M.Sc in Telecommunications

Reduced-complexity semi-blind channel estimation for LTE Downlink

Niccolò Franceschi

Supervisors:
Lars Christensen (Renesas Mobile)
Søren Christensen (Renesas Mobile)
Ole Winther (DTU IMM)

Technical University of Denmark
August 2012

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

Summary

The advance of MIMO techniques as a means of boosting data rate and reliability in wireless communications has challenged researchers to investigate new channel estimation methods. As MIMO multiplies the number of channel parameters, longer pilot-sequences need to be sent to attain the same accuracy as SISO. Of course, this increases the overhead resulting in a waste of channel capacity. *Semi-blind* channel estimators address this problem making use of both pilot-sequence and user data to enhance the quality of the estimate. Even though these methods are appealing in terms of mean squared error, they considerably raise the complexity of the receiver. This issue is even more severe if we consider that MIMO is expected to speed up the bit rate, meaning that an increasing amount of data has to be processed to produce the estimate.

The aim of this thesis is investigating low-complexity semi-blind estimation techniques, capable of improving the mean squared error and still computationally affordable. Firstly, the MIMO-LTE channel model is formulated, then we will discuss traditional pilot-only estimation and its limitations. Afterwards, the semi-blind problem is presented and expressed using two different approaches, one relying on the true discrete distribution of the data symbols and the other on a Gaussian approximation. Then, EM-based solutions are derived and compared with numerical techniques that are independent of the size of the data sequence. Finally, all these methods are tested through simulations assessing their accuracy and computational cost.

Acknowledgements

First of all, I would like to thank Dr. Lars Christensen and Dr. Niels Mørch for giving me the opportunity to work in such an inspiring environment as Renesas Mobile Corporation. A special thanks to my supervisors, Dr. Lars Christensen, Dr. Søren Christensen and Prof. Ole Winther for supporting me with many valuable comments and ideas. Thanks to Dr. Maja Loncar and Dr. Pedro Højen-Sørensen for all the precious insights and suggestions they gave me. A special thanks to Maja for reading the manuscript and providing thorough remarks.

I owe a debt of gratitude to my parents, Licia and Vinicio, and my grandparents for their total support and empathy. I want to thank Annalisa for bridging any distance with her tenderness. Thanks to my friends back home for their long-time affection (you know who you are!), and to the new friends that shared this experience with me, especially Miguel, Cristian and Andrea.

Contents

List of Figures

List of Algorithms

Notation

Scalars

k		discrete-time sample index
l		filter tap index
n		sub-carrier index
p		OFDM symbol index
R		number of receiving antennas
T		number of transmitting antennas
N		number of OFDM sub-carriers
CP		cyclic prefix length
K		total number of symbols
K_p		number of pilot symbols
K_d		number of data symbols

Vectors

$X_n[p]$	T	inputs on sub-carrier n and OFDM symbol p
$Y_n[p]$	R	outputs on sub-carrier n and OFDM symbol p
$x^p[k]$	T	p-th OFDM transmitted symbol
$y^p[k]$	R	p-th OFDM received symbol
$w^p[k]$	R	noise vector for the p-th OFDM symbol
X	$T \cdot K$	collection of transmitted symbols
Y	$R \cdot K$	collection of received symbols
W	$R \cdot K$	collection of noise samples
$X^{(\mathcal{P})}$	$T \cdot K_p$	collection of pilot inputs
$X^{(\mathcal{D})}$	$T \cdot K_d$	collection of data inputs
$Y^{(\mathcal{P})}$	$R \cdot K_p$	collection of outputs associated with pilot inputs
$Y^{(\mathcal{D})}$	$R \cdot K_d$	collection of outputs associated with data inputs
$W^{(\mathcal{P})}$	$R \cdot K_p$	collection of noise samples associated with pilot inputs
$W^{(\mathcal{D})}$	$R \cdot K_d$	collection of noise samples associated with data inputs

Matrices

h_l	$R \times T$	l-th tap of the MIMO channel matrix
H	$R \times T$	MIMO channel matrix in the frequency domain
Σ_x	$T \times T$	Symbol covariance matrix $\mathbb{E}\left[X[k]X[k]^H\right]$
$\Lambda_{yx}^{(\mathcal{P})}$	$R \times R$	$\sum_{k=0}^{K_p-1} Y^{(\mathcal{P})}[k]X^{(\mathcal{P})}[k]^H$
$\Lambda_{xx}^{(\mathcal{P})}$	$R \times R$	$\sum_{k=0}^{K_p-1} Y^{(\mathcal{P})}[k]X^{(\mathcal{P})}[k]^H$
$\Lambda_{yy}^{(\mathcal{D})}$	$R \times R$	$\sum_{k=0}^{K_d-1} Y[k]^{(\mathcal{D})}Y[k]^{(\mathcal{D})^H}$
Ω_{yx}	$R \times R$	$\Lambda_{yx}^{(\mathcal{P})} + \sum_{k=0}^{K_d-1} Y^{(\mathcal{D})}[k]\,\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]^H \middle\vert Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right]$
Ω_{xx}	$R \times R$	$\Lambda_{xx}^{(\mathcal{P})} + \sum_{k=0}^{K-1} \mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]X^{(\mathcal{D})}[k]^H \middle\vert Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right]$

Chapter 1

Introduction

Contemporary society has been defined by various thinkers as an *Information Society* [1]. Manipulation, distribution, control and diffusion of information play a central role in our globalized economy and culture. As many other changes, the advent of Information Society has been supported by a technological revolution. In particular, computer science and telecommunications allowed a tremendous amount of information to be transmitted and processed in real time. This handy and fast access to information paved the way for a new way of making use of knowledge. Especially, mobile communications are making information technology all-pervading and are creating unthought needs and behaviours with a huge business potential. In this context, it seems that the need of ubiquitous, faster and more reliable communication is a major economy driver.

Long Term Evolution (LTE) is a *3rd Generation Partnership Project* (3GPP) technology that attempts to push the limits of mobile communication a step further. Employing advanced physical layer techniques such as *Orthogonal Frequency Division Multiplexing* (OFDM) and *Multiple Input Multiple Output* (MIMO), LTE achieves 300 Mbps nominal downlink speeds and latencies below 5 ms. In this initial chapter we present the main limitations affecting wireless communication and we give reasons for MIMO and OFDM. Then, the problem of channel and noise estimation is formulated, stressing semi-blind channel estimation, which is the object of this thesis.

1.1 The wireless channel

In contrast to wired communications, wireless transmission is characterized by two significant aspects that make it a challenging topic. Firstly, channel strength varies abruptly over frequency and time; this phenomenon is called *fading* and consists of two com-

ponents: large scale fading, caused by power losses and large obstacles shadowing the receiver; small scale variations, which are the effect of interference among multiple replicas of the transmitted signal. Secondly, wireless communication is affected by multi-user interference: unlike wired technology, transmitters and receivers are not isolated from the rest of the environment. Additionally, as other electronic devices, wireless receivers are prone to thermal and shot noise, often modelled as *Additive White Gaussian Noise* (AWGN).

A thorough physical description of the medium can be provided using Maxwell's theory of electromagnetism. Given the coordinates of the receiver-transmitter pair, the structure of the environment and the analytical expression of the signal, it is possible to solve a system of partial differential equations and produce a full solution of the communication problem. However, even if this information were available, it would require an infeasible computational effort to be processed. Thus, a statistical approach is the most reasonable way of describing wireless transmission. This means that the nature of the channel becomes a black-box and its global effect is described as a linear system with transfer function given by a stochastic process.

Even from a probabilistic standpoint, it is still useful to make use of some approximate physical description for the key parameters of the model. In fact, having a rough physical representation of the medium enables us to select the statistical model that best suits the real channel. There are two general guidelines that can provide insight on the experienced channel conditions: *coherence time* and *coherence bandwidth*. Coherence time is the interval over which the channel can be considered constant; physically, this quantity is related to Doppler spread, i.e. the maximum Doppler shift experienced by interfering signal replicas. Coherence bandwidth is analogous to coherence time but in frequency domain: it is the frequency interval it takes to the received signal to change significantly. Coherence bandwidth is related to a quantity named delay spread, which is the maximum delay difference among different versions of the signal.

Coherence time and coherence bandwidth are not significant in an absolute sense, but only when related to the transmitted signal. In fact, the relation between frequency coherence and signal bandwidth influences the way the wireless channel is modelled: if frequency coherence is larger than the bandwidth, then the channel is called *flat fading* and can be represented as a discrete-time linear system with one-tap transfer function. In this case, delay spread is smaller than symbol time and adjacent symbols do not interfere with one another, i.e. *Inter Symbol Interference* (ISI) is negligible.

When compared to purely AWGN channels, wireless medium - often modelled as a Rayleigh fading channel - shows evident limitations: in fact, there is a significant

probability that the channel is in *deep fade*. Assuming channel taps are represented as zero mean Gaussian random variables, we call deep fade event the situation where the square norm of channel taps is smaller than the SNR. In that condition, the signal is heavily attenuated and the "instantaneous" SNR is much poorer than the nominal one, which is an average quantity. Hence, signal degradation is twofold in the wireless environment: both AWGN and channel fading contribute to impair performances. In general, it can be seen that bit error probability drops exponentially with SNR for AWGN channels, while for fading channels, if no measures are taken, only at a linear rate.

One way of overcoming these limitations is *diversity coding*. The idea is that the probability of deep fading decreases if symbols are transmitted through independently fading channels. This technique is employed in all wireless communication systems in use and makes the medium much more reliable. There are several ways of producing diversity: *time diversity* is obtained interleaving a codeword over different fades of the channel; *frequency diversity* exploits redundancy provided by multiple paths, which can be done utilising equalization techniques, spreading the symbols on a wide spectrum or coding the information over different frequency bands; *space diversity* makes use of multiple antennas placed more than one wavelength apart.

The focus of this thesis will be on MIMO and OFDM. OFDM is a transmission technique with a great advantage on wireless channels: it turns a frequency selective channel into a number of independent flat fading channels called subcarriers. This gives two main benefits: first, frequency diversity is gained because symbols are transmitted over independent frequency bands; second, the equalization of each subcarrier is straightforward because at each subcarrier the channel is flat fading. In the last years OFDM has become increasingly popular and it is used in all latest wireless systems, like ETSI DVB-T (Digital Television), IEEE 802.16 (WiMAX), IEEE 802.11 (Wi-Fi) and 3GPP LTE (4G).

MIMO is a versatile communication paradigm coming with two advantages: it boosts transmission speed and it provides space diversity. In particular, using T transmitting and R receiving antennas theoretically produces T degrees of freedom and up to $T \cdot R$ diversity; moreover the capacity of the channel is increased by a factor $\min(T, R)$. For an exhaustive theoretical derivation of MIMO channel capacity and diversity, see [2]. There are two main ways of operating a MIMO channel: *spatial multiplexing* techniques (such as V-BLAST) send independent symbol streams over different antennas; while *space-time coding* methods (such as Alamouti schemes or repetition codes) map each symbol of a data stream into a space-time coordinate. The model presented in this thesis is for spatial multiplexing; for each OFDM subcarrier, independent data streams

are encoded across T transmitting antennas. MIMO has been successfully implemented in a few systems: IEEE 802.11n (WiFi), IEEE 802.16e (WiMAX) and of course 3GPP LTE.

