Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Communication Technology Laboratory
Wireless Communications Group

Optimization of MIMO Channel

Subtitle of Master Thesis

Master Thesis
Spring 2014

Marc MÃ¼ller

Advisor: Yahia Hassan
Professor: Prof. Dr. Armin Wittneben

Abstract

Here comes the abstract...

Preface

This Master Thesis is part of the graduate study at the Department of Information Technology and Electrical Engineering (D-ITET) at the Swiss Federal Institute of Technology (ETH) Zurich.

The author certifies that this Master Thesis, and the research to which it refers, are the product of the author's own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referring practices of the discipline.

Marc MÃ¼ller

Acknowledgements

I would like to thank blablabla

Author:	Marc MÃ¼ller	marcmue@student.ee.ethz.ch
Advisor:	Yahia Hassan	yehia.hassan@nari.ee.ethz.ch
Professor:	Prof. Dr. Armin Wittneben	wittneben@nari.ee.ethz.ch

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Information Theory

1.1 general Channel

//by book of Tse
We use $x(t)$ as the transmitted signal and $y(t)$ as the received signal. Since the signal is affected by different objects in its path the received signal can be written as a function of the resulting different paths

$$y(t) = \sum_i a_i(f, t)x(t - \tau_i(t))$$

. $a_i(f, t)$ describes the attenuation of each path which is mostly dependent on frequency and time. we can omit these dependencies if we can assume sufficient narrow bandwidth for a frequency flat channel and with relatively stationary objects a time flat channel, objects are transmitters, receivers, scatterers and so on. $\tau_i(t)$ is the delay in time of the path and is constant in a time flat channel.
To distinguish between these different channels we introduce some measures:

Delay Spread T_d describes the time difference from the first to the last arrival of the signal at the receiver.

Doppler Spread D_s describes the maximum frequency difference which is introduced by Doppler shifts.

coherence Bandwidth W_c is the inverse of the delay spread and a measure of the frequency flatness of the channel. It is the bandwidth at which we can assume that the channel is frequency independent.

coherence Time T_c is the inverse of the Doppler spread and a measure for the time flatness of the channel e.g. it is the time over which the channel is time invariant.

// image

1.1.1 The equivalent baseband representation

If we have a frequency-flat and time-flat channel we can represent our channel with a time varying channel gain h which leads to

$$y(t) = h \cdot x(t)$$

for our input output relation. But it is often simpler to use a complex baseband model. if we sample the system at a sufficiently high rate we get

$$y[m] = \sum_l h_l \cdot x[m]$$

where m describes discrete time $x[m]$ and $y[m]$ the transmitted respectively the received signal.

1.1.2 Channel Capacity

the channel capacity is a good figure of merit to describe the optimal performance of a communication system. It gives a maximum data rate at which we could transmit error-free when using appropriate coding.

AWGN-channel: in an AWGN channel we experience no fading and the received signal consists just the sent signal with additive white Gaussian noise $y = x + n$. The capacity depends only on the transmit P versus noise Power N_0:

$$C = log\left(1 + \frac{P}{N_0}\right) bits/s/Hz$$

time- and frequency-flat fading channel: in a one tab fading channel the capacity is a function of the tab gain h:

$$C = log\left(1 + |h|^2 \frac{P}{N_0}\right) bits/s/Hz$$

1.2 MIMO

IN the thrive for higher data rates the limits of the basic single Antenna communication channels became obvious. The bandwidth limitation due to scarcity and cost and power constraints due to physics and regulations called for different approaches. MIMO uses the spatial dimension of the channel to further increase reliability and speed.

1.2.1 from SISO to MIMO

We are still using the assumptions of frequency- and time-flat channels. In the SISO case the discrete complex baseband channel tab h characterized the channel sufficiently. If we now use multiple antennas an the transmitter and receiver side of the channel and make sure that they are sufficiently spaced it establishes multiple distinct channels due to small-scale fading processes. In this case our discrete complex baseband channel tab h is no longer a single coefficient but a matrix $\mathbf{H} \in \mathbb{C}^{N \times M}$ with M number of transmit antennas and N number of receive antennas.

