MBS-SPS & TTE-ECR
Low power analog/digital array signal processing at the 60 GHz band

by J.H.C. van den Heuvel

Report of Master’s project
carried out from February 2006 to October 2006.

Supervisors:
Dr. R. Rietman (Philips)
Dr. ir. P.C.W. Sommen (TU/e)
Dr. ir. M.H.A.J. Herben (TU/e)

Professor:
Prof. dr. ir. E.R. Fledderus

The Faculty of Electrical Engineering of Eindhoven University of Technology disclaims all responsibility for the contents of traineeship and graduation reports.
Abstract

In conventional array signal processing every antenna element is sampled at, or above, the Nyquist rate. Unfortunately, this architecture is not suitable for the 60 GHz band due to power constraints. An option to reduce power consumption is to perform part of the signal processing in the analog domain. Previous research indicates that an architecture consisting of several receive antennas, analog filters and one AD-converter operating at the Nyquist rate can achieve good results in terms of signal-to-noise ratio (SNR), assuming that the receiver sets the filters for maximum ratio combining (MRC). In this report, we investigate how such a setting can practically be achieved. A hybrid architecture that can acquire information from the individual antennas, but still offers the low power benefits of the analog architecture is proposed. The hybrid architecture samples every antenna element below the Nyquist rate and a combined signal is sampled by an AD-converter operating at the Nyquist rate. The signals are processed by an analog filter before being combined. Since the antenna elements are sampled below the Nyquist rate, a channel estimation of each antenna element is no longer necessarily unique. To obtain a channel estimation, an iterative algorithm is used. The signals are under-sampled, therefore the problem is non-linear. The algorithm, which is used to solve the set of equations, is based on the non-linear Conjugate Gradient method. Constraints exist with regard to the minimal number of required samples; one sample per variable, estimated by the iterative algorithm, is required. In order to reduce the number of required samples and to obtain a direct estimate for the parameter values of the analog filters, the problem is approximated by a set of linear equations. The obtained parameter values of the filters do not necessarily achieve the global optimum in terms of SNR. Simulations, performed in a realistic environment, show excellent results in terms of SNR. A typical hybrid architecture, which can obtain an SNR equal to, or higher than, a conventional array, requires 86 % less power for the AD-conversion. However, the hybrid architecture requires twice the number of receive antennas. The most likely antenna element used on the 60 GHz band is the patch antenna element, due to production-cost and chip-integration considerations. Therefore, a small increase in the number of antenna elements will not lead to a considerable increase in production costs. Simulations also show that the algorithm solves the set of linear equations in just one iterative step, indicating simpler algorithms can be used to solve the set of equations.
# Contents

Abbreviations v

List of symbols vii

1 Introduction 1

2 System design considerations 3

2.1 Receiver model .................................................. 3

2.2 Signal model .................................................... 3

2.2.1 Legal transmit power ........................................ 3

2.2.2 Packet based .................................................. 4

2.2.3 OFDM ........................................................... 5

2.2.4 Noise ............................................................ 5

2.3 Channel models .................................................. 6

3 Effects of under-sampling and up-sampling 9

3.1 Introduction ...................................................... 9

3.2 Z-transform ...................................................... 9

3.2.1 Under-sampling .............................................. 10

3.2.2 Up-sampling .................................................... 11

4 Channel estimation 15

4.1 Hybrid architecture ............................................. 15
## 4.2 Unique solution

- 4.2.1 Z-transform
- 4.2.2 Polyphase decomposition
- 4.2.3 Multiple channels

## 4.3 Conclusions

## 5 Iterative channel estimation

- 5.1 Model
- 5.2 Iterative estimation
  - 5.2.1 Steepest Descent
  - 5.2.2 Conjugate Gradient
  - 5.2.3 Nonlinear Conjugate Gradient
- 5.3 Filter setting
- 5.4 Conclusions

## 6 Simulation Results

- 6.1 Introduction
- 6.2 Data handling
- 6.3 A $4 \times 4$ receiver array
  - 6.3.1 Line of sight
  - 6.3.2 Non line of sight
- 6.4 A $9 \times 1$ receiver array
  - 6.4.1 Line of sight
  - 6.4.2 Non line of sight
- 6.5 Conclusions

## 7 Conclusions and Recommendations

- 7.1 Future research
Abbreviations

AD analog-to-digital conversion
BPSK binary phase shift keying
CG Conjugate Gradient
FFT fast Fourier transform
IEEE Institute of Electrical and Electronic Engineers
IFFT inverse fast Fourier transform
LOS line-of-sight
MRC maximum ratio combining
NLOS non-line-of-sight
OFDM orthogonal frequency division multiplexing
RPS Radio Propagation Simulator
SNR signal-to-noise ratio
## List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>data structure consisting out of a set of columns of different length</td>
</tr>
<tr>
<td>( a_m )</td>
<td>complex amplitude</td>
</tr>
<tr>
<td>( a_q[n] )</td>
<td>signal in the time-discrete domain</td>
</tr>
<tr>
<td>( a_{m,q} )</td>
<td>element of data structure ( a )</td>
</tr>
<tr>
<td>( a_{m,q} )</td>
<td>complex amplitude</td>
</tr>
<tr>
<td>( A )</td>
<td>function used in non-linear CG to calculate A-orthogonal vectors</td>
</tr>
<tr>
<td>( A_q(z) )</td>
<td>signal in the ( z )-domain</td>
</tr>
<tr>
<td>( b )</td>
<td>a vector containing complex values</td>
</tr>
<tr>
<td>( b_q[n] )</td>
<td>signal in the time-discrete domain</td>
</tr>
<tr>
<td>( B )</td>
<td>bandwidth containing both negative and positive frequencies</td>
</tr>
<tr>
<td>( B_q(z) )</td>
<td>signal in the ( z )-domain</td>
</tr>
<tr>
<td>( c )</td>
<td>the speed of light in vacuum ( 3 \times 10^8 ) m/s</td>
</tr>
<tr>
<td>( \vec{c} )</td>
<td>a vector containing complex values</td>
</tr>
<tr>
<td>( c_q )</td>
<td>component of vector ( \vec{c} )</td>
</tr>
<tr>
<td>( c_q )</td>
<td>complex attenuation</td>
</tr>
<tr>
<td>( ^\circ C )</td>
<td>degrees Celsius</td>
</tr>
<tr>
<td>( d_{(i)} )</td>
<td>search vector of the ( i )th iteration</td>
</tr>
<tr>
<td>( dB )</td>
<td>decibel</td>
</tr>
<tr>
<td>( D )</td>
<td>element of a set of natural numbers not containing zero</td>
</tr>
<tr>
<td>( D )</td>
<td>up-sample or down-sample factor</td>
</tr>
<tr>
<td>( e )</td>
<td>base of natural logarithm ( e = 2.7183 )</td>
</tr>
<tr>
<td>( e_1, e_2 )</td>
<td>vectors of size ( n )</td>
</tr>
<tr>
<td>( f )</td>
<td>frequency (Hz)</td>
</tr>
<tr>
<td>( f_c )</td>
<td>carrier frequency</td>
</tr>
<tr>
<td>( f_q(t) )</td>
<td>filter in the time domain</td>
</tr>
<tr>
<td>( f_q[n] )</td>
<td>filter in the time-discrete domain</td>
</tr>
<tr>
<td>( F_q(f) )</td>
<td>filter in the frequency domain</td>
</tr>
<tr>
<td>( F_q(z) )</td>
<td>filter in the ( z )-domain</td>
</tr>
<tr>
<td>( F_{noise} )</td>
<td>noise figure</td>
</tr>
<tr>
<td>( g(t) )</td>
<td>signal in the time domain</td>
</tr>
<tr>
<td>( g[n] )</td>
<td>signal in the time-discrete domain</td>
</tr>
<tr>
<td>( G(f) )</td>
<td>signal in the frequency domain</td>
</tr>
<tr>
<td>( G(z) )</td>
<td>signal in the ( z )-domain</td>
</tr>
<tr>
<td>( h(t) )</td>
<td>transfer function in time domain</td>
</tr>
<tr>
<td>( h[n] )</td>
<td>transfer function in time-discrete domain</td>
</tr>
<tr>
<td>( H(f) )</td>
<td>transfer function in frequency domain</td>
</tr>
<tr>
<td>( H(z) )</td>
<td>transfer function in ( z )-domain</td>
</tr>
<tr>
<td>( i )</td>
<td>an integer</td>
</tr>
<tr>
<td>( j )</td>
<td>the imaginary number ( \sqrt{-1} )</td>
</tr>
<tr>
<td>( k )</td>
<td>Boltzmann’s constant ( 1.38 \times 10^{-23} ) joule/K</td>
</tr>
<tr>
<td>( k )</td>
<td>an integer</td>
</tr>
</tbody>
</table>
\( K \) is an integer
\( K \) number of frequencies
\( K \) degrees Kelvin (°C+273)
\( \bar{L} \) a vector
\( L_k \) component of vector \( \bar{L} \)
\( L_k \) OFDM data symbol
\( m \) an integer
\( M \) an integer
\( M \) number of rays arriving at a receive antenna
\( n \) an integer
\( N \) an integer
\( N \) number of samples
\( N \) noise power
\( p[n] \) periodic train of impulses
\( Q \) number of receive antennas
\( r_{(i)} \) residu of an iterative step
\( s(t) \) output signal in the time domain
\( s[n] \) output signal in the time-discrete domain
\( S(f) \) output signal in the frequency domain
\( S(z) \) output signal in the \( z \)-domain
\( t \) time
\( t_{1}...t_{10} \) short training symbols
\( T \) a time interval
\( T \) absolute temperature (Kelvin)
\( T_1, T_2 \) long training symbols
\( T_{Symbol} \) symbol time
\( u_i \) part of a set of linearly independent vectors
\( u_q[n] \) signal in the time-discrete domain
\( U_q(z) \) signal in the \( z \)-domain
\( v_q[n] \) signal in the time-discrete domain
\( V_q(z) \) signal in the \( z \)-domain
\( W \) Bandwidth containing only positive frequencies
\( x \) a variable
\( x(t) \) input signal in the time domain
\( x[n] \) input signal in the time-discrete domain
\( X(f) \) input signal in the frequency domain
\( X(z) \) input signal in the \( z \)-domain
\( y \) a variable
\( y(t) \) output signal in the time domain
\( y[n] \) output signal in the time-discrete domain
\( Y(f) \) output signal in the frequency domain
\( Y(z) \) output signal in the \( z \)-domain
\( z \) a variable
\( z \) \( z \)-domain with \( z = e^{j2\pi fT} \)
\( Z^{-q} \) delay in the time-discrete domain with \( Z = e^{j2\pi fT} \)
\( \alpha \) a constant
\( \beta \) a constant
\( |\Delta SNR|_{100} \) average of the absolute deviation of the SNR of 100 simulations
\( \theta \) a variable
\( \lambda \) wavelength
\( \lambda_c \) carrier wavelength
\( \pi \) 3.14159
\( \sigma \) a constant
τ: a delay time

Z: a data structure consisting of a set of columns of different length

τ_{m,q}: element of data structure Z

τ_{m,q}: a delay time

φ: a variable
Chapter 1

Introduction

The future of electronic devices such as computers, telephones and cameras is currently perceived as being “wireless”. The benefit of having wireless devices is obvious; the mesh of cables, needed to connect all devices, can be avoided. Next to the demand of having wireless devices, consumers desire them to achieve an ever higher “data-rate”. A problem of desiring wireless high data-rate devices is an increasing demand on limited spectral space. This could lead to a choking of existing systems, even more so when the amount of wireless devices rises. An option, to relieve the demand on limited spectral space, is to divide the urban environment into “micro-cells”. Typical examples of micro-cells are offices and rooms.

An option, for wireless high data-rate communication in a micro-cell environment, which enjoys a lot of attention at the moment is the 60 GHz band. A massive license free spectral space (approximately 5 to 7 GHz) is allocated worldwide, for dense wireless communication, in this band [1]. Due to the availability of the spectral space and the attenuation properties (oxygen absorbs radiation at 60 GHz thereby limiting the communication range), the 60 GHz band is considered the most likely candidate to fulfill our view of the near future: wireless, high data-rate, micro-cells. Although it may seem the choking of existing systems is avoided and a 60 GHz network can be build, still problems exist which need to be considered. A significant problem is the power consumption of AD-converters.

