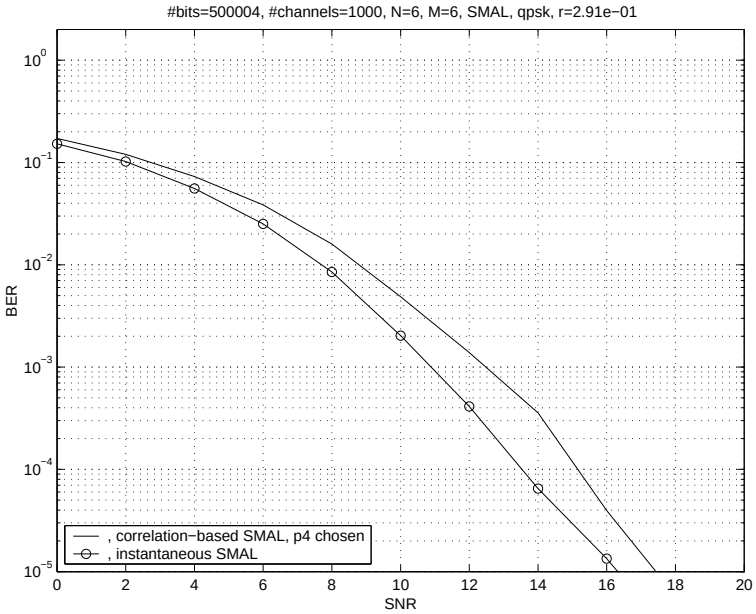

Figure 4.14: BER results for instantaneous SMAL and correlation-based SMAL (pattern p_4 chosen), $N = M = 6$, $r = 0.29$

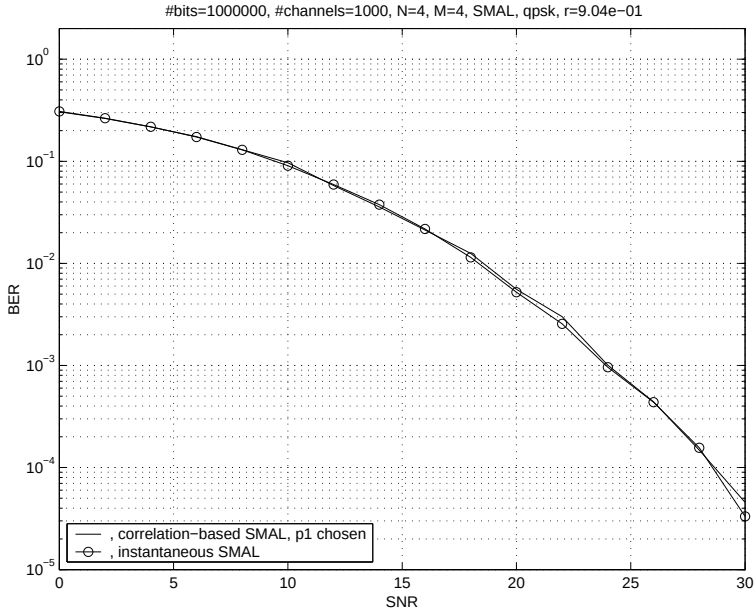

Figure 4.15: BER results for instantaneous SMAL and correlation-based SMAL (pattern p_1 chosen), $N = M = 4$, $r = 0.90$

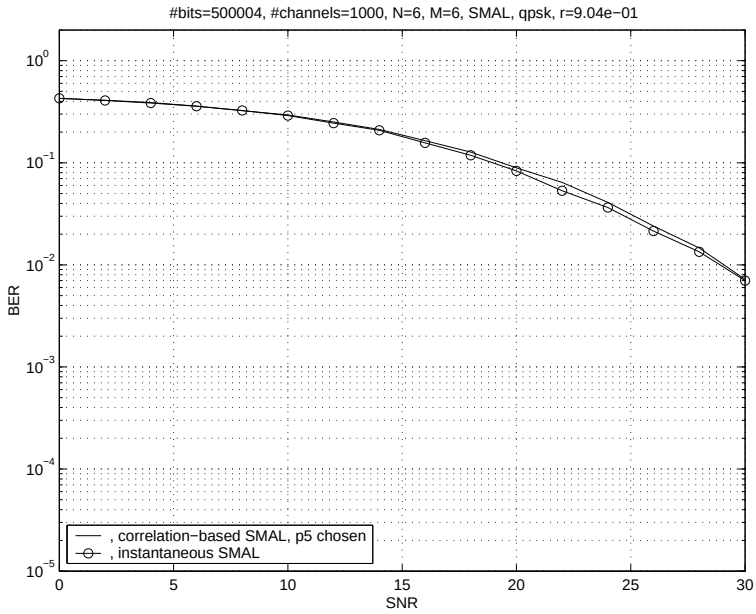

Figure 4.16: BER results for instantaneous SMAL and correlation-based SMAL (pattern p_5 chosen), $N = M = 6$, $r = 0.90$

in the case of heavily correlated fading channels. Then, computational capacity is better spent elsewhere than on keeping track of instantaneous channel state information. With knowledge of long-term correlation statistics, equal performance gain over the random pattern selection is achieved by choosing the best pattern over time.

Chapter 5

Conclusion

MIMO systems are used to increase both the capacity and the quality of wireless communication. Various algorithms presented here describe different ways in which to exploit the spatial dimension offered. These MIMO algorithms are often defined as belonging to one of two categories; either diversity-oriented transmission through space-time coding or spatial multiplexing.

In MIMO diversity schemes the multiple antennas are used to combat Rayleigh fading, in order to increase the quality as measured by the BER. Spatial multiplexing transmits independent sub-streams of data on each antenna, thereby increasing the data rate in a direct way.

So far, these approaches have been considered as competing, and we are only beginning to understand the trade-offs between them. We know that diversity schemes yield diminishing returns when the number of antennas is increased, and that spatial multiplexing with a simple receiver lags in performance due to a lack of diversity. Through plots of BER performance, we have confirmed this knowledge, which calls for simultaneous use of both approaches on the same array.

Furthermore, we have also seen that the two approaches react differently to correlated fading, in which case the BER performance of spatial multiplexing breaks down more abruptly than that of the diversity schemes. Because the level of correlated fading is assumed to differ between antenna pairs in a practical array, this also calls for a specific solution to combine multiplexing and diversity scheme over the transmit array.

We address this call by presenting the SMAL scheme, which combines Alamouti STC and SM in space, essentially by multiplexing independent blocks of Alamouti space-time coded symbols.

SMAL is developed in two versions, both for the case when instantaneous channel state information is known at the transmitter and the case when only long-term correlation statistics are available. The common

idea of both is to transmit using the best of a number of antenna assignment patterns, each representing a different mapping of the Alamouti blocks to the antennas.

We have shown that both versions offer significant performance improvements over an approach that picks a random antenna assignment pattern.

The instantaneously optimised SMAL chooses the best transmit pattern for every new channel realisation. This approach is shown to be especially useful for low levels of correlated fading. Given a neighbouring correlation level of $r = 0.29$ and $N = M = 4$, the instantaneous SMAL achieves a performance gain of 2 dB over the random pattern selection at a target BER of 10^{-4}. In the case of $N = M = 4$, the gain is found to be over 3 dB.

When correlation is strong, the correlation-based SMAL works just as well as the instantaneous approach, and their BER performance-curves are very similar. The reason is that long-term correlation statistics dominates over short-term variations.

For our tested levels of correlated fading, $r = 0.29$ and $r = 0.90$, the statistical SMAL-version has been shown to select the best transmit pattern. For $r = 0.90$ and a BER-level of 10^{-3} the gain of the correlation-based SMAL over the random pattern selection is almost 5 dB for $N = M = 4$, and we expect a similar result at the target BER of 10^{-4}. The improvement is substantial also for a system with $N = M = 6$.

We conclude that our proposed combination of SM and STC, the SMAL scheme, offers substantial gain when transmission is done using the best spatial pattern, either instantaneous or over time. This is particularly interesting under conditions of heavy correlation, when the pattern optimisation offers ample gain based on long-term statistics alone.

5.1 Open problems

One obvious extension to the proposed SMAL is to develop a systematic way of optimising the scheme. As the number of antennas increases, so does the number of possible patterns, and it becomes too computationally complex to find the best by investigating all of them. By being able to determine which patterns are worth examination and which are not, we could reduce the complexity substantially.

Another idea would be to to generalise the scheme, and optimise with respect to the transmit data rate as well as the antenna assignment patterns. Based on either instantaneous or statistical channel knowledge, such a generalisation could find the best combination of schemes (pure SM, a combination of SM and SMAL or others) and the best antenna map-

ping. This involves an obvious increase in complexity, which must be evaluated against the achievable gain.

Bibliography

[1] S. M. Alamouti. “A simple transmit diversity technique for wireless communications”. *IEEE Journal on Selected Areas in Comm.*, 16(8), 1998.

[2] J. Bach Andersen. “Array Gain and Capacity for Known Random Channels with Multiple Element Arrays at Both Ends”. *IEEE Journal on Selected Areas in Comm.*, 18(11):2172–2178, Nov. 2000.