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1M} \\ h_{21} & h_{22} & \dots & h_{2M} \\ \vdots & \vdots & \dots & \vdots \\ h_{N1} & h_{N2} & \dots & h_{NM} \end{bmatrix} \tag{1.1}$$

//figure
the input output relation of a MIMO channel is then written as

$$\mathbf{y}[m] = \mathbf{H}\mathbf{x}[m] + \mathbf{w}[m] \tag{1.2}$$

where $\mathbf{x} \in \mathbb{C}^M, \mathbf{y} \in \mathbb{C}^N$ and $\mathbf{w} \sim \mathcal{CN}(0, N_0\mathbf{I}_N)$

1.2.2 MIMO Capacity

the capacity of a MIMO channel without any precodeing can be written as:

$$C = \log_2 \det \left[\mathbf{I}_N + \frac{\text{SNR}}{M}\mathbf{H}\mathbf{H}^H\right] \tag{1.3}$$

with SNR as total transmit power P over noise power N_0.

1.3 Singular Value Decomposition

If we are in single user MIMO system and we can assume full CSI (Channel State Information at Transmitter and Receiver) we can fully diagonalize the Channel by using the SVD of $\mathbf{H}$:

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^H \tag{1.4}$$

with $\mathbf{U}$ and $\mathbf{V}$ unitary and square and $\mathbf{S}$ a diagonal matrix consisting of all singular Values of $\mathbf{H}$.
With this knowledge we can multiply our transmit and received vector as follows: $\tilde{\mathbf{y}} = \mathbf{U}^H\mathbf{y}$ and $\tilde{\mathbf{x}} = \mathbf{V}\mathbf{x}$. and our transmit equation changes as follows:

$$\begin{align}
\tilde{\mathbf{y}} &= \mathbf{U}^H\mathbf{y} \tag{1.5}\\
&= \mathbf{U}^H(\mathbf{H}\tilde{\mathbf{x}} + \mathbf{w}) \tag{1.6}\\
&= \mathbf{U}^H\mathbf{U}\mathbf{S}\mathbf{V}^H\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{1.7}\\
&= \mathbf{S}\mathbf{V}^H\mathbf{V}\mathbf{x} + \tilde{\mathbf{w}} \tag{1.8}\\
&= \mathbf{S}\mathbf{x} + \tilde{\mathbf{w}}. \tag{1.9}
\end{align}$$

Now we have created min(N,M) parallel SISO systems out of our MIMO system. this brings us to the question of how to distribute the transmit power optimally over these systems to achieve the maximal possible ergodic rate.

1.3.1 waterfilling

The well known answer to that question is the water filling Solution described in //tsebook as follows:

1.4 the Receiver

In this project we were mainly looking into optimizing on the transmitter side of the channel but the receiver structure has a huge impact at the performance of the system too. In this section we will use two different receiver structures:

1. Linear MMSE Receiver
   //formula

2. V-BLAST MMSE or SIC with MMSE (MMSE-SIC)
   //formula (algorithm)
   Generally speaking the SIC receivers provide better Throughput but are more complicated.

1.4.1 Linear MMSE

The goal of the MMSE equalizer is to minimize the mean error at the filter output. We use $\mathbf{K}_y = \mathbf{H}\mathbf{H}^H + \mathbf{K}_n$ and equal power allocation at the transmitter. Then we can write the error covariance matrix as

$$\mathbf{K}_e = \mathbf{G}\mathbf{K}_y\mathbf{G}^H - \mathbf{G}\mathbf{H} - \mathbf{H}^H\mathbf{G}^H + \mathbf{I}_N \tag{1.10}$$

$$= \left(\mathbf{G}\mathbf{K}_y - \mathbf{H}^H\right)\mathbf{K}_y^{-1}\left(\mathbf{G}\mathbf{K}_y - \mathbf{H}^H\right)^H - \mathbf{H}^H\mathbf{K}_y^{-1}\mathbf{H} + \mathbf{I}_N. \tag{1.11}$$