The power consumption of AD-converters is almost linear in their sample rate. The AD-converters sample a broad-band signal, which is transmitted over the 60 GHz band. Several channels (in the 5 to 7 GHz of spectral space) may be available, each channel being approximately 1 to 2 GHz wide. Due to a limit of legal transmit power, it is necessary to use smart antenna structures and thus multiple antennas [2, 3, 4, 5, 6, 7]. Target applications will involve mobile devices, with a limited power supply. In conventional array signal processing, every receive antenna is sampled at, or above, the Nyquist rate. Unfortunately, this is not possible due to power constraints. An option to reduce power consumption is to lower the sample rate the AD-converters are operating at and perform part of the signal processing in the analog-domain. The main focus is on the ability of smart antenna structures to estimate the channel, of every antenna branch, when the AD-converters are running below the Nyquist rate.
Chapter 2

System design considerations

A description of the considered system is given.

2.1 Receiver model

In previous research [8], an architecture consisting of several antennas, analog filters and one fast AD-converter was considered. The architecture is shown in Figure (2.1), where $F_q$ is the filter at the $q^{th}$ antenna. In this architecture the number of AD-converters is extremely reduced, resulting in power consumption savings for the AD-conversion. It is shown that for larger bandwidths, a filter consisting of a single phase shifter yields comparable results to more complex filters in terms of signal to noise ratio (SNR) at the AD-converter. This research compares a phase shifter and a single tapped delay line. More complex filters are considered unpractical, since they introduce more analog components which need to be set by the digital controller. If the single phase shifters are set for maximum ratio combining (MRC), the results are comparable to a matched filter. The loss in terms of SNR, for a worst case scenario, is approximately 3.0 dB.

An option to compensate the loss in terms of SNR, is increasing the number of antenna elements. This seems a costly solution to boost the SNR to a desired level. However, the antenna element most likely used, is the patch antenna [9, 10, 11, 12, 13, 14, 15]. This is due to production-cost and chip-integration considerations. A patch antenna, operating at the 60 GHz band, is a small metallic slap (approximately 1 to 4 mm$^2$ in size) which can be integrated on a chip. Therefore, a small increase in the number of patch antennas will not lead to a considerable increase in production costs.

2.2 Signal model

2.2.1 Legal transmit power

From literature [1] it is known that the legal transmit power limit, for the 60 GHz band, is 20 dBm.
2.2.2 Packet based

The most likely protocol for the 60 GHz band is a protocol similar to the IEEE 802.11a standard, which is currently in use for the 5 GHz band. In this protocol a number of data symbols are preceded by a preamble. During the preamble the receiver is trained and the channel is estimated. It is assumed the channel is time invariant, during the transmission of the data symbols. Such a system is called "packet based". For our protocol a similar packet based system is assumed. Figure (2.2) shows the orthogonal frequency division multiplexing (OFDM) training structure as is used in the IEEE 802.11a standard. In this figure, $t_1$ to $t_{10}$ denote short training symbols and $T_1$ and $T_2$ denote long training symbols. The preamble is followed by the SIGNAL field and DATA. The SIGNAL field contains information about the modulation and the packet length. The DATA contains the data symbols the transmitter sends to the receiver. The guard intervals $GI$ and $GI2$ are chosen such, that the data symbols and the training symbols $T_1$ and $T_2$ are a convolution with the channel in the time-domain.
2.2.3  OFDM

In an OFDM signal the bandwidth is divided into several subcarriers each having their own data symbol. The baseband equivalent OFDM signal is expressed as

\[ x(t) = \sum_{k=-K}^{K-1} L_k e^{(j2\pi B_k t)}, \quad (2.1) \]

where \( t \) is the time, \( B \) is the bandwidth of the signal, \( K \) is the number of OFDM frequencies and \( L_k \) is the data symbol of the \( k \)th OFDM frequency. In the case of a bandwidth \( B = 1 \) GHz and a number of OFDM frequencies \( K = 256 \) the symbol time equals \( T_{Symbol} = \frac{256}{10^9} = 256 \) ns.

The length of the long training symbols, \( T_1 \) and \( T_2 \), is the symbol time.

Two functions \( x_n(t) \) and \( x_m(t) \) are said to be orthogonal over the interval \( a < t < b \) if they satisfy the condition

\[ \int_{\alpha}^{\beta} x_n(t) x_m^*(t) dt = 0, \quad (2.2) \]

where \( m \neq n \). In an OFDM signal the subcarriers are spaced \( 1/T_{Symbol} \) Hz apart and the time of a data symbol is \( T_{Symbol} \). Because the data symbol time is \( T_{Symbol} = \frac{B}{K} \), it is assured the carriers satisfy the orthogonality criterion. This is easily shown with the following equations

\[ \int_{\alpha}^{\alpha+T_{Symbol}} x_n(t) x_m^*(t) dt = \int_{\alpha}^{\alpha+T_{Symbol}} e^{(j2\pi f_n t)} L_n L_n^* e^{(-j2\pi f_m t)} dt \]
\[ = L_n L_m^* \int_{\alpha}^{\alpha+T_{Symbol}} e^{(j2\pi (n-m) \frac{B}{K} t)} dt = L_n L_m^* e^{(j2\pi (n-m) \frac{B}{K} T_{Symbol})} \left( e^{(j2\pi (n-m) \frac{B}{K} T_{Symbol})} - 1 \right) = 0, \quad (2.3) \]

because \( m \neq n \) and \( e^{(j2\pi (n-m) \frac{B}{K} T_{Symbol})} = 1 \). The benefit, of ensuring the subcarriers are orthogonal, is that they will not interfere with each other. This quality is independent and unaffected by the channel which may exist between the transmitter and receiver; as long as the channel is constant during the symbol time and the guard interval is chosen sufficiently large to assume a convolution between the symbol and the channel.

2.2.4  Noise

The system is affected by white noise. The noise power at a receive antenna is equal to

\[ N = 10 \log(kTW F_{noise}), \quad (2.4) \]

where \( k \) is Boltzmann’s constant \( (k = 1.38 \cdot 10^{-23}) \), \( T \) is the equivalent noise temperature of the receiver \( (T = 290 \) K), \( W \) is the positive noise bandwidth and \( F_{noise} \) is the receiver noise.
figure. Assuming a receiver noise figure of about 10 dB and a positive noise bandwidth of 1 GHz, the receiver noise power amounts to $N = -74$ dBm. Note that the entire noise bandwidth is $B = 2W$.

### 2.3 Channel models

Several options to model a channel between a transmitter and a receiver exist. A simple model is the “two ray” model. This model assumes two rays make a connection between the transmitter and the receiver. This can either be a direct line of sight and a single reflection or two reflections, each having its own unique delay and complex amplitude. The channel is now defined as

$$h(t) = \sum_{m=1}^{2} a_m e^{j2\pi f(t-\tau_m)}, \tag{2.5}$$

where $a_m$ and $\tau_m$ are the complex amplitude and delay of the $m^{th}$ ray. The two ray model is a model that is often used in preliminary research.

A more complex way to model the channel, between a transmitter and a receiver, is using a ray-tracing software tool. Modeling a room or an urban area can be done using this tool. The software tool calculates the number of rays, which make a connection between the receiver and transmitter, before they are attenuated below a noise threshold. This channel model is defined as

$$h(t) = \sum_{m=1}^{M} a_m e^{j2\pi f(t-\tau_m)}, \tag{2.6}$$

where $a_m$ and $\tau_m$ are the complex amplitude and delay of the $m^{th}$ ray and $M$ is the total number of significant rays as specified by the ray-tracing software tool. The used ray-tracing software tool, for this research, is Radio Propagation Simulator (RPS) of Radioplan. In the ray tracing simulator a model of an existing room is constructed, which is modified for the 60 GHz band. In the original room, measurements on the 60 GHz band have been conducted [2]. Figure (2.4) and Figure (2.3) show the modeled room. The room contains a large cabinet stretching from the floor to the roof. This allows both line of sight (LOS) situations and non line of sight situations (NLOS). Figure (2.3) shows a top view of the room. Figure (2.4) shows a 3-dimensional view of the modeled room when used in RPS. RPS exports raw and modified data. In case of modified data, rays are put in a time bin of size $\frac{B}{2\pi}$, where $B$ is the bandwidth. If several rays arrive during a time bin, they are added up coherently. The raw data of a typical impulse response is shown in Figure (2.5). In figure (2.5) each square block corresponds to the amplitude and the time-of-arrival of a significant ray. Figure (2.6) shows the same typical impulse response with a bin-size of 0.5 ns. In this figure each circle corresponds to the data point of a time-bin.
Figure 2.3: Top view of the modeled room.

Figure 2.4: The room modeled in RPS.
Figure 2.5: Raw data of a typical impulse response for a receive antenna in the LOS position.

Figure 2.6: Modified data of a typical impulse response for a receive antenna in the LOS position with a time-bin of 0.5 ns.
Chapter 3

Effects of under-sampling and up-sampling

The effects of under-sampling and up-sampling on a digital signal are outlined.

3.1 Introduction

In conventional array signal processing every antenna element is sampled at, or above, the Nyquist rate and all signal processing is performed in the digital domain. In previous research \[8\] it is discussed a system with only one AD-converter and a set of analog filters achieves satisfying results in terms of SNR. However, the question of how the architecture should be modified so it is able to calculate the optimal values for the phase shifters remains? Let us consider a conventional receiver array. The problem of the array is the power consumption, which is linear in the sample rate. Therefore, a scaling to the 60 GHz band of a conventional array conflicts with power constraints. The main idea to overcome the problem is simple; lower the sample rate. A decrease of the sample rate arises some questions; “What are the effects of lowering the sample rate?” “What happens to the received signals if they are under-sampled?” To answer these questions the next few sections provide an introduction into the Z-transform and the effects of under-sampling and up-sampling.

3.2 Z-transform

Signal processing will occur in a digital controller. The signal, which will be analyzed by the digital controller is a stream of samples of a time continuous signal. If a time continuous signal $g(t)$ is sampled by an AD-converter at an interval $T$, the relation between the sampled signal and the time continuous signal is given by

$$g[n] = g(nT), \quad (3.1)$$

where $n$ is the $n^{th}$ sample of the time continuous signal. The relation between the sampled
Figure 3.1: Under-sampling of a signal represented by a fast AD-converter and a down-sampler.

A signal and its \( z \)-transform is given by

\[
G(z) = \sum_{n=-\infty}^{\infty} g[n] z^{-n}.
\]  

(3.2)

where \( z = e^{j2\pi f T} \), with \( f \) the frequency and \( T \) the sample time. Note that this corresponds to the Fourier transform.

### 3.2.1 Under-sampling

A signal sampled at, or above, the Nyquist rate can be reconstructed from the samples. Sampling below the Nyquist rate results in a loss of information. This information loss results in a mirroring of higher frequencies into lower frequencies, aliasing. An under-sampling AD-converter can be modeled as an AD-converter operating at the Nyquist rate which is down-sampled by a factor \( D \). Figure (3.1) shows how a time continuous input signal \( x(t) \) is sampled at the Nyquist rate and down-sampled with a factor \( D \). This scheme corresponds to an AD-converter operating at \( 1/D^{th} \) of the Nyquist rate. Consider a time discrete signal \( x[n] \) leaving the fast AD-converter and entering the down-sampler. The signal leaving the down-sampler \( y[m] \) is equal to [16]

\[
y[m] = x[mD].
\]  

(3.3)

this can be rewritten as

\[
y[m] = x[mD]p[mD] = \tilde{x}[mD],
\]  

(3.4)

where \( \tilde{x}[n] \) is defined as

\[
\tilde{x}[n] = \begin{cases} 
  x[n], & \text{for } n = 0, \pm D, \pm 2D, \ldots \\
  0, & \text{otherwise}
\end{cases}
\]  

(3.5)

with

\[
\tilde{x}[n] = x[n]p[n]
\]  

(3.6)

and

\[
p[n] = \begin{cases} 
  1, & \text{for } n = 0, \pm D, \pm 2D, \ldots \\
  0, & \text{otherwise}
\end{cases}
\]  

(3.7)
The closed form representation of \( y[m] \) is rewritten to include a function \( p[n] \). This is a mathematical method to find an easy way to calculate the \( z \)-transform of the signal. To clarify this, the Discrete Fourier series representation of \( p[n] \) is introduced. For clarity, all mathematical steps required to find the solution are included. The Discrete Fourier series representation of \( p[n] \) is given by

\[
p[n] = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{kn}{D}}.
\] (3.8)

The \( z \)-transform of the signal can be found with Equation [3.2] and equals

\[
Y(z) = \sum_{m=-\infty}^{\infty} y[m] z^{-m} = \sum_{m=-\infty}^{\infty} \tilde{x}[mD] z^{-m} = \sum_{m=-\infty}^{\infty} \tilde{x}[m] z^{-\frac{m}{D}} = \sum_{m=-\infty}^{\infty} x[m] \left[ \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{km}{D}} \right] z^{-\frac{m}{D}} = \frac{1}{D} \sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} x[m] (e^{-j2\pi \frac{k}{D} z} z^{-m}) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{-j2\pi \frac{k}{D} z}).
\] (3.9)

Now a closed form representation in the \( z \)-domain is found. Figure (3.2) shows the change in the spectrum of a time continuous signal which is sampled at twice the Nyquist rate and then down-sampled by a factor \( D \).

