Diplomarbeit

"Optimum training for MIMO wireless channels"

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs
unter der Leitung von

a.o. Univ. Prof. Dipl. Ing. Dr. Franz Hlawatsch
Institut für Nachrichtentechnik und Hochfrequenztechnik

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik von

Nicolai Czink
Matrikel Nr. 9925516
Eisstoßgasse 6, 1220 Wien

Datum

Unterschrift

Preface

First of all, I want to thank my family, and especially my father, for their steady loving support. Without the help of my family I could not have developed my ideas and realized my dreams. Moreover, special thanks go to my friends, simply because they were there for me and gave me foothold in turbulent times.

Furthermore, I want to thank my colleges from the Institut für Nachrichtentechnik und Hochfrequenztechnik for many valuable talks, which let this diploma thesis prosper. My advisers DI Dominik Seethaler and Dr. Harold Artés have always been anxious to encourage and aid me at their best.
Also, my colleges from the Institut für Grundlagen und Theorie der Elektrotechnik receive my thanks for accepting me in their mids to do interdisciplinary research. I was free to learn much under their advise.

Finally I want to express my deep gratitude to my teachers, who imparted their knowledge, but much more their character. It has been and still is an outstanding pleasure to learn and prosper under their advise. Those people are Ing. Werner Chmel, Prof. Franz Hlawatsch and Prof. Ernst Bonek.

"Time flies like an arrow,
fruit flies like a banana."
cited by Franz Hlawatsch

"Wissen Sie, ich sage Ihnen,
eine gesunde Beziehung
ist der beste Rückhalt im Leben!"
Ernst Bonek

"Verrichte sechs Jahre lang Deine Arbeit;
im siebenten aber gehe in die Einsamkeit
oder unter Fremde,
damit die Erinnerung Deiner Freunde
Dich nicht hindert zu sein,
was du geworden bist."
cited by Werner Chmel

Abstract

Recently, MIMO (multiple-input multiple-output) systems draw great attention because they offer high spectral efficiency and high reliability, concurrently. For exploiting those gains, channel coefficients need to be known, thus channel training is a crucial part for the performance of MIMO systems.

In the first chapter, this work offers a short summary on system models for MIMO systems. The second chapter summarizes conditions on optimal training symbol design [1] and placement [2] for spatially white channels, and it shows the connection between channel capacity and training.

The third chapter considers channel estimation for correlated channels. Channel correlations can be exploited in order to increase channel estimation, and thus to improve channel capacity. Conventional correlation estimators, which are used throughout the literature, are *biased* (e.g. [3]). Therefore, novel unbiased *corrected correlation estimators* are introduced. Simulations and performance discussion is provided.

Finally, the corrected correlation estimators are reformulated into iterative structures, and complexity orders are investigated.

Kurzfassung

MIMO Systeme (Mehrfachantennensysteme) erlangten in letzter Zeit besondere Aufmerksamkeit, da sie gleichzeitig eine hohe spektrale Effizienz und Zuverlässigkeit versprechen. Um diese Eigenschaften ausnützen zu können, müssen die Kanalkoeffizienten bekannt sein, deshalb ist das Kanaltraining ein kritischer Punkt für die Leistung von MIMO Systemen.

Im ersten Kapitel bietet diese Arbeit eine kurze Zusammenfassung über MIMO Systemmodelle. Das zweite Kapitel ist eine Zusammenfassung über Bedingungen für optimales Trainingssymboldesign [1] und deren Platzierung [2] für räumlich weiße Kanäle, und zeigt den Zusammenhang zwischen Kanaltraining und Kanalkapazität.

Das dritte Kapitel betrachtet Kanalschätzung für korrelierte Kanäle. Korrelationen des Kanals können ausgenützt werden, um die Kanalschätzung zu verbessern, und dadurch die Kanalkapazität zu erhöhen. Herkömmliche Korrelationsschätzer, die in der Literatur verwendet werden, sind *nicht erwartungstreu* (z.B. [3]). Deswegen werden neue, erwartungstreue *"korrigierte Korrelationsschätzer"* vorgestellt. Weitere Betrachtungen, Simulationen und Diskussion des Verhaltens der Korrelationsschätzer werden angestellt.

Abschließend werden die korrigierten Korrelationsschätzer in iterative Strukturen übergeführt und die Komplexitätsordnung untersucht.

Contents

1 Introduction

Multiple-input multiple-output (MIMO) Systems have recently been of great interest. The use of multiple antennas on both ends of a wireless link promises various gains.

- *Multiplexing gain:* A significant improvement of spectral efficiency is attained by spatial multiplexing.
- *Diversity gain:* The use of multiple antennas generates a higher diversity, and thus improves the link reliability significantly.
- *Array gain:* By coherent combining the coverage can be improved.
- *Reduction of co-channel interference:* By doing beamforming and placing zeros in the antenna pattern the cellular capacity can be increased.

For example, in the ideal case it can be shown, that the ergodic capacity increase is $min(N_t, N_r)$ bits per second per hertz for every $3dB$ increase of Signal to Noise Ratio (SNR) where N_t and N_r denote the number of transmit antennas and the number of receive antennas, respectively [4][1].

The theory of all those improvements is based on channel knowledge on either the transmitter or the receiver side or both. Thus, channel training and channel estimation are an essential part of communication systems, and have to be done carefully.

[1] In the single antenna AWGN channel, 1 bit per second per hertz can be achieved with every $3dB$ increase at high SNR.

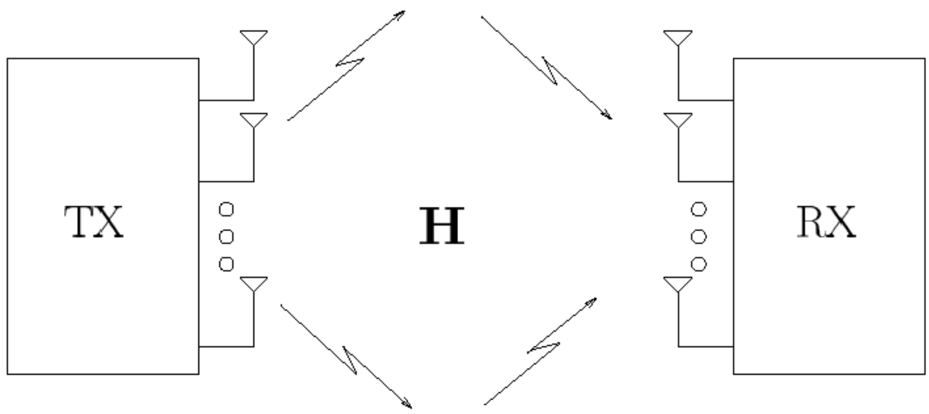


Figure 1.1: Point-to-point MIMO Channel with N_t transmit antennas and N_r receive antennas. **H** describes the scattering medium in between.

1.1 Notation

Matrices are represented in capital bold face letters (e.g. **H**), vectors in lowercase bold face letters (e.g. **s**). Furthermore, $(\cdot)^H$, $(\cdot)^T$ and $(\cdot)^*$ denote hermitian transpose, transpose and conjugate, respectively. Finally, $\mathrm{E}\{\cdot\}$ denotes the expectation, $\mathrm{tr}(\mathbf{A})$ the trace of matrix $\mathbf{A}$, $\mathbf{I}$ denotes the identity matrix of appropriate size, a_{ij} denotes the (i, j)th entry of matrix $\mathbf{A}$, and $\otimes$ is reserved for the Kronecker product.

Furthermore, $N_{(\cdot)}$ always denotes numbers of antennas and $L_{(\cdot)}$ always denotes block-lengths.

1.2 System Models

A point-to-point MIMO link consists of a multiple-antenna transmit array with N_t transmit antennas and a multiple-antenna receive array with N_r receive antennas and a scattering medium between them (see Figure 1.1).

By the scattering medium each signal from the transmit antenna array is scattered, refracted, diffracted, etc., and generally impinges with different amplitudes and phases on each antenna of the receiver array. Generally, the channel shows intersymbol interference (ISI) and thus be frequency-selective. In this thesis, I will assume *block-fading* MIMO channels. This means that the channel stays constant for the duration of a transmit block and then changes completely.

When describing this system, the equivalent discrete-time baseband representation is used. The model has to be able to describe a MIMO channel with frequency selective fading.

In the following, I will derive a MIMO system model out of some well known single input single output (SISO) system models.

1.2.1 SISO flat-fading system model

A SISO flat-fading system with block transmission can be modeled as [5]

$$\mathbf{y} = \mathbf{s}h + \mathbf{n} \tag{1.1}$$

where $\mathbf{s}$ denotes the transmit symbol vector with size $L_s \times 1$ representing the transmit block, h represents the channel, and $\mathbf{n}$ is a zero mean white Gaussian noise vector whose components have variance $\text{var}\{n_i\} = \sigma_n^2,\ i = 1...L_s$. This is the most basic model one can find for a single antenna point to point link for an ISI-free channel.

1.2.2 MIMO flat-fading system model

A multiple-input multiple-output flat-fading system with block transmission can be modeled as [1]

$$\mathbf{Y} = \mathbf{S}\mathbf{H} + \mathbf{N}, \tag{1.2}$$

where $\mathbf{S}$ denotes the $L_s \times N_t$ transmit matrix, where all symbol blocks for an antenna are stacked into a row of the matrix. So the column index of $\mathbf{S}$ represents the transmit antenna and the row index represents the time index. $\mathbf{H}$ is the $N_t \times N_r$ channel transfer matrix and describes the scattering medium. Its elements h_{ij} represent the fading coefficient from the ith transmit antenna to the jth receive antenna. Thus, $\mathbf{Y}$ is an $L_s \times N_t$ receive symbol matrix, where the column index of this matrix corresponds to the receive antenna and the row index to the time index [2]. $\mathbf{N}$ is a complex Gaussian noise matrix with independent identically distributed (iid) elements with zero mean and component-wise variance σ_n^2.

Often we use a normalized noise and channel, i.e. $\mathbf{N}$ is complex Gaussian with zero mean and *unit variance* ($\sigma_n^2 = 1$) and also the transmit symbols have unit mean power ($E\{s_{ij}^2\} = 1$). With this assumption the system model can be written as

$$\mathbf{Y} = \sqrt{\frac{\rho}{N_t}}\mathbf{S}\mathbf{H} + \mathbf{N}, \tag{1.3}$$

where ρ denotes the average signal to noise ratio (SNR) for a single receive antenna [1].

1.2.3 SISO frequency-selective system model

For frequency selective fading the assumption is used, that the impulse response is finite, and thus the input-output relation is given by the discrete-time convolution [5]

$$y_n = \sum_{i=0}^{L_h-1} h_i s_{n-i},$$

which can be written in vector-matrix-notation as

$$\mathbf{y} = \mathbf{S}\mathbf{h}.$$

Here $\mathbf{h} = [h_0 \quad h_1 \quad \cdots \quad h_{L_h-1}]^T$ is a $L_h \times 1$ vector containing the channel taps and $\mathbf{S}$ is a Toeplitz matrix of size $(L_s + L_h - 1) \times L_h$:

$$\mathbf{S} = \begin{bmatrix} s_0 & 0 & \cdots & 0 \\ s_1 & s_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ s_{L_h-1} & s_{L_h-2} & \cdots & s_0 \\ \vdots & \vdots & & \vdots \\ s_{L_s-1} & s_{L_s-2} & \cdots & s_{L_s-L_h} \\ 0 & s_{L_s-1} & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & s_{L_s-1} \end{bmatrix}. \tag{1.4}$$

Thus, the system model of a SISO frequency selective channel becomes

$$\mathbf{y} = \mathbf{S}\mathbf{h} + \mathbf{n}, \tag{1.5}$$

where $\mathbf{y}$ is the received symbol vector and $\mathbf{n}$ is again a zero mean white gaussian noise vector with component-wise variance σ_n^2. Now, this model is extended to the MIMO case.

1.2.4 MIMO frequency-selective system model

For a MIMO system with frequency-selective fading the model from section 1.2.3 can be extended in the following way.

Vector $\mathbf{h}$ from equation (1.5) is extended to a block matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1 & \ldots & \mathbf{H}_{L_h} \end{bmatrix}^T,$$

where $\mathbf{H}_i$ denotes the MIMO channel matrix as in the flat-fading model (1.2) for channel tap i. The Toeplitz matrix $\mathbf{S}$ is extended to a block-Toeplitz matrix of size $(L_s + L_h - 1) \times (L_h N_t)$ similar to (1.4) given by

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_0 & 0 & \cdots & 0 \\ \mathbf{s}_1 & \mathbf{s}_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{s}_{L_h-1} & \mathbf{s}_{L_h-2} & \cdots & \mathbf{s}_0 \\ \vdots & \vdots & & \vdots \\ \mathbf{s}_{L_s-1} & \mathbf{s}_{L_s-2} & \cdots & \mathbf{s}_{L_s-L_h} \\ 0 & \mathbf{s}_{L_s-1} & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{s}_{L_s-1} \end{bmatrix}. \tag{1.6}$$

where the vector-elements

$$\mathbf{s}_i = \begin{bmatrix} s_{i1} & s_{i2} & \cdots & s_{iN_t} \end{bmatrix}$$

are holding to the transmit symbols corresponding to the transmit antennas.

With this notation the system model can again be written as

$$\mathbf{Y} = \mathbf{S}\mathbf{H} + \mathbf{N}$$

which has the same structure as equation (1.2) in section 1.2.2. Again, $\mathbf{Y}$ is a $(L_s + L_h - 1) \times N_r$ receive symbol matrix and $\mathbf{N}$ models the system noise. Through this symmetry of these system models, they can be treated widely equally (e.g. for channel estimation). In fact, for the flat fading case, the channel block-matrix vector $\mathbf{H}$ collapses to the single matrix $\mathbf{H}_0$ and the symbol Toeplitz matrix $\mathbf{S}$ collapses to the first column (corresponding to only one channel tap). When there are other differences in treating flat-fading and frequency-selective channels, I will comment on it separately.

To regain data $\mathbf{S}$ out of the receive signal $\mathbf{Y}$ the channel matrix $\mathbf{H}$ has to be estimated. So channel modeling and channel estimation are crucial for the efficient design of wireless systems.

2 Training by Pilot Symbols

Training symbols (also called ”pilot symbols” or ”pilots”) are symbols that are known to the transmitter and to the receiver and thus are used for channel estimation.

This chapter is about training based schemes that use just the pilot symbols for channel estimation [1, 6] in contrast to using both, data and training symbols for channel estimation and subsequently for detection [7], or joint detection and decoding in the context of turbo decoders. This chapter also neglects the usage of second order statistics to improve channel estimation.

In the following chapters I will indicate training symbols with $(\cdot)_\tau$ and data symbols with $(\cdot)_d$ (e.g. $\mathbf{S}_\tau$ will denote a transmit symbol matrix containing only training symbols). These symbols are placed somewhere in the transmit block.