and since $\mathbf{K}_y$ is a non-negative definite matrix we need to minimize $\mathbf{G}\mathbf{K}_y - \mathbf{H}^H = \mathbf{0}$ with $\mathbf{K}_y = \sigma_n^2\mathbf{I}_N$. This leads us to the filter matrix $\mathbf{G}$ as

$$\mathbf{G} = \mathbf{H}^H\mathbf{K}_y^{-1} \tag{1.12}$$

$$= \mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H + \sigma_n^2\mathbf{I}_N\right)^{-1} \tag{1.13}$$

$$= \left(\mathbf{H}\mathbf{H}^H + \sigma_n^2\mathbf{I}_N\right)^{-1}\mathbf{H}^H. \tag{1.14}$$

where the last equation uses the matrix inversion lemma.
Now we use $\tilde{x} = \mathbf{G}\mathbf{y}$ to estimate the sent vector $\mathbf{x}$. This gives us

$$\mathbf{K}_{e,MMSE} = \mathbf{I}_N - \mathbf{H}^H\mathbf{K}_y^{-1}\mathbf{H} = \sigma_n^2\left(\mathbf{H}^H\mathbf{H} + \sigma_n^2\mathbf{I}_N\right)^{-1} \tag{1.15}$$

for the error covariance matrix and a channel capacity of

$$C = -\sum_{i=1}^{N}\log_2\left(\left[\mathbf{K}_{e,MMSE}\right]_{i,i}^{-1}\right). \tag{1.16}$$

The corresponding per-user SINR is then calculated with

$$\mathrm{SINR}_i = \frac{1}{\left[\mathbf{K}_{e,MMSE}\right]_{i,i}^{-1}} - 1. \tag{1.17}$$

1.4.2 MMSE-SIC

We can use the result of one decorrelator filter to aid the others which results in a successive cancellation strategy: we decode $x_1[m]$ first and then subtract the decoded stream from the received vector. So the second decoder sees only the interference of the streams $x_3, \ldots, x_m$.But be aware of error propagation, this means that an error in decoder 2 leads to erroneous results in all subsequent decoders.
The V-BLAST gives us

$$R < \log_2 \det \left(\mathbf{I}_N + \frac{1}{N_0} \mathbf{H} \mathbf{K}_{\mathbf{x}} \mathbf{H}^H \right) bits/s/Hz \tag{1.18}$$

with $\mathbf{K}_{\mathbf{x}} = \text{diag}\{P_1, \ldots, P_M\}$ as the covariance matrix of the transmitted signal. Which leads to a capacity of

$$C = \max_{\text{Tr}\,|\mathbf{K}_{\mathbf{x}}| \leq P} \log_2 \det \left(\mathbf{I}_N + \frac{1}{N_0} \mathbf{H} \mathbf{K}_{\mathbf{x}} \mathbf{H}^H \right) bits/s/Hz \tag{1.19}$$

which is the capacity of the fas fading MIMO channel.
in the case of CSIT we already discussed the optimal solution as the waterfilling. But with no CSI on the transmitter side //tse proves that an equal Power allocation, in other words equal power on each transmit antenna, gives us the best performance

$$\mathbf{K}_{\mathbf{x}} = \left(\frac{P}{M} \right) \mathbf{I}_N \tag{1.20}$$

which gives us a resulting capacity of

$$C = \log_2 \det \left(\mathbf{I}_N + \frac{\text{SNR}}{M} \mathbf{H} \mathbf{H}^H \right). \tag{1.21}$$

We see a power loss in comparison to full CSI of M which is achieved by using transmit beaming.
In general the decoding order of a this SIC receivers has no impact on the performance, but in our work we always decode the channel which provides the highest MMSE-SINR first. With this decoding order we can guarantee a solution for the later discussed max-min algorithm.
// picture
// code