### 3.2.2 Up-sampling

A signal can be up-sampled by adding zeros between the samples, Figure (3.4). The output signal \( y[n] \) of the up-sampler is equal to [16]

\[
y[n] = \begin{cases} 
  x[n/D], & \text{for } n = 0, \pm D, \pm 2D, \ldots \\
  0, & \text{otherwise}.
\end{cases}
\] (3.10)

The \( z \)-transform of the signal can be found with Equation [3.2] and equals

\[
Y(z) = \sum_{n=-\infty}^{\infty} y[n] z^{-n} = \sum_{m=-\infty}^{\infty} x[n/D] z^{-n} = \sum_{m=-\infty}^{\infty} x[n] z^{-nD} = X(z^D).
\] (3.11)
Figure 3.2: Change of the spectrum due to sampling and down-sampling: (a) spectrum of the original time continuous signal, (b) spectrum of the signal when sampled at twice the Nyquist rate, (c) and (d) spectrum of the signal when down-sampled with a factor $D = 2$ and $D = 3$.

Figure 3.3: Change of the spectrum due to sampling and up-sampling: (a) spectrum of the original time continuous signal, (b) spectrum of the signal when sampled at twice the Nyquist rate, (c) and (d) spectrum of the signal when up-sampled with a factor $D = 2$ and $D = 3$. 
Equation (3.11) shows that the only effect of up-sampling is a scaling of $z$ in $X$. This is caused by a change in the time constant, for $X$ it is $T$ and for $Y$ it is $\frac{1}{D} T$. Figure (3.3) shows the change in the spectrum of a time continuous signal which is sampled at twice the Nyquist rate and then up-sampled by a factor $D$. 

Figure 3.4: Up-sampling of a signal by adding zeros between the samples.
Chapter 4

Channel estimation

The possibility to find a unique channel estimation from aliased spectra is explored

4.1 Hybrid architecture

Let us consider a conventional antenna array architecture. In conventional array signal processing every antenna is sampled at, or above, the Nyquist rate (Figure (4.1)). This allows a digital channel estimation for every antenna. Using this estimation, a digital matched filter per antenna can be implemented in the digital controller. Unfortunately, this architecture is not suitable for the 60 GHz band due to the power consumption of the AD-converters. The proposed solution is to perform part of the signal processing with analog filters and sample the combined signal with an AD-converter operating at the Nyquist rate. This architecture is shown in Figure (2.1). However, this architecture is not complete since the values of the analog filters need to be set by the receiver. Figure (4.2) shows a hybrid architecture, where classical array signal processing and the architecture using analog filters are combined. In order to be more energy efficient with respect to the conventional architecture, the AD-converters directly attached to the antenna elements need to operate at a sample rate below the Nyquist rate.

4.2 Unique solution

The demand to sample every antenna below the Nyquist rate means all input signals are aliased. This leads to the question if it is possible to reconstruct the channel of every antenna and use this information to find optimal values for the analog filters in terms of SNR. First the possibility to find a unique solution for the channel estimation is considered. A unique solution is preferred because it guarantees the optimal solution is found. Let us take a closer look at the proposed architecture. To find a unique solution we would like to perform an FFT on the input signal of every receive antenna. Unfortunately, not enough samples are available to perform an FFT. However, it is possible to alter the architecture in such a way that there would at least be a sample at every time slot which is required to perform an FFT of a single branch. From literature [16] it is known that an array of slow AD-converters can be modified to mimic a fast AD-converter. This is called a polyphase decomposition scheme.
Figure 4.1: Architecture in which every antenna is sampled at, or above, the Nyquist rate.

Figure 4.2: Hybrid architecture in which every antenna is sampled below the Nyquist rate and a combined signal is sampled at the Nyquist rate.
The hybrid architecture is modified on the basis of the polyphase decomposition scheme (Figure (4.3)). In the architecture every antenna is progressively sampled by a “slow” AD-converter, which is symbolized in Figure (4.3) by a sufficiently fast AD-converter down-sampled by a factor \( D \). The delays, which symbolize the progressive sampling between the branches, are represented by \( Z^{-q} \), where \( q \) is the \( q^{th} \) antenna branch. The factor \( Z = e^{j2\pi fT} \), where \( f \) is the frequency and \( T = \frac{1}{f_{Nyquist}} \) is the sample time of the Nyquist rate. The output signals of the slow AD-converters are up-sampled by a factor \( D \) by adding zeros between the samples. Afterwards, they are delayed again to put every sample which is non zero in the correct time slot with respect to each other. The total delay of the sub-sample scheme will therefore be \( Z^{-(Q-1)} \), where \( Q \) is the total number of branches. Next to the under-sampling of every antenna branch, the combined signal of the antennas is sampled by one fast AD-converter operating at the Nyquist rate. The combined signal is of interest because, when the analog filters \( F_q \) are set for MRC, the signals add up constructively while unwanted signals are filtered. This leads to an improved SNR and therefore allows for higher data-rates to be transferred over the 60 GHz band. The slow AD-converters are used to assist in the channel estimation, while the single fast AD-converter is used to receive the broadband signal at a sufficient SNR. If the number of branches, \( Q \), equals the down sample factor, \( D \), the architecture is equivalent to the architecture shown in Figure (4.4). Here the progressive sampling of every branch is represented by a switch that cyclically connects to each branch. The switch connects to a branch, takes one sample, moves to the next branch, takes one sample and so on. The overall delay of the scheme \( Z^{-(Q-1)} \) is represented by a single delay in front of a single AD-converter. The AD-converter of this scheme is operating at the Nyquist rate.

### 4.2.1 Z-transform

To analyse the architecture depicted in Figure (4.3), all signals are considered in the time-discrete domain. In Figure (4.5) an equivalent scheme of the \( q^{th} \) branch is shown. The signal \( x[n] \) is a digital representation of the transmitted signal. The filter \( h_q[n] \) represents the channel model. The delay \( Z^{-q} \) and the down-sampler \( D \downarrow \) represent the slow AD-converter which takes a sample at a different time slot than the other branches. The up-sampler \( D \uparrow \) adds zeros to the signal and the delay \( Z^{-(Q-1-q)} \) places every non zero sample at the correct time slot with respect to the other branches. The method described in the previous chapter is used to calculate the spectrum of the branches. First, the spectrum of \( b_q[n] \), \( B_q(z) \) is calculated. The relation between the variables is given by

\[
v_q[n] = \sum_{i=0}^{\infty} h_q[i]x[n - i]
\]

\[
a_q[n] = v_q[n - q]
\]

\[
b_q[m] = a_q[mD].
\]  

(4.1)

The output of the down-sampler \( b_q[m] \) can be rewritten as

\[
b_q[m] = a_q[mD]p[mD] = \tilde{a}_q[mD],
\]  

(4.2)

where

\[
\tilde{a}_q[n] = \begin{cases} a_q[n], & \text{for } n = 0, \pm D, \pm 2D, \ldots \\ 0, & \text{otherwise} \end{cases}
\]  

(4.3)
Figure 4.3: Antenna array where simultaneously one AD-converter is sampling the combined input signal and every antenna is under-sampled progressively.

Figure 4.4: Equivalent architecture for $L = D$. 

18
Figure 4.5: Schematic of the \( l^{th} \) under-sampled branch.

\[
\tilde{a}_q[n] = a_q[n]p[n]
\] (4.4)

and

\[
p'[n] = \begin{cases} 
1, & \text{for } n = 0, \pm D, \pm 2D, \ldots \\
0, & \text{otherwise} 
\end{cases}
\] (4.5)

The Discrete Fourier series representation of \( p[n] \) is given by

\[
p[n] = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{k}{D}}.
\] (4.6)

The \( z \)-transform of signal \( B_q(z) \) can be found with Equation[3.2] and equals

\[
B_q(z) = \sum_{m=-\infty}^{\infty} b_q[m]z^{-m} = \sum_{m=-\infty}^{\infty} \tilde{a}_q[mD]z^{-m} = \sum_{m=-\infty}^{\infty} \tilde{a}_q[m]z^{-\frac{m}{D}} = \sum_{m=-\infty}^{\infty} a_q[m] \left[ \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{km}{D}} \right] z^{-\frac{m}{D}} = \frac{1}{D} \sum_{k=0}^{D-1} \sum_{m=-\infty}^{\infty} V_q[m-q]e^{j2\pi \frac{km}{D}}z^{-\frac{m}{D}}(e^{-j2\pi \frac{k}{D}z^{-\frac{1}{D}}})^{-\frac{m}{D}} = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{k}{D}} \sum_{m=-\infty}^{\infty} V_q[m-q](e^{-j2\pi \frac{k}{D}z^{-\frac{1}{D}}})^{-\frac{m}{D}} = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{k}{D}}H_q(e^{-j2\pi \frac{k}{D}z^{-\frac{1}{D}}})X(e^{-j2\pi \frac{k}{D}z^{-\frac{1}{D}}}).
\] (4.7)

The signal that leaves the up-sampler \( u_q[n] \) can be expressed as a function of the input signals,

\[
u_q[n] = p[n-q] \sum_{i=0}^{\infty} h_q[i]x[n-q-i],
\] (4.8)

where
\[
p[n] = \begin{cases} 1, & \text{for } n = 0, \pm D, \pm 2D, \ldots \\ 0, & \text{otherwise} \end{cases}. \tag{4.9}
\]

The Discrete Fourier series representation of \( p[n] \) is given by
\[
p[n] = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi \frac{kn}{D}}. \tag{4.10}
\]

The spectrum of \( u_q[n] \), \( U_q(z) \) now equals
\[
U_q(z) = \frac{1}{D} z^{-q} \sum_{k=0}^{D-1} e^{+j2\pi \frac{kq}{D}} H_q(e^{-j2\pi \frac{k}{D}} z) X(e^{-j2\pi \frac{k}{D}} z). \tag{4.11}
\]

The spectrum of \( y_q[n] \), \( Y_q(z) \) equals
\[
Y_q(z) = \frac{1}{D} z^{-(Q-1)} \sum_{k=0}^{D-1} e^{+j2\pi \frac{qk}{D}} H_q(e^{-j2\pi \frac{k}{D}} z) X(e^{-j2\pi \frac{k}{D}} z). \tag{4.12}
\]

If the outputs of all under-sampled branches are added up the total spectral function \( Y(z) \) equals
\[
Y(z) = \frac{1}{D} z^{-(Q-1)} \sum_{l=0}^{Q-1} \sum_{k=0}^{D-1} e^{+j2\pi \frac{ql}{D}} H_q(e^{-j2\pi \frac{k}{D}} z) X(e^{-j2\pi \frac{k}{D}} z). \tag{4.13}
\]

Now a closed form representation of the output of the polyphase decomposition scheme in the \( z \)-domain is found. Practical issues exist which need to be pointed out. Since the delays are implemented in the analog domain, they can never equal the exact value they are supposed to be. Unfortunately, a deviation from the exact values of the delays leads to non-linearities in the architecture. The effects of the non-linearities that occur due to imperfect delays in a polyphase decomposition scheme are not explained in this report, but the reader should be aware of this problem.