1.2 Channel Estimation

In the simple case of a Single Input Single Output (SISO) flat fading channel with no noise, the effect of the channel is just multiplying each symbol by a zero-mean normally distributed complex number. This is equivalent to a scalar real-valued multiplication of the magnitude followed by a rotation in the complex plane. It is easy to see that - assuming M-PSK modulation- the received symbol carries no information and it is totally useless for detection. In fact, the uniform rotation given by the channel completely destroyed the phase information. Thus, estimating the channel correctly and compensating for its effects is a crucial part of the receiver.

Channel estimation is usually performed transmitting a known sequence of symbols, called *pilot sequence*, and inferring the *Channel State Information* (CSI) from the observations on the receiver side. If $H \in \mathbb{C}^{R\times T}$ is a matrix containing the parameters to estimate, and $\hat{H}(y)$ is a matrix-valued function of the observed values used to estimate H (also called an *estimator* for y), a popular measure of its accuracy is the *Mean Square Error* (MSE):

$$\begin{aligned} MSE(\hat{H}) &\triangleq \frac{1}{RT}\mathbb{E}\left[\left\|\hat{H}(y)-H\right\|^2\right] = \frac{1}{RT}\sum_{i=1}^{R}\sum_{j=1}^{T}\mathbb{E}\left[\left|\hat{H}_{i,j}(y)-H_{i,j}\right|^2\right] \\ &= \frac{1}{RT}\mathrm{Tr}\left\{\mathbb{E}\left[\left(\hat{H}(y)-H\right)\left(\hat{H}(y)-H\right)^H\right]\right\} \end{aligned} \tag{1.1}$$

In this case we chose to use the Frobenius norm of the parameter matrix, which is the sum of the MSE of its components. In order to make the MSE comparable for different sizes of H, a normalization factor is introduced. Another important characterization is the estimation *bias*, defined as:

$$\mathcal{B}(\hat{H}) \triangleq \mathbb{E}\left[\hat{H}(y)-H\right] \tag{1.2}$$

Thus, if $\hat{H}(y)$ is unbiased, (1.1) becomes:

$$MSE(\hat{H}) = \frac{1}{RT}\mathrm{Tr}\left\{\mathbb{E}\left[\left(\hat{H}(y)-\mathbb{E}\left[\hat{H}(y)\right]\right)\left(\hat{H}(y)-\mathbb{E}\left[\hat{H}(y)\right]\right)^H\right]\right\} \tag{1.3}$$

In general, the *channel estimator* $\hat{H}(y)$ is a function of random variables and there is no unique way of constructing it; some different rationales can be used:

Minimum MSE (MMSE) estimators choose the value of H that minimizes the MSE:

$$\hat{H}_{MMSE}(y) \triangleq \arg\min_{\hat{H}}\{\mathbb{E}[\|\hat{H}(y) - H\|^2]\} = \mathbb{E}[H|y] \tag{1.4}$$

Note that the expectation in this case is a Bayesian operator; in other words, we assume the posterior distribution of H to be known, and we integrate (or sum) over it.

Maximum a Posteriori (MAP) determines the value of H maximizing the posterior probability of the observations:

$$\hat{H}_{MAP}(y) \triangleq \arg\max_{H}\{p(H|y)\} \tag{1.5}$$

As in the previous case, the distribution of H must be known. It is possible to think of MAP and MMSE as similar criteria: while MAP chooses the value of H producing the maximum value of $p(H|y)$, MMSE selects the one in correspondence to the expected value of $p(H|y)$. If the posterior distribution is symmetric these methods are the same.

Maximum Likelihood (ML) estimators choose H such that:

$$\hat{H}_{ML}(y) \triangleq \arg\max_{H}\{p(y;H)\} \tag{1.6}$$

In this case a semicolon symbol is used to stress the fact that this posterior probability is different from the one used before; in fact we do not make any assumptions on the distribution of H, which is treated as a deterministic parameter. Since in the most of this thesis we will deal with ML estimators, keeping this difference in mind the symbol "|" is used in place of ";".

Of course the length of the pilot sequence influences the accuracy of the estimation process. On the other hand, longer pilot sequences yield a higher overhead, since its transmission is a waste of channel capacity. So, there is a trade-off between accuracy and overhead in the estimation process.

One strategy to overcome this problem is *semi-blind* channel estimation. This approach not only relies on pilot symbols for estimating the CSI, but also exploits the unknown user data. This extra information can be used to attain the same estimation

accuracy reducing the number of pilots, or to improve the performance of the estimator keeping fixed number of pilots. Unfortunately, this enhancement comes at a price: semi-blind channel estimation requires the receiver to run iterative algorithms (typically EM) that can burden it consistently, especially at high data rates. Moreover, this kind of algorithms scale at least linearly with the size of the unknown data. Hence, another trade-off is found: larger data samples improve the pilot-based estimation but overload the receiver. The aim of this thesis is finding semi-blind estimation techniques with reduced complexity with respect to the existing ones.

1.2.1 Cramer-Rao Lower Bound

It is often useful to have an optimal reference to compare estimators. A well known lower bound for mean squared error is the *Cramer-Rao Lower Bound (CRLB)*. Although there exist several versions of CRLB, for both Bayesian and non-Bayesian estimators, in this thesis the CRLB for complex-valued unbiased non-Bayesian estimators is used. In particular, since the estimation is often performed in the complex domain, we will extend the usual real-valued CRLB to complex parameters, as presented in [3]. Let $\theta \in \mathbb{C}^K$ be a complex vector of parameters and $\hat{\theta}(Y) \in \mathbb{C}^K$ be an estimator for them. We can create two auxiliary vectors:

$$\gamma = \begin{pmatrix} \theta^T & \theta^H \end{pmatrix}^T \in \mathbb{C}^{2N} \qquad \hat{\gamma} = \begin{pmatrix} \hat{\theta}^T & \hat{\theta}^H \end{pmatrix}^T \in \mathbb{C}^{2N} \tag{1.7}$$

The *Fisher Information Matrix* (FIM) for complex parameters is defined as:

$$\mathcal{I}_\gamma \triangleq \mathbb{E}\left[\left(\frac{\partial \ln p\left(Y|\theta\right)}{\partial \gamma^T} \right)^H \frac{\partial \ln p\left(Y|\theta\right)}{\partial \gamma^T} \right] \tag{1.8}$$

Then, we have the following Cramer Rao Lower Bound for the covariance matrix of the estimator:

$$\mathrm{Cov}\left(\hat{\gamma}\right) \geq {\mathcal{I}_\gamma}^{-1} \tag{1.9}$$

where the matrix inequality means $z^H \mathrm{Cov}\left(\hat{\gamma}\right) z \geq z^H {\mathcal{I}_\gamma}^{-1} z$ for any $z \in \mathbb{C}^{2K}$.

Since dealing with a single scalar bound is more practical, we are interested in a one-dimensional relation possibly involving the MSE: assuming $\hat{\theta}$ unbiased and using

(1.3), we can write the MSE of the original parameter as:

$$MSE(\hat{\theta}) = \frac{1}{K}\mathrm{Tr}\left\{\mathrm{Cov}(\hat{\theta})\right\} = \frac{1}{2K}\mathrm{Tr}\left\{\mathrm{Cov}(\hat{\gamma})\right\} \tag{1.10}$$

As both $\mathrm{Cov}\,(\hat{\gamma})$ and ${\mathcal{I}_\gamma}^{-1}$ are positive semi-definite, from (1.9) and (1.10) this relation follows:

$$MSE(\hat{\theta}) \geq \frac{1}{2K}\mathrm{Tr}\left\{{\mathcal{I}_\gamma}^{-1}\right\} \tag{1.11}$$

This inequality will be used as a scalar bound for complex parameters throughout this thesis. In the rest of this chapter, our reference channel model is presented and a quantitative formulation of the estimation problem is given.

1.3 General MIMO-OFDM Model

We will start our model description presenting the general case of a $R \times T$ MIMO channel modelled as a discrete-time linear system. Than, a general expression for MIMO-OFDM will be derived. Let $x[k]$ be a vector of length T containing the symbols transmitted by the T transmitting antennas at time k. At the same time, an array of R antennas receive the signal $y[k]$. For the simple case of a flat fading channel, the relation between input and output can be written as:

$$y[k] = h[k]x[k] + w[k] \tag{1.12}$$

where $w[k]$ is a vector of R complex noise samples and h is the $R \times T$ channel matrix that specifies how signals from different antennas mix up on the receiver side. In the most general case, however, ISI affects the system, so that each received symbol depends on L previous other symbols:

$$y[k] = \sum_{l=0}^{L-1} h_l[k]x[k-l] + w[k] \tag{1.13}$$

Note that in this general scenario the channel matrices also depend on time. If we reasonably assume that the channel is constant over the observation time span (typically one symbol time), we can reduce the model to a *Linear Time Invariant* (LTI) system:

$$y[k] = \sum_{l=0}^{L-1} h_l x[k-l] + w[k] \tag{1.14}$$

In other words, we can express the effect of the channel in terms of an L-tap *Finite Impulse Response* (FIR) filter. In this thesis we accept the assumption that the channel does not change significantly during a symbol time, and we can represent it as a FIR filter.

OFDM systems have the attracting feature of splitting a multi-tap channel into several independent flat-fading channels operating at different frequencies. Each of these orthogonal channels is called a subcarrier. In this section an input-output relation for the subcarriers is given. Since transmission is independent across subcarriers, we can choose different strategies for feeding their input. One choice is to use only one bit-stream, map bits into symbols and finally create N subcarriers using serial-to-parallel conversion of blocks of N symbols. In this case, the orthogonal channels have a symbol time N times larger than the original one. Alternatively, subcarriers can be allocated up to N separate symbol streams; this approach is used in *Orthogonal Frequency-Division Multiple Access* (OFDMA), where users are given a subset of the total number of subcarriers. The latter approach - more general - is considered in this work.

Let $X_n[p]$ be the vector of T symbols transmitted on subcarrier n during OFDM symbol p. The p -th OFDM symbol $x^p[k]$ is generated multiplying each subcarrier by a complex exponential and then summing the branches. If the frequencies of these exponential functions are integers, they form an orthogonal set:

$$\sum_{n=0}^{N-1} e^{i2\pi jn/N} e^{-i2\pi kn/N} = \begin{cases} N & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \quad \text{with } k, j = 0, 1, \ldots, N-1 \tag{1.15}$$

Thus we can write the p -th OFDM symbol as a function of time k:

$$x^p[k] = \frac{1}{N} \sum_{n=0}^{N-1} X_n[p]\, e^{i2\pi kn/N} \quad \text{with } k = -CP, \ldots, N-1 \tag{1.16}$$

This is the inverse *Discrete Fourier Transform* (DFT) [4] of the subcarrier inputs $X_n[p]$. In the above expression, CP is the number of samples forming the cyclic prefix. Adding a cyclic prefix, in fact, is a commonly used technique in OFDM: it guarantees subcarrier orthogonality even when delayed replicas of the signal could generate discontinuities between two different OFDM symbols. In practice, the prefix is inserted before each symbol copying the last part of the OFDM signal before its start. To completely suppress the potentially catastrophic effects of ISI, the length of the cyclic prefix CP must be chosen longer or at least equal to the maximum delay of the channel. In this thesis we always assume $CP \geq L - 1$. After transmission, the prefix is simply discarded by the

receiver [5, chap. 2].