Chapter 2

Optimization

The question now is, is it possible to increase the Throughput of this channel with knowledge of the channel Matrix H. and to investigate this we use following Channel Representation and we are fully aware of its implications and imperfections.
// description of our channel

2.1 Multiuser Case

Since Water Filling diagonalizes the channel it does not matter what kind of decoder is used on the receiver side, the streams are already independent. But Waterfilling is only feasible in a single user case with full CSIT due to the fact that inter antenna data coordination is needed. So in a multi user scenario we need other methods to decode the channel. We use two well-known Receiver structures:

2.2 Optimizing for MMSE

2.2.1 gradient Search

The idea of the steepest descent algorithm is to find the global maximum or minimum of a function by searching along the gradient of the function. We used two ways to calculate the gradient at a certain point, numerical and analytical:

Numerical Gradient

The numerical Algorithm calculates the Gradient with numerical means. Which means to increase the input in each dimension one after another by a very small amount and recalculate output. With both results, the original one and the new ones, we can approximate the local gradient very precisely.

---

1. Start at one point 2. Calculate the Gradient 3. Use the gradient to get to the next point (Step) 4. Iterate

---

1. Calculate Starting Point 2. 1:K a. Increase input K b. Calculate Point 3. Calculate Gradient 4. Step to next point 5. Iterate

Pre: calculate analytical Gradient by hand 1. Calculate gradient at input Point with the precalculated gradient function 2. Step to the next point 3. Iterate

Analytical Gradient

The analytical Algorithm on the other hand uses analytical tools to calculate the gradient form the known function in advance. It uses the gradient function to calculate the exact gradient at each point.

optimizing algorithm

One Problem that occurred during the simulations was that at some point the gradient search got unstable, in our case it started to fluctuate and stopped moving towards the optimum. The reason of this bad and unwanted behavior lies in the step size of the function, and so there are two ways to fix it. The first and easy way is just to reduce the step size and increase the number of iterations accordingly, this way the problem occurs later and weaker and has less impact on the end result. The second much more complicated solution is to introduce adaptive step size.
Adaptive Step size means, that for each iteration we search the optimal step size which allows us to take the biggest step towards our goal. So it should not be possible to choose a step size which gives us a worse result than we had on the iteration before. With this Algorithm we can prevent any fluctuation in the gradient and force it to go the fastest possible way. We do not have to calculate the gradient as often and can thus reduce the needed calculation time. Another nice side effect is, that we implicitly know that we reached the maximum or minimum when the step size is decreased to zero without finding any better points than the previous one.

2.2.2 max sumRate

To find the maximal possible and feasible throughput with a given channel matrix $\mathbf{H}$ we can optimize the sumrate with a gradient search algorithm.
We Optimize

$$\mathbf{P} = \arg \max_{\text{trace}(\mathbf{P}) \leq P, p_k \geq 0 \, \forall p_k \in \text{diag}\{\mathbf{P}\}} \sum_k R \tag{2.1}$$

for a given total power P so that the diagonal power allocation matrix $\mathbf{P}$ is positive-semidefinite e.g. has no negative entries.
The gradient search step

$$\mathbf{P}^{n+1} = \mathbf{P}^n + \mu \nabla \text{sumRate}(\mathbf{P}) \tag{2.2}$$

Start at an arbitrary step size, calculate the next point If next Point is better than last, increase step size If next Point is worse than last, decrease step size Stop if increasing or decreasing does not improve the result anymore where $(\cdot)^n$ denotes the n-th iteration of the search algorithm and μ an arbitrary step size. We now calculate the gradient of

$$\left\|(R(\mathbf{P}))\right\|_P = \left(\sum_k (R_k)^p\right)^{1/p}. \tag{2.3}$$