[3] H. Bolcskei, D. Gesbert, and A. Paulraj. “On the capacity of wireless systems employing OFDM-based spatial multiplexing”. *IEEE Trans. on Comm.*, February 2002.

[4] G. D. Golden and G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky. “Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture”. *Electronics Letters*, 35(1):14–15, January 1999.

[5] G. J. Foschini and M. J. Gans. “On limits of wireless communications in a fading environment when using multiple antennas”. *Wireless Personal Communications*, 6(3):311, March 1998.

[6] J. Foschini. “Layered Space-time architecture for wireless communications in a fading environment”. *Bell Labs Technical Journal*, 1(2):41–59, 1996.

[7] D. Gesbert and J. Akhtar. “Breaking the barriers of Shannon’s capacity: An overview of MIMO wireless system”. *Telektronikk Telenor Journal*, January 2002.

[8] D. Gesbert, M. Shafi, D. Shiu, and P. Smith. “From theory to practice: An overview of space-time coded MIMO wireless systems”. *IEEE Journal on Selected Areas in Comm. (JSAC)*, 21(3):281–302, April 2003.

[9] D. Gore, R.W. Heath, and A.J. Paulraj. “Performance analysis of spatial multiplexing in correlated channels”. *IEEE Trans. on Communication*, submitted 2002.

[10] J. Salz (Member IEEE) and J. H. Winters (Senior Member IEEE). "Effect of Fading correlation on Adaptive Arrays in Digital Mobile Radio". *IEEE Transactions on vehicular technology*, 43(4):1049–1057, November 1994.

[11] S. Loyka (Member IEEE) and G. Tsoulos (Member IEEE). "Estimating MIMO system Performance Using the Correlation Matrix Approach". *IEEE Communications Letter*, 6(1):19–21, January 2002.

[12] R.W. Heath Jr. and A. J. Paulraj. "Diversity versus Multiplexing in Narrowband MIMO Channels: A Tradeoff Based on Euclidean Distance.". *IEEE Trans. on Communication*, submitted 2002.

[13] G. Lebrun, T. Ying, and M. Faulkner. "MIMO transmission over a time-varying TDD channel using SVD". *Electronics Letters*, 2001.

[14] J. G. Proakis. *Digital Communications*. McGraw-Hill Science/Engineering/Math, fourth edition, 2000.

[15] T. S. Rappaport. *Wireless communications*. Prentice Hall PTR, second edition, 2002.

[16] P. M. Shankar. *Introduction to Wireless Systems*. John Wiley Sons, Inc., 2002.

[17] H. Skjevling, D. Gesbert, and N.D. Christophersen. "Combining Space-Time Block Codes and Multiplexing in Correlated MIMO channels: An Antenna Assignment strategy". NORSIG-2003, submitted 2003.

[18] B. Sklar. "Rayleigh Fading Channels in Mobile Digital Communication Systems. Part 1: Characterization". *IEEE Communications Magazine*, pages 90–100, July 1997.

[19] V. Tarokh, H. Jafarkhani, and A. R. Calderbank. "Space-time block codes from orthogonal designs". *IEEE Trans. on Information Theory*, 45(5):1456–1467, July 1999.

[20] V. Tarokh, N. Seshadri, and A. R. Calderbank. "Space-time codes for high data rate wireless communication: Performance criterion and code construction". *IEEE Trans. on Information Theory*, 44(2):744–765, March 1998.

[21] H. L. Van Trees. *Optimum Array Processing*. John Wiley Pub., 2002.

[22] 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". In *Proc. ISSSE-98, Pisa, Italy*, September 1998.

Appendix A

Miscellaneous

A.1 SMAL channel matrices, $N = 6$

$$\tilde{\mathbf{H}}(p_1) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{h}_5 & \mathbf{h}_6 \\ \mathbf{h}_2^* & -\mathbf{h}_1^* & \mathbf{h}_4^* & -\mathbf{h}_3^* & \mathbf{h}_6^* & -\mathbf{h}_5^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_2) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_5 & \mathbf{h}_4 & \mathbf{h}_6 \\ \mathbf{h}_2^* & -\mathbf{h}_1^* & \mathbf{h}_5^* & -\mathbf{h}_3^* & \mathbf{h}_6^* & -\mathbf{h}_4^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_3) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_6 & \mathbf{h}_4 & \mathbf{h}_5 \\ \mathbf{h}_2^* & -\mathbf{h}_1^* & \mathbf{h}_6^* & -\mathbf{h}_3^* & \mathbf{h}_5^* & -\mathbf{h}_4^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_4) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_3 & \mathbf{h}_2 & \mathbf{h}_5 & \mathbf{h}_4 & \mathbf{h}_6 \\ \mathbf{h}_3^* & -\mathbf{h}_1^* & \mathbf{h}_5^* & -\mathbf{h}_2^* & \mathbf{h}_6^* & -\mathbf{h}_4^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_5) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_3 & \mathbf{h}_2 & \mathbf{h}_6 & \mathbf{h}_4 & \mathbf{h}_5 \\ \mathbf{h}_3^* & -\mathbf{h}_1^* & \mathbf{h}_6^* & -\mathbf{h}_2^* & \mathbf{h}_5^* & -\mathbf{h}_4^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_6) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_4 & \mathbf{h}_2 & \mathbf{h}_5 & \mathbf{h}_3 & \mathbf{h}_6 \\ \mathbf{h}_4^* & -\mathbf{h}_1^* & \mathbf{h}_5^* & -\mathbf{h}_2^* & \mathbf{h}_6^* & -\mathbf{h}_3^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_7) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_4 & \mathbf{h}_2 & \mathbf{h}_6 & \mathbf{h}_3 & \mathbf{h}_5 \\ \mathbf{h}_4^* & -\mathbf{h}_1^* & \mathbf{h}_6^* & -\mathbf{h}_2^* & \mathbf{h}_5^* & -\mathbf{h}_3^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_8) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_5 & \mathbf{h}_2 & \mathbf{h}_6 & \mathbf{h}_3 & \mathbf{h}_4 \\ \mathbf{h}_5^* & -\mathbf{h}_1^* & \mathbf{h}_6^* & -\mathbf{h}_2^* & \mathbf{h}_4^* & -\mathbf{h}_3^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_9) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_6 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{h}_5 \\ \mathbf{h}_6^* & -\mathbf{h}_1^* & \mathbf{h}_3^* & -\mathbf{h}_2^* & \mathbf{h}_5^* & -\mathbf{h}_4^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_10) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_6 & \mathbf{h}_2 & \mathbf{h}_4 & \mathbf{h}_3 & \mathbf{h}_5 \\ \mathbf{h}_6^* & -\mathbf{h}_1^* & \mathbf{h}_4^* & -\mathbf{h}_2^* & \mathbf{h}_5^* & -\mathbf{h}_3^* \end{bmatrix}$$

$$\tilde{\mathbf{H}}(p_11) = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_6 & \mathbf{h}_2 & \mathbf{h}_5 & \mathbf{h}_3 & \mathbf{h}_4 \\ \mathbf{h}_6^* & -\mathbf{h}_1^* & \mathbf{h}_5^* & -\mathbf{h}_2^* & \mathbf{h}_4^* & -\mathbf{h}_3^* \end{bmatrix}$$

A.2 The expected value of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)$

We develop the expectation of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}^H(p_k)$, as an extended result belonging to section 4.3.6. We show that this result is not dependent on the choice of antenna assignment pattern, which makes it useless as a building block to develop a criterion for pattern selection.

$$\begin{aligned}
\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H =&(\mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G})(\mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G})^H\\
=&(\mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1) + \mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G})(\mathbf{I}(p_k^1)^T\mathbf{H}^H\mathbf{I}_{r1}^T + \mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{H}^T\mathbf{I}_{r2}^T)\\
=&\mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1)\mathbf{I}(p_k^1)^T\mathbf{H}^H\mathbf{I}_{r1}^T + \mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1)\mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{H}^T\mathbf{I}_{r2}^T +\\
&\mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\mathbf{H}^H\mathbf{I}_{r1}^T + \mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G}\mathbf{G}^T\mathbf{I}(p_k^2)^T\mathbf{H}^T\mathbf{I}_{r2}^T
\end{aligned}$$

$$\mathbf{I}(p_k^1)\mathbf{I}(p_k^1)^T = \mathbf{I}(p_k^2)\mathbf{I}(p_k^2)^T = I, \quad \mathbf{G} = \mathbf{G}^T, \quad \mathbf{G}\mathbf{G}^T = I$$

$$\begin{aligned}
\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H =&\mathbf{I}_{r1}\mathbf{H}\mathbf{H}^H\mathbf{I}_{r1}^T + \mathbf{I}_{r1}\mathbf{H}\mathbf{I}(p_k^1)\mathbf{G}\mathbf{I}(p_k^2)^T\mathbf{H}^T\mathbf{I}_{r2}^T +\\
&\mathbf{I}_{r2}\mathbf{H}^*\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\mathbf{H}^H\mathbf{I}_{r1}^T + \mathbf{I}_{r2}\mathbf{H}^*\mathbf{H}^T\mathbf{I}_{r2}^T
\end{aligned}$$