2.1 Definitions

2.1.1 Decomposition of Symbols

Assuming that the training symbols are spread out in the symbol block, one can decompose the transmit symbol matrix into

$$\mathbf{S} = \mathbf{S}_d + \mathbf{S}_\tau, \tag{2.1}$$

and the receive symbol matrix into

$$\mathbf{Y} = \mathbf{Y}_d + \mathbf{Y}_\tau, \tag{2.2}$$

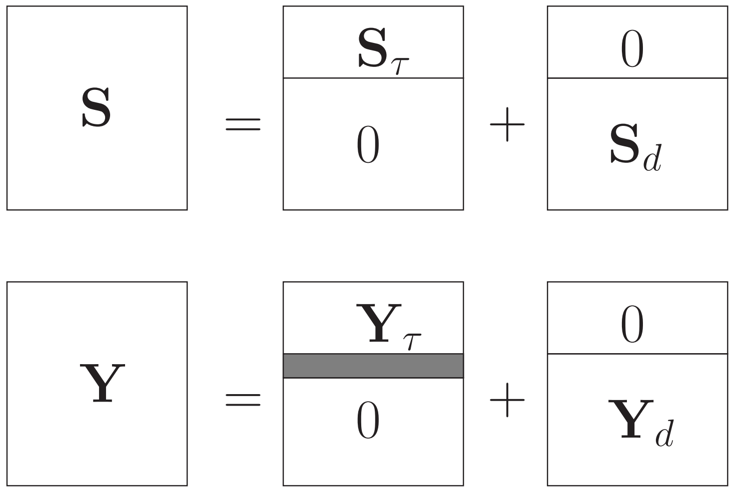


Figure 2.1: Separating training and data symbols for channel estimation. Matrices may overlap in frequency-selective fading environments

where $\mathbf{S}_d$ and $\mathbf{S}_\tau$ are the parts of $\mathbf{S}$ containing all the data symbols or training symbols, respectively, and the other elements are set to zero. Also, one can decompose the receive matrix $\mathbf{Y}$ into a matrix $\mathbf{Y}_d$ with contributions only based on the transmitted *data* symbols and the matrix $\mathbf{Y}_\tau$ containing contributions only based on the transmitted *training* symbols. This principle is shown in Figure 2.1

If a flat-fading scenario is assumed, the nonzero contributions from $\mathbf{S}_d$ and $\mathbf{S}_\tau$ do not interfere at the receiver whereas in the frequency-selective case, generally, there are elements of $\mathbf{Y}$ that contain contributions from both, data and training symbols. The receiver knows the training symbols as well as their positions in the transmission blocks and thus can use this knowledge for channel estimation.

2.1.2 Channel Estimation

There are various ways to estimate the channel matrix $\mathbf{H}$ out of the pilot symbols. The most commonly used methods are least-squares (LS) estimation and minimum mean square error (MMSE) estimation. The equations read as

$$\hat{\mathbf{H}}_{\mathrm{LS}} = (\mathbf{S}_\tau^H \mathbf{S}_\tau)^{-1} \mathbf{S}_\tau^H \mathbf{Y}_\tau, \tag{2.3}$$

$$\hat{\mathbf{H}}_{\mathrm{MMSE}} = (\mathbf{S}_\tau^H \mathbf{S}_\tau + \sigma_n^2 \mathbf{I}_{N_t})^{-1} \mathbf{S}_\tau^H \mathbf{Y}_\tau, \tag{2.4}$$

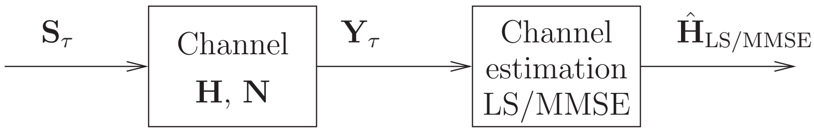


Figure 2.2: Block estimation of channel matrix $\mathbf{H}$. The training cluster $\mathbf{S}_\tau$ produces an output $\mathbf{Y}_\tau$. The channel estimator knows the training sequence, and thus the output of the channel can be used for channel estimation.

where only training symbols are used for channel estimation. Also, for the MMSE estimator, no further statistical knowledge of the channel is assumed to be known, so a spatially white channel is considered. A block diagram of this structure is show in Figure 2.2.

Thus, the channel can be decomposed into

$$\mathbf{H} = \hat{\mathbf{H}} + \tilde{\mathbf{H}}, \tag{2.5}$$

where $\hat{\mathbf{H}}$ denotes the channel estimate and $\tilde{\mathbf{H}}$ denotes the channel estimation error.

To obtain a meaningful estimate of $\mathbf{H}$, at least as many measurements as unknowns are needed [1]. This implies that $N_r \cdot L_\tau \geq N_r \cdot N_t \cdot L_h$ or $L_\tau \geq N_t \cdot L_h$, where L_τ is the number of training symbols per antenna. There are two things to note:

- The quality of the resulting estimation is of great interest. It depends heavily on the design, power allocation and placement of the training symbols.
- For frequency-selective fading equations (2.1) and (2.2) do not say anything about how to get the contributions $\mathbf{Y}_\tau$ out of the received matrix $\mathbf{Y}$.

These points will be a topic in the next sections.

2.2 Optimality criteria on training symbols

There are various ways for optimizing training with pilot symbols. All these methods can be subsumed in three categories:

- Training symbol design
- Power allocation
- Placement

In the next subsections I will describe these results and find conditions where those results hold true.

2.2.1 Optimum training symbol design

It can be shown that following criterion on the training symbols *minimizes the variance of the estimation error* [1]. First I denote the matrix $\bar{\mathbf{S}}_\tau$ as a matrix containing only training symbols without leading or trailing zeros and without any zeros in between the training symbols (so all all-zero columns or rows are cancelled). So the remaining matrix holds only the training symbols over time and space, closely stacked.

Now it can be shown that following *orthogonality condition*

$$\boxed{\bar{\mathbf{S}}_\tau^H \bar{\mathbf{S}}_\tau = const \cdot \mathbf{I}_{L_\tau}} \tag{2.6}$$

minimizes the the variance of the estimation error, where *const* denotes an arbitrary, real, nonzero factor. This means that the training signal must be a multiple of a matrix with orthonormal columns, or simply spoken *the training symbols have to be orthogonal to each other in time and space.* This result is also valid for frequency-selective channels. It has to be emphasized that this result *does not depend on the distribution of* $\mathbf{H}$ [2].

It is not astonishing that this conclusion can be gained by means of different methods of optimization; in [2] this result is achieved by minimizing the Cramer Rao Bound (CRB)[1]. A similar conclusion is drawn in [9] in training for BLAST.

[1] The Cramer Rao bound is a lower bound on the variance of the estimation error any estimator can achieve. An estimator is called optimal if its variance of estimation error achieves the CRB [2, 8].

2.2.2 Optimal Power allocation

This chapter closely refers to [1]. The following derivations are for the time being *only valid for block fading, flat fading channels*. Block fading means, that the channel does not change for the blocklength of the transmitted data. The normalized system model from equation (1.3) was used for deriving the following results. At the end of this section a result for optimum power allocation for frequency selective fading channels is provided for the sake of completeness.

A basic energy relation is given by

$$\rho L = \rho_d L_d + \rho_\tau L_\tau, . \tag{2.7}$$

where ρ was defined in (1.3), which denotes the average SNR of one symbol for one single receive antenna (corresponding to the average symbol power). L denotes the total transmit block length per transmit antenna, L_d and L_τ denote the number of *data* or *training* symbols, respectively, and ρ_d and ρ_τ denote the mean SNR of one *data* or of one *training* symbol, respectively.

Power allocation describes how much power is used for training or for data transmission in relation to the average power. One can define a power allocation factor as

$$\alpha = \frac{\rho_d L_d}{\rho L}, \tag{2.8}$$

which describes the average power of the data symbols in relation to the average total power. One can also read this formula as the relation between transmit energy used for data transmission and total transmit energy.

Optimization of power allocation can now be done in two different ways:

- Change the training symbol power ρ_τ and keep the number of training symbols L_τ constant.
- Change the number of training symbols L_τ and keep the power ρ_τ of those symbols constant.

Now a measure for optimizing the power allocation has to be defined. In [1] this measure is defined over the mutual information and is given by

$$C_\tau = \sup_{p_{\mathbf{S}_d}(\cdot),\ \mathrm{E}\{\|\mathbf{S}_d\|_F^2\} \le N_t L} \frac{1}{L}\, \mathrm{I}(\mathbf{Y}_\tau, \mathbf{S}_\tau, \mathbf{Y}_d;\ \mathbf{S}_d). \tag{2.9}$$

This measure defines a capacity C_τ (depending on the training symbols) for a block transmission channel with block length L by finding the maximum of mutual information $\mathrm{I}(\mathbf{Y}_\tau, \mathbf{S}_\tau, \mathbf{Y}_d;\ \mathbf{S}_d)$ over all transmit symbol distributions $p_{\mathbf{s}_d}(\cdot)$ with a given power constraint $E\{\|\mathbf{S}_d\|_F^2\} \le N_t L$.

For further derivation following assumptions are used:

- The channel is estimated by MMSE estimation over the training contributions only, so $\hat{\mathbf{H}} = f(\mathbf{S}_\tau, \mathbf{Y}_\tau)$ (see (2.4)).
- The pdf of the channel coefficients $p(\mathbf{H})$ is (left and right) rotationally invariant[2].

After some calculations one can find

$$C_\tau \ge \mathrm{E}\left\{ \frac{L - L_\tau}{L} \log\det\left(I_{N_r} + \frac{\rho_{\text{eff}}}{N_t} \cdot \bar{\mathbf{H}}^H \bar{\mathbf{H}} \right) \right\} \tag{2.10}$$

where

$$\rho_{\text{eff}} = \frac{\rho_d \sigma_{\hat{\mathbf{H}}}^2}{1 + \rho_d \sigma_{\tilde{\mathbf{H}}}^2} = \frac{1 + \rho_d}{1 + \rho_d \sigma_{\tilde{\mathbf{H}}}^2} - 1,$$

$$\bar{\mathbf{H}} = \frac{\hat{\mathbf{H}}}{\sigma_{\hat{\mathbf{H}}}}, \quad \sigma_{\tilde{\mathbf{H}}}^2 = \mathrm{E}\left\{ \frac{1}{N_t N_r} \mathrm{tr} \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} \right\}.$$

Let us take a closer look at the parameters of the derived capacity. Its value is mainly governed by two parameters, the first is the length of the training sequence L_τ, on which it depends *linearly*, and the second an "effective" SNR ρ_{eff}, on which it depends *logarithmically*. Now this training based capacity shall be optimized with respect to those two parameters.

For further derivations it is assumed that the channel matrix is spatially white, i.e. $\mathbf{H}$ has iid complex circular Gaussian entries with zero mean and unit variance, so $\mathbf{H} \sim \mathcal{CN}(0, \mathbf{I})$.

[2] One has to note that this assumption is a very restrictive constraint on the pdfs considered!

This is a very restrictive condition, because for radio channels this assumption is only fulfilled in rich scattering environments. However, this assumption is valid in the context of maximum entropy if no further knowledge of the channel is available [10]. With this final assumption one can go for the optimization of (2.10).

The first optimization step is to use the optimal symbol design rule from section 2.2.1. This condition minimizes the variance of the estimation error, and thus results in a better effective SNR ρ_{eff}.

With this optimization the effective SNR can be rewritten to[3]

$$\rho_{\text{eff}} = \frac{\rho_d \rho_\tau L_\tau}{N_t(1+\rho_d) + \rho_\tau L_\tau}.$$

With this result ρ_{eff} is further optimized. This can be done over power allocation and over the number of training symbols.

Power Allocation Optimizing over the power allocation yields the following result on optimal power distribution with previously defined power allocation factor α (cf. (2.8)):

$$\alpha = \begin{cases} \gamma - \sqrt{\gamma(\gamma-1)}, & \text{for } L_d \geq N_t, \\ \frac{1}{2}, & \text{for } L_d = N_t, \\ \gamma + \sqrt{\gamma(\gamma-1)}, & \text{for } L_d \leq N_t, \end{cases} \tag{2.11}$$

where

$$\gamma = \frac{N_t + \rho L}{\rho L(1 - \frac{N_t}{L_d})}.$$

It shows that for a large number of data symbols ($L_d > N_t$) the data symbols should be transmitted at a lower SNR than the training symbols, but the energy collected in the training symbols is always less than the energy collected in the data symbols, which sounds very reasonable, because one usually transmits more data symbols than training symbols. Further discussion is provided in the next paragraph.

[3]Note that unit noise variance was assumed.

Number of Training Symbols The last step is now to optimize over the number of training symbols L_τ. We already found that there must be at least N_t training symbols for proper channel estimation[4] (see section 2.1.2).

Increasing the number of training symbols further leads to a linear decrease of our previously defined capacity measure. So it follows that it is optimal to use as little training symbols as possible. Losses in accuracy in estimating $\mathbf{H}$ due to little number of training symbols can be compensated by increasing ρ_τ.

This leads to the result that the optimal training length is given by $L_\tau = N_t$ for all ρ and L.

So, one can summarize the found conditions on power allocation as

$$\begin{aligned} \rho_d \leq \rho \leq \rho_\tau \quad (L \geq 2N_t), \\ \rho_\tau \leq \rho \leq \rho_d \quad (L \leq 2N_t), \\ \rho_d = \rho = \rho_\tau \quad (L = 2N_t), \end{aligned} \tag{2.12}$$

valid for all SNR ρ with $L_\tau = N_t$.

This means that for equal training and data length (which matches the number of transmit antennas) the training and data powers shall also be equal. Usually one wants to transmit more than just N_t data symbols per transmit antenna. Thus, for optimality in this context, the training power shall be enhanced at the expense of the data power.

Figure 2.3 shows the power allocation factor alpha plotted for $N_r = N_t = 4$, an SNR of 10 dB and optimum training block length of $L_\tau = N_t = 4$.

One can observe that the data symbols carry most of the power. This is although the power of the data symbols is less than the power of the training symbols. But as there are much more data symbols than training symbols, this result is achieved.

Of course it is intuitive that training symbols should carry less power than data symbols, as one wants to use the channel for transmission.

[4] Assuming flat fading, so $L_h = 1$.