The received signal $y^p[k]$, which is associated to the p-th OFDM symbol, is given by the MIMO channel model (1.14). As we shall see, the DFT of the noise samples and the channel impulse response matrix h_l play an important role for the n-th subcarrier:

$$H_n \triangleq \sum_{l=0}^{L-1} h_l \; e^{-i2\pi ln/N} \tag{1.17}$$

$$W_n[p] \triangleq \sum_{k=0}^{N-1} w^p[k] \; e^{-i2\pi kn/N} \tag{1.18}$$

Note that H_n is a $R \times T$ matrix and W_n a vector of length R. At the receiver side, after removing the prefix, the OFDM signal is split in N branches. In each of these branches, the N samples of the signal are multiplied by a complex exponential with a different frequency and then summed to give the output symbol $Y_n[p]$. Since this is equivalent to performing a DFT of the received signal, it is always implemented using *Fast Fourier Transform* (FFT) Algorithms. Finally, under the assumptions of a long enough cyclic prefix, we can derive a very simple expression relating the inputs $X_m[p]$ and the outputs $Y_m[p]$ for the subcarrier m:

$$\begin{aligned}
Y_m[p] &= \sum_{k=0}^{N-1} y^p[k] \; e^{-i2\pi km/N} \\
&= \sum_{k=0}^{N-1} \left(\sum_{l=0}^{L-1} h_l x[k-l] + w[k] \right) \; e^{-i2\pi km/N} \\
&= \sum_{k=0}^{N-1} \left(\sum_{l=0}^{L-1} h_l \left(\frac{1}{N} \sum_{n=0}^{N-1} X_n[p] \; e^{i2\pi n(k-l)/N} \right) e^{-i2\pi km/N} \right) + \sum_{k=0}^{N-1} w^p[k] e^{-i2\pi km/N} \\
&= \frac{1}{N} \sum_{n=0}^{N-1} \left(\left(\sum_{l=0}^{L-1} h_l e^{-i2\pi ln/N} \right) X_n[p] \sum_{k=0}^{N-1} e^{i2\pi k(n-m)/N} \right) + W_m[p] \\
&= \sum_{n=0}^{N-1} (H_n X_n[p] \delta_{nm}) + W_m = H_m X_m[p] + W_m[p]
\end{aligned} \tag{1.19}$$

This last identity simplifies considerably our model: we can think of the OFDM system as a number of separate subcarriers where the input symbols are just multiplied by the channel matrix in the frequency domain. From a practical point of view, the fact that each subcarrier behaves as a flat-fading channel makes equalization much easier.

When the channel coherence time is longer than a symbol time, the estimation process

infers the channel matrix from more than one successive symbols; thus it is convenient to stack K observations of inputs and outputs and noise samples:

$$X_n = \begin{pmatrix} X_n[p] \\ \vdots \\ X_n[p+K] \end{pmatrix} \quad Y_n = \begin{pmatrix} Y_n[p] \\ \vdots \\ Y_n[p+K] \end{pmatrix} \quad W_n = \begin{pmatrix} W_n[p] \\ \vdots \\ W_n[p+K] \end{pmatrix} \tag{1.20}$$

where X_n is a TK length vector, Y_n is a RK length vector, and W_n a RK length vector. So, with this slightly different notation, the result (1.19) becomes:

$$Y_n = (I_K \otimes H_n)\, X_n + W_n \quad \text{with } n = 0, \ldots, N-1 \tag{1.21}$$

1.3.1 Gaussian Noise

Let us now analyse the noise term contained in (1.14) and its relation with (1.17). We make the common assumption that the noise samples $w[k]$ in (1.14) are drawn from a circular complex Gaussian distribution with zero-mean and covariance given by $\mathrm{Cov}(w[k]) = \mathbb{E}[w[k]w[k]^H]$. Notice that in this case the covariance is a $R \times R$ matrix. We also assume that this covariance does not change significantly during the period of interest, i.e. it is constant during the time it takes to collect the symbols used for estimation. In other terms, we are assuming that the noise is a wide-sense stationary complex Gaussian process: important quantities like mean, covariance and auto-correlation function do not change over time.

We have seen that (1.17) is the DFT of the noise samples associated to the same OFDM symbol. It can be shown that the DFT operator is linear and can be expressed in terms of a unitary matrix U; then for the receiving antenna r:

$$(W_0[p]_r\ , \ldots,\ W_{N-1}[p]_r)^T = U\,(w^p[k]_r, \ldots, w^p[k+N-1]_r)^T \quad r = 1, \ldots, R \tag{1.22}$$

where $W_n[p]_r$ and $w^p[k]_r$ are the r-th component of the vectors $W_n[p]$ and $w^p[k]$. Thus, it follows from the properties of the complex Gaussian distribution that these two vectors are identically distributed. That is, for each antenna r, the noise is Gaussian even in the frequency domain and it is independent across subcarriers. However, this does not necessarily mean that the noise is also independent across different antennas: in the most general case we do not make any assumptions on $\mathrm{Cov}(W_n[p]) = \mathbb{E}(W_n[p]W_n[p]^H)$. Since we assumed that the noise samples in the time domain are independent even for more than one OFDM symbol, this property still holds in their frequency domain

representation.

Summing up, we are assuming the noise vector $W_n[p]$ to be circular complex Gaussian with zero mean and covariance matrix given by $\mathrm{Cov}(W_n[p]) = \mathbb{E}(W_n[p]W_n[p]^H)$. Its distribution does not depend on the OFDM symbol p nor on the subcarrier n.

1.4 Assumptions

In this thesis we do not employ the general model for MIMO-OFDM: instead, some restrictive assumptions about the channel and the noise are made. Firstly, we consider a flat fading channel: this means that H_n is constant across all different subcarriers n. Thus, the subcarrier index can be simply dropped, resulting in the channel model:

$$\boxed{Y = (I_K \otimes H)\, X + W} \tag{1.23}$$

From now on, equation (1.23) will be the reference channel model for this thesis. As mentioned before, semi-blind channel and noise estimation relies on K_p known pilot symbols and K_d unknown user data. It is then useful to split the matrix X_n in these two contributions. Assuming that at any time either all the antennas or none of them are transmitting pilots, we can split the observation, symbol and noise vectors in two components:

$$X = \begin{pmatrix} X^{(\mathcal{P})} \\ X^{(\mathcal{D})} \end{pmatrix} \qquad Y = \begin{pmatrix} Y^{(\mathcal{P})} \\ Y^{(\mathcal{D})} \end{pmatrix} \qquad W = \begin{pmatrix} W^{(\mathcal{P})} \\ W^{(\mathcal{D})} \end{pmatrix} \tag{1.24}$$

Using this new notation, we can re-write (1.23) as:

$$Y^{(\mathcal{P})} = \left(I_{K_p} \otimes H\right) X^{(\mathcal{P})} + W^{(\mathcal{P})} \tag{1.25}$$

$$Y^{(\mathcal{D})} = (I_{K_d} \otimes H)\, X^{(\mathcal{D})} + W^{(\mathcal{D})} \tag{1.26}$$

Moreover, we do not assume any specific pattern for data symbols; ${X_n}^{(\mathcal{D})}$ is a collection of independent discrete uniform random variables, having as co-domain one of the constellations available for LTE, i.e. QPSK, 16-QAM or 64-QAM. Furthermore no assumptions are made about the structure of the pilot sequence: it is just an array of known complex symbols. The problem of designing a good symbol sequence is out of the scope of this thesis.

Secondly, the noise is not only considered independent of different OFDM symbols,

but it is also assumed uncorrelated across different antennas, thus giving:

$$\mathrm{Cov}(W[k]) = \mathbb{E}\left[W[k]W[k]^H\right] = \sigma^2 I_R \tag{1.27}$$

1.5 Problem Statement

After the model description, we are ready to formulate the estimation problem that we will try to tackle in this work:

Definition 1.1: (Semi-Blind Estimation in flat-fading MIMO-OFDM systems)
Let (1.23) *be the reference channel model for a flat-fading MIMO-OFDM system, consider the pilot-data notation introduced in* (1.24) *and* (1.25). *Assume the noise to be circular Gaussian, independent of OFDM symbols and having covariance* (1.27). *Given a matrix Y, containing the observations of R antennas for K different OFDM symbols, and a matrix $X^{(\mathcal{P})}$, comprising K_p symbols transmitted by T antennas, the Semi-Blind Channel Estimation problem consists of finding:*

- *an estimator $\hat{H}(Y, X^{(\mathcal{P})})$ for the channel matrix H;*
- *an estimator $\hat{\sigma^2}(Y, X^{(\mathcal{P})})$ for the noise variance σ^2.*

In the next chapters we will present some different approaches to solve this problem efficiently. The quality of the solutions will be evaluated as described in Section 1.2: considering Mean Square Error and computational complexity.

Chapter 2

Pilot-Only Estimation

Let us start our description by reviewing some important results for pilot-only estimation. In this chapter we simply neglect the data part of the observations and consider only pilot symbols, i.e. symbols whose value is known by the receiver. Even though pilot-only estimation is the easiest way of getting channel coefficients and noise variance, it will be extensively used as an initial guess for such iterative methods that also consider data symbols as part of the estimation. A Maximum Likelihood estimator is derived and its statistical properties are presented.

2.1 Maximum Likelihood Estimator

As described in Section 1.2, maximum likelihood estimation consists of choosing the parameter that maximizes the likelihood function associated to the problem. In the case of pilot-only estimation, assuming additive circular Gaussian noise at the receiver, we can write the likelihood function as:

$$\mathcal{L}(H,\sigma^2) = p(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2) \tag{2.1}$$

$$= \mathcal{CN}(Y^{(\mathcal{P})}|(I_{K_p}\otimes H)X^{(\mathcal{P})},\ \ \sigma^2 I_{RK_p}) \tag{2.2}$$

$$= \frac{1}{\pi^{RK_p}|\sigma^2 I_{RK_p}|}\exp\left(-\frac{\|Y^{(\mathcal{P})}-(I_{K_p}\otimes H)X^{(\mathcal{P})}\|^2}{\sigma^2}\right) \tag{2.3}$$

This expression must be interpreted as a function of H and σ^2. So, the ML estimator is:

$$(H_{ML}, \sigma^2_{ML}) = \arg\max_{H,\sigma} \left\{ \frac{1}{\pi^{RK_p} |\sigma^2 I_{RK_p}|} \exp\left(-\frac{\left\| Y^{(\mathcal{P})} - (I_{K_p} \otimes H) X^{(\mathcal{P})} \right\|^2}{\sigma^2} \right) \right\} \quad (2.4)$$

Since the logarithm is a strictly increasing function, an alternative to maximizing the likelihood function is to minimize its negative logarithm:

$$(H_{ML}, \sigma^2_{ML}) = \arg\min_{H,\sigma} \left\{ RK_p \ln \sigma^2 + \frac{\left\| Y^{(\mathcal{P})} - (I_{K_p} \otimes H) X^{(\mathcal{P})} \right\|^2}{\sigma^2} \right\} \quad (2.5)$$

where we also discarded the additive term not depending on H and σ^2. Differentiating this expression with respect to the complex conjugate of the channel matrix, H^*, and setting to zero gives:

$$\frac{\partial \ln \mathcal{L}(H, \sigma^2)}{\partial H^*} = \frac{1}{\sigma^2} \left(\sum_{k=0}^{K_p-1} Y^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H + H \sum_{k=0}^{K_p-1} X^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H \right) = 0 \quad (2.6)$$

Notice that this is a complex matrix derivative. For a review on this kind of operations see Appendix A and [6]. Solving for H produces:

$$\begin{aligned} H_{ML} &= \sum_{k=0}^{K_p-1} Y^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H \left(\sum_{k=0}^{K_p-1} X^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H \right)^{-1} \\ &= \Lambda^{(\mathcal{P})}_{yx} \left(\Lambda^{(\mathcal{P})}_{xx} \right)^{-1} \end{aligned} \quad (2.7)$$

where we introduced these two $R \times R$ matrices:

$$\Lambda^{(\mathcal{P})}_{yx} \triangleq \sum_{k=0}^{K_p-1} Y^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H \quad (2.8)$$

$$\Lambda^{(\mathcal{P})}_{xx} \triangleq \sum_{k=0}^{K_p-1} X^{(\mathcal{P})}[k] X^{(\mathcal{P})}[k]^H \quad (2.9)$$

It is well known that Maximum Likelihood estimation reduces to Least Squares estimation in the simple case of a linear model with Gaussian noise. Thus we can equally refer to this result as a ML or a LS estimator.