Substituting $\mathbf{H} = \sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}$ and using that $\sqrt{\mathbf{R}_t}$ and $\sqrt{\mathbf{R}_r}$ are real, symmetric matrices, yields

$$\begin{aligned}
\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H =& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T +\\
&\mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{G}\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T +\\
&\mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T +\\
&\mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\mathbf{R}_t\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T
\end{aligned}$$

For simplicity, we define the deterministic product $\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}$ as the deterministic matrix $\mathbf{C}_k$, and rewrites the expression.

$$\mathbf{C}_k = \sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}$$

$$\begin{aligned}
\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H =& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\mathbf{H}_0\mathbf{C}_k\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T +\\
&\mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\mathbf{C}_k^T\mathbf{H}_0^H\sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\mathbf{H}_0^*\mathbf{R}_t\mathbf{H}_0^T\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T
\end{aligned}$$

To study the expected value of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H$, we remember that the expectation of a sum equals the sum of each part's expectation. By also moving out all deterministic parts, the expected value may be written as

$$
\begin{aligned}
E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H) =& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \\
& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{G}\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T + \\
& \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \\
& \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0^*\mathbf{R}_t\mathbf{H}_0^T)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T
\end{aligned}
\tag{A.1}
$$

We know that $\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H = (\mathbf{H}_0^*\mathbf{R}_t\mathbf{H}_0^T)^T$ and $\mathbf{I}(p_k^1)\mathbf{G}\mathbf{I}(p_k^2)^T = \left(\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\right)^T$.

$$
\begin{aligned}
E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H) =& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \\
& \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0\mathbf{C}_k\mathbf{H}_0^T)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T + \\
& \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\, E(\mathbf{H}_0^*\mathbf{C}_k^T\mathbf{H}_0^H)\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r1}^T + \\
& \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}\, E((\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H))^T\, \sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T
\end{aligned}
\tag{A.2}
$$

First, we concentrate on the first and last of the four terms, and find the expected value

$$
E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H) \tag{A.3}
$$

where a certain coefficient in the mth row and nth column of the $M \times M$ matrix is given by

$$
E\Big(\sum_{k=1}^{N} h_{mk}\big(\sum_{l=1}^{N} r_{kl}h_{ln}^*\big)\Big) = \sum_{k=1}^{N}\sum_{l=1}^{N} r_{kl}\, E(h_{mk}h_{ln}^*) \tag{A.4}
$$

Recalling the results stated in equation (4.30), makes it clear that the only non-zero sums will appear when $m = n$, i.e. along the diagonal. In these sums, only the N terms when $l = k$ will contribute, so the result is that the expected value in the first term of $E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H)$ is given by

$$
E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H) = \sum_{k=1}^{N} r_{kk}\cdot I = NI. \tag{A.5}
$$

where I is the identity matrix. This result also holds for the expectation in the fourth term, because

$$
E(\mathbf{H}_0^*\mathbf{R}_t\mathbf{H}_0^T) = \left(E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H)^T\right) = \left(E(\mathbf{H}_0\mathbf{R}_t\mathbf{H}_0^H)\right)^T = NI. \tag{A.6}
$$

Next, we move our attention to the more complicated second and third terms; $E(\mathbf{H}_0\mathbf{C}_k\mathbf{H}_0^T)$ and $E(\mathbf{H}_0^*\mathbf{C}_k^T\mathbf{H}_0^H)$. We attempt to develop the expected value of $\mathbf{H}_0\mathbf{C}_k\mathbf{H}_0^T$ first. The coefficients in the mth row and nth column of $E(\mathbf{H}_0\mathbf{C}_k\mathbf{H}_0^T)$ may be written as the sum

$$E\Big(\sum_{k=1}^{N} h_{km}(\sum_{l=1}^{N} c_{kl}h_{ln})\Big) = \sum_{k=1}^{N}\sum_{l=1}^{N} c_{kl}E(h_{km}h_{ln}) \tag{A.7}$$

The coefficients c_{kl} of $\mathbf{C}_k$ are taken out of the equation because they are deterministic, assuming a pattern has been chosen. By recalling the results stated in equation (4.30) once more, it is clear that the expectation in the above equation is always zero, for all values of k, l, m and n. That result is also valid when the coefficients of $\mathbf{H}_0$ are complex conjugated (as for $E(\mathbf{H}_0^*\mathbf{C}_k^T\mathbf{H}_0^H)$). If $\mathbf{C}_k = \sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}$ is substituted back into the equation, the second and third terms of equation (A.1) are now given by

$$E(\mathbf{H}_0\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^1)\mathbf{I}(p_k^2)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^T) = E(\mathbf{H}_0^*\sqrt{\mathbf{R}_t}\mathbf{I}(p_k^2)\mathbf{G}\mathbf{I}(p_k^1)^T\sqrt{\mathbf{R}_t}\mathbf{H}_0^H) = 0 \tag{A.8}$$

With the results from the above (A.6) and (A.8), the expected value of $\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H$ from equation (A.1), is simplified and now given by

$$\begin{aligned} E(\tilde{\mathbf{H}}(p_k)\tilde{\mathbf{H}}(p_k)^H) &= \mathbf{I}_{r1}\sqrt{\mathbf{R}_r}NI\sqrt{\mathbf{R}_r}\mathbf{I}_{r1} + 0 + 0 + \mathbf{I}_{r2}\sqrt{\mathbf{R}_r}NI\sqrt{\mathbf{R}_r}\mathbf{I}_{r2}^T \\ &= N\Big(\mathbf{I}_{r1}\mathbf{R}_r\mathbf{I}_{r1}^T + \mathbf{I}_{r2}\mathbf{R}_r\mathbf{I}_{r2}^T\Big), \end{aligned} \tag{A.9}$$

a result that is not dependent on the particular pattern chosen, as both $\mathbf{I}_{r1}, \mathbf{I}_{r2}$ and $\mathbf{R}_r$ are deterministic with respect to that.

A.3 Extra performance plots for the SMAL scheme

A.3.1 BER-results of fixed patterns over time, $N = M = 6$

In the plots in figures A.1 and A.2, all patterns are shown for the case of $N = M = 6$, for the two chosen levels of correlation $r = 0.29$ and $r = 0.90$. These plots are bases for the $N = M = 6$ plots shown in section 4.3.

A.3.2 Best pattern for correlation-based SMAL, $N = M = 6$ and $r = 0.90$

Figure A.3 shows that the pattern chosen by the correlation-based SMAL at a correlation level between neighbouring elements of $r = 0.90$ is p_5.

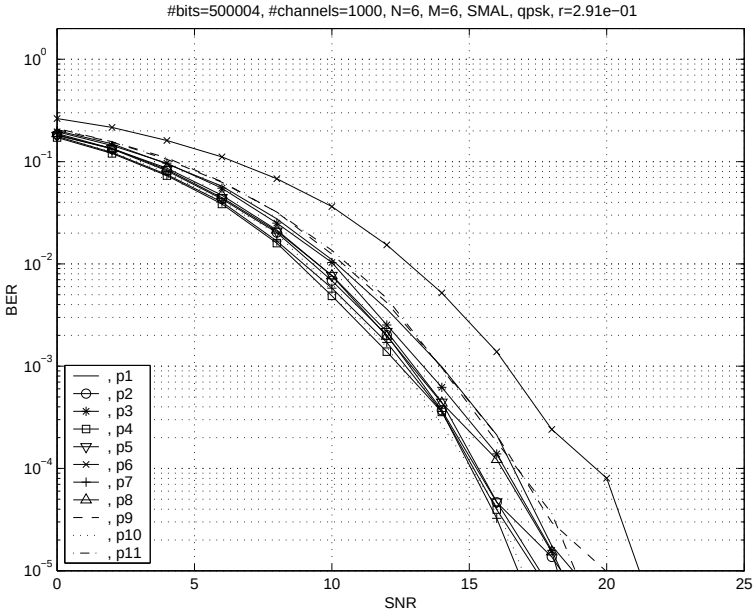

Figure A.1: Fixed patterns, one curve for each, $N = M = 6$, $r = 0.29$.

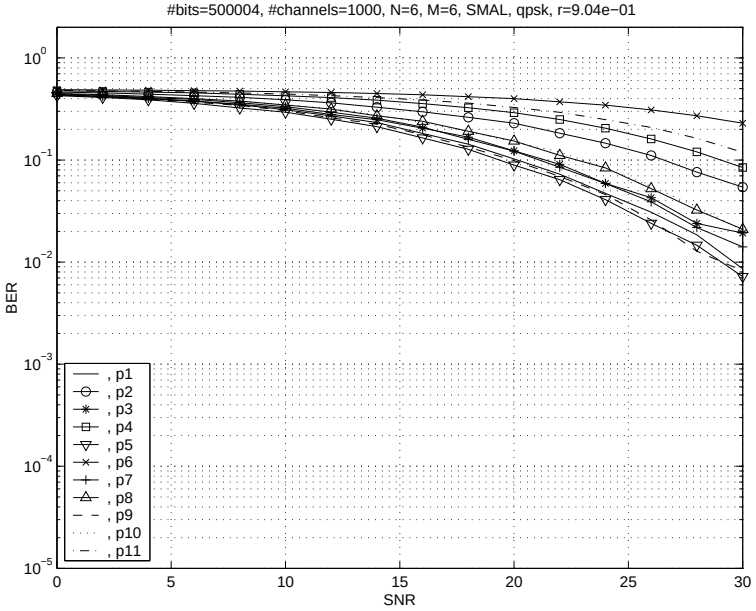

Figure A.2: Fixed patterns, one curve for each, $N = M = 6$, $r = 0.90$.