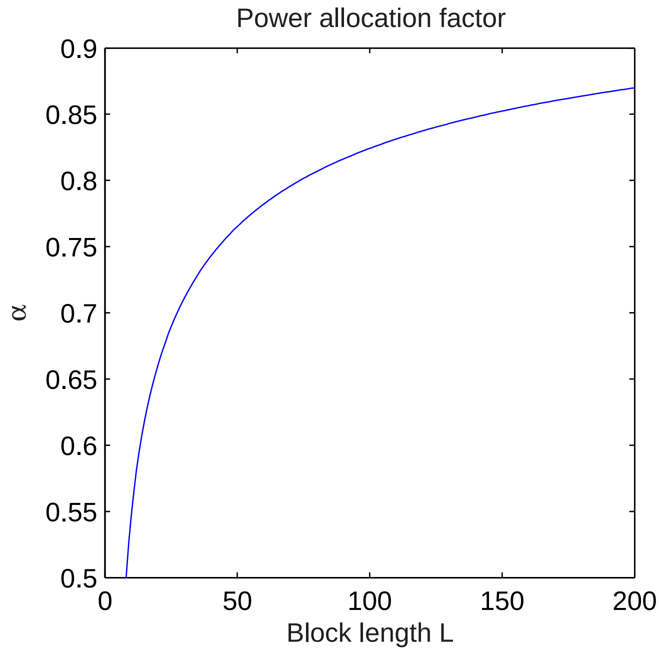


Figure 2.3: The power allocation factor α as a function of the training block length L_τ for $N_r = N_t = L_\tau = 4$ at a mean SNR of $\rho = 10$ dB. For a large number of data symbols, the whole power is concentrated in the data symbols, even though the training symbols have more instantaneous power than the data symbols (see equation 2.12).

Equal Data and Training Power In communication systems one often does not have the luxury of varying the power during the training and data phases [1]. So we have to assume that $\rho = \rho_d = \rho_\tau$.

Choosing a larger value of L_τ improves the channel estimation, and hence the channel capacity increases logarithmically (through ρ_{eff}). Using a too large value of L_τ results in a linear decrease of capacity (by the factor $\frac{L-L_\tau}{L}$). So, there is a tradeoff between those values.

One can find an optimum value of the capacity (2.10) by evaluating the lower bound for the given parameters (either analytically or via Monte Carlo simulation) for various values of L_τ.

Frequency Selective Fading Channel For completeness, I want to cite the results for optimum power allocations for frequency selective MIMO channels ($L_h > 1$) from [6] without going into detail.

Here it is assumed that only one training symbol per antenna is transmitted, and that the transmitter is able to send training symbols with different powers than data symbols.

The optimum power allocation for iid. channel coefficients in the sense of optimizing channel capacity is then given by

$$\alpha = \frac{\beta - \sqrt{\beta(\beta - (1-\lambda))}}{1-\lambda}$$

with

$$\beta = 1 + N_t(L_h + 1)/\rho, \quad \text{and } \lambda = N_t(L_h + 1)/L_d.$$

The assumption of different powers for data and training symbols is usually not valid, thus, this result is more of theoretical interest.

2.2.3 Simulations

Simulations were done for a MIMO channel with $N_t = N_r = 4$ antennas. Equation (2.10) has been evaluated and plotted over the block length L for different SNR values [1]. The orthogonality condition on training symbols from section 2.2.1 has been used, and also the conditions on training symbol power are included in simulations.

Figure 2.4 shows the training based lower bound on capacity as a function of L with an SNR of 10 dB (Figure 2.4a), and with an SNR of 6 dB (Figure 2.4b).

The red curves describe the capacity for the perfectly known channel, whereas the black curves show the lower bound of the training based capacity, when one can exploit optimal power allocation. Note that for this setting, the number of training symbols is fixed to the number of transmit antennas, so $L_\tau = N_t$.

One can see that the training based capacity is much lower than the optimum capacity provided by the channel. For large block lengths L the difference is of about 1 bit/channel use.

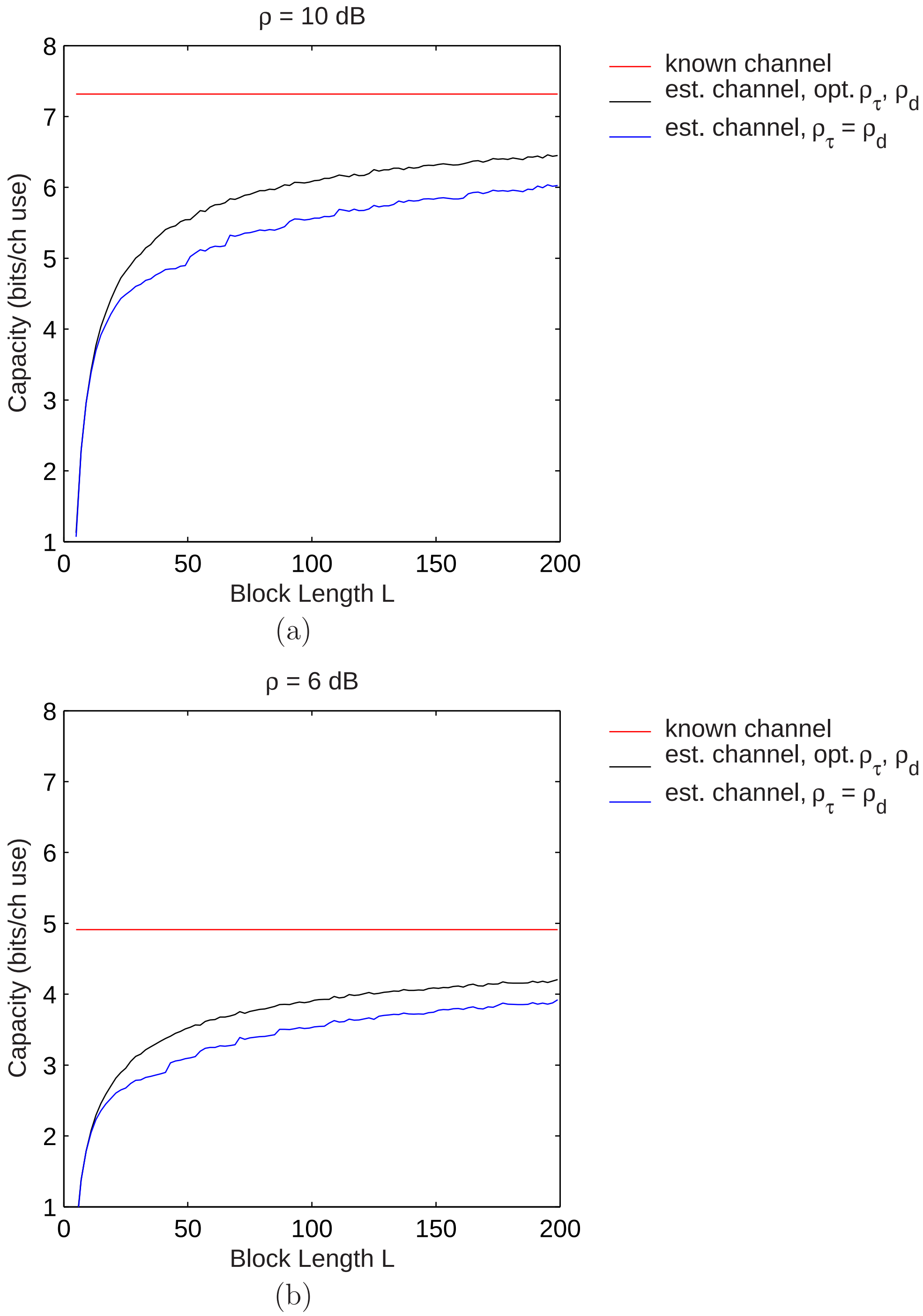


Figure 2.4: Lower bound on training based capacity as a function of block length L for $N_t = N_r = 4$ for optimal power allocation (black curves) and for equal training symbol and data symbol power (blue curves): (a) for $\rho = 10$ dB and (b) for $\rho = 6$ dB. The curves for equal training symbol and data symbol power were plotted for optimized training length L_τ. The red curve gives the upper bound of channel capacity when no training is needed. One can observe the loss of capacity because of training. For increasing block-length this loss decreases.

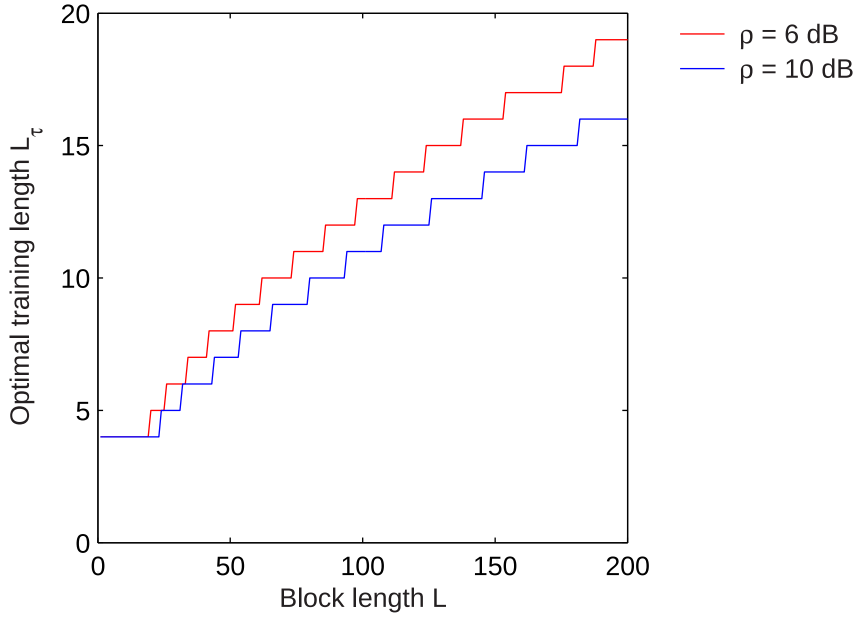


Figure 2.5: Optimum number of training symbols L_τ as a function of the block length L for $N_r = N_t = 4$ and two values of SNR: $\rho = 6$ dB and $\rho = 10$ dB. For higher SNR a smaller number of training symbols is needed.

As optimal power allocation can usually not be performed, the blue curves show the training based capacity for $\rho_\tau = \rho_d = \rho$ with the optimum number of training symbols L_τ.

One can see that for different SNRs the principle behaviour stays the same, the curves are only shifted to higher capacities.

The optimum number of training symbols has been evaluated by maximizing equation (2.10) numerically for arbitrary L_τ. The result can be seen for different SNRs in Figure 2.5.

2.2.4 Optimal Placement

After deciding how much power and how many training symbols to use, one can concentrate on placing them in the data stream. This section closely refers to [2] and thus the derivations are done for *frequency selective block fading channels*. The system model used is discussed in section 1.2.4.

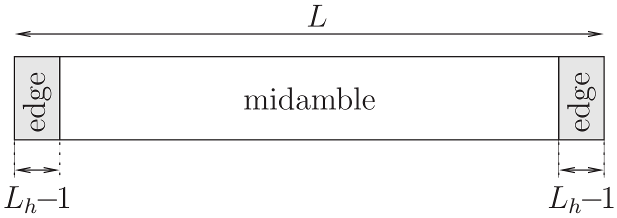


Figure 2.6: Midamble and edge positions in symbol block

First, I will subsume the results for optimum placement when considering training with all (data and training) symbols, derived in [2] shortly without going into details, because the essence of the referred paper seems very intuitive. Then I will describe how to use these results for performing channel training only with training symbols.

I will assume that the training symbols have at least as much total power as one data symbol ($L_\tau \rho_\tau \geq \rho_d$). The assumption on the *total* power is done, since training symbols may be zero and thus have zero power, too. Finally, only the total training power is of interest. Also, it is assumed that one has a sufficient number of training symbols available (so, one can spend as many training symbols as needed).

First, "midamble" and "edge" positions are defined in our block with length L. Edge positions are the first and the last L_h positions in the block, where L_h denotes the length of the channel impulse response (see figure 2.6). Now one can show two things:

(i) Since we do not have previous knowledge of the symbols sent before our block starts ($\mathbf{s}_{-1}$, $\mathbf{s}_{-2}, \ldots$ are unknown), the first $L_h - 1$ symbols do not only depend on the symbols sent in our block. Placing training symbols there would be very unproductive, because of those unknown contributions. Thus, these positions shall not be used for channel estimation. Note that this approach also considers inter-block-interference (IBI), additionally to ISI.

(ii) Furthermore, placing training symbols in the last $L_h - 1$ positions would be ineffective, because the impulse response of these training symbols would not fit into our receive block, and thus valuable channel information gets lost.

Therefore, it would be the best to set the edge positions of the block to zero so that they

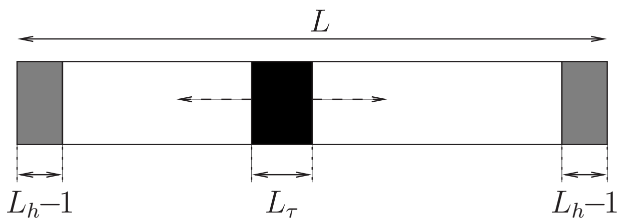


Figure 2.7: Training cluster placement on midamble-positions is shift-invariant

do not interfere with the rest of the block (i) and do not waste any power (ii). For this scheme we have to spend exactly $2(L_h - 1)$ training symbols to set the edges to zero.

From those considerations follows that training symbols should be placed somewhere in the midamble of the transmit block. It would be intuitive to group all pilot symbols into one single orthogonal cluster, which is in fact the best one can do when using only pilot symbols for channel estimation. The problem with this approach is that it is quite difficult to design long orthogonal training symbol sequences for channel estimation out of a given code alphabet [11].

One can do better when using all (data and training) symbols for channel estimation. This can be done by feedback structures or other techniques, however, in [2] only the optimal case of perfect usage of all symbols for channel estimation is considered. In the referred paper it is shown that certain symbol placements minimize the CRB.

Furthermore, it is shown for the introduced assumptions above (sufficient training power, sufficient number of training symbols) that *any placement in midamble positions is shift-invariant as long as the orthogonality condition is fulfilled*, which means that one can place the training symbols anywhere in the midamble (see figure 2.7). It has to be emphasized that this conclusion is only valid when using *all* symbols for channel estimation!

As it is quite difficult to generate long orthogonal sequences, one can achieve orthogonality of the training sequence by spreading many small orthogonal training clusters throughout the transmit block. Orthogonal clusters are small clusters of training symbols that do not overlap in space (antennas) and time. By doing this, those training clusters are mutually orthogonal in the time domain. From these thoughts has emerged the idea of quasi-periodic placement (QPP-α) schemes. The placement is done in following way:

- Design orthogonal training clusters with length α
- Place them as periodically as possible throughout the midamble positions of the transmit block

According to [2] it has been shown that the QPP schemes are optimal in the sense of maximizing mutual information as well as in the sense of minimizing the average MSE, when a decision feedback equalizer (DFE) is used. It turns out that, for QPP-1 schemes, the orthogonality condition on training symbols is the easiest to satisfy. When the training blocks furthermore are now spaced at least with distance $L_h - 1$, the QPP-1 scheme is optimal for channel estimation.

When using constant-modulus signals, all conclusions from above still hold true. Of course, by giving away the usage of different powered symbols, the CRB for optimal channel estimations increases.

Training by pilot symbols only. The referred paper [2] gives an intuitive framework of how to place training symbols, but it neglects the case of using only training symbols for channel estimation. Often it is not possible to use all symbols for channel estimation (e.g. for complexity reasons), and thus also the case of training by pilot symbols only has to be considered as an important case.