For the noise variance, the real-valued partial derivative is computed:

$$\frac{\partial \ln \mathcal{L}(H,\sigma^2)}{\partial \sigma^2} = \frac{RK_p}{\sigma^2} - \frac{1}{\sigma^4}\|Y^{(\mathcal{P})} - (I_{K_p} \otimes H_{ML})X^{(\mathcal{P})}\|^2 = 0 \tag{2.10}$$

Solving for σ^2 gives a simple solution:

$$\sigma^2_{ML} = \frac{\|Y^{(\mathcal{P})} - (I_{K_p} \otimes H_{ML})X^{(\mathcal{P})}\|^2}{RK_p} \tag{2.11}$$

2.1.1 Statistical Properties

In order to evaluate the performance attained by the ML pilot-only estimator, two important statistical quantities are derived: its bias and its variance.

Bias

By definition, the bias of our estimator is:

$$\mathcal{B}(H_{ML}) \triangleq \mathbb{E}\left[H - H_{ML}(Y^{(\mathcal{P})})\right] = H - \mathbb{E}\left[H_{ML}(Y^{(\mathcal{P})})\right] = H - \mathbb{E}\left[\Lambda^{(\mathcal{P})}_{yx}\right]\Lambda^{(\mathcal{P})^{-1}}_{xx} \tag{2.12}$$

As explained in Section 1.2, in this case the expectation is computed integrating over the observations $Y^{(\mathcal{P})}$. Expanding the last expression we get:

$$\begin{aligned}
\mathbb{E}\left[\Lambda^{(\mathcal{P})}_{yx}\right] &= \sum_{k=0}^{K_p-1} \mathbb{E}\left[\left(HX^{(\mathcal{P})}[k] + W[k]\right)X^{(\mathcal{P})}[k]^H\right] \\
&= H\sum_{k=0}^{K_p-1} X^{(\mathcal{P})}[k]X^{(\mathcal{P})}[k]^H + \sum_{k=0}^{K_p-1}\mathbb{E}\left[W[k]X^{(\mathcal{P})}[k]^H\right] \\
&= H\Lambda^{(\mathcal{P})}_{xx}
\end{aligned} \tag{2.13}$$

In the last step we used the assumption of uncorrelated zero-mean noise. Thus we have shown that $\mathbb{E}[H_{ML}] = H$, i.e. the ML estimator is unbiased:

$$\mathcal{B}(H_{ML}) = H - \mathbb{E}[\Lambda^{(\mathcal{P})}_{yx}]\Lambda^{(\mathcal{P})^{-1}}_{xx} = H - H\Lambda^{(\mathcal{P})}_{xx}\Lambda^{(\mathcal{P})^{-1}}_{xx} = 0 \tag{2.14}$$

Similarly, for the noise variance:

$$\begin{aligned}
\mathbb{E}\left[\sigma^2_{ML}\right] &= \frac{1}{RK_p}\mathbb{E}\left[\|Y^{(\mathcal{P})} - (I_{K_p} \otimes H_{ML})X^{(\mathcal{P})}\|^2\right] \\
&= \frac{1}{RK_p}\mathrm{Tr}\left\{\sum_{k=0}^{K_p-1}\mathbb{E}\left[W[k]W[k]^H\right]\right\} \\
&+ \frac{1}{RK_p}\mathrm{Tr}\left\{\mathbb{E}\left[(H-H_{ML})^H(H-H_{ML})\right]\sum_{k=0}^{K_p-1}X^{(\mathcal{P})}[k]X^{(\mathcal{P})}[k]^H\right\} \\
&= \frac{1}{RK_p}\mathrm{Tr}\left\{\sum_{k=0}^{K_p-1}\sigma^2 I_R\right\} + \frac{1}{RK_p}\mathrm{Tr}\left\{\mathbb{E}\left[(H-H_{ML})^H(H-H_{ML})\right]\Lambda^{(\mathcal{P})}_{xx}\right\} \qquad (2.15) \\
&= \sigma^2 + \frac{1}{RK_p}\mathrm{Tr}\left\{\mathbb{E}\left[(H-H_{ML})^H(H-H_{ML})\right]\Lambda^{(\mathcal{P})}_{xx}\right\} \qquad (2.16)
\end{aligned}$$

Thus we have shown that H_{ML} is unbiased regardless of the estimate of the noise, while σ^2_{ML} is biased as far as $H_{LS} \neq H$.

MSE

We continue the derivation with an expression of the estimation MSE. Notice that, since the estimate is unbiased, it is possible to simplify (1.1):

$$\begin{aligned}
MSE(H_{ML}) &= \frac{1}{RT}\mathrm{Tr}\left\{\mathbb{E}\left[(H-H_{ML})(H-H_{ML})^H\right]\right\} \\
&= \frac{1}{RT}\mathrm{Tr}\left\{-HH^H + \mathbb{E}\left[H_{ML}{H_{ML}}^H\right]\right\} \qquad (2.17)
\end{aligned}$$

The expectation can be expanded as follows:

$$\begin{aligned}
\mathbb{E}\left[H_{ML}{H_{ML}}^H\right] &= \mathbb{E}\left[\Lambda^{(\mathcal{P})}_{yx}{\Lambda^{(\mathcal{P})}_{xx}}^{-1}\left(\Lambda^{(\mathcal{P})}_{yx}{\Lambda^{(\mathcal{P})}_{xx}}^{-1}\right)^H\right] \\
&= \sum_{k=0}^{K_p-1}\sum_{p=0}^{K_p-1}\mathbb{E}\left[Y^{(\mathcal{P})}[k]X^{(\mathcal{P})}[k]^H{\Lambda_{xx}}^{-1}{\Lambda_{xx}}^{-1}X^{(\mathcal{P})}[p]Y^{(\mathcal{P})}[p]^H\right] \\
&= HH^H + \sigma^2 R{\Lambda_{xx}}^{-1} \qquad (2.18)
\end{aligned}$$

Inserting the above expression in (2.17), we obtain an expression for the MSE:

$$MSE(H_{ML}) = \frac{\sigma^2}{T}\mathrm{Tr}\left\{{\Lambda_{xx}}^{-1}\right\} \tag{2.19}$$

Thus, given a number of receiving and transmitting antennas, the MSE grows linearly with the variance of the noise. For the noise variance a similar result is derived:

$$\begin{aligned} MSE(\sigma^2_{ML}) &= \mathbb{E}\left[\left|\sigma^2_{ML} - \sigma^2\right|^2\right] \\ &= \mathbb{E}\left[\sigma^4_{ML}\right] - \sigma^4 - \frac{2}{RK_p}\mathrm{Tr}\left\{\mathbb{E}\left[(H - H_{ML})^H(H - H_{ML})\right]\Lambda^{(\mathcal{P})}_{xx}\right\} \end{aligned} \tag{2.20}$$

To simplify the derivation of the noise MSE, even if in reality it is never the case, perfect knowledge about the channel matrix H is assumed (the estimator becomes unbiased). After substituting (2.11) in the expression above, the first term can be expanded recognizing a χ^2 distribution and using its first two moments:

$$\begin{aligned} \mathbb{E}\left[\sigma^2_{ML}\right] &= \frac{1}{R^2K_p^2}\mathbb{E}\left[\left\|Y^{(\mathcal{P})} - (I_{K_p}\otimes H)X^{(\mathcal{P})}\right\|^4\right] = \frac{1}{R^2K_p^2}\mathbb{E}\left[\left(\sum_{k=0}^{K_p-1}\|W[k]\|^2\right)^2\right] \\ &= \frac{\sigma^4}{4R^2K_p^2}\mathbb{E}\left[\left(\underbrace{\sum_{k=0}^{K_p-1}\frac{2}{\sigma^2}\|W[k]\|^2}_{\chi^2(2K_pR)}\right)^2\right] = \frac{\sigma^4}{RK_p} + \sigma^4 \end{aligned} \tag{2.21}$$

Thus, merging these expressions:

$$MSE(\sigma^2_{ML}) = \frac{\sigma^4}{RK_p} \tag{2.22}$$

Notice that, even if the MSE does not explicitly depend on the channel matrix, the derivation assumes perfect knowledge of H. Since we always have an estimate on H, an extra error term needs to be added to (2.22). Thus, we can conclude that there is coupling between the errors on the channel matrix and the noise variance.

2.2 CRLB for Pilot-Only Estimation

In this section we will give an expression for the *Fisher Information Matrix* (FIM) connected to pilot-only estimation. This theoretical result is very helpful to better

understand the structure of the problem and the interaction among different variables. We will refer to the discussion about CRLB given in Section 1.2.1 and especially to the definition of FIM in (1.8).

First of all, we need to arrange the complex parameters H and σ^2 in a vector γ defined as follows:

$$\begin{aligned}\gamma &= \left(\text{row}_1(H), \ldots, \text{row}_R(H), \text{row}_1(H^*), \ldots, \text{row}_R(H^*), \sigma^2\right)^T \in \mathbb{C}^{2RT+1} \\ &= \left(\text{row}(H), \text{row}(H^*), \sigma^2\right)^T \end{aligned} \tag{2.23}$$

where we introduced the following notation: $\text{row}_i(H)$ is the i-th row of H and $\text{row}(H)$ is a vector collection of the rows of H: $\text{row}(H) = (\text{row}_1(H), \ldots, \text{row}_R(H))$. Then, from (1.8) the Fisher information matrix $\mathcal{I}_\gamma \in \mathbb{C}^{(2RT+1)\times(2RT+1)}$ is partitioned as follows:

$$\mathcal{I}_\gamma = \left[\begin{array}{cc|c} \mathcal{I}_{H^*H} & \mathcal{I}_{H^*H^*} & \mathcal{I}_{H^*\sigma^2} \\ \mathcal{I}_{HH} & \mathcal{I}_{HH^*} & \mathcal{I}_{H\sigma^2} \\ \hline \mathcal{I}_{\sigma^2 H} & \mathcal{I}_{\sigma^2 H^*} & \mathcal{I}_{\sigma^2\sigma^2} \end{array}\right] \tag{2.24}$$

Let us start from the terms only involving H and its conjugate. Expanding the sub-matrices we can see that the terms $\mathcal{I}_{HH^*}$ and $\mathcal{I}_{H^*H}$ vanish: this comes from the fact that the following second-order derivatives are null:

$$\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H_{ij}\,\partial H_{pq}} = 0 \qquad\qquad \frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H^*_{ij}\,\partial H^*_{pq}} = 0 \tag{2.25}$$