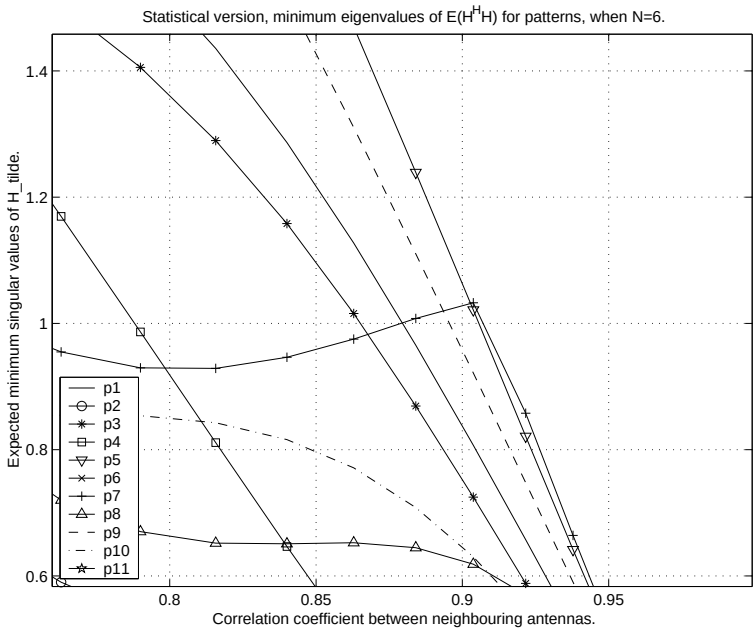

Figure A.3: Minimum singular values of $E(\tilde{\mathbf{H}}^H(p_k)\mathbf{H}(p_k))$, for $k = 1,\ldots,11$ and $N = 6$

Appendix B

List of acronyms and mathematical notations

B.1 List of acronyms

BLAST	Bell Labs Space-Time
LOS	Line of Sight
MIMO	Multiple-Input Multiple-Output
MISO	Multiple-Input Single-Output
MD	MIMO diversity
MRC	Maximum Ratio Combining
N-LOS	Non Line of Sight
SIMO	Single-Input Multiple-Output
SISO	Single-Input Single-Output
SM	Spatial Multiplexing
SMAL	Spatial Multiplexing of ALamouti
STBC	Space-time block code
STC	Space-time code

SVA	The Singular Vector Approach
SVD	Singular Value Decomposition

B.2 Mathematical notations and list of symbols

Notation

$a, \mathbf{a}, \mathbf{A}$	notations for a scalar, a vector (boldface) and a matrix (capital boldface)
$\mathbf{a}_k$	the kth element of the vector **a**.
$\mathbf{A}_{kl}$	the element in the intersection between the kth row and the lth column of the matrix **A**.
$\mathbf{U}, \Sigma, \mathbf{V}$	factors of a singular value decomposition of the matrix **A**, $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$
$a^*, \mathbf{a}^*, \mathbf{A}^*$	the conjugate of the complex scalar a, vector **a** or matrix **A**
$\mathbf{a}^H, \mathbf{A}^H$	the conjugate transpose of the complex vector **a** or matrix A
$\mathbf{a}^T, \mathbf{A}^T$	the transpose of the vector or matrix A
$\mathbf{A}^{-1}$	the inverse of the square matrix A
$\mathbf{A}^{\#}$	the pseudo-inverse of the matrix A
$\|\mathbf{a}\|_2$	the length/norm of the vector A, either complex: $\|A\|_2 = \sqrt{A^H A} = \sqrt{a_1^* a_1 + \ldots + a_n^* a_n}$ or real: $\|A\|_2 = \sqrt{A^T A} = \sqrt{a_1^2 + \ldots + a_n^2}$.
$E(x) = \mu_x$	the mean or expected value of a random variable x
$E(\|x\|^2) = \sigma_x^2$	the variance of a random variable x

σ_x	the standard deviation of a random variable x

List of common symbols

N	number of transmit antennas
M	number of receive antennas
$s, \mathbf{s}, \mathbf{S}$	symbol (scalar, vector, matrix)
$y, \mathbf{y}, \mathbf{Y}$	received symbol (scalar, vector, matrix)
$h, \mathbf{h}, \mathbf{H}$	channel coefficient (scalar, vector, matrix)
$v, \mathbf{v}, \mathbf{V}$	noise symbol (scalar, vector, matrix)

Appendix C

Matlab simulations, background and code

C.1 An overview of the simulation framework

This section presents the practical part of my thesis, which has been the writing and testing of a simulator. The simulator is built to investigate how different transmit/receive algorithms and multiple antennas on both sides can improve the bit-error rate for given values of the signal-to-noise ratio.

The simulator is developed and tested for all the algorithms described in section 3.1 and the new SMAL scheme, presented in chapter 4. The simulator is written in Matlab, and all the function files are found under C.2.

Regardless of what algorithm is used to transmit/receive the information, the main logical structure is the same. This basis is shown in listing C.1. Only the essential function calls are included, and, in these calls, only vital parameters are shown. Hence, the figure does not reproduce exactly what happens in my simulation, but rather gives a general overview.

Listing C.1: Overview of the program

```
% initialise variables
% N,M                          #transmit and #receive antennas
% maxChannelIndex              #independent channels per SNR
% n                            #bits transmitted per channel
% minSNR,maxSNR,deltaSNR       defines the x-axis

SNR = minSNR:deltaSNR:maxSNR;
ber = zeros(length(SNR),1);
```

```
for SNRindex = 1:length(SNR)

  errors = 0;
  % variance of noise, given by current SNR
  var_n = 1/(power(10, SNR(k)/10));

  for j = 1:maxChannelIndex

    % H: channel matrix, possibly correlated
    H = channel_generator(N,M);

    % b: n random bits
    b = bitgenerator(n);

    % s: bits combined into symbols
    s = modulator(b);

    % simulate transmission with algorithm ABC
    % s_hat: array of received symbols
    s_hat = transceiver_ABC(s,H,var_n);

    % b_hat: symbols of s_hat sliced back to bits
    b_hat = slicer(s_hat);

    % errors: #bit errors between b and b_hat
    errors =  errors + BER_calculation(b, b_hat);
  end;

  ber(SNRindex,1) = errors/(maxChannelIndex*n);

end;

% plot the BER-values (y-axis) for all SNR-values (x-axis)
```

As seen from the pseudocode, we start out by defining necessary variables, such as the number of antennas and the interval of SNR-values we want to simulate for. For every SNR-value in dB, we determine the variance of the noise through the relation in (2.33). The channel coefficients of H are random, complex numbers, so to ensure that their statistical characteristics appear, we realize a large number of channels; typically `maxChannelIndex` = 1000.

In section 2.1.3, the channel was defined as quasi-static, which is modelled in the simulator by keeping it constant during transmission of

n bits at the time. The bits are generated by the function `bitgenerator.m`, which returns an array of length n. In `modulator.m` the bits are modulated into complex symbols and returned in the array s, which is $n/\log_2(M)$ long. M is the number of points in the constellation of the chosen modulation method, as described in section 2.1.1.

Given the symbol array s, the variance of the noise var_n (earlier denoted σ_n^2) and the channel matrix H, we call upon a transmit function to simulate wireless transmission using a certain pre-decided algorithm. The simulation framework contains a separate m-file for each of the algorithms that are implemented, and the choice of scheme is made before running the program.

The transceiver-function builds the noise matrix N and the symbol matrix S, as described by the corresponding transmit scheme, see the algorithm presentations in section 3.1. Next, it simulates transmission over the channel, given in general by equation 2.35. After reception, the chosen transmit algorithm describes how to retrieve an approximation of the original signal array, using knowledge of the channel, and this is what is returned to the main-function in the array s_hat.

In `slicer.m`, the approximated symbol values are sliced to bits, after which `ber_computation.m` computes the number of errors between the original and the estimated bit arrays. For every SNR-value, the sum of all the errors is normalized by the product of the number of channels and the number of bits, yielding the bit-error rate (BER). A plot of the BER-values on the y-axis versus the SNR-values on the x-axis illustrates the result of a simulation, and such plots will be seen in the next section.