### 4.2.2 Polyphase decomposition

Assuming the analog delays are exact, is it possible to find a unique solution for every antenna from the input signals? First, a case where the channel is a lossless transmission line is considered. It would be convenient if the sum of the outputs, of the under-sampled branches, is equal to the input. Looking closely to Figures (4.3) and (4.4) it is clear it is possible if \( Q = D \).

In Equation (4.13) the channel \( H_q(z) \) is considered to be a lossless transmission line with no delay. The equation can now be written as
\[
Y(z) = \frac{1}{D} z^{-(Q-1)} \sum_{q=0}^{Q-1} \sum_{k=0}^{D-1} e^{+j2\pi \frac{qk}{D}} X(e^{-j2\pi \frac{k}{D}} z). \tag{4.14}
\]
The spectrum of interest is \( X(z) \). The exponential function in \( X(z) \) is equal to one for \( k = 0 \). Therefore, the equation is rewritten as

\[
Y(z) = \frac{Q}{D} z^{-(Q-1)} X(z) + \frac{1}{D} z^{-(Q-1)} \sum_{q=0}^{Q-1} \sum_{k=1}^{D-1} e^{+j2\pi \frac{kq}{D}} X(e^{-j2\pi \frac{k}{D}} z). \tag{4.15}
\]

The second term needs to be zero for all values of \( k \), when \( D - 1 \geq k \geq 1 \), to recover the spectrum of \( X(z) \) directly from the input signal. Therefore

\[
\frac{1}{D} z^{-(Q-1)} \left( \sum_{q=0}^{Q-1} e^{+j2\pi \frac{kq}{D}} \right) = 0. \tag{4.16}
\]

The equation equals zero if

\[
p[k] = \sum_{q=0}^{Q-1} e^{+j2\pi \frac{qk}{D}} = 0. \tag{4.17}
\]

This function is similar to the \( p[n] \) of the previous section, only \( k \) has been replaced by \( q \) and \( n \) by \( k \). The function is zero for all \( k \) when \( D - 1 \geq k \geq 1 \) if

\[
Q = D, 2D, 3D, 4D, \ldots \tag{4.18}
\]

Therefore, the spectrum can be recovered if \( Q \) is an integer number \( N \) times the down-sample factor \( D \). The output spectrum is given by

\[
Y(z) = N z^{-(Q-1)} X(z). \tag{4.19}
\]

The architecture is now equal to a polyphase decomposition scheme [16]. Since the architecture is equal to a polyphase decomposition scheme, the channel can be calculated from the combined samples; if each receive antenna has an equal channel and given the constraint that the delays in the analog domain are exact.

### 4.2.3 Multiple channels

Let us assume we have two receivers in our polyphase decomposition scheme, each having a separate channel. The transmitted signal is an OFDM signal of which the training symbols are known. Since we are interested in under-sampling AD converters, the down sample factor \( D \) is set to be equal to two. The output of each branch now equals

\[
Y_0(z) = \frac{1}{2} z^{-1} H_0(z) X(z) + \frac{1}{2} z^{-1} H_0(-z) X(-z) \tag{4.20}
\]

\[
Y_1(z) = \frac{1}{2} z^{-1} H_1(z) X(z) - \frac{1}{2} z^{-1} H_1(-z) X(-z),
\]
where $H_0(z)$ and $H_1(z)$ are unknown channels and $X(z)$ is the known OFDM training symbol. If we consider the output spectra $Y_0(z)$ and $Y_1(z)$ in the discrete-time domain, they both have half the samples required to perform an FFT for each branch. Is it possible to find a unique solution of the channels?

An OFDM signal containing 256 subcarriers and a bandwidth of 1 GHz is assumed. To find a unique solution for each of the channels an FFT needs to be performed on 256 samples per antenna and a sample time of 1 ns is required, so a total of 512 samples is needed. Unfortunately, only 128 samples are available per receive antenna. This means an obtained solution of the channel is no longer necessarily unique. Performing an FFT on the combined signal means they need to be unraveled later on, adding unnecessarily complexity to the system. If the information of the single fast AD-converter is used enough samples are available to find a unique solution, since then there are $128 + 128 + 256 = 512$ samples available. Unfortunately, in a case of more than two receive antennas, too few samples are available to guarantee a unique solution.

### 4.3 Conclusions

Given the proposed architecture, in which every receive antenna is under-sampled and the combined signal is sampled at the Nyquist rate, it is not possible to find a unique solution for the channel; if the number of available samples is smaller than the number of samples required to perform an FFT per antenna branch.
Chapter 5

Iterative channel estimation

Iterative methods are introduced to estimate a channel from an under-sampled input signal.

5.1 Model

If the number of available samples is smaller than the number of samples required to perform an FFT per antenna branch, an estimate of the channel is not necessarily unique. Known is that every single sample holds all information about the spectrum in the time domain. Let us review the problem, the interest is in finding optimal parameter values for the analog filters. So the real interest is an estimate from the samples which can be used to set the analog filter.

We would like to find a closed form representation of the signals at the antenna elements in the time domain. The transmitted signal is an OFDM signal with a finite number of frequency components as described in Section (2.2). On baseband the signal is

\[ x(t) = \sum_{k=-\frac{K}{2}}^{\frac{K}{2}-1} L_k e^{(i2\pi \frac{kB}{K} t)}, \quad (5.1) \]

where \( t \) is the time, \( B \) is the bandwidth of the signal, \( K \) is the number of OFDM frequencies and \( L_k \) is the data symbol of the \( k^{th} \) OFDM frequency. It is assumed the receiver receives a training symbol. The training symbol is a convolution with the channel in the time domain. Therefore, the analog spectral representation of the signal at the \( q^{th} \) receive antenna is

\[ Y_q(f) = X(f)H_q(f), \quad (5.2) \]

where \( X(f) \) is the transmitted signal, \( H_q(f) \) is the channel of the antenna and \( Y_q(f) \) is the signal at the antenna. At every receive antenna a slow running AD-converter is taking samples at a known time. The problem of non-exact analog delays, which was pointed out in the previous chapter, is avoided by assuming the slow AD-converters run synchronized. For the combined signals an AD-converter is running at, or above, the Nyquist rate. The architecture is shown in Figure (5.1). The signal received by one antenna in the time domain equals
Figure 5.1: Hybrid architecture in which every antenna is sampled below the Nyquist rate and a combined signal is sampled at the Nyquist rate.

\[ y_q(t) = \int_{\tau = -\infty}^{\infty} x(\tau) h_q(t - \tau) d\tau. \tag{5.3} \]

A channel at the \( q \)th antenna is assumed with \( M_q \) rays, as presented in Section 2.3

\[ h_q(t) = \sum_{m=1}^{M_q} a_{m,q} e^{j2\pi f(t - \tau_{m,q})}, \tag{5.4} \]

where \( a_{m,q} \) is the complex attenuation and \( \tau_{m,q} \) is the delay of the \( m \)th ray at the \( q \)th antenna. The signal at the \( q \)th antenna now equals

\[ y_q(t) = \sum_{k=-\frac{K}{2}}^{\frac{K}{2} - 1} \sum_{m=1}^{M_q} L_k a_{m,q} e^{j2\pi k\frac{f}{f_s}(t - \tau_{m,q})}, \tag{5.5} \]

which is a closed form representation of the signal in the time-domain.

### 5.2 Iterative estimation

In the previous chapter is discussed that a solution of the channel from an under-sampled signal is not necessarily unique, yet a method to find a solution was not mentioned. An option, to find a solution of the channel, is to use an iterative method. An iterative method requires a
minimal number of input samples to find a solution. In the previous section a closed form representation in the time-domain of the signals at the antenna elements is found (Equation (5.5)). In the closed form representation a number of rays per antenna, each having its own unique delay and complex amplitude, describe the channel between the transmitter and the \(q^{th}\) receive antenna. The closed form representation is used in the function the iterative algorithm has to solve. To obtain an estimate for the delays and complex amplitudes of the rays, from the iterative method, at least one sample per variable per receive antenna is required. It should be noted that a solution obtained from the iterative method is not necessarily the correct solution, since the problem is non-linear.

Let us assume enough samples are taken, at every antenna element, to use an iterative algorithm. The samples are stored in a vector \(\vec{b}\) and a function \(A(\vec{L}, a, \tau)\) is known which describes the system. Function \(A(\vec{L}, a, \tau)\) contains as many equations, similar to Equation (5.5), as there are samples stored in vector \(\vec{b}\). The time and corresponding antenna element of the samples in vector \(\vec{b}\) is known. It is assumed \(\vec{L}, a\) and \(\tau\) contain unknown variables which have to be estimated iteratively with the aid of \(\vec{b}\). \(\vec{L}\) is a vector of size \(K\)

\[
\vec{L} = \begin{pmatrix}
L_1 \\
L_2 \\
L_3 \\
\vdots \\
L_K 
\end{pmatrix},
\]

(5.6)

where \(L_k\) is the \(k^{th}\) component of \(\vec{L}\) and \(K\) is the total number of OFDM symbols. \(L_k\) is the variable describing the \(k^{th}\) data symbol of the \(k^{th}\) OFDM frequency. \(a\) is a structures of \(Q\) columns each having its own unique length \(M_q\)

\[
a = \begin{pmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,Q} \\
a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,Q} \\
a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,Q} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{M_1,1} & a_{M_1,2} & a_{M_1,3} & \cdots & a_{M_1,Q} \\
a_{M_2,1} & a_{M_2,2} & a_{M_2,3} & \cdots & a_{M_2,Q} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\end{pmatrix},
\]

(5.7)

where \(Q\) is the number of receive antennas and \(M_q\) is the number of incoming rays of the \(q^{th}\) antenna. The element \(a_{m,q}\) of the structure \(a\) is the variable describing the complex amplitude of the \(m^{th}\) incoming rays of the \(q^{th}\) receive antenna. \(\tau\) is a similar structure as \(a\). The element \(\tau_{m,q}\) of the structure \(\tau\) is the variable describing the delay corresponding to the \(m^{th}\) ray of the \(q^{th}\) receive antenna. It is assumed \(\vec{L}, a\) and \(\tau\) contain the variables that need to be estimated, \(A(\vec{L}, a, \tau)\) contains a set of equations similar to Equation (5.5) and \(\vec{b}\) contains samples taken at every antenna at a known time. To find the solution for the variables in \(\vec{L}, a\) and \(\tau\) the equation

\[
A(\vec{L}, a, \tau) = \vec{b},
\]

(5.8)

needs to be solved. The equation is a nonlinear multidimensional problem. Therefore, the solution is not necessarily unique.
Now an example, to calculate the minimal amount of samples needed for a solution, is given. It is assumed every antenna receives just one ray and all variables in \( \vec{L}, a \) and \( \tau \) are unknown. Without a loss of generality it can be assumed that \( a_{1,1} = 1 + i0 \) and \( \tau_{1,1} = 0 \). Since \( \vec{L} \) is complex there actually are \( 2K \) variables, where \( K \) is the number of OFDM frequencies. Also \( a_{1,q} \) is complex resulting in \( 2Q \) variables, where \( Q \) is the number of receive antennas. And there are \( Q \) delays. The total number of variables now equals: \( 2K + 2Q + Q = 2K + 3Q \). Since it is assumed \( a_{1,1} = 1 + i0 \) and \( \tau_{1,1} = 0 \) the total number of remaining unknowns equals: \( 2K + 3(Q - 1) \). For the estimation of \( a_{m,q} \) and \( \tau_{m,q} \) per antenna except for the \( q = 1^{th} \) antenna 3 samples are required. For the estimation of \( L_k \), at least 1 sample is needed from the \( q = 1^{th} \) antenna and \( K - 1 \) samples resulting arbitrarily from the entire array. Note there is no condition for the time at which the samples are taken, as long as they are taken while the transmitted signal exists and the time at which they are taken is known by the receiver.