On the other hand, using the derivatives computed for calculating the ML solution, we write that:

$$\mathbb{E}\left[\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H^*_{ij}\,\partial H}\right] = \frac{1}{\sigma^2}\delta_{ij}\Lambda^{(\mathcal{P})^*}_{xx} \quad \mathbb{E}\left[\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H_{ij}\,\partial H^*}\right] = \frac{1}{\sigma^2}\delta_{ij}\Lambda^{(\mathcal{P})}_{xx} \tag{2.26}$$

where δ_{ij} is the $R \times R$ matrix with zero entries except for a one in position (i,j). Then, having the expressions for the second-order derivatives, we can infer the corresponding information matrices using the fact that $H_{ij} = \text{row}(H)((i-i)R + j)$.

$$\text{row}_{(i-1)R+j}(\mathcal{I}_{HH^*}) = \text{row}\left(\frac{1}{\sigma^2}\delta_{ij}\Lambda^{(\mathcal{P})}_{xx}\right) = \frac{1}{\sigma^2}\left(\underbrace{0,\ldots,0}_{T(i-1)}, \text{row}_j(\Lambda^{(\mathcal{P})}_{xx}), \underbrace{0,\ldots 0}_{T(R-i)}\right) \tag{2.27}$$

With similar arguments for $\mathcal{I}_{H^*H}$, the global expression for these matrices is given:

$$\mathcal{I}_{H^*H} = I_R \otimes \frac{1}{\sigma^2}\Lambda_{xx}^{(\mathcal{P})^*} \qquad \mathcal{I}_{HH^*} = I_R \otimes \frac{1}{\sigma^2}\Lambda_{xx}^{(\mathcal{P})} \tag{2.28}$$

Moreover, there are four second-order cross-derivatives:

$$\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial\sigma^2\,\partial H^*} = \frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H^*\,\partial\sigma^2} = \frac{1}{\sigma^4}\left(\Lambda_{yx}^{(\mathcal{P})} - H\Lambda_{xx}^{(\mathcal{P})}\right) \tag{2.29}$$

$$\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial\sigma^2\,\partial H} = \frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial H\,\partial\sigma^2} = \frac{1}{\sigma^4}\left(\Lambda_{xy}^{(\mathcal{P})^T} - H^*\Lambda_{xx}^{(\mathcal{P})^*}\right) \tag{2.30}$$

Even if these derivatives are non-zero, they vanish when we take the expectation. In fact, in (2.13) it is shown that $\mathbb{E}\left[\Lambda_{yx}^{(\mathcal{P})}\right] = H\Lambda_{xx}^{(\mathcal{P})}$, thus we have:

$$\mathbb{E}\left[\frac{\partial^2 \ln p\left(Y^{(\mathcal{P})}|H,\sigma^2\right)}{\partial\sigma^2\,\partial H^*}\right] = \frac{1}{\sigma^4}\left(\mathbb{E}\left[\Lambda_{yx}^{(\mathcal{P})}\right] - H\Lambda_{xx}^{(\mathcal{P})}\right) = 0 \tag{2.31}$$

Conversely, the second-order derivative with respect to σ^2 does not vanish:

$$\begin{aligned}\mathcal{I}_{\sigma^2\sigma^2} &= -\frac{KR}{\sigma^4} + \frac{2}{\sigma^6}\mathbb{E}\left[\|Y^{(\mathcal{P})}[k] - (I_{K_p}\otimes H)X\|^2\right] = -\frac{KR}{\sigma^4} + \frac{1}{\sigma^4}\mathbb{E}\left[\chi^2(2KR)\right] \\ &= \frac{KR}{\sigma^4}\end{aligned} \tag{2.32}$$

Summing up, the Fisher Information Matrix for pilot-only joint estimation takes this form:

$$\mathcal{I}_\gamma = \left[\begin{array}{cc|c} \mathcal{I}_{HH} & 0 & 0 \\ 0 & \mathcal{I}_{H^*H^*} & 0 \\ \hline 0 & 0 & \mathcal{I}_{\sigma^2\sigma^2} \end{array}\right] \tag{2.33}$$

with $\mathcal{I}_{\sigma^2\sigma^2}$ given by (2.32) and $\mathcal{I}_{HH^*}$ and $\mathcal{I}_{H^*H}$ given by (2.28). From the shape of the FIM, we can infer two important properties of the pilot-only unbiased estimation problem:

- the accuracy of the channel estimation is not dependent on the accuracy of noise estimation. Notice that this was not the case for the ML estimator of the previous section: the expression of H_{ML} was independent of σ^2, but the estimate of the noise σ^2 was influenced by H_{ML}.
- The channel estimate can be decoupled across different antennas; in principle, it

may be possible to write a set of R estimators that guess the rows of H independently.

Given the structure of the information matrix we can split the CRLB into two separate inequalities for channel and noise; moreover, the FIR can be inverted block by block. Using (1.11) and the fact that $\Lambda_{xx}^{(\mathcal{P})}$ and its inverse are Hermitian, it is easy to show that:

$$MSE(H) \geq \frac{\sigma^2}{T} \mathrm{Tr}\left\{\Lambda_{xx}^{(\mathcal{P})^{-1}}\right\} \tag{2.34}$$

$$MSE(\sigma^2) \geq \frac{\sigma^4}{RK_p} \tag{2.35}$$

Thus, comparing these expressions with (2.19) and (2.22), we see that the maximum likelihood estimator achieves the CRLB. So, ML is the optimal (non-Bayesian) unbiased estimator. However, the derivation of the noise MSE assumed perfect knowledge of the channel H, this means that the noise estimator approaches the bound only asymptotically, as the channel MSE vanishes.

Even if the next chapters will deal with semi-blind channel estimators, pilot-only CRLB is a good reference for comparing other types of estimators. In fact, since semi-blind estimators exploit additional information other than pilot observations, we expect them to perform at least as well as the pilot-only case. On the other hand, we can infer what the maximum improvement of this extra knowledge could be: if we assume all the data symbols are known at the receiver, their estimation reduces to ML pilot-only discussed in this chapter. So, no semi-blind estimator can achieve lower MSE than a pilot-only estimator where all the symbols (pilot and data) are known. In Chapter 6 further details are given and a thorough performance comparison is presented.

Chapter 3

Semi-Blind Estimation

In the previous chapters, we estimated the channel matrix H and the noise variance σ^2 given a set of observations and pilot symbols. In this chapter, the problem of semi-blind estimation is formulated. The main difference is that, in addition to the observations relating to pilots, we will also try to make use of received data symbols. Since the transmitted symbols are not known by the receiver, the complexity of the problem increases considerably. In fact, it will be clear that a closed-form solution is no longer possible. However, in Chapter 4 and Chapter 5 we will show that the problem can be solved iteratively.

In particular, we start our discussion by considering the true discrete distribution of the transmitted symbols and derive a formula for the log-likelihood function. Then, in order to reduce the complexity of the maximization, the assumption about the discrete distribution is relaxed, and we introduce the Gaussian approximation for the data symbols. Moreover, we will give an expression for the Cramer-Rao lower bound, which is the best possible MSE that can be achieved by any semi-blind channel estimator.

3.1 Discrete Symbol Distribution

Let us consider the transmission of K symbols from an arbitrary constellation set C consisting of $|C|$ complex symbols. Assume there are K_p pilots and K_d data symbols. The vector of observations Y can be split in a pilot and data component, as in (1.25).

Thus, the total likelihood is written as:

$$\begin{aligned}\mathcal{L}(H,\sigma^2) &= p(Y|X^{(\mathcal{P})},H,\sigma^2) = p(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2)\; p(Y^{(\mathcal{D})}|H,\sigma^2) \\ &= p(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2) \sum_{X^{(\mathcal{D})}} p(Y^{(\mathcal{D})}|X^{(\mathcal{D})},H,\sigma^2)p(X^{(\mathcal{D})}) \end{aligned} \tag{3.1}$$

where, as seen in the general model and in (1.25), there is no influence of pilot symbols on the observation relating to the data. Assuming additive circular complex Gaussian white noise, with covariance matrix $\sigma^2 I_R$ for each received symbol, we can write these two probabilities as:

$$p(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2) = \mathcal{CN}\left(Y^{(\mathcal{P})}\middle|(I_{K_p}\otimes H)X^{(\mathcal{P})},\ \sigma^2 I_{RK_p}\right) \tag{3.2}$$

$$p(Y^{(\mathcal{D})}|X^{(\mathcal{D})},H,\sigma^2) = \mathcal{CN}\left(Y^{(\mathcal{D})}\middle|(I_{K_d}\otimes H)X^{(\mathcal{D})},\ \sigma^2 I_{RK_d}\right) \tag{3.3}$$

Thus, since $X^{(\mathcal{D})}$ is a random vector whose samples are independently drawn from a discrete uniform distribution, we have:

$$\begin{aligned} p(Y^{(\mathcal{D})}|H,\sigma^2) &= \frac{1}{|C|^{TK_d}} \sum_{X^{(\mathcal{D})}} \mathcal{CN}\left(Y^{(\mathcal{D})}\middle|(I_{K_d}\otimes H)X^{(\mathcal{D})},\ \sigma^2 I_{RK_d}\right) \\ &= \frac{1}{|C|^{TK_d}} \sum_{X^{(\mathcal{D})}} \prod_{k}^{Kd} \mathcal{CN}\left(Y^{(\mathcal{D})}[k]\middle|HX^{(\mathcal{D})}[k],\ \sigma^2 I_{RK_d}\right) \end{aligned} \tag{3.4}$$

In general the sum and product over the symbols cannot be interchanged, thus the observation samples are not independently distributed. However, if the constellation set preserves the symmetry with respect to the origin, it is possible to interchange them. In general, the likelihood function is:

$$\begin{aligned}\mathcal{L}(H,\sigma^2) = \frac{1}{\pi^{RK_p}|\sigma^2 I_{RK_p}|} \exp\left\{-\frac{1}{\sigma^2}\|Y^{(\mathcal{P})} - (I_{K_p}\otimes H)X^{(\mathcal{P})}\|^2\right\} \times \frac{1}{|C|^{TK_d}} \\ \times \frac{1}{\pi^{RK_d}|\sigma^2 I_R|^{K_d}} \sum_{X^{(\mathcal{D})}} \prod_{k}^{Kd} \exp\left\{-\frac{1}{\sigma^2}\|Y^{(\mathcal{D})}[k] - HX^{(\mathcal{D})}[k]\|^2\right\} \end{aligned} \tag{3.5}$$

Once again, dealing with the negative log-likelihood is more practical; in addition, terms not depending on H and σ^2 can be discarded:

$$-\ln \mathcal{L}(H,\sigma^2) = RK_p \ln \sigma^2 + \frac{1}{\sigma^2}\|Y^{(\mathcal{P})} - (I_{K_p} \otimes H)X^{(\mathcal{P})}\|^2 + RK_d \ln \sigma^2 - \ln\left(\sum_{X^{(\mathcal{D})}} \prod_k \exp\left\{-\frac{1}{\sigma^2}\|Y[k]^{(\mathcal{D})} - HX^{(\mathcal{D})}[k]\|^2\right\}\right) \quad (3.6)$$

This expression is extremely difficult to handle. In fact, unlike the pilot-only case, taking the logarithm did not simplify the exponential in the second term. A direct minimization of this function is not possible in closed-form. However, we notice that its high complexity is due to the marginalization of the unknown data in (3.2). In fact, if we assumed perfect knowledge of the transmitted symbols, we would not have the sum in (3.6) and optimization would be the same as for the pilot-only case. This observation is the rationale behind the *Expectation Maximization (EM)* algorithm discussed in Chapter 4, which alternates symbol estimation and closed-form optimization of H and σ^2.