One point to be made about the simulations is that they are quite time-consuming, at least if we desire to run it employing several antennas, efficient algorithms or for high SNR-values, so as to produce very low BER-results. The reason is that when errors are less likely to occur (low BER), we need more iterations to produce properly averaged data, which obviously takes more time. A reasonable rule-of-thumb is that the number of bits n should approximate

$$\frac{1}{BER_{exp}} 100$$

where BER_{exp} is the lowest BER-value we expect. To help reduce the time spent waiting for simulations to finish, the program is also adjusted to be run in parallell on several computers, one for each SNR-value. However, whether the simulations are run serially or in parallell is of no consequence to the results.

C.2 Matlab functions

```
function [tx, rx, min_SNR, max_SNR, delta_SNR, ...
          distance, maxChannelIndex, c, d, exp_BER, ...
          lambda, combocase] = parameters;

% PARAMETERS Setting the parameters of a simulation
% PARAMETERS returns the value of important parameters
%            to a main-function


% number of transmit and receive antennas
tx = 4;
rx = 4;

% SNR-values over which to run the simulation
min_SNR = 0;
max_SNR = 30;
delta_SNR = 2;


% wavelength lambda, corresponding to the
% frequency 2 GHz (in use in 3G)
lambda = 1.5*10^(-1);


% distance between antenna elements, measured in
% wavelengths lambda and in meters, respectively
distance_coef = 1.0*10^(-1);
distance = lambda*distance_coef;

% number of channel realisations to average over
maxChannelIndex = 1000;


% modulation scheme; QPSK (2) or 16QAM (3)
c = 2;

% choice of transmit scheme:
% diversity, transmit/receive MRC or SVA      (0)
% spatial multiplexing                        (1)
% the alamouti scheme (2xM case)              (2)
% the SMAL scheme                             (3)
```

```
d = 3;

% expected BER is used to determine the number of bits
% to be sent for channel realisation, see num_bits.m
exp_BER = 10^(-4);


% combocase, determines case of combining for SMAL
% for tx=4: combocase = [1,3],  tx=6: combocase = [1,11]
% combocase = -1 => instantaneous adaptive selection
% combocase = 0  => random selection of a pattern
combocase = 0;
```

```
%
% main: in charge of running the BER-simulations, non-parallel
%
% tx: number of tranmitter antennas
% rx: number of receiver antennas
%

clear all;
format short e;

t_start = clock;

par = 0; % not run in parallel
num_SNR = -1; % irrelevant, only useful when run in parallel



[values_per_SNR, errors, SNR, total, maxChannelIndex, ...
 tx, rx, c, d, distance, lambda, Rt, Rr, avg_sing_val, ...
 combocase, descr] = main_common(par, num_SNR);

min_SNR = SNR(1);
max_SNR = SNR(length(SNR));
delta_SNR = (max_SNR-min_SNR)/(length(SNR)-1);


db_impr = 10*log10(avg_sing_val);
disp(db_impr);

t_end = etime(clock,t_start);
t_end = t_end/((max_SNR-min_SNR)/delta_SNR + 1);

% plot
filnavn = sprintf('data_%s_%d_%d_%d_%d_%d_%d.mat',descr,total, ...
                  maxChannelIndex,max_SNR, delta_SNR, tx, rx);

save(filnavn, 'SNR', 'values_per_SNR', 'errors', 'min_SNR', ...
     'max_SNR', 'delta_SNR','total', 'maxChannelIndex','tx', ...
     'rx','c','d', 'distance','lambda','Rt','Rr','db_impr', ...
     't_end','par','combocase');
```

```
function main_parallell(num_SNR)

%
% main_parallel: runs the BER-simulations for a
% given SNR-value
%
% tx: number of tranmitter antennas
% rx: number of receiver antennas
%
% 0 <= num_SNR <= N-1


format short e;

t_start = clock;

% setting the random seed to something different,
% ensuring random numbers on all computers
num = sum(100*clock);
rand('state', num);
randn('state', num);


par = 1; % run in parallel


[values_per_SNR, errors, SNR, total, maxChannelIndex, ...
 tx, rx, c, d, distance, lambda, Rt, Rr, avg_sing_val, ...
 combocase,descr] = main_common(par, num_SNR);

min_SNR = SNR(1);
max_SNR = SNR(length(SNR));
delta_SNR = (max_SNR-min_SNR)/(length(SNR)-1);

t_end = etime(clock,t_start);

% plot
numFile = num_SNR;
value_curr_SNR = values_per_SNR;
filnavn = sprintf('comb_%s_%d_%d_%d_%d_%d_%d_%d.mat', ...
                  descr,total,maxChannelIndex,max_SNR, ...
                  delta_SNR,tx,rx,numFile);

save(filnavn, 'value_curr_SNR','errors', 'min_SNR', ...
```

```
'max_SNR', 'delta_SNR','total', 'maxChannelIndex', ...
'tx','rx','c','d','distance','lambda','Rt','Rr', ...
'avg_sing_val','t_end','combocase');
```

```
function [values_per_SNR_, errors, SNR, total, maxChannelIndex, tx, rx, ...
          c, d, distance, lambda, Rt, Rr, avg_sing_val, combocase, descr] = ...
    main_common(par, num_SNR);

%
% Runs the main loops of the program, as shown in the pseudocode
%


% run the parameters routine, defining and initializing important variables
[tx, rx, min_SNR, max_SNR, delta_SNR, distance, maxChannelIndex, ...
 c, d, exp_BER, lambda, combocase] = parameters;

SNR = min_SNR:delta_SNR:max_SNR;

if par == 0
  startIndexSNR = 1;
  maxIndexSNR = length(SNR);
else
  startIndexSNR = num_SNR+1;
  maxIndexSNR = startIndexSNR;
end;



% default value for normalizing the channel matrix; to be used when
% computing singular values
tx_sing = tx;
C_albl = 0;

% decide on function (-name) to use for transceiving data
if d == 0
  if tx == 1
    fname = sprintf('transceiver_RxMRC');
    funcdescr = sprintf('RxMRC');
  elseif rx == 1
    fname = sprintf('transceiver_TxMRC');
    funcdescr = sprintf('TxMRC');
  else
    fname = sprintf('transceiver_SVA');
    funcdescr = sprintf('SVA');
  end;
elseif d == 1
  fname = sprintf('transceiver_BLAST');
```

```
  funcdescr = sprintf('BLAST');
elseif d == 2
  fname = sprintf('transceiver_ALAMOUTI');
  funcdescr = sprintf('ALAM');
  if tx ~= 2
    disp('OBS: Feil! Alamouti fungerer kun med 2 transmit antennas.');
  end;
elseif d == 3

  H_manip  = zeros(2*rx,tx);

  if combocase == -1
    % simulation should choose antenna pattern based on channel
    % realizations, so may change

    mod_dist = eucldist_modulation(c);
    % finding all possible permutations of combinations
    allperms = permutation_matrix(tx);

    stats = zeros(size(allperms,1),1);
    t1_index = zeros(1,tx);
    t2_index = zeros(1,tx);

    fname = sprintf('transceiver_SMAL');
    funcdescr = sprintf('SMAL%s','adapt');
  elseif combocase == 0
    % simulation should choose a random pattern
    fname = sprintf('transceiver_SMAL');
    funcdescr = sprintf('SMAL%s','rand');

    % finding all possible permutations of combinations
    allperms = permutation_matrix(tx);

    t1_index = zeros(1,tx);
    t2_index = zeros(1,tx);

  else
    fname = sprintf('transceiver_SMAL');
    funcdescr = sprintf('SMAL%d',combocase);

    C_albl = get_pattern(combocase,tx);
  end;
end;
```

```
print_dist = distance/lambda;
corrdescr = sprintf('%.02e',print_dist);

if c == 1
  moddescr = sprintf('bpsk');
elseif c == 2
  moddescr = sprintf('qpsk');
elseif c == 3
  moddescr = sprintf('16qam');
end;

descr = sprintf('%s_%s_%s',funcdescr,corrdescr,moddescr);
disp(descr);


[Rt, Rr] = correlation(tx,rx,lambda,distance);

% compute the number of bits we will generate every time,
% and how many we'll send each time in inner for-loop, to avoid
% stalling because matlab needs to swap
[n,numTimes] = num_bits(c,rx,tx,d,exp_BER,maxChannelIndex)
total = n*numTimes

% Array to be plotted at the end of the simulation
if par == 0
  values_per_SNR = zeros(1,length(SNR));
  errors = zeros(1,length(SNR));
else