In contrast to the previous context, where data symbols were treated as known (e.g. by some DFE structure), they now have to be considered as completely *unknown*. This introduces following constraints:

- One needs at least training symbol blocks with length L_h per transmit antenna to obtain a meaningful estimate of all channel coefficients.
- When spreading out training symbol clusters throughout the transmit block (as it is done in the QPP-α schemes), then the CRB is invariant to training block shifts, as long as the spacing between the blocks is at least $L_h - 1$.
- Splitting up training clusters and spreading them over the transmit block *leads to a higher estimation error than before.* This is because each block must have at

least length L_h that the channel impulse response is only dependent on training symbols. Thus, by this ”guard-period”, there is always valuable training symbol energy, because one cannot use the contributions from the interfering data symbols.

From this follows that the QPP-1 scheme (which was optimal for using all symbols for channel estimation) *is now the worst one can do*! In fact, it is prohibited by the statements above.

This leads to the result that *one single (orthogonal) training symbol cluster is optimal*, and this cluster is still shift-invariant throughout midamble positions. This result is again quite intuitive, for one is ”wasting” only once the guard period and can use all other training symbols for channel estimation.

For *frequency flat fading channels* (i.e. $L_h = 1$) all previous precautions can be dropped. As the response of the channel is immediate, no overlapping in time needs to be considered. Thus, training clusters can be chosen with arbitrary lengths and shift invariance throughout the whole transmit block is valid. Note that in this case there are no edge or midamble positions (as there is no interference between the transmitted symbols at the receiver). In this optimum case one only needs to take care that the training symbols are orthogonal to each other in the antenna domain.

2.2.5 Summary

For *flat fading* scenarios training symbol placement is not relevant to optimal channel training. One can fully concentrate on training symbol design and power allocation. When one freely can choose transmit powers, the optimum number of training symbols is equal to the number of transmit antennas ($N_t = L_\tau$) and the optimum power allocation is given by (2.11). It also shows that in the low SNR region the optimum power allocation is given by $\alpha = 0.5$, which is also a quite intuitive result.

Furthermore, the training based capacity depends on the number of training symbols and on the effective SNR. In this chapter training by pilot symbols only was considered. For a better channel estimation one can also take into account second order statistics for channel estimation. This further improves the effective SNR. More details of this approach are discussed in the next chapter.

When dealing with *frequency selective fading* channels, one has to take care about training symbol placement, where edge positions shall be avoided. Depending on which symbols are used for channel estimation (only training symbols in contrast to all symbols), there has to be taken care of the optimum placement.

When doing training with all transmitted symbols, the placement of the training symbols is completely shift-invariant within midamble positions with regard to the CRB of channel estimation, the optimum symbol placement is the QPP-1 scheme described above. Using only training symbols for channel estimation, this total shift-invariance is no longer given and for power considerations it is the optimum to group all training symbols in one single orthogonal cluster. The difficulty of creating a single training symbol cluster is the design of long orthogonal sequences out of a given symbol alphabet [11].

Using constant modulus signals, these results stay the same, although the error variances of channel estimation rise due to lost optimality in training symbol design. Thus, power allocation becomes tricky, as there is a trade-off between optimum symbol power and number of training symbols.

All these results were shown for block-fading channels. Assuming that the channel does not change at all over time, it is true that the training sequence can be placed at any position. If this condition is weakened, so that the channel varies only very slowly over time (this means that the channel varies only very little during one block length L), it is again intuitive for a single-cluster scheme to place this cluster in the middle of the symbol block. For multiple-cluster schemes it is intuitive to spread those clusters as far as possible throughout the transmit block, both to get the best average over the channel that is possible.

3 Channel estimation using pilot symbols and second-order statistics

In contrast to chapter 2, in which no special statistical knowledge about the was channel assumed, this chapter emphasizes on the use of second order statistics for channel estimation.

First, I will derive the MMSE channel estimator using second order statistics. Since the estimator in (2.4) is a matrix-estimator, which assumes a spatially white channel statistically, it has to be adapted for deploying full second order statistics.

The quality of channel estimation using second order statistics strongly depends on the quality of the statistical knowledge and on the channel itself. If the knowledge of the second order statistics is poor, channel estimation may be worse compared to the use of no statistical knowledge. Thus, I will introduce some methods for estimating the second order statistics, as well as novel *"corrected estimators"* for estimating the second order statistics. Then, I will conclude this chapter with comparative simulations of the derived results.

3.1 Channel estimation using second-order statistics

First we define the channel correlation matrix $\mathbf{R_H}$ as

$$\mathbf{R_H} \triangleq \mathrm{E}\left\{\mathrm{vec}(\mathbf{H})\mathrm{vec}(\mathbf{H})^H\right\}, \tag{3.1}$$

where the vec$(\cdot)$ operator converts a matrix into a vector by stacking the columns. The diagonal elements of $\mathbf{R_H}$ represent the mean powers of the propagation paths and the

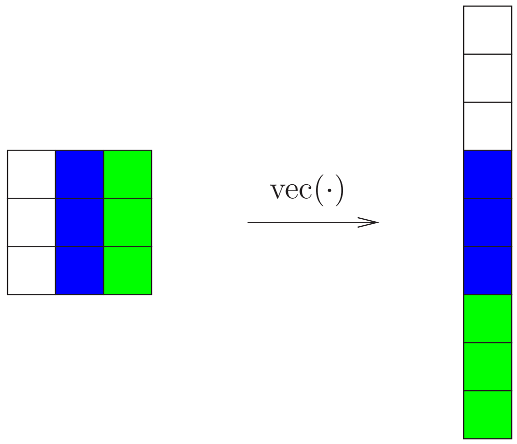


Figure 3.1: The vec(·) operator stacks a matrix into a vector.

off-diagonal elements represent the correlations between all fading coefficients of the channel matrix $\mathbf{H}$. For a spatially white channel, we have $\mathbf{R}_{\mathbf{H}} = \mathbf{I}$. Note that $\mathbf{R}_{\mathbf{H}}$ is hermitian by definition.

The use of the vec(·) operator is demonstrated in figure 3.1.

Then we introduce a block-index n and use the vec(·) operator on the MIMO system model (1.2) to obtain a vector representation:

$$\mathrm{vec}(\mathbf{Y}_\tau[n]) = \mathrm{vec}(\mathbf{S}_\tau[n]\mathbf{H}[n] + \mathbf{N}[n])$$
$$\Rightarrow \mathrm{vec}(\mathbf{Y}_\tau[n]) = (\mathbf{I} \otimes \mathbf{S}_\tau[n])\mathrm{vec}(\mathbf{H}[n]) + \mathrm{vec}(\mathbf{N}[n]),$$

which can be rewritten to

$$\mathbf{y}_\tau[n] = \underline{\mathbf{S}}_\tau[n]\mathbf{h}[n] + \mathbf{n}[n], \tag{3.2}$$

where $(\cdot)[n]$ represents the variable for the nth received block. Thus, $\mathbf{y}_\tau[n]$ represents the stacked receive training matrix of the nth transmitted block, $\underline{\mathbf{S}}_\tau[n]$ represents the Kronecker-multiplied training symbol matrix, $\mathbf{n}[n]$ the stacked noise matrix, and $\mathbf{h}[n]$ represents the stacked channel matrix. In further derivations I will omit the block index for better readability, when not necessary. Note that (3.1) can now be rewritten to $\mathbf{R}_{\mathbf{H}} = \mathrm{E}\{\mathbf{h}\mathbf{h}^H\}$.

One can derive an MMSE vector-estimator on this system model for the channel $\mathbf{h}$ as [8, 12]

$$\hat{\mathbf{h}} = \mathbf{R_H}\underline{\mathbf{S}}_\tau^H \left(\underline{\mathbf{S}}_\tau \mathbf{R_H} \underline{\mathbf{S}}_\tau^H + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}_\tau, \tag{3.3}$$

which can also be written as

$$\hat{\mathbf{h}} = \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \mathbf{R_H}^{-1}\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}_\tau, \tag{3.4}$$

where $\hat{\mathbf{h}}$ is the estimate of the stacked channel matrix using the statistical knowledge $\mathbf{R_H}$. The equality of (3.3) and (3.4) can be shown by the matrix inversion lemma[1]. I will also use the LS estimator for comparison to other estimators, as it does not use any statistical knowledge of the channel. One can obtain the LS estimator by setting $\sigma_n^2 = 0$ in equation (3.3), which results in

$$\hat{\mathbf{h}}_{\mathrm{LS}} = \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}_\tau.$$

An operational block diagram of the vectorized system is given in figure 3.2. The training symbols are sent over the channel and produce an output $\mathbf{y}_\tau$. Those received values are now used for correlation estimation and for channel estimation. Note that these two parts are uncoupled in this scheme. Correlation estimation itself is usually done by first estimating channel coefficients by some appropriate estimator (which is usually not the same estimator as the channel estimator) and afterward build an estimate for the channel correlation. This correlation estimate is fed to the channel estimator, which finally produces the estimate of the channel coefficients with equation (3.3). This cycle is done for every received symbol block.

For examination of the MMSE of channel estimation, the estimation error is defined by [8]

$$\tilde{\mathbf{h}} = \hat{\mathbf{h}} - \mathbf{h}, \tag{3.5}$$

and the covariance matrix of the channel estimation error is obtained as

$$\mathrm{cov}\left\{\tilde{\mathbf{h}}\right\} = \left(\mathbf{R_H}^{-1} + \sigma_n^2 \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\right)^{-1}. \tag{3.6}$$

The trace of the error covariance matrix tells about the minimum MSE achievable when knowing the second order statistics $\mathbf{R_H}$ of the channel perfectly. This trace can be

[1]Matrix inversion lemma: $\left(\mathbf{A}^{-1} + \mathbf{BCD}\right)^{-1} = \mathbf{A} - \mathbf{AB}(\mathbf{DAB} + \mathbf{C}^{-1})^{-1}\mathbf{DA}$.

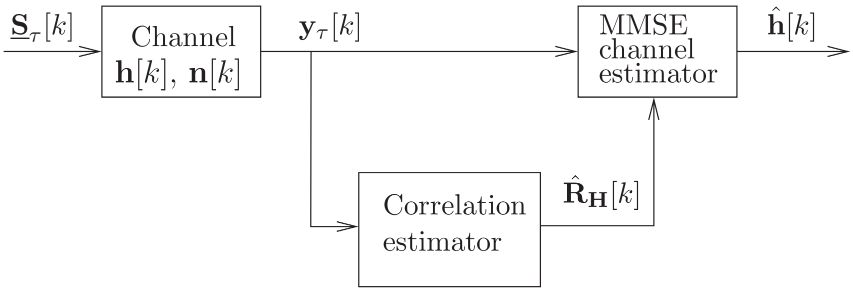


Figure 3.2: Using 2^{nd} order statistics for channel estimation. The training symbols are sent over the channel and produce an output $\mathbf{y}_\tau$. This channel output is used for estimating 2^{nd} order statistics by a correlation estimator. The correlation estimate is improved with each received symbol block and then used by the channel estimator. Using correlation knowledge of the channel significantly improves channel estimation for correlated channels.

minimized by setting $\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau = const \cdot \mathbf{I}$ (see chapter 2.2.1). For more correlated channels (i.e. $\mathbf{R}_\mathbf{H}$ has only a little number of significant eigenvalues), the MSE decreases and the estimation gets better. For rich scattering, the channel coefficients are uncorrelated (spatially white), and the estimator in (3.3) is equivalent to the estimator given in (2.4).

3.1.1 Simulations

Simulations were done for a 4×4 MIMO system. Correlation matrices were either generated synthetically or taken from a measurement campaign [13].

Channel realizations were generated by the formula

$$\mathbf{h} = \mathbf{R}_\mathbf{H}^{1/2} \mathbf{g}, \tag{3.7}$$

where $\mathbf{g}$ is a complex Gaussian random variable with zero mean and unit variance, hence $\mathbf{g} \sim \mathcal{CN}(0, \mathbf{I})$. With this setting, all channel realisations $\mathbf{h}$ have same statistics and are distributed $\mathbf{h} \sim \mathcal{CN}(0, \mathbf{R}_\mathbf{H})$. This can be shown by calculating

$$\mathrm{E}\{\mathbf{h}\mathbf{h}^H\} = \mathbf{R}_\mathbf{H}^{1/2} \underbrace{\mathrm{E}\{\mathbf{g}\mathbf{g}^H\}}_{\mathbf{I}} \mathbf{R}_\mathbf{H}^{H/2} = \mathbf{R}_\mathbf{H}.$$

Various propagation scenarios were assumed. Note that the propagation scenario completely determines the correlation matrix [14]! The joint angular power spectrum (APS), which includes the directions of departure (DOD) and directions of arrival (DOA) of each propagation scenario is shown in figure 3.5, which was obtained by a Bartlett Beamforming algorithm [14]. Figure 3.5a shows the APS for the synthetic channel, which has only one propagation path and can be seen as pin-hole channel. Figure 3.5b and c show the APS for the measured channels used.

All correlation matrices were normalized, such that all channels have the same power by setting $\mathrm{tr}\{\mathbf{R_H}\} = N_r$. The training symbols where chosen such that $\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau = \mathbf{I}$.

The MSE of channel estimation for different signal-to-noise ratios (SNR) was simulated, which means that the MSE was plotted for different values of the SNR. The MSE is defined as the mean square error of channel estimation, so

$$\mathrm{MSE} \triangleq \mathrm{E}\{\|\mathbf{h} - \hat{\mathbf{h}}\|^2\}.$$

The SNR is defined as the average signal to noise ration per receive antenna, so

$$\mathrm{SNR} \triangleq \frac{P_{\mathbf{S}_\tau} \mathrm{tr}(\mathbf{R_H})/N_r}{\sigma_n^2},$$

where $P_{\mathbf{S}_\tau}$ denotes the mean power of one transmit symbol.

The channel was estimated in three ways:

- LS estimation,
- MMSE estimation with white prior,
- MMSE estimation with perfectly known 2^{nd} order statistics.

Figure 3.3 shows results for a *synthetic propagation scenario* with only one propagation path which carries the whole power (further information can be found in [14]). Note that this corresponds to a pin-hole channel! The channel matrix has rank 1. So, all channel coefficients are fully correlated.