### 5.2.1 Steepest Descent

One option, to solve Equation (5.8) is the method of Steepest Descent [17]. The method of Steepest Descent is a straightforward approach. An arbitrary starting-point is selected and with the aid of the derivative the solution slides to a minimum of the solution surface. The method of Steepest Descent works for quadratic forms where the solution is the minimum of the function. Therefore, Equation (5.8) is rewritten as

\[
||A(\vec{L}, a, \tau) - \vec{b}||^2 = 0. \tag{5.9}
\]

It should be noted the solution is still not unique. The answer depends on the starting point since this will determine to which minimum the solution will slide on the solution surface.

#### Numerical results

A number of simulations have been done with the aid of the method of Steepest Descent. A benefit of the method is that it is easy to program and implement. A major drawback of the method is that it requires a huge amount of iterations. The initial program requires in the order of \( 10^5 \) iterations for a number of variables in the order of \( 10^7 \). The ratio is somewhat improved by limiting the values of the delays. In order to limit the values of the delays it is assumed the antennas are spaced at half the wavelength of the carrier frequency (60GHz). The ratio is further improved by optimizing the value of the “friction” the solution experiences while sliding down the solution surface. Figure (5.2) shows a typical problem. The number of variables is 14. It should be noted that although for this case the variables convert to the “correct” solution this is not guaranteed and depends solely on the starting-point.

In the program the derivative is calculated numerically. The original values of the variables in \( \vec{L}, a \) and \( \tau \) which are used to generate the input vector \( \vec{b} \) of the program are

\[
a_{1,q} = (0.5 + i0.3)^q \\
\tau_{1,q} = l + 0.01 \ast delay \\
L_k = 0.3^{k+1} \\
Q = 3 \\
K = 4. \tag{5.10}
\]
Figure 5.2: Mean square error as a function of the number of iterations.

The value of the maximal progressive delay between the antenna elements is \( \frac{1}{60} \)th of a sample time. Therefore, in the program \( \text{delay} = \frac{1}{m} \). Because the delays are so small their influence, when using the method of Steepest Descent, is only “felt” if the error is already very small \(< 10^{-12}\). To obtain accurate values for the delays, in the cases where the method converges to the “correct” answer, an error \(< 10^{-15}\) is required. This is yet another drawback of the method of Steepest Descent.

5.2.2 Conjugate Gradient

Another option, to calculate the solution iteratively is the Conjugate Gradient (CG) method [17]. The conjugate gradient is a method in which the residues are used, in effect the error caused by the current and all previous steps, to calculate the direction of the next step on the solution surface. The second derivative is used to calculate the optimal step in a given direction along the solution surface. The method is constructed in such a way that once a direction has been used it is never used again. Theoretically, if the computer was infinitely accurate the method would yield the correct solution, for a linear system, in as many steps as there are variables. Unfortunately, the equation is nonlinear and the solution is not unique. Therefore, an advanced form of the Conjugate Gradient is used to deal with this nonlinearities. A detailed description of the Conjugate Gradient method can be found in [17]. The next part of this section will give a short introduction into the concepts of the Conjugate Gradient method.
Conjugant

In the method of Steepest Descent the solution is sliding down the solution surface with the help of the derivative. If the solution “overshoots” the answer or slides slightly up-hill on its descent it is forced down again by the derivative. The method is best compared to a man sitting on a tire sliding down a snowy mountain (the solution surface) towards the valley (the solution) under the forces of gravitation (the derivative). The man has little control over his path and his speed. The speed of the man depends on the gravitational constant and the friction (In Steepest Descent this is the magnitude in which the derivative is used for a certain direction, because our problem is not linear this factor is chosen to be constant). If the man is sliding too fast he can miss the bottom of the valley and slide up a different slope to be stopped again by the forces of gravitation and be forced down again, towards the bottom of the valley. The method of Steepest Descent works best if the surface on which the man is sliding is smooth and if there are no holes in the mountain-surface in which the man can get stuck when his speed is too slow (so called local minima). So wouldn’t it be nice if our man had more control over his tire and wouldn’t slide helplessly down the mountain? If he would have some ski’s so he could slide down towards the valley in controlled straight lines?

The inefficiency of the method of Steepest Descent is caused by a reuse of directions. The man slides down on his tire up another slope, down again, up yet another slope and so on. If a set of orthogonal directions is considered (directions along which the man is allowed to ski) and if the method would use every direction only once and takes the optimal step in that direction (telling the man at which spot he should turn to take another direction) the method would be at the bottom of the valley in minimally as many steps as there are variables/directions. Unfortunately, we would have to know the answer in advance to construct a set of orthogonal directions.

The solution to this problem is to construct a set of directions which are $A$-orthogonal or conjugate instead of orthogonal. Two vectors $e_1$ and $e_2$ of size $n$ are $A$-orthogonal, or conjugate if

$$ e_1^T A e_2 = 0, \quad (5.11) $$

where $A$ is the matrix of the equation

$$ \|Ax - \vec{b}\|^2 = 0 \quad (5.12) $$

and has a size of $n \times n$ containing constant values, $x$ is a vector of size $n$ containing the variables and $\vec{b}$ is a vector of size $n$ containing constant values. In our case the formula is of the form

$$ \|A(\vec{L}, \vec{a}, \vec{z}) - \vec{b}\|^2 = 0. \quad (5.13) $$

What does it mean if vectors are $A$-orthogonal or conjugant? It means that if you stretch and scale the space a certain amount they all become orthogonal. To find out more about $A$-orthogonality you can find additional information in [17]. We will not go into more detail now. The basic idea of the Gradient Directions is to view the error term as a vector which is build-up out of conjugate vectors, or the search directions. With every iteration one conjugate vector of the error term is clipped of until all vectors have been cut and the error term has vanished.
Theoretically this is in minimally as many steps as there are variables. Unfortunately, due to limited precision and floating point roundoff errors the error term might not vanish completely.

**Conjugate Gradient method**

A simple way to generate a set of $A$-orthogonal search directions is a conjugate Gram-Schmidt process

$$d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}, \quad (5.14)$$

where $\beta_{ik}$ are defined for $i > k$

$$\beta_{ik} = -\frac{u_i^T A d_{(k)}}{d_{(k)}^T A d_{(k)}}. \quad (5.15)$$

Where $u_i$ is part of a set of linearly independent vectors, for example the coordinate axes and $d_{(i)}$ is the search vector of the $i^{th}$ iteration. The main drawback of this method is that every previous direction needs to be kept in memory to construct a new one. Another drawback, is that if the search vectors are constructed by conjugation of the axial unit vectors, the method becomes equivalent to performing Gaussian elimination. The Conjugate Gradient method solved these problems by using the residuals to construct the search vectors. Because the residual has the nice property of being orthogonal to the previous search directions it is guaranteed always to produce a new, linearly independent search direction. Another important property of using the residual is that a Krylov subspace is created. This is a subspace created by repeatedly applying a matrix to a vector. Therefore, the residual is already $A$-orthogonal to all search directions which came before except for the current one. Therefore, only the current search direction needs to be kept in memory to construct the next search vector.

**5.2.3 Nonlinear Conjugate Gradient**

The Conjugate Gradient (CG) method can be used to find the minimum point of any continuous function for which the gradient can be computed. To derive nonlinear CG there are three changes made in the algorithm of linear CG. First the residual is no longer calculated iteratively but is obtained from the derivative. Secondly the step size is more difficult to compute and requires the calculation of the second derivative. In the used algorithm this is done with the Secant method

$$\frac{d^2}{d\sigma^2} f(x + \alpha d) \approx \frac{[\frac{d}{d\sigma} f(x + \alpha d)]_{\sigma = \sigma} - [\frac{d}{d\sigma} f(x + \alpha d)]_{\sigma = 0}}{\sigma} = \frac{[f'(x + \sigma d)]^T d - [f'(x)]^T d}{\sigma \neq 0} \quad (5.16)$$

which becomes a better approximation of the second derivative if $\alpha$ and $\sigma$ approach zero. Thirdly there are several choices within the conjugate Gram-Schmidt process to compute the conjugate directions. In linear CG these are equivalent expressions. In nonlinear CG these expressions are no longer equivalent. Which method is preferable is still open for debate. Two choices are
the Fletcher-Reeves formula, which is used in linear CG for its ease of computation, and the Polak-Ribiére formula

\[
\beta_{i+1}^{FR} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \quad \beta_{i+1}^{PR} = \frac{r_{(i+1)}^T (r_{(i+1)} - r_{(i)})}{r_{(i)}^T r_{(i)}},
\]

where \( r_{(i)} \) is the residu of an iterative step, \( i \) is the \( i^{th} \) iterative step and \( \beta \) is a factor used in the conjugate Gram-Schmidt process to compute the next conjugate direction. The Fletcher-Reeves method converges if the starting point is sufficiently close to the desired minimum, whereas the Polak-Ribiére method can in rare cases cycle infinitely without converging. However, Polak-Ribiére is known to converge more quickly. Polak-Ribiére can be guaranteed to converge by choosing \( \beta \) equal to zero when the value of \( \beta \) becomes negative. If the value of \( \beta \) becomes zero the iterative process is restarted, all past search directions are forgotten and the process restarts at its current location in the direction of steepest descent. Because CG can only generate \( n \) conjugate vectors in an \( n \)-dimensional space it makes sense to restart CG every \( n \)-iterations.

Pre-conditioning

Most nonlinear CG algorithms use a preconditioning matrix. The aim of the precondition matrix is to redistribute the eigenvalues of the matrix \( A \) in such a way that it is easier to solve iteratively, thereby reducing the number of required iterations. There are a lot of options for preconditioning, the option chosen in our algorithm is a Hessian matrix where only the values along the diagonal are used. The precondition matrix can only be used if it is positive definite. Because we are dealing with a nonlinear problem the precondition matrix is updated every iterative step and is verified to be positive definite. Next to the precondition matrix the values of the delays are limited according to the same criteria as mentioned in the previous section. More details about nonlinear CG can be found in [17].

Algorithm

So what does the used programme look like in practice? The flowchart of the used algorithm can be found in Figure (5.3). The input of the system is a set of samples stored in \( \vec{b} \) which are generated by another program according to Equation (5.5). The algorithm ends if the maximum number of iterations is reached, if the residue is sufficiently small or if the program stops converging for more than five iterations. The criteria to end the algorithm can be changed depending on the nature of the function that needs to be solved.

Numerical results

From simulations it became clear that the Fletcher-Reeves method is more robust and the Polak-Ribiére method converges faster. The numerical problem of the previous section was calculated with the nonlinear CG method. The results can be found in Figure (5.4). As can be seen in the figure the fastest algorithm finds the solution for the 14 variables in just 25 steps. If this number of steps is compared to the number of iterative steps required by the method of Steepest Descent, the saving is enormous. Unfortunately, the accuracy of the nonlinear CG method is limited, compared to the method of Steepest Descent, because both the derivative and the second derivative are estimated numerically. On the other hand, the values of the delays start
Figure 5.3: Flowchart of the used nonlinear CG program.
converging along with all other variables from the first step on. Disadvantage of this method is that the solution is not unique so the solution can converge to another minimum. Whether or not the method converges to the “correct” solution depends solely on the starting-point of the iterative process.

5.3 Filter setting

Although we showed in the previous section that the algorithm can find a solution for the channel iteratively, this is not the final goal of our algorithm. Our real interest is in finding optimal parameter values for our filter. In previous research [8] it was shown that a single phase shifter yielded good results in terms of SNR. Therefore, only a value for the phase shifter needs to be found. Which leads to the question how accurate the estimation of the channel needs to be, because the used filter is in fact a crude estimation of the channel. Let us review the steps until now. We under-sample to reduce power consumption. Due to the under-sampling we cannot find a unique solution of the channel. To obtain a solution we use an iterative method. The found solution of the channel will be used to set the filter. So if we are already making an estimate of the channel only to use this estimate to make a simpler estimate, we are effectively doing things double. Therefore, it is more sensible to calculate the values for the crude estimate directly from the sampled signal. What we need to alter to our iterative algorithm to find this values is the assumption that the channel is a complex amplitude. We have chosen a complex amplitude and not a phase shift, because by taking this form we assume the problem is linear.
The benefit of writing the problem as a set of linear equations is that it is easier to solve by the iterative method. A down side to approximating a problem that is clearly non-linear, with a set of linear equations, is that the error no longer necessarily converges towards zero. Therefore, other ending criteria for the algorithm are used. In our case we used the normalized difference between iterative steps, if it is smaller than $10^{-10}$ the algorithm ends. The formulas in the estimation algorithm are now rewritten as

\[ h_q(t) = c_q, \quad (5.18) \]

where $c_q$ is a complex attenuation at the $q^{th}$ receive antenna. $c_q$ is a component of the vector $\vec{c}$. Further it is assumed the training symbol is known to the receiver. Now the minimal number of samples needed is reduced to 2 per receiver antenna. $A$ is now only a function of $\vec{c}$. Therefore

\[ ||A(\vec{c}) - \vec{b}||^2 \quad (5.19) \]

needs to be minimized by the algorithm, where $\vec{b}$ is a vector representing the actual samples taken at every antenna. Although the equation will not necessarily come close to zero, by minimizing the function in $\vec{c}$ an optimal value for the phase shifters is found. According to the matched filter criterion the filter should be the conjugate of the channel. Therefore, filter settings are calculated according to

\[ F_q(f) = \frac{c_q^*}{|c_q|}, \quad (5.20) \]

where $F_q(f)$ represents the $q^{th}$ antenna element in the frequency-domain. However, due to non-linearities in the actual function it is not guaranteed that these are equal to the settings of the global optimum in terms of SNR.