  values_per_SNR = zeros(1,1);
  errors = zeros(1,1);
end;

% arrays keeping the bits, both original and computed
b = zeros(1,n);
b_hat = zeros(1,n);

%determining length of symbol array, based on coding parameter c
l = length_symbolarray(c, length(b));

% arrays keeping the symbols, both original and computed
l_arr = 1:l;
s = zeros(1,l);
```

```
s_hat = zeros(1,l);

var_n = 0;
sing_vals = 0;


for k = startIndexSNR:maxIndexSNR

  % compute the variance (var_n) of the noise,
  % given by the input argument snr, so that

  var_n = 1/(power(10, SNR(k)/10));
  errors_all_channels = 0;

  for j = 1:maxChannelIndex

    % generate the channel
    H = channel_generator(tx,rx,c,Rt,Rr);

    if d == 3 & combocase == -1
      [C_albl, index] = get_pattern_inst(H, H_manip, ...
                                         allperms, ...
                                         mod_dist);
      stats(index,1) = stats(index,1) + 1;
    elseif d == 3 & combocase == 0
      [C_albl, index] = get_pattern_rand(tx, rx, allperms);
    end;

    % compute and sum singular values
    sing_vals = sing_vals+sing_val(H * (1/sqrt(tx_sing)))^2;

    % extra loop, to limit swapping of memory
    for q=1:numTimes

      % generate the signals for transmission
      b = bitgenerator(n);

      % modulate signals using scheme in c
      s = modulator(c,b,l_arr);

      % simulate transmission, feval() is slightly slower
      % than if-tests, but more elegant.
      s_hat = feval(fname,rx,tx,s,H,var_n,l,C_albl);
```

```
        % convert back to bits
        b_hat = slicer(s_hat,c,n);

        % compute BER and store in array
        errors_all_channels = errors_all_channels + ...
            BER_calculation(b, b_hat);
      end;

    end;

    % errors_all_channels, average over all values
    ber_percent = errors_all_channels/(maxChannelIndex*total);

    if par == 0
      values_per_SNR(k) = ber_percent;
      errors(k) = errors_all_channels;
    else
      values_per_SNR = ber_percent;
      errors = errors_all_channels;
    end;

    disp(SNR(k));

end;

if d == 3 & combocase == -1
    for lc=1:size(stats,1)
      stat=sprintf('%s:  %8d',num2str(allperms(lc,:)),stats(lc));
      disp(stat);
    end;
    %size(allperms)
    %sum(stats)
end;
if par == 0
    avg_sing_val = sing_vals/(maxIndexSNR*maxChannelIndex);
else
    avg_sing_val = sing_vals/maxChannelIndex;
end;


values_per_SNR_ = values_per_SNR;
```

```
function b = bitgenerator(n)
%
% BITGENERATOR     Generates random bit-values
% BITGENERATOR(n)  Generates n random bits and returns
%                  them in array b
%


% generate n uniformly distributed random numbers between 0 and 1

% round them off to 0 and 1, possible to view them as bits
b = round(rand(1,n));
```

```
function s = modulator(c,b,l_arr)
%
% Generates n random bits and returns them in array b
%
% Maps the bits in array b to complex symbols,
% using the coding scheme c (either QPSK, 16QAM or binary).
% The result is returned in the nx2 matrix s, n is equal
% to the length of the bit-array b and the two parts
% (real and imaginary) are stored in separate coloumns.
%
%
%


switch c
 case 1
  s = b;
 case 2
  s = modulator_qpsk(b,l_arr);
 case 3
  s = modulator_16qam(b,l_arr);
 otherwise
  disp('Invalid coding scheme.');
end;
```

```
function s = modulator_qpsk(b,l_arr)
%
% qpsk: Maps bits to complex symbols,
% using the modulating scheme QPSK
%
% The values in array b (the bits) is first
% mapped and stored in b1 as follows:
%   0 is mapped to -1 and 1 is mapped to 1
%
% Then log2(M) entries from b is mapped to a
% complex symbol (M = 4, so 2 bits gives one symbol),
% in the following way:
%   m(k) = b1(2*k-1) + b1(2*k)*i;
%
%
%


b1 = 2*b - 1;

s = (b1(2*l_arr-1) + b1(2*l_arr)*i)/sqrt(2);
```

```
function s = modulator_16qam(b,l_arr)
%
% Maps bits to complex symbols, using the modulating scheme 16-QAM
%

M = 16;
bitsPerSymbol = log2(M);

% 0000 : 1+i
% 0001 : 1+3i
% 0010 : 3+i
% 0011 : 3+3i

% 0100 : 1-i
% 0101 : 1-3i
% 0110 : 3-i
% 0111 : 3-3i

% 1000 : -1+i
% 1001 : -1+3i
% 1010 : -3+i
% 1011 : -3+3i

% 1100 : -1-i
% 1101 : -1-3i
% 1110 : -3-i
% 1111 : -3-3i

% Algorithm : Reorganize bitsPerSymbol bits at the time by
% switching bits number 2 and 3. Then the first two bits
% and the last two bits are both interpreted as follows:
% 01 :  3d
% 00 :   d
% 10 : - d
% 11 : -3d

l = length(l_arr);


bit_mat = zeros(l,bitsPerSymbol);


b = b.';
```

```
bit_mat = 2*[-(b(bitsPerSymbol*(l_arr-1) + 1)), ...
         b(bitsPerSymbol*(l_arr-1) + 3), ...
         -(b(bitsPerSymbol*(l_arr-1) + 2)), ...
         b(bitsPerSymbol*(l_arr-1) + 4)] + repmat([1,1,1,1],l,1);


s_ = bit_mat(:,1).*bit_mat(:,2) + bit_mat(:,3).*bit_mat(:,4)*i;


% Skalere, skal dele åp roten av average power, som er
% (summen av power for alle symboler fra 1 til 16)^(1/2)

avg_pow = 10;
s = (s_.*(1/sqrt(avg_pow))).';
```

```
function [Rt, Rr] = correlation(tx,rx,lambda,distance);

%
% CORRELATION(tx,rx,lambda,distance) returns the correlation
%                                    matrices
%
% lambda is the carrier wave-length
% distance is the inter-element antenna distance
%

if distance/lambda > 100 % regner over 100 som Inf
  Rt = eye(tx,tx);
  Rr = eye(rx,rx);
else

  Rt=zeros(tx,tx);
  Rr=zeros(rx,rx);

  Positions_tx = 0:distance:distance*(tx-1);
  Positions_rx = 0:distance:distance*(rx-1);

  %generate correlation matrices

  %%%%TX correlation matrix%%%%%%%%%%%%%%%%%%%%%%%%

  for k=1:tx

    for l=1:tx

      Rt(k,l)=bessel(0,2*pi*(Positions_tx(k)-Positions_tx(l))/lambda);

    end;
  end;


  %%%%RX correlation matrix%%%%%%%%%%%%%%%%%%%%%%%%

  for k=1:rx

    for l=1:rx

      Rr(k,l)=bessel(0,2*pi*(Positions_rx(k)-Positions_rx(l))/lambda);

    end;
```

```
    end;
end;
```

```
function H = channel_generator(tx,rx,c,Rt,Rr)
%
% CHANNEL_GENERATOR channel matrix
% CHANNEL_GENERATOR(tx,rx,c,Rt,Rr) returns the correlated
%                                  channel matrix
%
% c is the choice of modulation scheme (c=1 is binary)

if c == 1
  H0 = randn(rx,tx);
else
  H0 = (randn(rx,tx) + randn(rx,tx)*i)/sqrt(2);
end;

H = sqrtm(Rr)*H0*sqrtm(Rt);
```

```
function s_hat = transceiver_ALAMOUTI(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel,
% using the Alamouti scheme.
%
% rx: the number of transmitter antennas
% tx: the number of transmitter antennas
% s:  signal to be sent
% h:  channel matrix
% snr: Signal-to-Noise-Ratio

% H: rx*tx
% s: l*1, l er antall symboler som skal sendes
% y: rx*l
% s_hat: 1*l
% wt: tx*1 og wr: rx*1

% using alamouti directly, we have [y1 y3; y2 y4],
% where y1 and y3 are
% received at Rx1 - in time slot 1 and 2,
% and y2 and y4 are received at
% Rx2.
% With the manipulated H used here, what we get
% is rather y_ = [y1;-y3*;y2;-y4*], because we
% manipulate H and not the symbols. This is
% corrected by using the same manipulated H-matrix
% to invert the system.
% y = H_tilde * s_mat + n;
% s_hat = (H_tilde'*y)/div_norm;
% know that we need to normalize with
% (abs(h11)+abs(h12)+abs(h21)+abs(h22))
% => same as
% sum(sum(H_tilde'*H_tilde))/2 = trace(H_tilde'*H_tilde)/2


% generating noise
n = (randn(rx*2,l/2) + randn(rx*2,l/2)*i)*sqrt(var_n/2);
s_mat = reshape(s,tx,l/2);

% scaling the channel matrix H, to maintain total power equal 1, although
% sending on tx transmit antennas
H = H.*(1/sqrt(tx));
```

```
% manipulating H into the right form
H_t2 = [conj(H(:,2)) -conj(H(:,1))];
H_tilde = [H;H_t2];

div_norm = sum(sum(H.*conj(H)));