3.2 Gaussian Approximation

The likelihood function (3.6) has an intrinsic limitation: its complexity depends dramatically on the number of observations in the data sequence and on the size of the constellation $|C|$. In fact, the sum over $X^{(\mathcal{D})}$ consists of $|C|^{K_d}$ terms. Unfortunately, with the current assumptions there is no work around it; an effective way of reducing the complexity is relaxing the discreteness of the transmitted symbols, assuming they are drawn from a circular complex Gaussian distribution. It will be clear that this supposition greatly simplifies the expression of the likelihood function.

Once we decided to approximate the symbol distribution with a multivariate circular Gaussian, we need a way of choosing a specific member from this family, i.e. a sensible value for mean and covariance. The *Kullback-Leibler (KL) divergence* is a practical way of evaluating how similar two distributions are. Let $p(X^{(\mathcal{D})})$ be the true discrete distribution of symbols and $\tilde{p}$ be a pdf belonging to the Gaussian family; then KL divergence is defined as:

$$KL(p||\tilde{p}) = \sum_{X^{(\mathcal{D})}} p(X^{(\mathcal{D})}) \ln\left(\frac{p(X^{(\mathcal{D})})}{\tilde{p}(X^{(\mathcal{D})})}\right) \quad (3.7)$$

A reasonable strategy is choosing $\tilde{p}(X)$ such that it minimizes the KL divergence. Differ-

entiating the functional and setting the derivative to zero produces this intuitive result:

$$\begin{cases} \mu_{\tilde{p}} = \mathbb{E}\left[X^{(\mathcal{D})}\right] = 0 \\ \Sigma_{\tilde{p}} = \mathbb{E}\left[X^{(\mathcal{D})}X^{(\mathcal{D})H}\right] = I_{K_d} \otimes \mathbb{E}\left[X[k]^{(\mathcal{D})}X[k]^{(\mathcal{D})H}\right] = I_{K_d} \otimes \Sigma_x \end{cases} \tag{3.8}$$

where Σ_x is the covariance matrix of a single observation (as usual they are assumed i.i.d.). Thus, we can re-write the likelihood function (3.1) as:

$$\begin{aligned} \mathcal{L}(H,\sigma^2) &= p\left(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2\right)\int_{\mathbb{C}^{K_d}} p\left(Y^{(\mathcal{D})}|X^{(\mathcal{D})},H,\sigma^2\right)\tilde{p}\left(X^{(\mathcal{D})}\right)dX^{(\mathcal{D})} \\ &= \mathcal{CN}\left(Y^{(\mathcal{P})}|(I_{K_p}\otimes H)X^{(\mathcal{P})},\sigma^2 I_R\right) \\ &\times \int_{\mathbb{C}^{K_d}} \mathcal{CN}\left(Y^{(\mathcal{D})}|(I_{K_d}\otimes H)X^{(\mathcal{D})},\sigma^2 I_R\right)\times\mathcal{CN}\left(X^{(\mathcal{D})}|\,0,I_{K_d}\otimes\Sigma_x\right)dX^{(\mathcal{D})} \end{aligned} \tag{3.9}$$

It can be proved (see for instance [7, p. 93]) that, if x is normally distributed and we have the following conditional Gaussian distribution for y given x:

$$p(x) = \mathcal{CN}(x|\mu,\Lambda) \tag{3.10}$$

$$p(y|x) = \mathcal{CN}(y|Ax,L) \tag{3.11}$$

then, the marginal distribution of y is given by:

$$p(y) = \mathcal{CN}\left(y|A\mu,L+A\Lambda A^H\right) \tag{3.12}$$

Using this result we obtain a final expression for the likelihood:

$$\begin{aligned} &\mathcal{L}(H,\sigma^2) = \mathcal{CN}\left(Y^{(\mathcal{P})}|(I_{K_p}\otimes H)X^{(\mathcal{P})},\sigma^2 I_R\right)\times\mathcal{CN}\left(Y^{(\mathcal{D})}|0,I_{K_d}\otimes(\sigma^2 I_R+H\Sigma_x H^H)\right) \\ &= \frac{1}{\pi^{RK_p}|\sigma^2 I_{RK_p}|}\exp\left\{-\frac{1}{\sigma^2}\|Y^{(\mathcal{P})}-(I_{K_p}\otimes H)X^{(\mathcal{P})}\|^2\right\} \\ &\times\frac{1}{\pi^{RK_d}|\sigma^2 I_{RK_p}+H\Sigma_x H^H|^{K_d}}\exp\left\{-\mathrm{Tr}\left[I_{K_d}\otimes\left(\sigma^2 I_R+H\Sigma_x H^H\right)^{-1}Y^{(\mathcal{D})}Y^{(\mathcal{D})H}\right]\right\} \end{aligned} \tag{3.13}$$

It is practical to define a symbol for the $R \times R$ sample covariance matrix:

$$\Lambda_{yy}^{(\mathcal{D})} \triangleq \sum_{k=0}^{K_d-1} Y[k]^{(\mathcal{D})} Y[k]^{(\mathcal{D})^H} \tag{3.14}$$

With this notation we can derive an easy formula for the negative log-likelihood:

$$\begin{aligned} -\ln \mathcal{L}(H, \sigma^2) = RK_p \ln \sigma^2 + \frac{1}{\sigma^2} \| Y^{(\mathcal{P})} - (I_{K_p} \otimes H) X^{(\mathcal{P})} \|^2 \\ + K_d \ln |\sigma^2 I_R + H \Sigma_x H^H| + \operatorname{Tr}\left[\left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \right] \end{aligned} \tag{3.15}$$

As for the discrete distribution case, it is not possible to find closed-form solutions for H and σ^2 that minimize (3.15). However, this Gaussian approximation has a big advantage compared to the discrete case: $\Lambda_{yy}^{(\mathcal{D})}$ is a sufficient statistic for H and σ^2. Intuitively, it means that the sample covariance matrix $\Lambda_{yy}^{(\mathcal{D})}$ is a "summary" that provides the same information about H and σ^2 as the original observation set $Y^{(\mathcal{D})}$.

Observation 3.1: (Sufficiency of the sample covariance)
Assuming the transmitted symbols can be approximated by a Gaussian distribution, then the sample covariance matrix $\Lambda_{yy}^{(\mathcal{D})}$ of the data is a sufficient statistic for H and σ^2.

Proof. From the likelihood of the observations,

$$p(Y^{(\mathcal{D})} | H \sigma^2) = \underbrace{\frac{1}{\pi^{RK_d}}}_{h(Y^{(\mathcal{D})})} \times \underbrace{\frac{1}{|\sigma^2 I_{RK_p} + H \Sigma_x H^H|^{K_d}} \exp\left\{ -\operatorname{Tr}\left[\left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \right] \right\}}_{g(\Lambda_{yy}^{(\mathcal{D})}, H, \sigma^2)} \tag{3.16}$$

it is straightforward to see that the first factor is independent of either H and σ^2, while the second one only depends on $Y^{(\mathcal{D})}$ through the covariance matrix $\Lambda_{yy}^{(\mathcal{D})}$, which is a statistic for it. Thus, due to the Neyman Factorization Theorem [8, p. 289], $\Lambda_{yy}^{(\mathcal{D})}$ is sufficient statistic for $Y^{(\mathcal{D})}$. □

This is the key property that will be exploited to try to reduce the computational complexity of the estimate. Practically, it means we do not need to store the $R \cdot K_d$ long vector of observations and process it iteratively, but just handle a $R \times R$ sample covariance matrix.

As we have seen, the Gaussian assumption produces some advantages: the likelihood function does not depend on the type of constellation used and its complexity does not scale with the number of data symbols taken into account. However, one can wonder

how closely it mimics the original discrete distribution: the answer depends on the modulation used and the noise level. In general, since the received symbols are distributed as a mixture of Gaussians, but we pretend there is only one zero-mean Gaussian, the approximation performs better if the original components are more overlapping. Intuitively this situation happens for high noise variances or high-order constellation.

3.3 Further work: Gram-Charlier Series Expansion

As we shall see in Chapter 6, Gaussian approximation manages to capture the essential features of the data distribution when additive noise is quite high. Unfortunately, for low-noise scenarios this distribution fails its task and does not produce any advantage over pilot-only ML. In order to address this issue, we want to replace the normal approximation used for data symbols with another continuous distribution that is more representative of the true discrete one; this strategy can improve performances at expenses of some additional complexity.

In this section we propose a method to find different refinements for the symbol discrete distribution; this is not meant to be an exhaustive discussion providing a complete solution of the problem; instead, we just sketch an idea that could be investigated in further studies. Firstly, the total log-likelihood is written as in (3.1):

$$\mathcal{L}(H,\sigma^2) \;=\; p(Y^{(\mathcal{P})}|X^{(\mathcal{P})},H,\sigma^2)\sum_{X^{(\mathcal{D})}} p(Y^{(\mathcal{D})}|X^{(\mathcal{D})},H,\sigma^2)p(X^{(\mathcal{D})}) \tag{3.17}$$

We focus on the $p(X^{(\mathcal{D})})$ term: it is a discrete probability distribution given by the modulation scheme in use. We know that it can be factorized over time and also over transmitting antennas, if we assume that independent streams are sent. Additionally, since LTE constellations (QPSK, 16QAM, 64QAM) have symmetrical distributions, we can also separate real and imaginary parts:

$$p(X^{(\mathcal{D})}) = \prod_k \prod_t p\left(\Re X[k]_t\right)\; p\left(\Im X[k]_t\right) \tag{3.18}$$

where $X[k]_t = \Re X[k]_t + i\Im X[k]_t$ is a complex symbol transmitted at OFDM symbol k and at antenna t. The idea is approximating the distributions $p\left(\Re X[k]_t\right)$ and $p\left(\Im X[k]_t\right)$ by means of a *Gram-Charlier series* expansion. The Gram-Charlier series is a way of expressing a probability distribution in terms of an infinite sum of known distributions (typically Gaussian), for instance, the forth-order Gram-Charlier expansion of a general

pdf $p(x)$ is given by:

$$p(x) \approx \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \left[1 + \frac{\kappa_3}{3!\sigma^3} H_3\left(\frac{x-\mu}{\sigma}\right) + \frac{\kappa_4}{4!\sigma^4} H_4\left(\frac{x-\mu}{\sigma}\right)\right]; \quad (3.19)$$

with μ, σ being the mean and standard deviation of $f(x)$, and κ_i the i-th order cumulant. $H_i(x)$ stands for the i-th-order Hermite polynomial. Notice that from expression (3.19) it follows that the Gaussian approximation used in Section 3.2 was just a specific case of first-order Gram-Charlier expansion. In this sense we can say that this approach generalizes what was done before. Then, we can obtain different levels of fidelity to the original distribution just adding more terms to the series expansion.