For training this is the most ideal case, but note that this channel does not provide any diversity! The curves show that using 2^{nd} order statistics shows great advantages to the

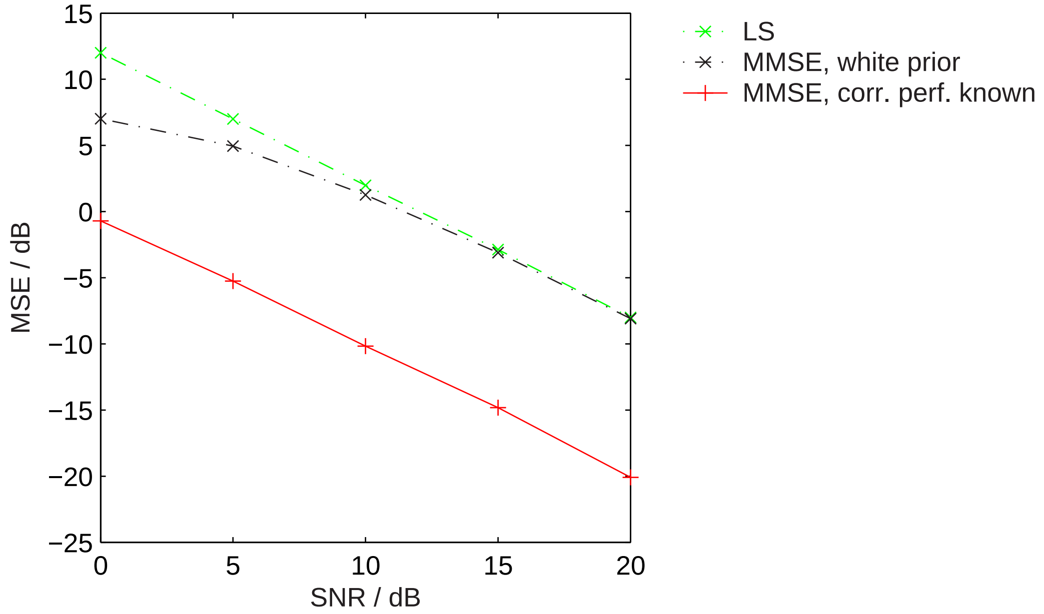


Figure 3.3: MSE versus SNR for channel estimation of a synthetic channel with only one propagation path. One can see that using an MMSE estimator exploiting full knowledge of channel correlation shows significant advantages compared to other estimators for correlated channels. The APS of this synthetic channel is provided in 3.5a.

other estimators. The optimum gain for a 4×4 MIMO system is around 10 dB. This gain rises when using more antennas.

Figures 3.4a and b show channel estimatino behavior for different *measured propagation scenarios*. Training is still much better, when one uses channel statistics, although there is not such a big trade-off as in the fully correlated case.

For a completely uncorrelated channel (which would correspond to the case of a rich scattering propagation scenario), the red curve for perfectly known second order statistics would approach the black curve, which means that the channel is spatially white.

3.2 Estimating channel statistics

The following section is devoted to estimating the second order statistics of the channel. Referring to Figure 3.2, this chapter describes the block "correlation estimator".

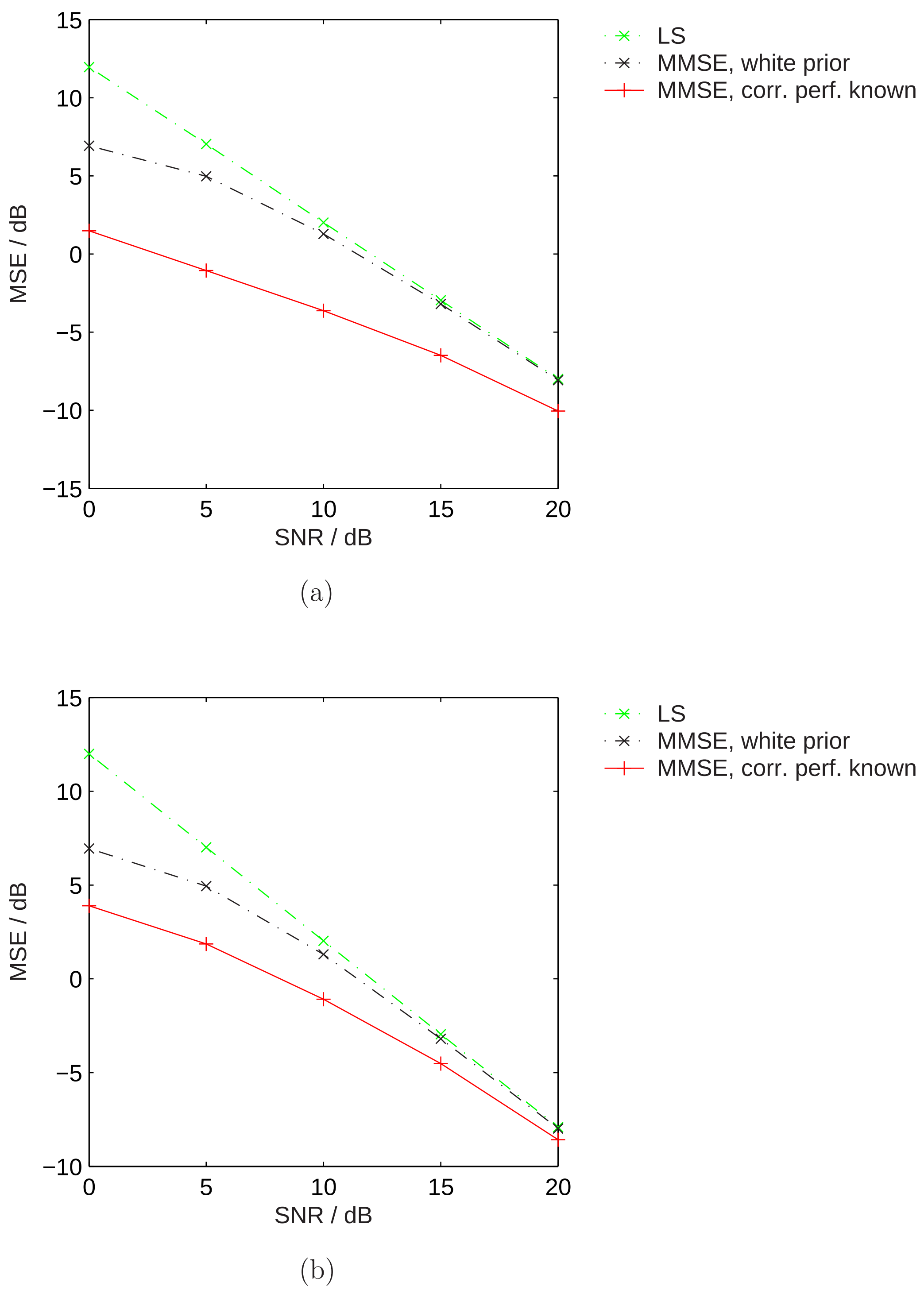
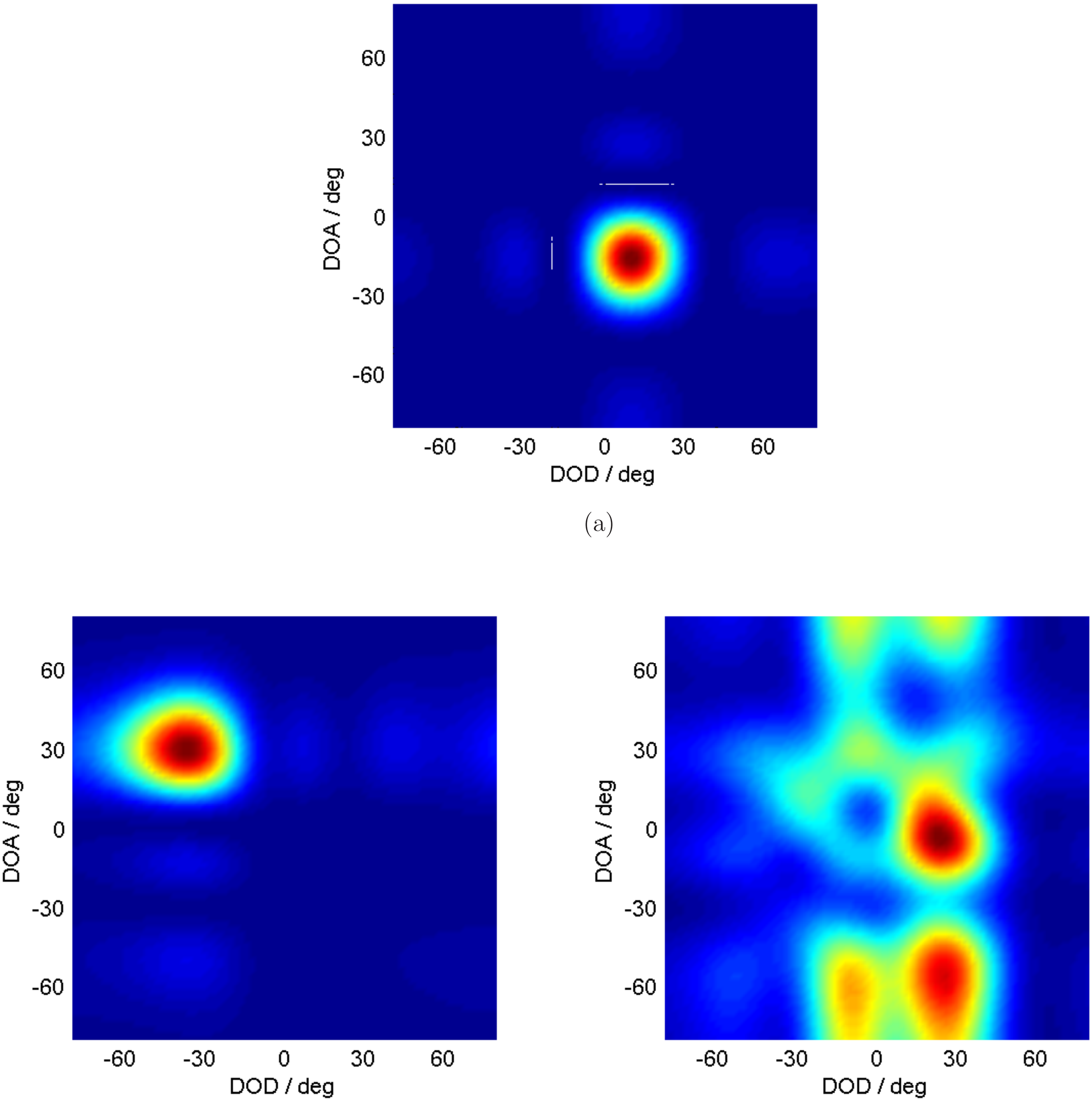


Figure 3.4: MSE versus SNR for channel estimation of two measured channels with different propagation scenarios. The channel in (a) shows more spatial correlation than in (b). APS of the used channels are provided in figure 3.5.



(a)

(b) (c)

Figure 3.5: APS of channels used for simulation: (a) shows a synthetic channel with only one DOD-DOA path (it could be interpreted as perfect pin-hole channel, hence, this channel is fully correlated). (b) and (c) show measured channels. One can see that the channel in (b) shows more spatial correlation than the channel in (c).

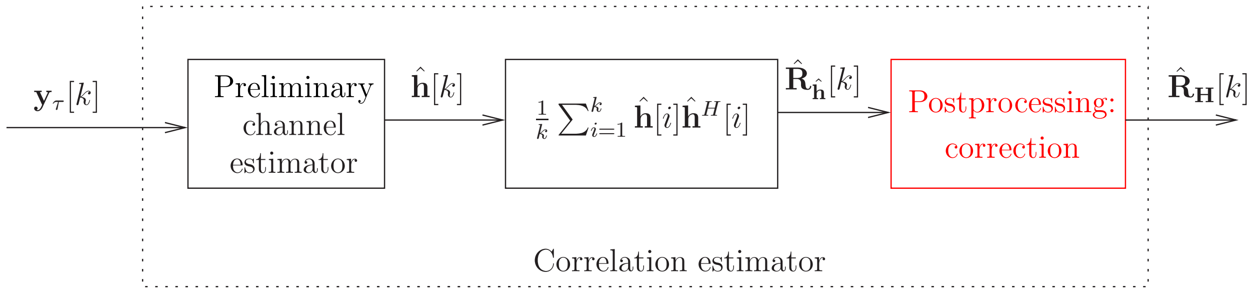


Figure 3.6: Basic principle of correlation estimation. First a channel estimate is built with some appropriate estimator. Those values are accumulated in an estimation of the channel correlation. As it is described later, correcting this estimate significantly improves correlation estimation.

Correlation estimation is done by first estimating channel coefficients by some appropriate estimator and then building a correlation estimate of those estimated channel coefficients. The problem of estimating channel statistics evolves from the fact that those statistics are calculated with *estimates* of the channel coefficients. It strongly depends on the correlation estimator, if the calculated (estimated) statistics are biased, or not. This principle is shown in figure 3.6

Considering an ergodic, wide sense stationary channel[2], the second order statistics $\mathbf{R}_{\hat{\mathbf{h}}}$ of the *estimated channel coefficients* can be estimated by averaging over time. This results in following formula for estimating the statistics:

$$\hat{\mathbf{R}}_{\hat{\mathbf{h}}} = \frac{1}{M}\sum_{i=1}^{M}\hat{\mathbf{h}}[i]\hat{\mathbf{h}}^H[i], \tag{3.8}$$

$\hat{\mathbf{R}}_{\hat{\mathbf{h}}}$ is the estimation of the correlation matrix of the *estimated channel coefficients* $\hat{\mathbf{h}}$ that are estimated by some appropriate estimator and M is the number of transmit blocks that are used for estimating the channel statistics by averaging. When M grows large, this value converges to the correlation matrix of the *estimated channel coefficients* ($\mathrm{E}\{\hat{\mathbf{R}}_{\hat{\mathbf{h}}}\} = \mathbf{R}_{\hat{\mathbf{h}}}$). So, for theoretical investigation I will concentrate on $\mathbf{R}_{\hat{\mathbf{h}}}$ and omit the summation. In the following I will term such estimators that can be used to estimate second order statistics as *"correlation estimators"* (CE). Depending on the way $\hat{\mathbf{h}}$ is computed, it will result in an "LS correlation estimator" (LS CE) or an "MMSE correlation estimator" (MMSE CE).

[2]i.e. the second order statistics $\mathbf{R}_{\mathbf{H}}$ of the channel matrix do not change over time

It is noteworthy that the *correlation matrix of the estimated channel coefficients is in general not equal to the true correlation matrix of the channel!* So, in general,

$$\boxed{\mathbf{R}_{\hat{\mathbf{h}}} \neq \mathbf{R}_{\mathbf{H}}.} \tag{3.9}$$

This can be shown in the following way:

$$\mathrm{E}\{\hat{\mathbf{h}}\hat{\mathbf{h}}^H\} = \mathrm{E}\{(\mathbf{h}+\tilde{\mathbf{h}})(\mathbf{h}+\tilde{\mathbf{h}})^H\} = \mathrm{E}\{\mathbf{h}\mathbf{h}^H\} + \mathrm{E}\{\mathbf{h}\tilde{\mathbf{h}}^H\} + \mathrm{E}\{\tilde{\mathbf{h}}\mathbf{h}^H\} + \mathrm{E}\{\tilde{\mathbf{h}}\tilde{\mathbf{h}}^H\} =$$
$$= \mathbf{R}_{\mathbf{H}} + \mathbf{R}_{\mathbf{h}\tilde{\mathbf{h}}} + \mathbf{R}_{\tilde{\mathbf{h}}\mathbf{h}} + \mathbf{R}_{\tilde{\mathbf{h}}\tilde{\mathbf{h}}}$$

Only if the last three correlation terms compensate to zero, the estimation of the second order statistics is unbiased. An estimator is unbiased, iff $\mathrm{E}\{\mathbf{a} - \hat{\mathbf{a}}\} = 0$ for some random vector $\mathbf{a}$ and its estimate $\hat{\mathbf{a}}$. In the sense mentioned here, the condition on an unbiased correlation estimates reads as $\mathrm{E}\{\mathbf{h}\mathbf{h}^H - \hat{\mathbf{h}}\hat{\mathbf{h}}^H\} = 0$.