s_hat = (H_tilde' * (H_tilde*s_mat + n))/div_norm;
s_hat = reshape(s_hat,1,l);


%Normal ALAMOUTI stc
% s_coded = zeros(2,l);
% s_coded(:,1:2:l-1) = s_mat;
% s_coded(1,2:2:l) = -conj(s_mat(2,:));
% s_coded(2,2:2:l) = conj(s_mat(1,:));
% n = reshape(n,rx,l);
% y=H*s_coded + n;
% factor = repmat([1 -1],rx,1); % know tx = 2
% s_hat = (H'*y(:,1:2:l-1) + (factor.*fliplr(H)).'*conj(y(:,2:2:l)))/ ...
%         div_norm;
% s_hat = reshape(s_hat,1,l);
```

```
function s_hat = transceiver_BLAST(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel.
% INDEPENDENT DATA ON ALL TRANSMIT ANTENNAS
%
% rx: the number of transmitter antennas
% tx: the number of transmitter antennas
% s:  signal to be sent
% h:  channel matrix
% snr: Signal-to-Noise-Ratio
%
%

% H    : rx*tx
% s    : l*1, l er antall symboler som skal sendes
% y    : rx*l
% s_hat: 1*l


num_transmits = floor(l/tx);

% generate random noise, stat. different for every signal
n = (randn(rx,num_transmits) + ...
     randn(rx,num_transmits)*i)*sqrt(var_n/2);

% scaling the channel matrix h, to maintain total
% power equal 1, although sending on tx transmit antennas
H = H.*(1/sqrt(tx));


% vektormetode
s_mat = reshape(s,tx,num_transmits); % tx*num_transmits
y_mat = H*s_mat + n;                 % rx*num_transmits


s_hat = pinv(H)*y_mat;           % tx*num_transmits
s_hat = reshape(s_hat, 1,l);
```

```
function s_hat = transceiver_RxMRC(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel.
%
% rx: the number of receiver antennas
% tx: the number of transmitter antennas
% s:  signal to be sent
% h:  channel matrix
% snr: Signal-to-Noise-Ratio
%

% generate random noise, multiplying with sqrt(var_n/2) to ensure
% 1) first that variance == 1 for the whole complex noise symbol
%    by factor sqrt(1/2), and then manipulating it to
% 2) variance == var_n by factor sqrt(var_n), the standard
%    deviation.

% OBS   sum(abs(power(h,2))) == h*h' = h*transpose(conj(h))

n = (randn(rx,l) + randn(rx,l)*i)*(sqrt(var_n/2));

s_hat = sqrt(H'*H)*s + (H'/(sqrt(H'*H)))*n;
```

```
function s_hat = transceiver_TxMRC(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel.
%
% rx: the number of transmitter antennas
% s:  signal to be sent
% h:  channel matrix
% snr: Signal-to-Noise-Ratio
%
%

% h: rx*tx = 1*tx
% s: l*1, l er antall symboler som skal sendes
% y: rx*l
% s_hat: 1*l

% generate random noise, stat. different for every signal
% naming the noise matrix y, for reuse of variables
% generated for every symbol to be sent
n = (randn(1,l) + randn(1,l)*i)*sqrt(var_n/2);

% No need to scale the channel matrix, automatically
% done when weighting symbols on transmit antennas

% simulate sending over the channel, received signal is
% y = H*w.'*s + n,
% where
% w.' = H'/sqrt(H * H')
% the weight array the signals are multiplied by
% before transmit

% The demand is that |w|^2 = (sqrt(w*w'))^2 == 1;

wt = H'/sqrt(H * H');

% what actually happens
%sw = wt*s;
%s_hat = H*sw + n;

% to save operations
wH = H*wt; % a scalar
s_hat = (wH * s) + n;
```

```
function s_hat = transceiver_SVA(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel.
%
% rx: the number of transmitter antennas
% tx: the number of transmitter antennas
% s:  signal to be sent
% H:  channel matrix
% snr: Signal-to-Noise-Ratio

% H: rx*tx
% s: l*1, l er antall symboler som skal sendes
% y: rx*l
% s_hat: 1*l
% wt: tx*1 og wr: rx*1

%H = (1/sqrt(tx))*H;

% Generating the transmit and receive weights
[U,S,V] = svd(H);

% These vectors are normalized by the svd
wt = V(:,1);
wr = conj(U(:,1));

% generate random noise, stat. different for every signal
% naming the noise matrix y, for reuse of variables
n = (randn(rx,l) + randn(rx,l)*i)*sqrt(var_n/2);

% the normalization is done to get s_hat values as
% close to s values as possible, but this is not
% crucial for the slicer (without norm., the
% symbols are just even further apart).
symb_fact = wr.' * H * wt;
s_hat = (symb_fact * s + wr.' * n);
```

```
function s_hat = transceiver_SMAL(rx,tx,s,H,var_n,l,C_albl)
%
% transceiver: Models a sending and receiving over a channel,
% using a combination of the Alamouti STC and SM schemes.
%
% rx: the number of transmitter antennas
% tx: the number of transmitter antennas
% s:  signal to be sent
% H:  channel matrix
% snr: Signal-to-Noise-Ratio

% H: rx*tx
% s: l*1, l er antall symboler som skal sendes
% y: rx*l
% s_hat: 1*l
% wt: tx*1 og wr: rx*1

% because of alamouti on 2 and 2 antennas
eff_spatial_diversity = tx/2;
num_transmits = l/eff_spatial_diversity;


s_mat = reshape(s,tx,l/tx);

% scaling the channel matrix H, to maintain
% total power equal 1, although
% sending on tx transmit antennas
H = H.*(1/sqrt(tx));

H_manip = zeros(2*rx,tx);



t1_index = C_albl{1};
t2_index = C_albl{2};
t2_len = length(t2_index);

% manipulating H into the right form
H_manip(1:2:2*rx-1,:) = H(:,t1_index);

H_manip(2:2:2*rx,1:2:tx-1) = -conj(H(:,t2_index(1:2:t2_len-1)));
H_manip(2:2:2*rx,2:2:tx)   =  conj(H(:,t2_index(2:2:t2_len)));

% generating noise that fits in size with the product H_manip*s_mat
```

```
n = (randn(2*rx,num_transmits/2) + ...
     randn(2*rx,num_transmits/2)*i)*sqrt(var_n/2);


y = H_manip * s_mat + n;
s_hat = pinv(H_manip)*y;

s_hat = reshape(s_hat,1,l);
```

```
function b_hat = slicer(s_hat,c,n)
%
% slicer: Maps the values in s_hat to bits using MRC
% returns bitvalues in b_hat
%
%

switch c
 case 1
  b_hat = slicer_binary(s_hat);
 case 2
  b_hat = slicer_qpsk(s_hat);
 case 3
  b_hat = slicer_16qam(s_hat);
 otherwise
  disp('Invalid coding scheme.');
end;
```

```
function b_hat = slicer_qpsk(s_hat,n)
%
% Takes the received symbols in s_hat and
% makes the decisions on what bits they are,
% slices them back to bitform and returns them
% in b_hat.
%
%


n_len = length(s_hat)*2;
b_hat_tmp = zeros(1,n_len);

u=1:2:(n_len-1);


b_hat_tmp(u) = real(s_hat(1,ceil(u/2)));
b_hat_tmp(u+1) = imag(s_hat(1,ceil(u/2)));

b_hat_tmp = b_hat_tmp./abs(b_hat_tmp);

b_hat = b_hat_tmp + abs((b_hat_tmp-1)/2);
```

```
function b_hat = slicer_16qam(s_hat)
%
% Takes the received symbols in s_hat and makes the decisions on
% what bits they are, slices them back to bitform and returns them
% in b_hat.
%
%

% Algorithm : Reorganize bitsPerSymbol bits at the time by
% switching bits number 2 and 3. Then the first two bits
% and the last two bits are both interpreted as follows:
% 01 :  3d
% 00 :   d
% 10 : - d
% 11 : -3d

% That means (going the other way)
% Know that 3 was normalized to .98487, and 1 to .3162
%  .98487 :  3 => 01
% -.98487 : -3 => 11
%  .3162  :  1 => 00
% -.3162  : -1 => 10

% Trick: Split in real and imaginary parts, take sign,
% and then round off, giving the following mapping:
% 0 : 0
% 1 : 1
% + : 0
% - : 1

M = 16;
bitsPerSymbol = log2(M);
b_len = length(s_hat) * bitsPerSymbol;


tmp_real_sign = -.5*(sign(real(s_hat))-1);
tmp_imag_sign = -.5*(sign(imag(s_hat))-1);


abs_all = [abs(real(s_hat)) abs(imag(s_hat))];
div_point = abs(mean(abs_all));

% finding what side of the "mean value"
% (in separate dimensions) each point is
```

```
tmp_real = sign(abs(real(s_hat))-div_point);
tmp_imag = sign(abs(imag(s_hat))-div_point);

tmp_real = (tmp_real+1)/2;
tmp_imag = (tmp_imag+1)/2;

b_hat = zeros(1,b_len);
b_arr = 1:bitsPerSymbol:b_len-(bitsPerSymbol-1);

b_hat(1,b_arr)   = tmp_real_sign;
b_hat(1,b_arr+1)= tmp_imag_sign;
b_hat(1,b_arr+2)= tmp_real;
b_hat(1,b_arr+3)= tmp_imag;

b_hat = round(b_hat./(b_hat + 0.001));
```

```
function ber = BER_calculation(b,b_hat)