Using the norm gives us the possibility to reuse the same calculation later for the throughput equalization (max minRate). We know that the gradient is calculated as

$$\nabla\left\|(R(\mathbf{P}))\right\|_P = \nabla\left[\left(\sum_k (R_k)^p\right)^{1/p}\right] \tag{2.4}$$

$$= \begin{bmatrix} \partial_{p_1} \\ \partial_{p_2} \\ \vdots \\ \partial_{p_M} \end{bmatrix} \cdot \left(\sum_k (R_k)^p\right)^{1/p} \tag{2.5}$$

but first we need to convert our bounded optimization problem into an unbounded one. First to guarantee $p_k \geq 0 \, \forall k \in \operatorname{diag}\{\mathbf{P}\}$ we can introduce a diagonal matrix $\mathbf{X}$ as $x_k^2 = p_k$ with x_k is the k-th diagonal element of $\mathbf{X}$. Secondly we need to guarantee $\operatorname{trace}(\mathbf{P}) \leq P$ to achieve this we tighten the condition to $\operatorname{trace}(\mathbf{P}) = P$.

$$R_k = -\log_2\left(\mathbf{\Phi}_{k,k}^{-1}\right) \tag{2.6}$$

$$= -\log_2\left(\left[\sqrt{\mathbf{P}}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\sqrt{\mathbf{P}} + \mathbf{I_M}\right]_{k,k}^{-1}\right) \tag{2.7}$$

$$= -\log_2\left(\left[\sqrt{\mathbf{P}\frac{P}{\operatorname{trace}(\mathbf{P})}}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\sqrt{\mathbf{P}\frac{P}{\operatorname{trace}(\mathbf{P})}} + \mathbf{I_M}\right]_{k,k}^{-1}\right) \tag{2.8}$$

$$= -\log_2\left(\left[\mathbf{X}\sqrt{\frac{P}{\operatorname{trace}(\mathbf{X}^2)}}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\sqrt{\frac{P}{\operatorname{trace}(\mathbf{X}^2)}} + \mathbf{I_M}\right]_{k,k}^{-1}\right) \tag{2.9}$$

$$= -\log_2\left(\left[\frac{P}{\operatorname{trace}(\mathbf{X}^2)}\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X} + \mathbf{I_M}\right]_{k,k}^{-1}\right) \tag{2.10}$$

We can calculate the partial derivatives of the Rates as follows:

$$\partial_{x_j}\left\|(R(\mathbf{P}))\right\|_P = \partial_{x_j}\left[\left(\sum_k (R_k)^p\right)^{1/p}\right] \tag{2.11}$$

$$= \frac{1}{p}\left(\sum_k (R_k)^p\right)^{1/p-1} \cdot \sum_k\left[p(R_k)^{p-1} \cdot \partial_{x_j}(R_k)\right]. \tag{2.12}$$

The partial derivative of the k-th users Rate R_k is calculated as

$$\partial_{x_j}(R_k) = \partial_{x_j}\left(-\log_2 \mathbf{\Phi}_{k.k}^{-1}\right) \tag{2.13}$$

$$= \frac{-1}{\log 2}\frac{1}{\mathbf{\Phi}_{k,k}^{-1}} \cdot \partial_{x_j}\left(\mathbf{\Phi}_{k,k}^{-1}\right) \tag{2.14}$$

and the derivative of the of the inverse matrix $\mathbf{\Phi}$

$$\partial_{x_j}\left(\mathbf{\Phi}^{-1}_{k,k}\right)=\left(-\mathbf{\Phi}^{-1}\cdot\partial_{x_j}\left(\mathbf{\Phi}^{-1}\right)\cdot\mathbf{\Phi}^{-1}\right)_{k,k}. \tag{2.15}$$