Now the question arises which properties a correlation estimator should have. Simulations will show that only unbiased estimators converge to the minimum MSE achievable of channel estimation (see equation 3.6). Nevertheless, even biased estimators perform better than using a white channel estimate from equation (2.4).

3.2.1 LS correlation estimator (LS CE)

Considering the system model from (3.2), we are using now an LS estimator for estimating the channel coefficients:

$$\hat{\mathbf{h}}_{\mathrm{LS}} = \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}_\tau. \tag{3.10}$$

Building the correlation matrix of these estimated channel coefficients gives us the correlation estimate

$$\mathbf{R}_{\hat{\mathbf{h}}_{\mathrm{LS}}} = \mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{LS}}\, \hat{\mathbf{h}}_{\mathrm{LS}}^H\right\}. \tag{3.11}$$

The question arises, if this correlation estimate converges to the true value of $\mathbf{R_H}$ for $M \rightarrow \infty$. This can be calculated by inserting (3.10) and (3.2) into (3.11), which yields

$$\begin{aligned}
&\mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{LS}}\ \hat{\mathbf{h}}_{\mathrm{LS}}^H\right\} = \\
&= \mathrm{E}\left\{\left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}\mathbf{y}^H \underline{\mathbf{S}}_\tau \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\right)^{-1}\right\} = \\
&= (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\ \mathbf{R_H}\ \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} + \sigma_n^2 (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} = \\
&= \mathbf{R_H} + \sigma_n^2 (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}.
\end{aligned} \tag{3.12}$$

Obviously, this correlation estimator is *biased*.

I want to emphasize that this estimator is computationally very efficient, as the training matrix $\underline{\mathbf{S}}_\tau$ is known, and thus the pseudo-inverse $(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} \underline{\mathbf{S}}_\tau^H$ can be calculated once beforehand and then used for estimation. When different training matrices are used, this inverses can still be calculated beforehand, only with adaptive training schemes these inversions have to be done in realtime.

Note that, if the training symbols are orthogonal (see Section 2.2.1), so $\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau = \mathbf{I}$, the LS estimator from (3.10) breaks down to a matched filter estimator. This means that for preliminary estimating the channel coefficients only a matrix-vector multiplication has to be performed!

More comments on estimation of Second Order Statistics using LS estimates

Throughout the literature (e.g. [3] and references therein), this correlation estimator is used for estimating second order statistics, since the estimate $\hat{\mathbf{h}}_{\mathrm{LS}}$ is unbiased, so $\mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{LS}}\right\} = \mathbf{h}$.

This does not mean that estimated second order statistics are also unbiased!

Simulations will show that the use of this correlation estimator will yield worse results in the low-SNR regions compared to more sophisticated correlation estimators.

Further discussion and simulations about the performance of this correlation estimator will be provided in Section 3.3.

3.2.2 Corrected LS Correlation Estimator (C-LS CE)

The result from (3.12) showed, that when estimating the second order statistics of a channel by using LS estimates of the channel coefficients, the correlation estimate does not converge to the true value of $\mathbf{R_H}$, even for an infinite number of training blocks. On the other hand, (3.12) shows only a constant systematic error which can be canceled, if the noise variance σ_n^2 is known. Thus, I introduce a *new "corrected LS correlation estimator"* that is given by

$$\boxed{\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{LS}}} = \hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{LS}}} - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}.} \tag{3.13}$$

This estimator is now unbiased and converges to the true value of $\mathbf{R_H}$. Still, this estimator *does not need any matrix inversions for estimation*! The pseudo-inverse needed for estimation of $\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{LS}}}$ can be calculated beforehand, because the training symbol matrix $\underline{\mathbf{S}}_\tau$ is fixed and known. Note that also the noise power is known, since we use an MMSE estimator as channel estimator.

Further discussion and simulations about the performance of this estimator will be provided in Section 3.3.

3.2.3 MMSE correlation estimator (MMSE CE) using white prior

One can also use MMSE estimates of the channel coefficients for estimating the second order statistics of the channel.

So, estimating the channel coefficients by an MMSE estimator with white prior yields

$$\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}} = \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \mathbf{I}\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}_\tau. \tag{3.14}$$

Again, considering the system model from (3.2), and defining the correlation estimate as the correlation matrix of the estimated channel coefficients as

$$\mathbf{R}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}} = \mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}\, \hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}^H\right\}, \tag{3.15}$$

one can go for investigation of the performance of this correlation estimator.

Inserting (3.14) and (3.2) into (3.15) yields

$$\begin{aligned}\mathbf{R}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}} &= \mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}\ \hat{\mathbf{h}}^{H}_{\mathrm{MMSE},\mathbf{I}}\right\} = \\ &= \mathrm{E}\left\{\left(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I}\right)^{-1}\underline{\mathbf{S}}_{\tau}^{H}\mathbf{y}\mathbf{y}^{H}\underline{\mathbf{S}}_{\tau}\left(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I}\right)^{-1}\right\} = \\ &= (\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau}\ \mathbf{R}_{\mathbf{H}}\ \underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1} + \\ &+ \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}.\end{aligned} \tag{3.16}$$

Using the condition on optimal symbol design from Section 2.2.1, stating that $\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} = \mathbf{I}$ and inserting this into (3.16), yields

$$\mathbf{R}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}} = \frac{1}{(1+\sigma_n^2)^2}\ \mathbf{R}_{\mathbf{H}} + \frac{\sigma_n^2}{(1+\sigma_n^2)^2}\ \mathbf{I}. \tag{3.17}$$

One can see that, again, these correlation estimators in (3.16) and (3.17) do not converge to the true value of $\mathbf{R}_{\mathbf{H}}$.

It has to be emphasized that this estimator also needs no periodic matrix inversions in a wide sense stationary environment. Only one matrix inversion is needed for calculating the inverse term $(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}$.

Further discussion and simulations about the performance of this estimator is provided in Section 3.3.

3.2.4 Corrected MMSE Correlation Estimator (C-MMSE CE) using white prior

As in the LS case, the result from (3.16) shows that when estimating the second order statistics of a channel by using only MMSE estimates, the estimator does not converge to the true value of $\mathbf{R}_{\mathbf{H}}$. On the other hand, (3.16) again shows only a systematic, constant error, which can also be canceled. Thus, I introduce a *new "corrected MMSE correlation estimator"* that is given by

$$\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{I}}} = \mathbf{T}_{\mathbf{I}}\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}}\mathbf{T}_{\mathbf{I}}^{H} - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} \tag{3.18}$$

with

$$\mathbf{T}_{\mathbf{I}} = (\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I}). \tag{3.19}$$

Note that the subscipt $(\cdot)_{\mathbf{I}}$ denotes that a white prior was used for estimation of the channel coefficients. The inversion in matrix $\mathbf{T}_{\mathbf{I}}$ needs to be calculated only once in a wide sense stationary environment, and thus no periodic matrix inversions are needed for estimating the corrected channel statistics.

When the orthogonality condition on training symbol design is fulfilled, the derivation of the estimator is straight-forward and can be derived directly out of (3.18) and (3.19), which yields

$$\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{I}}} = \left(1+\sigma_n^2\right)^2 \cdot \hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}} - \sigma_n^2\mathbf{I}. \tag{3.20}$$

Another interesting fact is that the C-MMSE CE is *completely identical* to the C-LS CE.

This can be shown by simply inserting (3.14) and (3.8) into (3.20), which yields

$$\begin{aligned}
\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE,I}}} &= \mathbf{T}_{\mathbf{I}}\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{MMSE,I}}}\mathbf{T}_{\mathbf{I}}{}^{H} - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} \\
&= \tfrac{1}{M}\textstyle\sum_{i=1}^{M}\mathbf{T}_{\mathbf{I}}\hat{\mathbf{h}}_{\mathrm{MMSE}}[i]\hat{\mathbf{h}}_{\mathrm{MMSE}}^{H}[i]\mathbf{T}_{\mathbf{I}}{}^{H} - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} \\
&= \tfrac{1}{M}\mathbf{T}_{\mathbf{I}}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}\underline{\mathbf{S}}_{\tau}^{H}\left(\textstyle\sum_{i=1}^{M}\mathbf{y}[i]\mathbf{y}^{H}[i]\right)\underline{\mathbf{S}}_{\tau}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau} + \sigma_n^2\mathbf{I})^{-1}\mathbf{T}_{\mathbf{I}}{}^{H} - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} \\
&= \tfrac{1}{M}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1}\underline{\mathbf{S}}_{\tau}^{H}\left(\textstyle\sum_{i=1}^{M}\mathbf{y}[i]\mathbf{y}^{H}[i]\right)\underline{\mathbf{S}}_{\tau}(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1} \\
&= \tfrac{1}{M}\textstyle\sum_{i=1}^{M}\mathbf{h}_{\mathrm{LS}}[i]\mathbf{h}_{\mathrm{LS}}^{H}[i] - \sigma_n^2(\underline{\mathbf{S}}_{\tau}^{H}\underline{\mathbf{S}}_{\tau})^{-1}.
\end{aligned}$$

The last line of this derivation is obviously equal to the C-LS CE.

Again, further discussion and simulations are provided in Section 3.3.

3.2.5 Recursive MMSE correlation estimator (RMMSE CE) using previous correlation as prior

All previously introduced correlation estimators were estimating the channel statistics by first estimating the channel coefficients themselves with an additional estimator. When using the estimation results from the estimator in (3.3), which finally estimates the

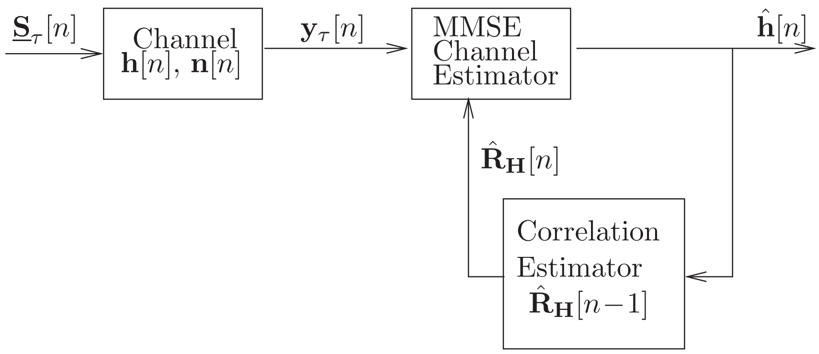


Figure 3.7: Using already estimated channel coefficients for estimating second order statistics. The estimated channel coefficients from the channel estimator are fed back to the correlation estimator.

channel coefficients (with inserted previously estimated channel statistics), one could use these estimates for again estimating the channel statistics, and thus iterating to better results. A block diagram demonstrating the idea is shown in figure 3.7.

Therefore, from the viewpoint from the correlation estimator, the channel estimator can be written as

$$\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{R}}[n] = \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1]\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}_\tau[n], \tag{3.21}$$

where $\mathbf{R}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1]$ is the correlation matrix estimated at iteration step $(n-1)$.

Calculating the value to which this estimator converges reads as

$$\begin{aligned}
&\mathbf{R}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{R}}}[n] = \mathrm{E}\left\{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{R}}\ \hat{\mathbf{h}}^H_{\mathrm{MMSE},\mathbf{R}}\right\} = \\
&= \mathrm{E}\left\{\left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1]\right)^{-1} \underline{\mathbf{S}}_\tau^H \mathbf{y}\mathbf{y}^H \underline{\mathbf{S}}_\tau \left(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1]\right)^{-1}\right\} = \\
&= (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1])^{-1} \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau\ \mathbf{R}_{\mathbf{H}}\ \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1])^{-1} + \\
&\quad + \sigma_n^2 (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1])^{-1} \underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau (\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau + \sigma_n^2 \hat{\mathbf{R}}^{-1}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n-1])^{-1}.
\end{aligned} \tag{3.22}$$

Again, this estimator does not converge to the true value of $\mathbf{R}_{\mathbf{H}}$.

3.2.6 Corrected Recursive MMSE correlation estimator (C-RMMSE CE) using previous correlation as prior

As discussed before, when using estimates of channel coefficients that were obtained by an MMSE estimator that includes a correlation matrix, the estimate is biased.

As done in the white case, one can again correct the result from (3.22) to obtain an unbiased estimator for the second order statistics. The result reads as

$$\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{R}}}[n] = \left(\mathbf{T}_{\mathbf{R}[\mathbf{n-1}]}\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{R}}}[n]\mathbf{T}^{H}_{\mathbf{R}[\mathbf{n-1}]} - \sigma_n^2(\underline{\mathbf{S}}_\tau^H\underline{\mathbf{S}}_\tau)^{-1}\right) \tag{3.23}$$

with

$$\mathbf{T}_{\mathbf{R}[\mathbf{n-1}]} = (\underline{\mathbf{S}}_\tau^H\underline{\mathbf{S}}_\tau)^{-1}(\underline{\mathbf{S}}_\tau^H\underline{\mathbf{S}}_\tau + \sigma_n^2\mathbf{R}[n-1]^{-1}). \tag{3.24}$$

This time, for each iteration step a matrix inversion for a matrix with size $(N_tN_r \times N_tN_r)$ has to be done. This is computationally more complex compared to the previous estimators, which needs no matrix inversions! Hence, it is more efficient to use an additional estimator for estimating second order statistics that does not need to convert huge matrices.

Again, performance discussion and simulations are provided in Section 3.3.

3.3 Performance of estimators for second order statistics

In this section I will compare the performance of previously introduced correlation estimators. It will be shown that unbiased estimators will yield better results than biased estimators. Another question is the convergence speed of the estimators.