A single phase shifter seems like a crude method to filter the signal, because at a single antenna for a single frequency component a single phase shift might cause a great error. However, because the filters are set for maximum ratio combining, mitigation of the errors occurs. Every antenna has its own unique channel consisting of several rays arriving at different times at the antenna element. Therefore, the spectrum of every channel is different. Therefore, if a single frequency component at a single antenna is attenuated more than other frequency components, this is not necessarily the case at the other receive antennas. This is illustrated in Figure (5.5). For this simulation 256 OFDM sub-carriers (Section (2.2)), a bandwidth of 1 GHz, a line of sight situation in the modeled room (Section (2.3)) and two samples per antenna were used. The total transmit power was 20 dBm and the noise power at each receiver was -74 dBm.
5.4 Conclusions

Although it is not possible to find a unique solution for the channel from an under-sampled signal it is possible to find an optimum value for the filters iteratively. Due to non-linearities the iterative algorithm does not necessarily converge to the global optimum. Since the interest is in finding optimal values for the parameters of the analog filters, the assumed channel model can be simplified. By simplifying the assumed channel model into a set of linear equations the number of required samples is significantly reduced. Since the algorithm estimates a non-linear function by a set of linear equations, the error no longer necessarily converges towards zero. Therefore, the criteria to end the iterative algorithm need to be altered. The constraint used to end the iterative algorithm is the normalized difference between iterative steps.
Chapter 6

Simulation Results

Simulations are done to test the performance of the hybrid architecture

6.1 Introduction

The channel of a single receive antenna can be estimated iteratively. Unfortunately, there is no guarantee that the found solution is also the best possible solution. So the question arose how the algorithm will perform in a realistic environment, in terms of the SNR which is achieved for the combined signal. For this purpose, a receiver array is defined in the modeled room in RPS and the channel models of the individual antennas are calculated. These channel models are used to generate the received signal at the receive antennas and noise is added. From these signals, a few samples are used as an input for the estimation algorithm. The parameter values the algorithm generates for the analog filters are used to calculate the SNR of the combined signal. The results are compared to the SNR of a matched filter, which gives the best SNR possible. Although it is considered unpractical to create an analog matched filter for a highly reflective environment, its SNR is used as a benchmark. Another benchmark is the SNR achieved by utilizing the filter settings which are obtained when the entire noise free signal is used as an input of the algorithm.

6.2 Data handling

From the ray tracing software tool, the raw impulse response data is extracted for each receive antenna. It is assumed that the OFDM training symbol is a convolution with the channel (Section (2.2)). Therefore, the Fourier transform of the time-continuous signal at the $q^{th}$ antenna corresponds to

$$Y_q(f) = X(f)H_q(f), \quad (6.1)$$

where $X(f)$ is the Fourier transform of the OFDM training symbol and $H_q(f)$ is the Fourier transform of the channel between the transmitter and the $q^{th}$ receive antenna. The $z$-transform
Figure 6.1: Cartesian and spherical coordinates

of the signals corresponds to

\[ Y_q(z) = X(z)H_q(z), \]  

where \( X(z) \) is the z-transform of the OFDM training symbol and \( H_q(z) \) is the z-transform of the channel between the transmitter and the \( q^{th} \) receive antenna. To increase the speed of the simulations an FFT of the impulse response is performed. The FFT of the impulse response is then multiplied with an FFT of the input signal. The result is transformed back to the discrete-time domain by an IFFT to obtain the discrete-time domain signal at a receive antenna. On top of the signal, at every receive antenna, white noise is added. From the resulting discrete-time domain signals several samples are used as an input of the iterative algorithm. The iterative algorithm knows which sample-time and antenna element corresponds to each sample. To obtain an FFT of the raw data of the impulse response, the data is put in a time-bin. An algorithm is written to put the raw data in time-bins of \( \frac{1}{N} \) of \( \frac{1}{B} \), where \( B \) is the bandwidth and \( N \) is an arbitrarily constant. The length of the FFT of the impulse response is equal to the length of the FFT of the transmitted signal. The time bins are chosen to be considerably smaller than \( \frac{1}{B} \) because a large time-bin would result in an inaccurate impulse response for each antenna. In other words; too many arriving rays would be added up coherently in a time-bin, resulting in a loss of accuracy.

6.3 A 4 × 4 receiver array

6.3.1 Line of sight

First, simulations are done for a line of sight (LOS) situation. In the simulations a bandwidth \( B = 1 \) GHz, a number of OFDM frequencies \( K = 256 \) and a noise power of -77 dBm at each receive antenna is assumed. The transmit power is varied between 20 dBm and -1 dBm. RPS is set to generate a beam every \( \frac{1}{16} \) of a degree in spherical coordinates (Figure 6.1) for \( 0^\circ \leq \theta \leq 180^\circ \) and \( 0^\circ \leq \phi < 360^\circ \). RPS follows the beams until they are under a noise floor of -120 dBm. As a receive array a 4 × 4 array is assumed. The antenna elements are spaced half a wavelength apart, where the wavelength \( \lambda_c = \frac{c}{f_c} \), \( c \) is the speed of light and the carrier frequency \( f_c = 60 \) GHz. The antenna elements are assumed to be isotropic. Figure 6.2 shows
a top view of the modeled room and the modeled receiver. The transmitter and receiver are at a height of 1.5 m. The receiver array is placed in the LOS position. For the long OFDM training symbols (Section (2.2)), every subcarrier $L_k$ has a random BPSK symbol (-1 or +1). Further it is assumed that the receiver will set the analog filters during the first long training symbol $T_1$ and that the fast AD-converter will make a channel estimate during the second long training symbol $T_2$. Since the duration of the training symbols $T_1$ and $T_2$ are set to be equal to the OFDM symbol time $T_{Symbol}$ their length is $T_{Symbol} = \frac{256}{10^9} = 256$ ns. The slow AD-converters only take one complex sample while the training symbol $T_1$ exists, the sample rate of the slow AD-converters therefore equals $\frac{1}{256} \times 10^9 = 3.91$ MHz. It should be noted that taking one complex sample is effectively taking two samples, one real sample and one imaginary sample. Several random long training symbols were tested and one was chosen which had good characteristics. More information about the design of preambles can be found in [18, 19, 20].

Figure (6.3) shows the results of a simulation. The figure shows the normalized gain pattern, for the carrier frequency $f_c$, after the array is set and shows the impulse response as was generated by RPS. In the figure it can be seen that the main beam of the antenna array points in the direction of the strongest ray. Therefore, the rays originating from that direction are added up constructively, while rays originating from other directions are suppressed. Due to constructive adding of rays the SNR is improved. Unfortunately, not all power of arriving rays can be utilized. Therefore, the SNR is improved, but is smaller than the SNR that could have been realized if all incoming rays were added constructively; as is done with a matched filter. In this case the SNR at the antenna elements was on average 15 dB, the SNR at the fast AD converter after the phase shifters were set was 24.0 dB and the SNR that can be achieved if all rays are added constructively is 26.5 dB. Figure (6.4) shows the normalized gain pattern, for the carrier frequency $f_c$ in the plane $\theta = 90^\circ$, of the isotropic receiver array when the transmit power is lowered. It can be seen in the figure that the ability of the array to set the main beam in the direction of the strongest ray deteriorates when the SNR at the single antenna elements decreases.

Table (6.1) shows the results of simulations where the transmit power is lowered and one complex sample per antenna is taken. Each value is averaged over 100 simulations. The first column of Table (6.1) shows the transmit power. The second column shows the value of the SNR at the single fast AD-converter when the filters are set. The third column shows the average absolute deviation from the average SNR at the single fast AD-converter in percentages, $|\Delta\text{SNR}|_{100}$. The fourth column shows the SNR achieved by utilizing the filter settings which are obtained when the entire noise free signal is used as an input of the algorithm. The fifth column shows the SNR that could have been achieved by a matched filter if it was set for MRC.

Table (6.1) shows that for lower values of the average SNR per receive antenna the average deviation $|\Delta\text{SNR}|_{100}$ becomes larger. Table (6.2) shows the results of simulations where the transmit power is lowered and two complex samples per receive antenna are taken. Comparing Table (6.1) to (6.2) shows that the performance in terms of SNR improves if more samples are taken. Also the difference in the results $|\Delta\text{SNR}|_{100}$ reduces, indicating more stable behavior.

By simulations, done with the $4 \times 4$ array in a LOS situation, it is shown that the iterative channel estimation improves the SNR. The achieved results are comparable to that of a matched filter, loosing only a couple of dB in SNR. The system becomes more stable and accurate if the number of samples increases. The algorithm can even improve the SNR at the single fast AD-converter when the power of the input signal at a single receive antenna is lower than the noise power. For the receiver, the noise seems to come from every direction while the strongest received rays have a direction of arrival (DOA) which is similar for all receivers. The DOA of the rays is so similar, for all antenna elements, because the spacing between the antennas is very small compared to the distance the rays travel. Another interesting result is that the non-linear CG algorithm solves the equations in just one iterative step. This indicates that the function can be
Figure 6.2: Top view of the modeled room and the modeled receiver in RPS.
Table 6.1: SNR as a function of the transmit power for a $4 \times 4$ receiver array (LOS) with one sample per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR$_{antenna}$</th>
<th>SNR$_{fastAD}$</th>
<th>$\Delta$SNR$_{100}$</th>
<th>SNR$_{entire}$</th>
<th>SNR$_{matched}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>18.0 dB</td>
<td>25.2 dB</td>
<td>2.2%</td>
<td>27.2 dB</td>
<td>29.5 dB</td>
</tr>
<tr>
<td>17.0 dBm</td>
<td>15.0 dB</td>
<td>22.2 dB</td>
<td>3.7%</td>
<td>24.2 dB</td>
<td>26.5 dB</td>
</tr>
<tr>
<td>7.0 dBm</td>
<td>5.0 dB</td>
<td>11.3 dB</td>
<td>14.9%</td>
<td>14.2 dB</td>
<td>16.5 dB</td>
</tr>
<tr>
<td>2.0 dBm</td>
<td>0.0 dB</td>
<td>4.6 dB</td>
<td>25.0%</td>
<td>9.2 dB</td>
<td>11.5 dB</td>
</tr>
<tr>
<td>0.0 dBm</td>
<td>-2.0 dB</td>
<td>2.1 dB</td>
<td>32.8%</td>
<td>7.2 dB</td>
<td>9.4 dB</td>
</tr>
<tr>
<td>-1.0 dBm</td>
<td>-3.0 dB</td>
<td>0.0 dB</td>
<td>35.0%</td>
<td>6.2 dB</td>
<td>8.6 dB</td>
</tr>
<tr>
<td>-2.0 dBm</td>
<td>-4.0 dB</td>
<td>-0.9 dB</td>
<td>37.2%</td>
<td>5.2 dB</td>
<td>7.5 dB</td>
</tr>
<tr>
<td>-3.0 dBm</td>
<td>-5.0 dB</td>
<td>-2.4 dB</td>
<td>36.6%</td>
<td>4.2 dB</td>
<td>6.5 dB</td>
</tr>
<tr>
<td>-4.0 dBm</td>
<td>-6.0 dB</td>
<td>-3.6 dB</td>
<td>39.7%</td>
<td>3.2 dB</td>
<td>5.5 dB</td>
</tr>
</tbody>
</table>