With

$$\begin{aligned}
\partial_{x_j}\left(\mathbf{\Phi}^{-1}\right) &= \partial_{x_j}\left(\frac{P}{\operatorname{trace}\left(\mathbf{X}^2\right)}\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}+\mathbf{I}_{\mathbf{M}}\right) && (2.16)\\
&= \partial_{x_j}\left(\frac{P}{\operatorname{trace}\left(\mathbf{X}^2\right)}\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\right) && (2.17)\\
&= P\cdot\partial_{x_j}\left(\frac{\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}}{\operatorname{trace}\left(\mathbf{X}^2\right)}\right) && (2.18)\\
&= P\cdot\left(\frac{\partial_{x_j}\left(\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\right)}{\operatorname{trace}\left(\mathbf{X}^2\right)}-\frac{\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\cdot\partial_{x_j}\left(\operatorname{trace}\left(\mathbf{X}^2\right)\right)}{\left(\operatorname{trace}\left(\mathbf{X}^2\right)\right)^2}\right) && (2.19)\\
& && (2.20)
\end{aligned}$$

and since

$$\begin{aligned}
\partial_{x_j}\left(\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\right) &= \mathbf{E}_{j,j}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}+\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{E}_{j,j} && (2.21)\\
\partial_{x_j}\left(\operatorname{trace}\left(\mathbf{X}^2\right)\right) &= 2\mathbf{X}\mathbf{E}_{j,j} && (2.22)
\end{aligned}$$

it all combines together to

$$\partial_{x_j}(R_k)=\frac{-1}{\log 2}\frac{1}{\mathbf{\Phi}^{-1}_{k,k}}\cdot P\cdot\left(\frac{\mathbf{E}_{j,j}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}+\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{E}_{j,j}}{\operatorname{trace}\left(\mathbf{X}^2\right)}-\frac{\mathbf{X}\mathbf{H}^H\frac{1}{\sigma^2}\mathbf{H}\mathbf{X}\cdot 2\mathbf{X}\mathbf{E}_{j,j}}{\left(\operatorname{trace}\left(\mathbf{X}^2\right)\right)^2}\right). \tag{2.23}$$

2.2.3 max minRate

// calculations: formulas step by step
Implementation with the norm and not just the sum allows us to use the same function in two different ways. Firstly with a //p value of 1, we just calculate the sum Rate for the throughput maximization or sum Rate maximization. Secondly we can set the //p value to something very small like -30 to create a max-min optimizer. The Optimum would be the //min norm but this is not feasible in the numerical domain and is prone to get stuck in local minimas, here -30 is a good approximation, we tested the algorithm with -100 and achieved results which were closer to the optimum that each channel gets the same Rate but that's not save to use.

// calculation

2.3 optimizing VBlast

2.3.1 jindal

[Jindal] The Paper proposes an algorithm which optimizes the throughput with an SIC receiver, the papers goal is to solve the problem of beam forming in a Broadcast or downlink

Channel, so exactly the opposite direction than we are discussing in this report. But as //paper has described there is a duality of these two channels and we are able to use the optimized results of the MAC to optimize the BC. So his work is using this duality to solve the easer problem of the MAC and then transform it back to the BC problem, so his work is highly relevant to us.
// Duality description
// abstract of Jindal
// Jindal algorithm

2.3.2 numeric Gradient

max sumRate

max minRate

2.4 optimizing for SINR target

2.4.1 fodor

2.4.2 The Algorithm

1. calculate the effective interference for MMSE processing:

$$\zeta_{k,s} = \left\{ \left(d_{k,k}^{-} \chi_{k,k} H_{k,k}^{H} \left(\sum_{j} P_j d_{k,j}^{-} \chi_{k,j} H_{k,j} T_j T_j^H H_{k,j}^H + N_t \sigma_n^2 I \right)^{-1} H_{k,k} + \frac{1}{P_k} I \right)^{-1} \right\}^{(s,s)} \tag{2.24}$$

2. calculate the optimal loading matrix

$$(T_k)^{(s,s)} = \sqrt{\frac{\zeta_{k,s} N_t}{\sum_{j=1}^{N_t} \zeta_{k,j}}} \; \forall s \in [1, N_t] \tag{2.25}$$

3. calculate used Power

$$P_k = \frac{\zeta_{k,s}}{|(T_k)^{(s,s)}|^2} (\gamma_{tgt} + 1) \; \forall k \tag{2.26}$$

n. until no more change

2.4.3 analytical gradient

Chapter 3

results

Bibliography