3.3.1 Overall Performance

The simulation setup is the same as described in Section 3.1.1. The MMSE channel estimator from equation (3.4) was used, correlation matrices were either taken from the

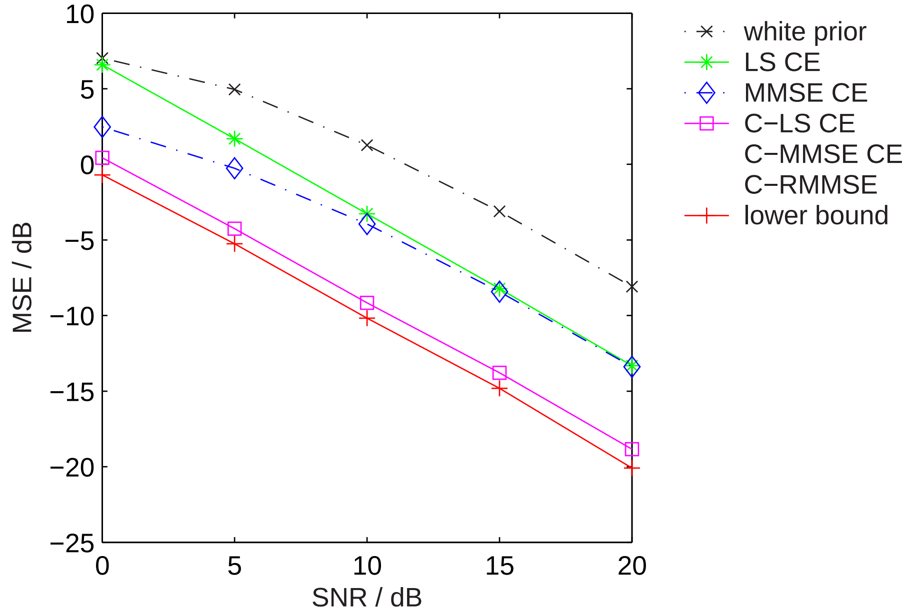


Figure 3.8: MIMO channel estimation with *MMSE channel estimator* using different correlation estimators on a synthetic channel (see APS in figure 3.5a) with $N_t = N_r = 4$ and training length of 1024 receive blocks. The MSE versus the SNR is shown. One can observe the performance of previously introduced correlation estimators. Using correlation estimates shows significant advantages in channel estimation. The LS CE and MMSE CE are biased correlation estimators for but still perform better compared to using a white prior. All corrected correlation estimators (C-LS CE, C-MMSE CE and C-RMMSE CE) converge to the lower bound. The C-LS CE and the C-MMSE CEare identical by definition. The C-RMMSE CE behaves equally well.

correlation estimators described above (LS CE, MMSE CE, C-LS CE, C-MMSE CE, C-RMMSE CE), or chosen constant (white prior, $\mathbf{R_H}$ perfectly known).

Correlation matrices were trained by the correlation estimators over a period of $64 \cdot N_t N_r$ blocks, which makes a total number of 1024 blocks.

In the first simulation, the mean square error of the channel estimation with usage of different correlation estimators was plotted over the SNR. Figure 3.8 shows the results for a synthetic channel with only one propagation path. This scenario corresponds to a totally correlated channel and shows the principle behavior of the algorithms. Simulations for measured channels are provided in Figures 3.9.

One can see that using a white prior is worst for channel training, compared to using correlation estimates.

Throughout many papers, the LS correlation estimator is used, which yields better results compared to using no statistical information, but still the errors are quite large.

The MMSE CE behaves much better than the LS CE at low SNR regions, which is quite intuitive, because the estimator accounts for the noise in the system.

For larger training intervals the corrected correlation estimators (C-LS CE, C-MMSE CE and C-RMMSE CE) all *approach the lower bound* of the channel estimation error, which is given by using the perfectly known (true) channel correlation matrix. The C-LS CE and C-MMSE CE are identical by definition and behave equally, and the C-RMMSE CE also behaves equally well.

At lower correlations (see Figure 3.9), using channel statistics for training is still much better than the use of a white prior, but only at higher correlations the different correlation estimators differ much in performance. Anyway, Figure 3.9 shows that the corrected estimators still outperform the biased estimators.

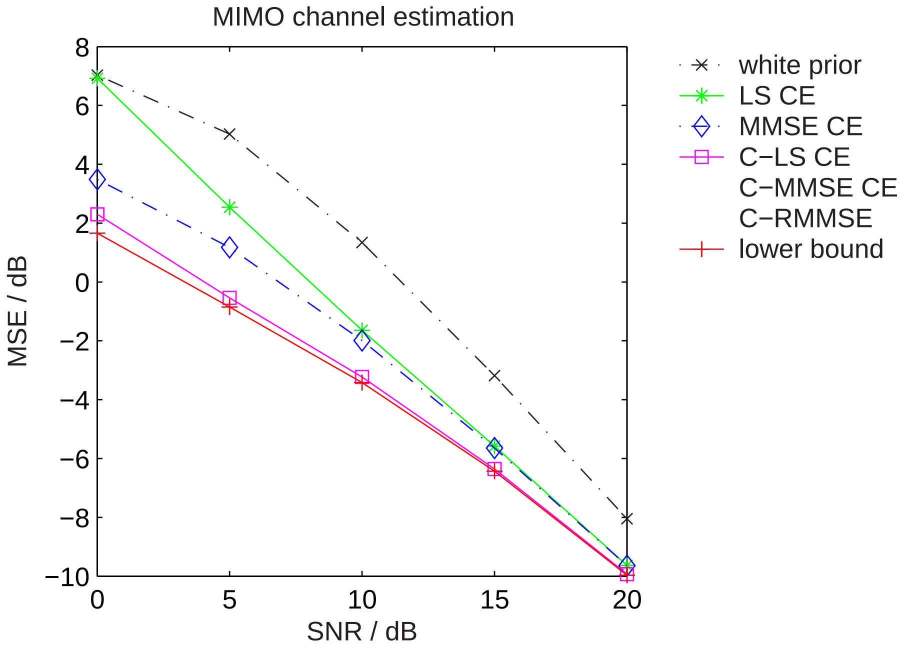
3.3.2 Convergence Speed

Another interesting question is how fast the MSE converges to its final value. The simulation setup is the same as in the section above.

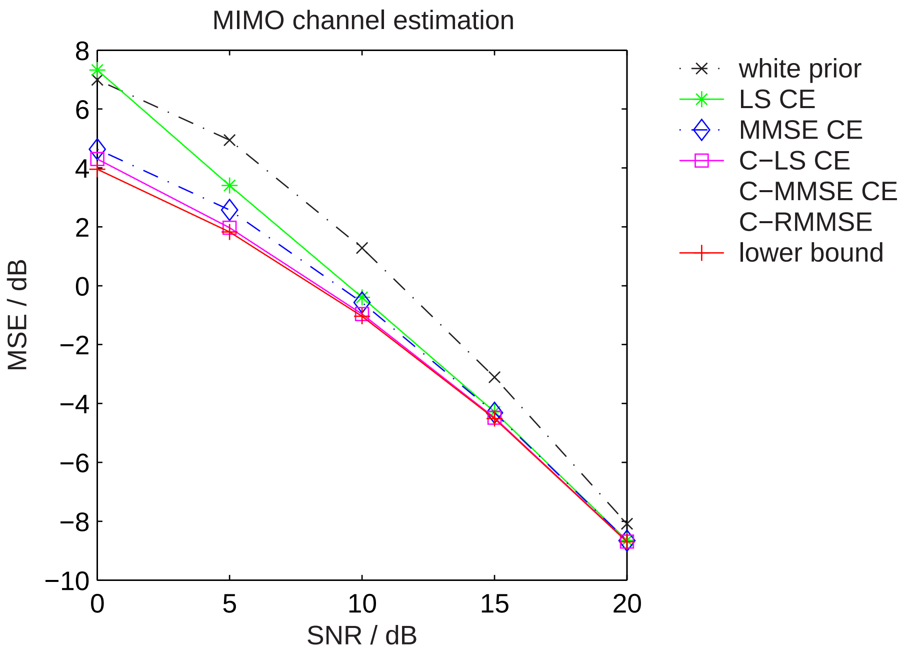
Simulations were done for the MSE of the channel estimation at different training lengths of the correlation matrices. The SNR was set to 0 dB, since here, one can observe the learning effect best.

Correlation matrix training was done in two phases:

Conditioning phase. As the correlation matrices are estimated by averaging over $\mathbf{h}[i]\mathbf{h}^H[i]$ those correlation matrices are singular when starting the algorithm, until a sufficient number of training intervals have passed. When using badly conditioned correlation matrices for estimating the channel coefficients, the estimation errors get worse.



(a)



(b)

Figure 3.9: Same as Figure 3.8 for two measured channels. The channel in (a) shows more spatial correlation than the one in (b). APS of these channels are provided in Figures 3.5b and 3.5c

Therefore, during the conditioning phase, an MMSE estimator with white prior was used for estimating the first channel coefficients. In these simulations a conditioning phase with a length of $(4N_rN_t) = 64$ training blocks was used.

Furthermore, the corrected MMSE estimator with previous correlation as prior (see Section 3.2.5) needs the inverse of the correlation matrix. Building the inverse is not possible, as long as the correlation matrix is singular. Also the iteration does not converge, when the correlation estimate is rank deficient.

Simulations have shown that a training length of at least $4N_rN_t$ is required for the iteration to work.

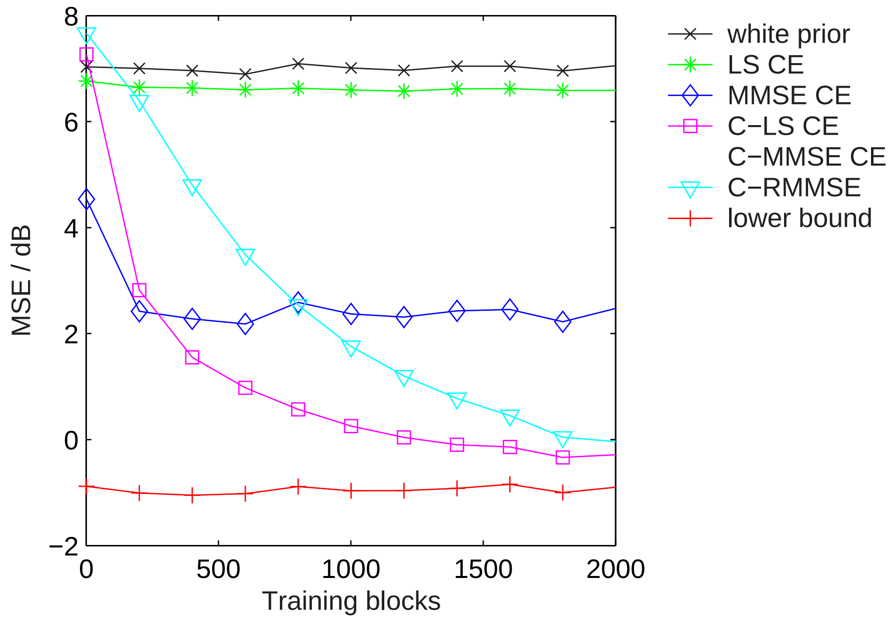
Training phase. During the training phase the correlation estimate is further improved, but channel estimation is already done with the correlation estimates. In communication systems, each training block improves the correlation estimate, so correlation estimation is always done (this would correspond to a training phase with infinite length). For these simulations, the training phase was set to 2000 blocks.

Figure 3.10(a) shows simulation results for a synthetic channel with only one propagation path, figure 3.10(b) is the same for a measured channel (APS in Figure 3.5b) [13].

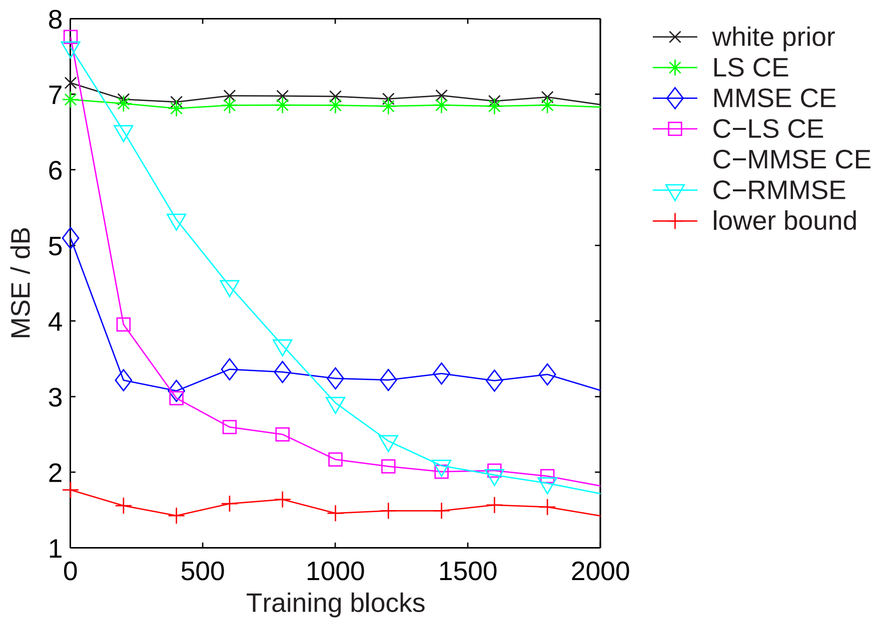
Obviously, the C-LS CE and the C-MMSE CE must behave exactly equal (see Section 3.2.4). Because of the equality of those two estimators, I will furthermore only refer to the C-LS CE as it has less complexity than the C-MMSE CE. Further discussion about complexity is provided in Section 3.4.

The C-RMMSE CE shows a different behavior. It necessarily needs the conditioning phase, otherwise it does not converge. Also, it shows different behavior in convergence time and speed. For the fully correlated channel at low SNR, the convergence time is quite large. When the correlation decreases or the SNR rises, the C-RMMSE CE shows its advantages by faster convergence to a smaller MSE compared to the C-LS CE, but the difference is negligible.

All uncorrected correlation estimators quickly steady on an error level that is far above the MMSE which is again given by using the perfectly known channel statistics.



(a)



(b)

Figure 3.10: Learning curves for MIMO channel estimation at an SNR of 0 dB. The MSE of channel estimation using different correlation estimators is plotted over the number of training blocks, which were used to improve the correlation estimate. The channel in (a) was a synthetic channel with only one propagation path (for APS see Figure 3.5a), (b) shows training for a measured channel (APS in Figure 3.5b). One can see that for lower correlation as in (b) the recursive algorithm (C-RMMSE CE) converges faster, but not as fast as the non-recursive methods (C-LS CE and C-MMSE CE).

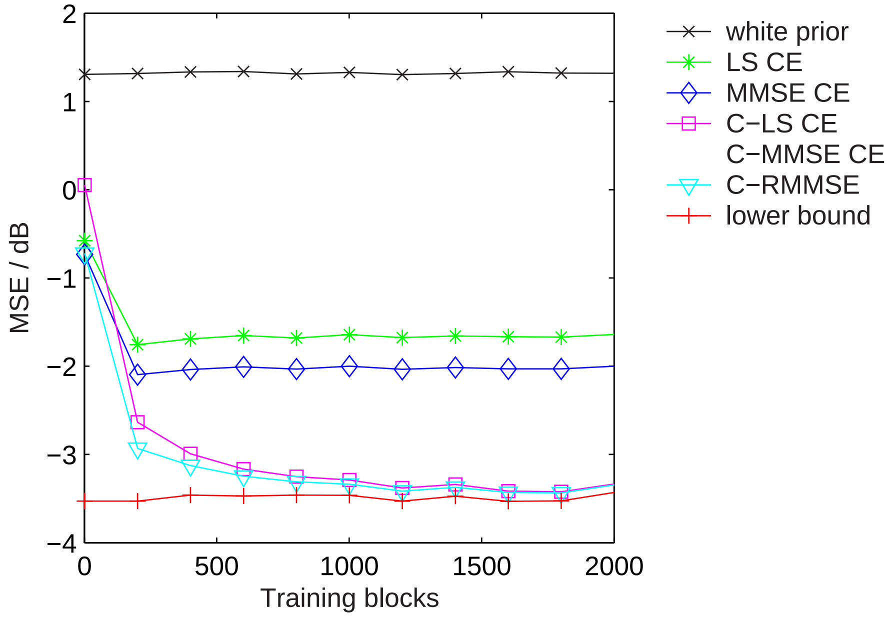


Figure 3.11: Same as Figure 3.10b but for an SNR of 10 dB. It can be observed that in a higher SNR region the recursive algorithm (C-RMMSE CE) converges faster than the non-recursive methods (C-LS CE and C-MMSE CE) and converge to smaller errors faster. The gain of the recursive algorithm is only about 0.1 dB, thus it is negligible.