%
% BER_CALCULATION   Calculates the BER for each transmission
% BER_CALCULATION(b,b_hat) returns the number of differences
% between the bit-values in b and b_hat. The original bits
% are contained in b and b_hat holds recovered bits at the
% receive side.
%


% the bit error rate (BER) is the percentage of bits in error,
% usually expressed as ten to a negative power. For example,
% BER = 10^(-6) means that 1 out of 1,000,000 bits is
% erroneously determined.

diff = b - b_hat;

ber = sum(abs(diff));
```

```
function allperms = permutation_matrix(tx)

%
% calling on permutate to generate all permutations,
% returns them in permutation_matrix
%

antenna_array = 1:tx;

alam_pairsize = 2;
num_alamblocks = tx/alam_pairsize;

pairs = nchoosek(1:tx,2);
num_pairs = size(pairs,1);

res=permutate(antenna_array,pairs);

[u,v] = find(res(:,tx)== 0);
res(u,:) = [];

allperms = res;
```

```
function patterns = permutate(antenna_array, poss_pairs)

%
% generates all possible patterns for SMAL given
% all permutations of two numbers between 1 and tx
%

curr_tx = length(antenna_array);
num_sp_comb = curr_tx - 1;
sp_combs_index = antenna_array(1,2:curr_tx);

sp_combs = [antenna_array(1)*ones(num_sp_comb,1) ...
            sp_combs_index'];


if curr_tx == 2
  num_patterns = 1;
elseif curr_tx == 4
  num_patterns = 3;
elseif curr_tx == 6
  num_patterns = 11;
elseif curr_tx == 8
  num_patterns = 58;
elseif curr_tx == 10
  num_patterns = 100;
end;

patterns = zeros(num_patterns,curr_tx);

% cut-off
if num_patterns == 1 & size(poss_pairs,1) ~= 0
  patterns = poss_pairs(1,:);
elseif num_patterns == 1 & size(poss_pairs,1) == 0
  patterns = [];
else

  next_antenna_array = zeros(1,curr_tx-2);
  next_tx = curr_tx - 2;
  num_next_poss_pairs = factorial(next_tx)/...
      (factorial(next_tx-2)*factorial(2));
  next_poss_pairs = zeros(num_next_poss_pairs,2);


  counter = 1;
```

```
for sp = 1:num_sp_comb

  % finding the current start pair
  curr_sp = sp_combs(sp,:);

  % finding the rest, possible pairing pairs
  ind_pairs = find((curr_sp(1,1)~=poss_pairs(:,1)) & ...
                   (curr_sp(1,2)~=poss_pairs(:,1)) & ...
                   (curr_sp(1,1)~=poss_pairs(:,2)) & ...
                   (curr_sp(1,2)~=poss_pairs(:,2)));

                   %(inv_curr_sp(1,1)~=poss_pairs(:,1)) & ...
                   %(inv_curr_sp(1,2)~=poss_pairs(:,2)));

  next_poss_pairs = poss_pairs(ind_pairs,:);


  % finding the antennas allowed to appear in
  % these next possible pairs
  ind_array = find(curr_sp(1,1)~=antenna_array(1,:) ...
                   & curr_sp(1,2)~= antenna_array(1,:));
  next_antenna_array = antenna_array(ind_array);

  ret_patterns = permutate(next_antenna_array,next_poss_pairs);

  % must fit the returned patterns with the current
  % start-pair curr_sp
  if size(ret_patterns,1) ~= 0
    curr_ind_pat = size(ret_patterns,1) * (counter - 1) + 1;

    patterns(curr_ind_pat:curr_ind_pat+size(ret_patterns,1)-1,:) = ...
        [repmat(curr_sp,size(ret_patterns,1),1) ret_patterns];
    counter = counter +1;

    % finding the inverse of that start pair,
    % to remove it from the group, but only if
    % the start pair was used, i.e. not cut off with
    % no pattern
    last_tx = antenna_array(curr_tx);
    % finding the right type of start pair, expressed by 1?
    sp_type = curr_sp - curr_sp(1) + 1;

    inv_curr_sp = fliplr([last_tx last_tx] - sp_type + [1 1]);
```

```
        [invind_pairs,c,d] = intersect(inv_curr_sp,poss_pairs,'rows');

        % removing this pair, know it is at most 1 appereance
        if size(invind_pairs,1) ~= 0
          poss_pairs(d,:) = [];
        end;

      end;


    end;


end;
```

```
function C_albl = get_pattern(combocase,tx)

%
% GET_PATTERN(combocase,tx) returns the requested pattern
%                           for a given tx, used with SMAL
%

t1_index = zeros(1,tx);
t2_index = zeros(1,tx);


% combocases4 = [1 2 3 4;1 3 2 4;1 4 2 3];
% combocases6 = [1 2 3 4 5 6; 1 2 3 5 4 6;
%                1 2 3 6 4 5; 1 3 2 5 4 6;
%                1 3 2 6 4 5; 1 4 2 5 3 6;
%                1 4 2 6 3 5; 1 5 2 6 3 4;
%                1 6 2 3 4 5; 1 6 2 4 3 5;
%                1 6 2 5 3 4];


% finding all transmission patterns for a given tx
all_patterns = permutation_matrix(tx);

t1_index = all_patterns(combocase,:);

t2_index(1,1:2:tx-1) = t1_index(1,2:2:tx);
t2_index(1,2:2:tx)   = t1_index(1,1:2:tx-1);

C_albl = {t1_index t2_index};
```

```
function [C_albl, index] = get_pattern_inst(H, H_manip, ...
                                            allperms, mod_dist)

% SMAL: finds the best assignment pattern for transmission
% under the current channel (knows the instantaneous channel)

tx = size(H,2);
rx = size(H,1);
H = H.*(1/tx);

num_patterns = size(allperms,1);
min_distance = zeros(num_patterns,1);
alam_pairsize = 2;

for counter=1:num_patterns

  t1_index = allperms(counter,:);

  t2_index = zeros(size(t1_index));
  t2_index(1,1:2:tx-1) = t1_index(1,2:2:tx);
  t2_index(1,2:2:tx)   = t1_index(1,1:2:tx-1);

  H_manip(1:2:2*rx-1,:) = H(:,t1_index);

  t2_len = length(t2_index);
  H_manip(2:2:2*rx,1:2:tx-1) = -conj(H(:,t2_index(1:2:t2_len-1)));
  H_manip(2:2:2*rx,2:2:tx)   =  conj(H(:,t2_index(2:2:t2_len)));

  % checking bound
  sing_vals = svd(H_manip);
  % sing_vals(min(tx,2*rx)) is the smallest singular value
  min_distance(counter) = sqrt(sing_vals(min(tx,2*rx))^2*mod_dist^2);
end;


[max_dist,index]=max(min_distance);

t1_index = allperms(index,:);
t2_index(1,1:2:tx-1) = t1_index(1,2:2:tx);
t2_index(1,2:2:tx)   = t1_index(1,1:2:tx-1);

C_albl = {t1_index t2_index};
```

```
function [C_albl, index] = get_pattern_rand(tx, rx, allperms)

% finds a random assignment pattern for transmission
% under the current channel. Used for comparison with SMAL

num_patterns = size(allperms,1);

% choose the pattern first in random
% permutation of all patterns

permutation = randperm(num_patterns);
pattern = permutation(1);
t1_index = allperms(pattern,:);
t2_index = zeros(size(t1_index));
t2_index(1,1:2:tx-1) = t1_index(1,2:2:tx);
t2_index(1,2:2:tx)   = t1_index(1,1:2:tx-1);

index = pattern;
C_albl = {t1_index t2_index};
```

```
function mod_dist = eucldist_modulation(c)

%
% EUCL_DIST(c) returns the euclidean distance of the
%              modulation scheme given by c
%

if c == 1
  mod_dist = 2; % bpsk
elseif c == 2
  mod_dist = sqrt(2); % qpsk or 4QAM
elseif c == 3
  mod_dist = 1/sqrt(5);
end;
```

```
function [n,numTimes] = num_bits(c,rx,tx,d,exp_BER,maxChannelIndex)
%
% Computes the number of bits to be generated for each new
% channel matrix


eachTime = 500000; % to avoid swapping
total = round(((1/exp_BER)*100));


if c == 1
  M = 2;
elseif c == 2
  M = 4;
elseif c == 3
  M = 16;
else
  disp('Hmmm');
end;

bitsPerSymbol = log2(M);

% diffSymbolsPerRound is how many different symbols
% are sent at once
if d == 0
  diffSymbolsPerRound = 1;
elseif d == 1
  diffSymbolsPerRound = tx;
elseif d == 2
  diffSymbolsPerRound = 2; % average 1
elseif d == 3
  diffSymbolsPerRound = tx; % average 1
end;

eachTime = round(eachTime/(bitsPerSymbol * diffSymbolsPerRound)) * ...
      (bitsPerSymbol*diffSymbolsPerRound);

if bitsPerSymbol*tx > total & d == 1
  disp('Problems, need more bits.');
end;


if floor(total/eachTime) == 0
  numTimes = 1;
```

```
  n = round(total/(bitsPerSymbol * diffSymbolsPerRound)) * ...
      (bitsPerSymbol*diffSymbolsPerRound);
else
  numTimes = floor(total/eachTime);
  n = eachTime;
end;
```