Table 6.2: SNR as a function of the transmit power for a $4 \times 4$ receiver array (LOS) with two samples per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR$_{antenna}$</th>
<th>SNR$_{fastAD}$</th>
<th>$\Delta$SNR$_{100}$</th>
<th>SNR$_{entire}$</th>
<th>SNR$_{matched}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>18.0 dB</td>
<td>26.9 dB</td>
<td>1.0%</td>
<td>27.2 dB</td>
<td>29.5 dB</td>
</tr>
<tr>
<td>17.0 dBm</td>
<td>15.0 dB</td>
<td>23.9 dB</td>
<td>0.9%</td>
<td>24.2 dB</td>
<td>26.5 dB</td>
</tr>
<tr>
<td>7.0 dBm</td>
<td>5.0 dB</td>
<td>13.6 dB</td>
<td>3.6%</td>
<td>14.2 dB</td>
<td>16.5 dB</td>
</tr>
<tr>
<td>2.0 dBm</td>
<td>0.0 dB</td>
<td>7.7 dB</td>
<td>10.9%</td>
<td>9.2 dB</td>
<td>11.5 dB</td>
</tr>
<tr>
<td>0.0 dBm</td>
<td>-2.0 dB</td>
<td>5.0 dB</td>
<td>17.6%</td>
<td>7.2 dB</td>
<td>9.5 dB</td>
</tr>
<tr>
<td>-1.0 dBm</td>
<td>-3.0 dB</td>
<td>3.8 dB</td>
<td>16.9%</td>
<td>6.2 dB</td>
<td>8.5 dB</td>
</tr>
<tr>
<td>-2.0 dBm</td>
<td>-4.0 dB</td>
<td>2.4 dB</td>
<td>22.8%</td>
<td>5.2 dB</td>
<td>7.5 dB</td>
</tr>
<tr>
<td>-3.0 dBm</td>
<td>-5.0 dB</td>
<td>0.9 dB</td>
<td>22.7%</td>
<td>4.2 dB</td>
<td>6.5 dB</td>
</tr>
<tr>
<td>-4.0 dBm</td>
<td>-6.0 dB</td>
<td>-0.6 dB</td>
<td>30.4%</td>
<td>3.2 dB</td>
<td>5.5 dB</td>
</tr>
</tbody>
</table>
Figure 6.3: Top and side view of: (a,c) the normalized gain pattern of the $4 \times 4$ isotropic antenna array and (b,d) the impulse response of the $6^{th}$ antenna element generated by RPS in LOS position.
Figure 6.4: Normalized gain pattern as a function of $\phi$ for $\theta = 90^\circ$ of the $4 \times 4$ isotropic antenna array (LOS) with an average SNR at the antenna elements of: (a) 15 dB, (b) 0 dB, (c) -3 dB, (d) -4 dB.
solved by simpler algorithms. The algorithm is able to solve the equations in one step due to the linearization of the problem. It should be noted that the error of the algorithm not necessarily converges to zero, but the error stops converging after the first iterative step.

6.3.2 Non line of sight

Simulations are done for the same receiver array in the none line of sight (NLOS) position (Figure (6.2)). Figure (6.5) shows the normalized gain of the receiver array and the impulsive response of the 6th antenna element for a simulation, where two complex samples per antenna are used in the estimation algorithm. In this simulation the transmitted power is 17 dBm, the average SNR at a single antenna element is 9.3 dB, the SNR after the phase shifters are set at the single fast AD-converter is 19.6 dB and the SNR of a matched filter is 21.0 dB. The figure shows that one of the two main beams points in the direction of the strongest arriving rays. These rays are subsequently added up constructively and contribute to the improved SNR at the single fast AD-converter.

The simulation is done for various transmit powers and for each transmit power is repeated a hundred times. The results of the simulations are found in Table (6.3) and (6.4). The tables show that the ability to maximize the SNR is improved if more samples are taken. The tables also show that if more samples are taken the $|\Delta\text{SNR}|_{100}$ is reduced, indicating stabler behavior. The SNR achieved by the single phase shifters comes closer to the SNR of a matched filter for the NLOS situation when two samples are taken. Comparing the NLOS situation to the LOS situation, on average, more rays contribute to the power of the signal at the single fast AD-converter. This can be seen graphically by comparing Figure (6.3) to Figure (6.5).
Figure 6.5: Top and side view of: (a,c) the normalized gain pattern of the $4 \times 4$ isotropic antenna array and (b,d) the impulse response of the $6^{th}$ antenna element generated by RPS in NLOS position.
Table 6.3: SNR as a function of the transmit power for a 4 × 4 receiver array (NLOS) with one sample per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR\textsubscript{antenna}</th>
<th>SNR\textsubscript{fastAD}</th>
<th>∆SNR\textsubscript{100}</th>
<th>SNR\textsubscript{entire}</th>
<th>SNR\textsubscript{matched}</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>12.3 dB</td>
<td>16.0 dB</td>
<td>17.0%</td>
<td>23.3 dB</td>
<td>23.9 dB</td>
</tr>
<tr>
<td>12.7 dBm</td>
<td>5.0 dB</td>
<td>8.8 dB</td>
<td>30.2%</td>
<td>16.0 dB</td>
<td>16.6 dB</td>
</tr>
<tr>
<td>7.7 dBm</td>
<td>0.0 dB</td>
<td>2.3 dB</td>
<td>45.7%</td>
<td>11.0 dB</td>
<td>11.6 dB</td>
</tr>
<tr>
<td>5.7 dBm</td>
<td>-2.0 dB</td>
<td>0.6 dB</td>
<td>50.6%</td>
<td>9.0 dB</td>
<td>9.6 dB</td>
</tr>
<tr>
<td>4.7 dBm</td>
<td>-3.0 dB</td>
<td>-2.0 dB</td>
<td>50.6%</td>
<td>8.0 dB</td>
<td>8.6 dB</td>
</tr>
<tr>
<td>3.7 dBm</td>
<td>-4.0 dB</td>
<td>-2.0 dB</td>
<td>50.9%</td>
<td>7.0 dB</td>
<td>7.6 dB</td>
</tr>
<tr>
<td>2.7 dBm</td>
<td>-5.0 dB</td>
<td>-3.5 dB</td>
<td>60.0%</td>
<td>6.0 dB</td>
<td>6.6 dB</td>
</tr>
<tr>
<td>1.7 dBm</td>
<td>-6.0 dB</td>
<td>-4.9 dB</td>
<td>55.1%</td>
<td>5.0 dB</td>
<td>5.6 dB</td>
</tr>
</tbody>
</table>

Table 6.4: SNR as a function of the transmit power for a 4 × 4 receiver array (NLOS) with two samples per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR\textsubscript{antenna}</th>
<th>SNR\textsubscript{fastAD}</th>
<th>∆SNR\textsubscript{100}</th>
<th>SNR\textsubscript{entire}</th>
<th>SNR\textsubscript{matched}</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>12.3 dB</td>
<td>23.0 dB</td>
<td>1.6%</td>
<td>23.3 dB</td>
<td>23.9 dB</td>
</tr>
<tr>
<td>12.7 dBm</td>
<td>5.0 dB</td>
<td>15.3 dB</td>
<td>4.8%</td>
<td>16.0 dB</td>
<td>16.6 dB</td>
</tr>
<tr>
<td>7.7 dBm</td>
<td>0.0 dB</td>
<td>8.9 dB</td>
<td>16.3%</td>
<td>11.0 dB</td>
<td>11.6 dB</td>
</tr>
<tr>
<td>5.7 dBm</td>
<td>-2.0 dB</td>
<td>5.8 dB</td>
<td>27.4%</td>
<td>9.0 dB</td>
<td>9.6 dB</td>
</tr>
<tr>
<td>4.7 dBm</td>
<td>-3.0 dB</td>
<td>4.3 dB</td>
<td>28.3%</td>
<td>8.0 dB</td>
<td>8.6 dB</td>
</tr>
<tr>
<td>3.7 dBm</td>
<td>-4.0 dB</td>
<td>2.9 dB</td>
<td>34.9%</td>
<td>7.0 dB</td>
<td>7.6 dB</td>
</tr>
<tr>
<td>2.7 dBm</td>
<td>-5.0 dB</td>
<td>1.0 dB</td>
<td>36.0%</td>
<td>6.0 dB</td>
<td>6.6 dB</td>
</tr>
<tr>
<td>1.7 dBm</td>
<td>-6.0 dB</td>
<td>-0.2 dB</td>
<td>47.4%</td>
<td>5.0 dB</td>
<td>5.6 dB</td>
</tr>
</tbody>
</table>
6.4 A $9 \times 1$ receiver array

6.4.1 Line of sight

Now, simulations are performed for a $9 \times 1$ receiver array in a line of sight situation. The receiver array is placed in the LOS position of Figure (6.2). The antenna elements are spaced half a wavelength apart in the x-direction, where the wavelength is $\lambda_c = \frac{c}{f_c}$ with the carrier frequency $f_c = 60$ GHz and $c$ the speed of light. The antenna array is not placed at exactly the same location as the $4 \times 4$ array. Therefore, the channels of the antenna elements differ from the channels of the antenna elements of the $4 \times 4$ array. Results can be found in Table (6.5) and Table (6.6). The tables show that the performance of the system, in terms of SNR, improves if the number of samples increases. The tables also show that the stability improves when more samples are taken, because $|\Delta \text{SNR}|_{100}$ becomes smaller.

6.4.2 Non line of sight

The $9 \times 1$ receiver array is placed in the NLOS position of Figure (6.2). The antenna elements are spaced half a wavelength apart in the x-direction, where the wavelength is $\lambda_c = \frac{c}{f_c}$ with the carrier frequency $f_c = 60$ GHz and $c$ is the speed of light. The antenna array is not placed at exactly the same location as the $4 \times 4$ array. Therefore, the channels of the antenna elements differ from the channels of the antenna elements of the $4 \times 4$ array. The results are found in Table (6.5) and Table (6.6).

Interesting outcome of the simulations is that for all performed simulations the non-linear CG algorithm solves the set of equations in just one step. This indicates that simpler algorithms can be used to solve the set of equations in just one step because the problem is rewritten as set of linear functions. It should be noted that although the algorithm stops converging after one iterative step, the error does not necessarily converge to zero. Another interesting outcome is that if the SNR at the single fast AD-converter is compared to the SNR of a matched filter, the loss is between 0.9 dB and 3.0 dB. This assuming the transmit power is equal to the legal transmit power limit of 20 dBm and two complex samples per receive antenna are used. Let us assume the loss in SNR is 3.0 dB and that a conventional array with a fast AD-converter at every antenna is equal to an array with matched filters. To obtain an SNR equal to the SNR of a conventional array consisting out of 8 antennas we need an array of 16 antennas. Instead of having 8 AD converters running at 1 GHz we now have 1 AD-converter running at 1 GHz and 16 running at 7.81 MHz. In terms of power consumption this means we only use 14% of the power consumption of the conventional array for the AD-conversion, thus saving 86%. During the remaining of the packet, the slow AD-converter can be switched off. This reduces the power consumption for the AD-conversion even further to 12.5% of the power consumption of a conventional antenna array.
Table 6.5: SNR as a function of the transmit power for a $9 \times 1$ receiver array (LOS) with one samples per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR$_{antenna}$</th>
<th>SNR$_{fastAD}$</th>
<th>$\Delta$SNR$_{100}$</th>
<th>SNR$_{entire}$</th>
<th>SNR$_{matched}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>18.5 dB</td>
<td>23.7 dB</td>
<td>2.7%</td>
<td>24.6 dB</td>
<td>27.3 dB</td>
</tr>
<tr>
<td>17.0 dBm</td>
<td>15.5 dB</td>
<td>20.7 dB</td>
<td>3.3%</td>
<td>21.6 dB</td>
<td>24.3 dB</td>
</tr>
<tr>
<td>7.0 dBm</td>
<td>5.5 dB</td>
<td>9.9 dB</td>
<td>12.9%</td>
<td>11.6 dB</td>
<td>14.3 dB</td>
</tr>
<tr>
<td>2.0 dBm</td>
<td>0.5 dB</td>
<td>3.5 dB</td>
<td>29.4%</td>
<td>6.6 dB</td>
<td>9.3 dB</td>
</tr>
<tr>
<td>0.0 dBm</td>
<td>-1.5 dB</td>
<td>0.7 dB</td>
<td>35.5%</td>
<td>4.6 dB</td>
<td>7.3 dB</td>
</tr>
<tr>
<td>-1.0 dBm</td>
<td>-2.5 dB</td>
<td>-0.3 dB</td>
<td>36.0%</td>
<td>3.6 dB</td>
<td>6.3 dB</td>
</tr>
<tr>
<td>-2.0 dBm</td>
<td>-3.5 dB</td>
<td>-1.6 dB</td>
<td>40.3%</td>
<td>2.6 dB</td>
<td>5.3 dB</td>
</tr>
<tr>
<td>-3.0 dBm</td>
<td>-4.5 dB</td>
<td>-2.8 dB</td>
<td>37.5%</td>
<td>1.6 dB</td>
<td>4.3 dB</td>
</tr>
<tr>
<td>-4.0 dBm</td>
<td>-5.5 dB</td>
<td>-4.3 dB</td>
<td>46.6%</td>
<td>0.6 dB</td>
<td>3.3 dB</td>
</tr>
</tbody>
</table>