3.3.3 Summary and Conclusions

The novel corrected correlation estimators show significant advantages to conventional correlation estimators, as they are unbiased. Simulations have shown that when using second order statistics for channel training, unbiased correlation estimators can achieve the MMSE bound, which is given by equation (3.6). Channel estimation using biased correlation estimators do not achieve this lower bound. However, channel estimation using biased correlation estimates still improve the estimate in contrast to using a white prior.

It has to be emphasized that *the corrected estimators have the same complexity order as the biased correlation estimators!* This means that one can achieve much better results at (practically) no additional costs as the corrections are only of linear order.

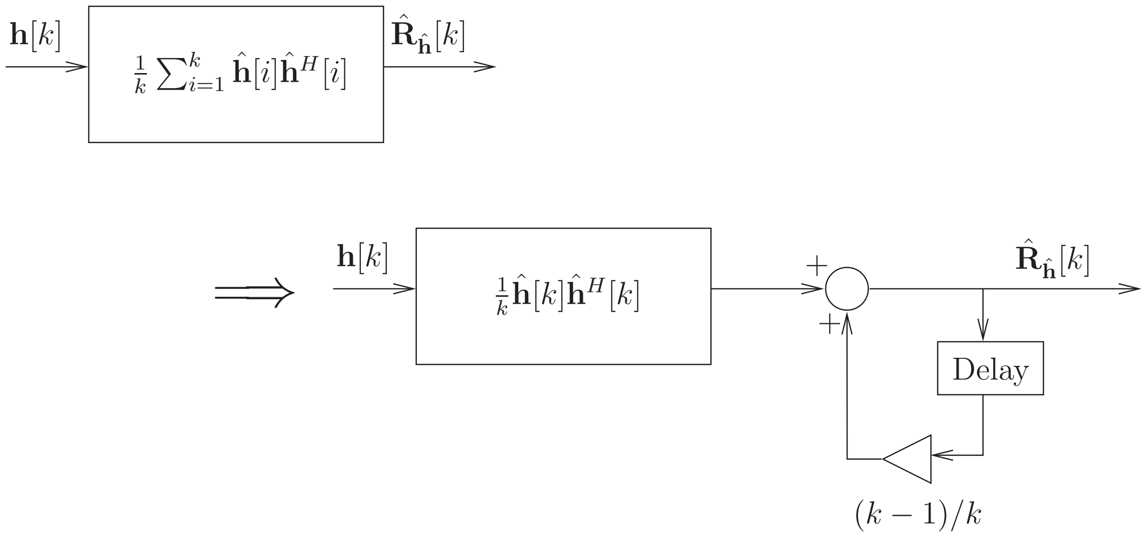


Figure 3.12: Iterative correlation estimation by breaking up the sum (see (3.25)).

3.4 Iterative estimation

The estimation of second order statistics was introduced by averaging over contributions to the correlation matrix (3.8). For a good estimate of the correlation matrix, many channel realizations have to be trained. When considering (3.8), this would result in a large number of outer vector multiplications and a long summation for building the mean. Also, all channel estimates would have to be memorized at each iteration step. This is not very economic, so an iterative way of correlation estimation would be preferable.

The sum from equation (3.8) can be reformulated into a recursive equation. A block diagram demonstrating this idea is presented in figure 3.12.

$$\begin{aligned}\hat{\mathbf{R}}_{\hat{\mathbf{h}}}[k] &= \frac{1}{k}\sum_{i=1}^{k}\hat{\mathbf{h}}[i]\hat{\mathbf{h}}^H[i] = \\ &= \frac{1}{k}\Big((k-1)\mathbf{R}_{\hat{\mathbf{h}}}[k-1] + \hat{\mathbf{h}}[k]\hat{\mathbf{h}}^H[k]\Big)\,.\end{aligned} \tag{3.25}$$

As it was stated above, the correlation of the estimated channel coefficients will not approach the true correlation. Therefore, iteration formulas have to be found for the corrected estimators.

One has to take care that the channel estimation still needs a conditioning phase. Thus, correlation estimates should only be used after a sufficient training phase.

3.4.1 Iterative Corrected LS Correlation Estimator

The iterative corrected LS correlation estimator can be found by inserting (3.25) into (3.13). This yields

$$\begin{aligned}\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{LS}}}[k] &= \hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{LS}}}[k] - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} \\ &= \tfrac{1}{k}\Big((k-1)\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{LS}}}[k-1] + \hat{\mathbf{h}}_{\mathrm{LS}}[k]\hat{\mathbf{h}}_{\mathrm{LS}}^H[k]\Big) - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1} \\ &= \tfrac{1}{k}\Big((k-1)(\hat{\mathbf{R}}_{\hat{\mathbf{h}}_{\mathrm{LS}}}[k-1] - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}) + \hat{\mathbf{h}}_{\mathrm{LS}}[k]\hat{\mathbf{h}}_{\mathrm{LS}}^H[k] - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}\Big) \\ &= \tfrac{1}{k}\Big((k-1)\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{LS}}}[k-1] + \hat{\mathbf{h}}_{\mathrm{LS}}[k]\hat{\mathbf{h}}_{\mathrm{LS}}^H[k] - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}\Big)\,.\end{aligned} \tag{3.26}$$

This is a simple update equation for the next estimate of the correlation estimator. For the conventional LS correlation estimator only the last term $(\sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1})$ has to be canceled.

Of course, this iterative estimator is completely equal to the estimator introduced in equation (3.13).

So, one update has the complexity $\mathcal{O}(N_r N_t)^2$ which results from the outer multiplication of the two vectors and the multiplication with the factor $(k-1)/k$. The extension to the corrected LS only needs to consider the term $\sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}$, this corresponds to an additional number of $(N_r N_t)$ subtractions (which has linear complexity and thus is negligible).

3.4.2 Iterative Corrected MMSE Correlation Estimator with white prior

As for the C-LS CE one can derive an update equation for the C-MMSE CE. The update equation reads as

$$\begin{aligned}\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{I}}}[k] = \ & \tfrac{1}{k}\Big((k-1)\hat{\mathbf{R}}_{\mathbf{H}_{\mathrm{MMSE},\mathbf{I}}}[k-1] + \\ & + \mathbf{T}_{\mathbf{I}}\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}[k]\hat{\mathbf{h}}_{\mathrm{MMSE},\mathbf{I}}^H[k]\mathbf{T}_{\mathbf{I}}^H - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}\Big)\end{aligned} \tag{3.27}$$

with $\mathbf{T_I}$ from equation (3.19).

This estimator has a higher complexity than the C-LS CE because of the multiplications with the correction matrices $\mathbf{T_I}$. The complexity is of order $\mathcal{O}(N_r N_t)^3$. However, if one can use the condition for optimal symbol design ($\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau = \text{const} \cdot \mathbf{I}$), this matrix multiplications with $\mathbf{T_I}$ breaks down to a scalar multiplication, and the complexity is again equal to the C-LS CE.

Note that the C-MMSE CE is exactly equal to the C-MMSE CE. In this sense it is reasonable to use the C-LS CE instead.

3.4.3 Iterative Corrected Recursive MMSE Correlation Estimator

Also for the C-RMMSE CE an update equation can be derived. It reads as

$$\begin{aligned} \hat{\mathbf{R}}_{\mathbf{H}_{\text{MMSE},\mathbf{R}}}[k] = \; & \tfrac{1}{k}\Big((k-1)\hat{\mathbf{R}}_{\mathbf{H}_{\text{MMSE},\mathbf{R}}}[k-1] + \\ & + \mathbf{T_{R[k-1]}}\; \hat{\mathbf{h}}_{\text{MMSE},\mathbf{R}}[k]\hat{\mathbf{h}}_{\text{MMSE},\mathbf{R}}^H[k]\; \mathbf{T_{R[k-1]}^H} - \sigma_n^2(\underline{\mathbf{S}}_\tau^H \underline{\mathbf{S}}_\tau)^{-1}\Big) \end{aligned} \tag{3.28}$$

with $\mathbf{T_{R[k-1]}}$ from equation (3.24).

This correlation estimator has a complexity of order $\mathcal{O}(N_r N_t)^3$, because it needs a matrix inversion for calculating $\mathbf{T_{R[k-1]}}$. So, the complexity order has increased compared to the other correlation estimators! It shows only minor advantages to the previously introduced correlation estimators as discussed in Section 3.3, and thus is not recommendable for deploying in mobile communication devices.

3.4.4 Conclusions

This section has shown a way for iterative correlation estimation. The iterative C-LS CE does not need any matrix inversions and has complexity order $\mathcal{O}(N_r N_t)^2$. It also shows a fast convergence to the true value of the channel correlation matrix $\mathbf{R_H}$.

The iterative C-MMSE CE and C-RMMSE CE have both complexity order $\mathcal{O}(N_r N_t)^3$, which evolves from matrix multiplications and in the second case from an additional matrix inversion. As discussed in Section 3.3.2, the C-MMSE CE is identical to the C-LS CE and has thus to be disregarded. The C-RMMSE CE shows faster convergence for less correlated channels and high SNR but is computationally inefficient compared to the C-LS CE.

Thus, the iterative corrected LS correlation estimator turns out to be a good choice for correlation estimation.

Appendix A

List of Symbols

Symbol	Description
N_r	number of receive antennas
N_r	number of transmit antennas
L	length of transmit block
L_d	number of data symbols (mostly equal to length of data block)
L_τ	number of training symbols (mostly equal to length of training block)
L_h	length of impulse response
$\mathbf{I}$	identity matrix of appropriate size
$\mathbf{H}$	channel matrix
$\mathbf{S}$	transmit matrix
$\mathbf{Y}$	receive matrix
$\mathbf{N}$	additive noise matrix
σ_n^2	noise power per receive antenna
$\hat{\mathbf{H}}$	estimated channel matrix (obtained by matrix estimator)
$\tilde{\mathbf{H}}$	estimation error of $\hat{\mathbf{H}}$
$(\cdot)_\tau$	denotes training symbols
$(\cdot)_d$	denotes data symbols

Symbol	Description
ρ	mean SNR
ρ_τ	mean power of training symbols
ρ_d	mean power of data symbols
α	power allocation factor
C_τ	training based capacity
ρ_{eff}	effective SNR used in C_τ
$\mathbf{h}$	stacked channel vector
$\underline{\mathbf{S}}_\tau = (\mathbf{I} \otimes \mathbf{S}_\tau)$	training symbol matrix
$\mathbf{y}_\tau$	stacked receive vector of training symbols
$\mathbf{n}$	stacked noise vector
$\hat{\mathbf{h}}$	estimated stacked channel vector (obtained by vector estimator)
$\tilde{\mathbf{h}}$	estimation error of $\hat{\mathbf{h}}$
$\mathbf{R}_\mathbf{H}$	Correlation matrix of the true channel
$\hat{\mathbf{R}}_\mathbf{H}$	estimated (unbiased) correlation matrix
$\mathbf{R}_{\hat{\mathbf{h}}}$	correlation matrix of estimated channel coefficients
$\hat{\mathbf{R}}_{\hat{\mathbf{h}}}$	estimated correlation matrix of estimated channel coefficients
SNR	Signal-to-noise ratio
MSE	Mean square error
APS	Angular Power Spectrum
DOA	Direction of Arrival
DOD	Direction of Departure
LS	Least squares (estimation algorithm)
MMSE	Minimum mean square error (estimation algorithm)
C-LS CE	Corrected LS *correlation* estimator
C-MMSE CE	Corrected MMSE *correlation* estimator
C-RMMSE CE	Corrected recursive MMSE *correlation* estimator

Bibliography

[1] B. Hassibi and B. Hochwald, "How much training is needed in multiple-antenna wireless links?," *IEEE Transactions on Information Theory*, vol. 49, pp. 951 – 963, Apr. 2003.

[2] M. Dong and L. Tong, "Optimal design and placement of pilot symbols for channel estimation," *IEEE Transactions on Signal Processing*, vol. 50, pp. 3055 – 3069, Dec. 2002.

[3] F. Dietrich and W. Utschik, "Pilot assisted channel estimation based on second order statistics," *Accepted for IEEE Transactions on Signal Processing*, 2004.

[4] I. E. Telatar, "Capacity of multi-antenna gaussian channels," Tech. Rep. BL0112170-950615-07TM, AT&T Bell Laboratories, June 1995.

[5] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, 3rd ed., 1995.

[6] X. Ma, L. Yang, and G. Giannakis, "Optimal training for mimo frequency-selective fading channels," *Submitted to IEEE Trans. on Wireless Comm.*, Oct. 2002.

[7] S. Adireddy and L. Tong, "Detection with embedded known symbols: Optimal symbol placement and equalization," *ICASSP Proceedings*, pp. 2541–2543, Jun 2000.

[8] F. Hlawatsch, *Statistical Signal Processing: Parameter Estimation Methods.* Lecture notes, Vienna University of Technology, 2003.

[9] T. Marzetta, "BLAST training: Estimating channel characteristics for high-capacity space-time wireless," in *Proc. 37th Allerton Conf. on Commun., Contr., Comput.*, Sept. 1999.

[10] M. Debbah and R. Müller, "MIMO channel modelling and the principle of maximum entropy: An information theoretic point of view," *Submitted to IEEE Transactions on Information Theory*, Sept. 2003.

[11] J. Balakrishnan, M. Rupp, and H. Visvanathan, "Optimal channel training for multiple antenna systems," in *Proc. Conf. Multiple Access, Mobility and Teletraffic for Wireless Communications*, Dec. 2000.

[12] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Englewood Cliffs (NJ): Prentice Hall, 1993.

[13] M. Herdin, H. Oezcelik, H. Hofstetter, and E. Bonek, "Variation of measured indoor MIMO capacity with receive direction and position at 5.2 GHz," *Electronics Letters*, vol. 38, pp. 1283–1285, Oct. 2002.

[14] W. Weichselberger, *Spatial Structure of Multiple Antenna Radio Channels - A Signal Processing Viewpoint.* PhD thesis, Technische Universität Wien, Dec. 2003.