Let us consider a QPSK constellation for instance, we have:

$$p\left(\Re X[k]_t\right) = \begin{cases} 1/2 & \text{if } \Re X[k]_t = \sqrt{2}/2 \\ 1/2 & \text{if } \Re X[k]_t = -\sqrt{2}/2 \end{cases} \quad (3.20)$$

of course it has zero mean and $1/2$ variance. Furthermore its characteristic function is:

$$\Phi_{\Re X[k]_t}(f) = \mathbb{E}\left[e^{if\,\Re X[k]_t}\right] = \frac{1}{2}\left(e^{i\sqrt{2}/2f} + e^{-i\sqrt{2}/2f}\right) = \cos(\sqrt{2}/2f) \quad (3.21)$$

Defining $K(f) \triangleq \ln \Phi_{\Re X[k]_t}(f)$, it can be shown that the cumulants of $\Re X[k]_t$ are simply given by the following expression:

$$\kappa_n = \frac{K^{(n)}(0)}{i^n} \quad (3.22)$$

Deriving $K(f)$ and using (3.22) we get the first four cumulants of a QPSK distribution:

$$\kappa_1 = 0 \qquad \kappa_2 = 1/2 \quad (3.23)$$

$$\kappa_3 = 0 \qquad \kappa_4 = -1/2 \quad (3.24)$$

Thus, substituting in (3.19), we obtain the approximate distribution of the real part of $X[k]_t$.

$$p(\Re X[k]_t) \approx \frac{1}{\pi} \exp\left[-x^2\right] \left[-\frac{1}{3}x^4 + x^2 + \frac{3}{4}\right]; \quad (3.25)$$

Using this formula with (3.18) and performing the integral in (3.17) produces the new forth-order likelihood function. Unfortunatelly, due to the lack of time, we have not been

able to compute the integral. However, we think that it should be possible to calculate it analytically. As a further study, it would be interesting to try to optimize the refined expression of the likelihood function using the methods illustrated in Chapter 4 and 5. If that is possible, performances in terms of MSE are expected to improve for high-SNR settings.

Chapter 4

EM-Based Solutions

In the previous chapters we have studied how to model the semi-blind estimation problem. In particular, we focused on a maximum likelihood approach and derived two cost functions we would like to minimize. In the following two chapters we will present different strategies to do it. The aim of this chapter is to give an overview of the Expectation Maximization (EM) algorithm and discuss how it can be used as an iterative estimator. We will start by reviewing the main ideas behind EM, then the general formulation is derived making no assumptions about the distribution of the unknown symbols. Then, the update equations for both the actual discrete distribution and the Gaussian approximation are given. Finally, we will review an alternative version of the algorithm, *Relaxed EM*, which relaxes some of the convergence properties but approaches the solution considerably faster.

4.1 Introduction to EM

Expectation Maximization [9] is a very popular algorithm for solving maximum likelihood problems that are not directly tractable in closed-form. In its full generality, it relies on the concept of hidden variable: a quantity that could greatly simplify the matter but is not directly observable. Let Y be the set of observations and θ the parameter we are interested in estimating. X is a set of hidden variables related to the problem. Our aim is to maximize the likelihood function $p(Y|\theta)$ with respect to θ. If this task is too difficult, we can think of marginalizing the likelihood over the latent variables X:

$$p(Y|\theta) = \sum_{X} p(Y, X|\theta) \tag{4.1}$$

The EM algorithm results in a remarkable simplification only if the joint probability $p(Y, X|\theta)$ is much easier to optimize than the original likelihood. This is the case in semi-blind channel estimation; it is easy to think of the unknown transmitted symbols $X^{(\mathcal{D})}$ as latent variables, and of course Y is the vector of observations. Then, the problem of optimizing the joint likelihood function is simply a pilot-only estimate, which has been successfully solved in Chapter 2. Naming $q(X)$ the distribution of the hidden variables, it can be proved that it is possible to rewrite the log-likelihood as:

$$\ln p(Y|\theta) = \mathcal{Q}(q, \theta) + KL(q||p) \tag{4.2}$$

where we defined:

$$\mathcal{Q}(q, \theta) = \sum_X q(X) \ln \left\{ \frac{p(Y, X|\theta)}{q(X)} \right\} \tag{4.3}$$

$$KL(q||p) = -\sum_X q(X) \ln \left\{ \frac{p(X|Y, \theta)}{q(X)} \right\} \tag{4.4}$$

Notice that $KL(q||p)$ is the Kullback–Leibler divergence between the actual distribution of the hidden variables and its posterior distribution. As we shall see, the actual distribution is unknown and it will be estimated iteratively. $\mathcal{Q}(q, \theta)$ is the so-called bound function, because, as $KL(q||p) \geq 0$, it is a lower bound for the global log-likelihood function $\ln p(Y|\theta)$.

EM is an iterative algorithm; this means that the optimization of the likelihood function is not performed in one step, but through several iterations. Each of these iterations is initialized with a value θ_{old} and terminates after producing a refined estimate θ_{new}. In turn, any iteration consists of two steps: the Expectation step (E-step) and the Maximization step (M-step). During the E-step, θ_{old} is kept fixed and $\mathcal{Q}(q, \theta_{old})$ is maximized with respect to the distribution $q(X)$. Since the likelihood $p(Y|\theta)$ does not depend on $q(X)$ and the KL divergence is non-negative, the maximum of $\mathcal{Q}(q, \theta_{old})$ is attained when $KL(q||p) = 0$. This means that $q(X) = p(X|Y, \theta_{old})$.

Conversely, during the M-step $q(X)$ is kept fixed to its previous value and $\mathcal{Q}(q, \theta)$ is maximized with respect to θ. Thus, after E and M step, the bound function is:

$$\begin{aligned} \mathcal{Q}(q, \theta) &= \sum_X p(X|Y, \theta_{old}) \ln p(Y, X|\theta) - \sum_X p(X|Y, \theta_{old}) \ln p(X|Y, \theta_{old}) \\ &= \mathbb{E}_X \left[\ln p(Y, X|\theta) | Y, \theta_{old} \right] - H(X|Y, \theta_{old}) \end{aligned} \tag{4.5}$$

where the first term is the only one that depends on θ and needs to be maximized, since

the other one is a constant equal to the conditional entropy of the hidden variables.

Thus it is possible to summarize the execution of a complete EM iteration as follows:

$$\theta_{new} = \arg\max_{\theta} \{\mathbb{E}_X [\ln p(Y, X|\theta)|Y, \theta_{old}]\} \tag{4.6}$$

as discussed before, the whole discussion makes sense only if the maximization in (4.6) can be carried out easily. Fortunately, this is our case.

4.2 General EM Solution

In this section we apply the general EM update rule (4.6) to the case of interest. For the sake of generality, no assumptions are made about the symbol distribution. The most obvious choice is using the transmitted symbols $X^{(\mathcal{D})}$ as hidden set X, and the couple (H, σ^2) as parameter θ. First of all, we can write the argument of the expectation (4.6) as:

$$\ln p(Y, X^{(\mathcal{D})}|X^{(\mathcal{P})}, H, \sigma^2) = \ln p(Y|X^{(\mathcal{P})}, X^{(\mathcal{D})}, H, \sigma^2) + \ln p(X^{(\mathcal{D})}) \tag{4.7}$$

Thus, since the second term does not depend on H or σ^2, the general equation becomes:

$$\left(H_{new}, \sigma^2_{new}\right) = \arg\max_{H,\sigma^2} \left\{\mathbb{E}_{X^{(\mathcal{D})}} \left[\ln p(Y|X^{(\mathcal{P})}, X^{(\mathcal{D})}, H, \sigma^2) \middle| Y, X^{(\mathcal{P})}, H_{old}, {\sigma^2}_{old}\right]\right\} \tag{4.8}$$

From the general model we know that $p(Y|X^{(\mathcal{P})}, X^{(\mathcal{D})}, H, \sigma^2)$ is distributed as a circular Gaussian with mean $(I_K \otimes H)X$ and variance $\sigma^2 I_{RK}$. Then, changing the signs and replacing the maximization with a minimization problem, the EM step is:

$$\left(H_{new}, \sigma^2_{new}\right) = \arg\min_{H,\sigma^2} \left\{\mathbb{E}_{X^{(\mathcal{D})}} \left[RK \ln \sigma^2 + \frac{\|Y - (I_K \otimes H)X\|^2}{\sigma^2} \middle| Y, X^{(\mathcal{P})}, H_{old}, \sigma^2_{old}\right]\right\} \tag{4.9}$$

Expanding the terms in the equation above and deriving with respect to H^* yields:

$$\begin{aligned}\frac{\partial}{\partial H^*} = &- \frac{1}{\sigma^2}\left(\Lambda^{(\mathcal{P})}_{yx} + \sum_{k=0}^{K_d-1} Y^{(\mathcal{D})}[k]\, \mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]^H \middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right]\right) \\ &- \frac{1}{\sigma^2} H\left(\Lambda^{(\mathcal{P})}_{xx} + \sum_{k=0}^{K_d-1} \mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k] X^{(\mathcal{D})}[k]^H \middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right]\right)\end{aligned} \tag{4.10}$$

Notice that we have split the expectation in K terms corresponding to K different observations. However, since the expectation is taken with respect to the data symbols, if one of the terms refers to a pilot observation, then that specific expectation reduces to the value of the corresponding pilot. This is the reason why the terms Λ_{yx} and Λ_{xx} appeared in the above expression. It is practical to introduce the following notation:

$$\Omega_{yx} \triangleq \Lambda_{yx}^{(\mathcal{P})} + \sum_{k=0}^{K_d-1} Y^{(\mathcal{D})}[k]\, \mathbb{E}_{X^{(\mathcal{D})}[k]} \left[X^{(\mathcal{D})}[k]^H \middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old} \right] \tag{4.11}$$

$$\Omega_{xx} \triangleq \Lambda_{xx}^{(\mathcal{P})} + \sum_{k=0}^{K-1} \mathbb{E}_{X^{(\mathcal{D})}[k]} \left[X^{(\mathcal{D})}[k] X^{(\mathcal{D})}[k]^H \middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old} \right] \tag{4.12}$$

Thus, we can write the channel matrix update as:

$$H_{new} = \Omega_{yx} \left(\Omega_{xx} \right)^{-1} \tag{4.13}$$

Notice that, as expected, this formula closely resembles (2.7) for the pilot-only case; the difference here is that we base our estimate on the posterior expectation of the symbols we do not know. With similar arguments it is possible to derive the expression for the noise variance σ^2:

$$\sigma^2_{new} = \frac{1}{RK} \left(Y^H Y - 2\, \Re \left(\mathrm{Tr} \left[H^H_{new} \Omega_{yx} \right] \right) + \mathrm{Tr} \left[H^H_{new} H_{new} \Omega_{xx} \right] \right) \tag{4.14}$$

Summing up, for each EM iteration the E step consists of computing the conditional expectations (4.11) and (4.12), then, the M step produces new values for the channel matrix and the noise variance through (4.13) and (4.14).

Notice that EM, as any other iterative algorithm, needs an initial guess for the parameters to be refined; in this case, pilot-only ML estimation given by (2.7) and (2.11) is a sensible choice. Finally, note that our discussion did not specify how to calculate the expectations in the E step, that will be the topic of the next few sections.

4.3 Discrete distribution

In the following two sections we will give some practical expressions for the computation of Ω_{yx} and Ω_{xx}. Of course, this derivation has to take into account how the transmitted symbols are distributed. We will start by assuming the case of a discrete alphabet of symbols, as in Section 3.1; it is clear that in this case the distribution also depends on the modulation in use.