Table 6.6: SNR as a function of the transmit power for a $9 \times 1$ receiver array (LOS) with two samples per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR$_{antenna}$</th>
<th>SNR$_{fastAD}$</th>
<th>$\Delta$SNR$_{100}$</th>
<th>SNR$_{entire}$</th>
<th>SNR$_{matched}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>18.5 dB</td>
<td>24.5 dB</td>
<td>1.0%</td>
<td>24.6 dB</td>
<td>27.3 dB</td>
</tr>
<tr>
<td>17.0 dBm</td>
<td>15.5 dB</td>
<td>21.3 dB</td>
<td>1.0%</td>
<td>21.6 dB</td>
<td>24.3 dB</td>
</tr>
<tr>
<td>7.0 dBm</td>
<td>5.5 dB</td>
<td>11.3 dB</td>
<td>3.2%</td>
<td>11.6 dB</td>
<td>14.3 dB</td>
</tr>
<tr>
<td>2.0 dBm</td>
<td>0.5 dB</td>
<td>5.6 dB</td>
<td>11.4%</td>
<td>6.6 dB</td>
<td>9.3 dB</td>
</tr>
<tr>
<td>0.0 dBm</td>
<td>-1.5 dB</td>
<td>3.1 dB</td>
<td>17.4%</td>
<td>4.6 dB</td>
<td>7.3 dB</td>
</tr>
<tr>
<td>-1.0 dBm</td>
<td>-2.5 dB</td>
<td>1.7 dB</td>
<td>17.5%</td>
<td>3.6 dB</td>
<td>6.3 dB</td>
</tr>
<tr>
<td>-2.0 dBm</td>
<td>-3.5 dB</td>
<td>0.4 dB</td>
<td>24.8%</td>
<td>2.6 dB</td>
<td>5.3 dB</td>
</tr>
<tr>
<td>-3.0 dBm</td>
<td>-4.5 dB</td>
<td>-1.0 dB</td>
<td>29.3%</td>
<td>1.6 dB</td>
<td>4.3 dB</td>
</tr>
<tr>
<td>-4.0 dBm</td>
<td>-5.5 dB</td>
<td>-2.4 dB</td>
<td>31.2%</td>
<td>0.6 dB</td>
<td>3.3 dB</td>
</tr>
</tbody>
</table>
Table 6.7: SNR as a function of the transmit power for a $9 \times 1$ receiver array (NLOS) with one sample per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR\textsubscript{antenna}</th>
<th>SNR\textsubscript{fastAD}</th>
<th>$\Delta$SNR\textsubscript{100}</th>
<th>SNR\textsubscript{entire}</th>
<th>SNR\textsubscript{matched}</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>13.5 dB</td>
<td>16.2 dB</td>
<td>25.2%</td>
<td>22.1 dB</td>
<td>22.7 dB</td>
</tr>
<tr>
<td>12.7 dBm</td>
<td>6.2 dB</td>
<td>8.1 dB</td>
<td>38.6%</td>
<td>14.8 dB</td>
<td>15.4 dB</td>
</tr>
<tr>
<td>7.7 dBm</td>
<td>1.2 dB</td>
<td>3.2 dB</td>
<td>47.6%</td>
<td>9.8 dB</td>
<td>10.4 dB</td>
</tr>
<tr>
<td>5.7 dBm</td>
<td>-0.8 dB</td>
<td>0.7 dB</td>
<td>54.8%</td>
<td>7.8 dB</td>
<td>8.4 dB</td>
</tr>
<tr>
<td>4.7 dBm</td>
<td>-1.8 dB</td>
<td>-0.3 dB</td>
<td>52.7%</td>
<td>6.8 dB</td>
<td>7.4 dB</td>
</tr>
<tr>
<td>3.7 dBm</td>
<td>-2.8 dB</td>
<td>-1.7 dB</td>
<td>55.7%</td>
<td>5.8 dB</td>
<td>6.4 dB</td>
</tr>
<tr>
<td>2.7 dBm</td>
<td>-3.8 dB</td>
<td>-2.2 dB</td>
<td>64.9%</td>
<td>4.8 dB</td>
<td>5.4 dB</td>
</tr>
<tr>
<td>1.7 dBm</td>
<td>-4.8 dB</td>
<td>-3.7 dB</td>
<td>56.4%</td>
<td>3.8 dB</td>
<td>4.4 dB</td>
</tr>
</tbody>
</table>

Table 6.8: SNR as a function of the transmit power for a $9 \times 1$ receiver array (NLOS) with two samples per antenna averaged over 100 simulations.

<table>
<thead>
<tr>
<th>Transmit power</th>
<th>SNR\textsubscript{antenna}</th>
<th>SNR\textsubscript{fastAD}</th>
<th>$\Delta$SNR\textsubscript{100}</th>
<th>SNR\textsubscript{entire}</th>
<th>SNR\textsubscript{matched}</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.0 dBm</td>
<td>13.5 dB</td>
<td>21.7 dB</td>
<td>2.1%</td>
<td>22.1 dB</td>
<td>22.7 dB</td>
</tr>
<tr>
<td>12.7 dBm</td>
<td>6.2 dB</td>
<td>14.2 dB</td>
<td>5.4%</td>
<td>14.8 dB</td>
<td>15.4 dB</td>
</tr>
<tr>
<td>7.7 dBm</td>
<td>1.2 dB</td>
<td>8.5 dB</td>
<td>13.2%</td>
<td>9.8 dB</td>
<td>10.4 dB</td>
</tr>
<tr>
<td>5.7 dBm</td>
<td>-0.8 dB</td>
<td>5.8 dB</td>
<td>21.1%</td>
<td>7.8 dB</td>
<td>8.4 dB</td>
</tr>
<tr>
<td>4.7 dBm</td>
<td>-1.8 dB</td>
<td>4.3 dB</td>
<td>31.1%</td>
<td>6.8 dB</td>
<td>7.4 dB</td>
</tr>
<tr>
<td>3.7 dBm</td>
<td>-2.8 dB</td>
<td>2.8 dB</td>
<td>29.4%</td>
<td>5.8 dB</td>
<td>6.4 dB</td>
</tr>
<tr>
<td>2.7 dBm</td>
<td>-3.8 dB</td>
<td>1.2 dB</td>
<td>32.6%</td>
<td>4.8 dB</td>
<td>5.4 dB</td>
</tr>
<tr>
<td>1.7 dBm</td>
<td>-4.8 dB</td>
<td>0.0 dB</td>
<td>42.2%</td>
<td>3.8 dB</td>
<td>4.4 dB</td>
</tr>
</tbody>
</table>
6.5 Conclusions

The algorithm is able to solve the set of equations in just one step, indicating that simpler algorithms can be used to estimate the parameter values of the filters. The system is able to obtain positive values for the SNR at the single fast AD-converter even if the samples, used to set the analog filters, are buried in noise. The overall system performance improves when more samples are used. A typical hybrid architecture, which can obtain an SNR equal to, or higher than, a conventional array, requires 86% less power for the AD-conversion. However, the hybrid architecture requires twice the number of receive antennas.
Chapter 7

Conclusions and Recommendations

Conventional array architectures sample every antenna element at, or above, the Nyquist rate. Due to power constraints, at the 60 GHz band, it is not possible to implement this architecture. A possibility to overcome this, is to perform part of the signal processing in the analog domain. Therefore, a hybrid architecture is proposed. The hybrid architecture samples every receive antenna below the Nyquist rate and samples a combined signal at the Nyquist rate. Before combining the signals and present them to the AD-converter operating at the Nyquist rate, they are filtered in the analog domain. The parameters of the analog filters are set using the under-sampled signals. An analog matched filter is considered unpractical, due to the large amount of components. From previous research [8] it is known that phase shifters can yield good results in terms of SNR.

To set the parameters of the analog filters, an iterative algorithm is used. The iterative algorithm estimates the channel from sampled signals. Since the signals are under-sampled the function is non-linear and obtained channel estimates are not necessarily unique. Therefore, calculated parameter values, for the analog filters, are not necessarily equal to the global optimum in terms of SNR.

To reduce the number of variables that are estimated by the iterative algorithm and obtain a direct estimate for the analog filters, the function is rewritten as a set of linear equations. Since a clearly non-linear problem is estimated with a set of linear equations, the error no longer necessarily converges towards zero. Therefore, other constraints to end the iterative algorithm need to be implemented. The implemented constraint, to end the algorithm, is the normalized difference between iterative steps.

Simulations, performed in a realistic environment, show that a simplified channel estimate achieved good results in terms of SNR. A typical hybrid architecture, which can obtain an SNR equal to, or higher than, a conventional array, requires 86 % less power for the AD-conversion. However, the hybrid architecture requires twice the number of receive antennas. The most likely antenna element used on the 60 GHz band is the patch antenna element, due to production-cost and chip-integration considerations. Therefore, a small increase in the number of antenna elements will not lead to a considerable increase in production costs. The simulations also show that the algorithm solves the set of linear equations in one iterative step, indicating simpler algorithms can be implemented.
7.1 Future research

In the simulations isotropic antenna elements are considered. Unfortunately, an isotropic antenna does not exist. Therefore, further research is required on the effects of realistic antenna elements on system performance.

The simulations indicate simpler algorithms can be implemented. This could lead to a reduction of overhead, resulting in faster algorithms.

An interesting application of the hybrid architecture is its ability to upgrade the values of the analog filters during a packet. Thereby, enabling the antenna array to follow the transmitter when it moves. In turn, leading to the assumption the channel estimate, of the single fast AD-converter, can be used for a longer period of time. If correct, this will allow for larger packet-lengths. If longer packets are used the overhead of the system is reduced.
Acknowledgements

At the end of this document, I want to credit the people who made a substantial contribution to the success of this project.

First of all, I would like to thank Ronald Rietman for being my daily supervisor. Our weekly talks, his advice and daily assistance shaped this project to the current form. I also would like to thank Piet Sommen for his insights in digital signal processing, his useful and practical advice and for raising new questions and other perspectives in our monthly talks. Furthermore, I would like to thank Matti Herben and Erik Fledderus for arranging the possibility to graduate at Philips Research Eindhoven and for their support during my studies at the TU/e.

I would like to thank the room-mates I had during my stay at Philips for providing jokes, good discussions and a good working atmosphere: Admar Schoonen, Sören Schulz, Stephan Wesemann, Erik van der Kuip, Teun van Berkel and in particular Luca Zappaterra for providing me with a steady flow of wine-gums! I would also like to thank my colleagues at the CoSiNe group, in particular Maurice Draaijer, Ludo Tolhuizen and Stan Baggen, for making me feel at home and for the discussions during lunch time.

I would like to thank the organization of Philips and especially Jean-Paul Linnartz for providing the opportunity to carry out this project and for their sponsorship.

Finally, I would like to state a special word of gratitude to my family, girlfriend and friends for their patience and continuing support.

Johan van den Heuvel, Eindhoven October 2006
Bibliography