Algorithm 1: EM

Input: an initial guess H_0 and σ_0^2; $\Lambda_{yx}^{(\mathcal{P})}$ and $\Lambda_{xx}^{(\mathcal{P})}$; max number of iterations i_{max}; termination threshold ϵ

$L_0 \leftarrow -\ln \mathcal{L}(H_0, \sigma_0^2)$ and $L_{-1} \leftarrow +\infty$;

$\delta \leftarrow \epsilon$;

$i \leftarrow 0$;

while $\delta \geq \epsilon$ *and* $i < i_{max}$ **do**

`// E-step:`

$\Omega_{yx} \leftarrow \Omega_{yx}\left(H_i, \sigma_i\right)$; `// Use` (4.11) `with` H_i, σ_i

$\Omega_{xx} \leftarrow \Omega_{xx}\left(H_i, \sigma_i\right)$; `// Use` (4.12)`with` H_i, σ_i

`// M-step:`

$H_{i+1} \leftarrow \Omega_{yx}\left(\Omega_{xx}\right)^{-1}$;

$\sigma_{i+1}^2 \leftarrow \frac{1}{RK}\left(Y^H Y - 2\,\Re\left(\mathrm{Tr}\left[H_{i+1}^H \Omega_{yx}\right]\right) + \mathrm{Tr}\left[H_{i+1}^H H_{i+1} \Omega_{xx}\right]\right)$;

$L_{i+1} \leftarrow -\ln \mathcal{L}(H_{i+1}, \sigma_{i+1}^2)$;

$\delta \leftarrow \left(L_i - L_{i+1}\right)/|L_{i+1}|$;

$i \leftarrow i + 1$;

end

First of all, we notice that, since any constellation set in consideration is symmetrical with respect to the real axis, the expectations in (4.11) and (4.12) are transparent to conjugation:

$$\begin{aligned}
&\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]^H \,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right] = \mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k] \,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right]^H \\
&= \left\{ \sum_{X^{(\mathcal{D})}[k]} X^{(\mathcal{D})}[k]\; p\left(X^{(\mathcal{D})}[k] \middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right) \right\}^H
\end{aligned} \tag{4.15}$$

Moreover, using Bayes' theorem and the sum rule of probability, we can write the posterior first-order moment as follows:

$$\begin{aligned}
&\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k] \,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right] \\
&= \frac{\sum_{X^{(\mathcal{D})}[k]} X^{(\mathcal{D})}[k]\;\; p\left(Y^{(\mathcal{D})}[k] \middle| X^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right) p\left(X^{(\mathcal{D})}[k]\right)}{\sum_{X^{(\mathcal{D})}[k]}\; p\left(Y^{(\mathcal{D})}[k] \middle| X^{(\mathcal{D})}[k], H_{old}, \sigma_{old}^2\right) p\left(X^{(\mathcal{D})}[k]\right)}
\end{aligned} \tag{4.16}$$

Since we know from Section 3.1 that $X^{(\mathcal{D})}[k]$ and $Y^{(\mathcal{D})}[k]|X^{(\mathcal{D})}[k]$ are distributed according to:

$$p(X^{(\mathcal{D})}[k]) = \mathcal{CN}\left(X^{(\mathcal{D})}\Big|\, 0,\ \Sigma_x\right) \tag{4.17}$$

$$p\left(Y^{(\mathcal{D})}[k]\Big|X^{(\mathcal{D})}[k], H_{old},\ \sigma^2_{old}\right) = \mathcal{CN}\left(Y^{(\mathcal{D})}[k]\Big|H_{old}X^{(\mathcal{D})}[k],\ \sigma^2_{old}I_R\right) \tag{4.18}$$

we obtain the posterior first order moment:

$$\begin{aligned}&\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]\,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right]\\ &= \frac{\sum_{X^{(\mathcal{D})}[k]} X^{(\mathcal{D})}[k]\ \ \exp\left\{-\frac{1}{\sigma^2_{old}}\left\|Y^{(\mathcal{D})}[k] - H_{old}X^{(\mathcal{D})}[k]\right\|^2\right\}}{\sum_{X^{(\mathcal{D})}[k]}\ \exp\left\{-\frac{1}{\sigma^2_{old}}\left\|Y^{(\mathcal{D})}[k] - H_{old}X^{(\mathcal{D})}[k]\right\|^2\right\}}\end{aligned} \tag{4.19}$$

and posterior second-order moment:

$$\begin{aligned}&\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]X^{(\mathcal{D})}[k]^H\,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right]\\ &= \frac{\sum_{X^{(\mathcal{D})}[k]} X^{(\mathcal{D})}[k]X^{(\mathcal{D})}[k]^H\ \ \exp\left\{-\frac{1}{\sigma^2_{old}}\left\|Y^{(\mathcal{D})}[k] - H_{old}X^{(\mathcal{D})}[k]\right\|^2\right\}}{\sum_{X^{(\mathcal{D})}[k]}\ \exp\left\{-\frac{1}{\sigma^2_{old}}\left\|Y^{(\mathcal{D})}[k] - H_{old}X^{(\mathcal{D})}[k]\right\|^2\right\}}\end{aligned} \tag{4.20}$$

From these expressions it is clear that the discrete version of the EM algorithm is very demanding in terms of computational load: each moment requires a number of operations growing like $\mathcal{O}\left(|C|^T\right)$. Even more importantly, the number of posterior moments to compute is equal to the size of the data set. Thus, it turns out that for MIMO transmisison and high order modulations like 16QAM or 64QAM, employing the discrete data distribution becomes infeasible even for relatively small data sets.

4.4 Gaussian distribution

In this section, the general EM formulation is applied to the simplified case of a Gaussian distribution of symbols. As we have already discussed in Section 3.2, this model is the most appealing for real applications because of its reduced complexity.

We can write each expectation term in (4.11) as follows. In the rest of this section we will prove that the posterior distribution of the data symbols is circular Gaussian, so

we can take the conjugation outside the expectation symbol.

$$\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]^H \,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right] = \mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k] \,\middle|\, Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right]^H$$
$$= \left\{\int X^{(\mathcal{D})}[k]\; p\left(X^{(\mathcal{D})}[k]\middle| Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right) dX^{(\mathcal{D})}[k]\right\}^H \tag{4.21}$$

A known result for multivariate Gaussian distributions is used to compute the expectation without solving analytically the integral: it can be proved that [7, p. 93], given two normal distributions like:

$$p(x) = \mathcal{CN}\left(x|\, 0,\ \Lambda\right) \tag{4.22}$$
$$p(y|x) = \mathcal{CN}\left(y|\, Ax,\ L\right) \tag{4.23}$$

then the conditional distribution of x given y is:

$$p(x|y) = \mathcal{CN}\left(x\,|\,\Gamma A^H L^{-1} y\,,\ \Gamma\right), \text{ with } \Gamma = (\Lambda^{-1} + A^H L^{-1} A)^{-1} \tag{4.24}$$

Remembering that data is distributed as (4.17) and the conditional distribution of the observation is given by (4.18), we can directly apply result (4.24) for normal distributions, obtaining:

$$p\left(X^{(\mathcal{D})}[k]\middle| Y^{(\mathcal{D})}[k], H_{old},\ \sigma^2_{old}\right) = \mathcal{CN}\left(X^{(\mathcal{D})}\middle|\mu^{(\mathcal{D})}_{X|Y},\ \Sigma^{(\mathcal{D})}_{X|Y}\right) \tag{4.25}$$

where the conditional mean and covariance matrix are given by:

$$\mu^{(\mathcal{D})}_{X|Y} := \frac{1}{\sigma^2_{old}}\ \Sigma^{(\mathcal{D})}_{X|Y}\ H^H_{old}\ Y^{(\mathcal{D})}[k] \tag{4.26}$$
$$\Sigma^{(\mathcal{D})}_{X|Y} := \left(\Sigma_x^{-1} + \frac{1}{\sigma^2_{old}} H^H_{old} H_{old}\right)^{-1} \tag{4.27}$$

Hence, it follows immediately that the posterior first-order moment (4.21) is just the mean $\mu^{(\mathcal{D})}_{X|Y}$ of this circular Gaussian distribution:

$$\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k] \,\middle|\, Y, X^{(\mathcal{P})}, H_{old}, \sigma^2_{old}\right]$$
$$= \mu^{(\mathcal{D})}_{X|Y} = \frac{1}{\sigma^2_{old}}\left(\Sigma_x^{-1} + \frac{1}{\sigma^2_{old}} H^H_{old} H_{old}\right)^{-1} H^H_{old}\ Y^{(\mathcal{D})}[k] \tag{4.28}$$

Similarly, for getting Ω_{xx} in (4.12) we have to compute the posterior second-order mo-

ment:

$$\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]X^{(\mathcal{D})}[k]^H\middle|\, Y^{(\mathcal{D})}[k],\, H_{old},\, \sigma^2_{old}\right] = \int X^{(\mathcal{D})}[k]\, X^{(\mathcal{D})}[k]^H\, p\left(X^{(\mathcal{D})}[k]\middle|Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right) dX^{(\mathcal{D})}[k] \tag{4.29}$$

Once again, we can avoid solving this integral by employing (4.25):

$$\mathbb{E}_{X^{(\mathcal{D})}[k]}\left[X^{(\mathcal{D})}[k]X^{(\mathcal{D})}[k]^H\middle|Y^{(\mathcal{D})}[k], H_{old}, \sigma^2_{old}\right] = \Sigma^{(\mathcal{D})}_{X|Y} + \mu^{(\mathcal{D})}_{X|Y}\ \mu^{(\mathcal{D})\ H}_{X|Y} \tag{4.30}$$

In this section we derived the expressions for the first and second-order posterior moments in case of Gaussian approximation. Thus, during the E step, Ω_{yx} and Ω_{xx} can be computed using (4.28) and (4.30).

4.5 Adaptive Overrelaxed EM

The convergence properties of EM algorithm have been extensively studied in literature: in many practical situations this algorithm seems to be excruciatingly slow [10]. This impression is confirmed by the simulations carried out in Chapter 6. The reason for this slowness is that, in many cases, the $Q(q, \theta)$ function is a loose bound for the log-likelihood and it takes many iterations to maximize it.

Salakhutdinov and Roweis [11] proposed a modified version of EM called *Adaptive overrelaxed EM* (AEM) that is meant to solve this problem. The idea is introducing a learning rate parameter $\eta_i \geq 1$ that adaptively extends the EM step; if $\eta_i = 1$ the i-th OEM step reduces to a regular EM iteration, otherwise, if $\eta_i > 1$, a longer step is taken. For instance, for the channel matrix H, the M step becomes:

$$H_{i+1} = H_i + \eta_i \left(H^{EM}_{i+1} - H_i\right) \tag{4.31}$$

where H^{EM}_{i+1} indicates the channel matrix given by a regular M step. However, while using a lerning rate parameter improves the speed of the solution, it unfortunately voids the primary convergence property: iterations are no longer guaranteed to decrease the negative log-likelihood function. AEM solves this issue very simply: before taking the E-step, we verify whether the "enhanced" step would cause a reduction in the log-likelihood, if that is the case, we proceed to the M-step and η_{i+1} is increased; otherwise, η_{i+1} is set equal to 1 and a regular EM step is taken. A detailed description of AEM is

given in Algorithm 2.

Algorithm 2: Adaptive Overrelaxed EM (AEM)

Input: initial guess for channel and noise H_1 and σ_1; $\Lambda_{yx}^{(\mathcal{P})}$ and $\Lambda_{xx}^{(\mathcal{P})}$; termination threshold ϵ; max number of iterations i_{max}; expansion factor $\alpha > 1$

$L_0 \leftarrow -\infty$;

$H_1^{EM} \leftarrow H_1$ and $\sigma_1^{EM} \leftarrow \sigma_1$;

$\eta_1 \leftarrow 1$ and $\delta \leftarrow \epsilon$;

$i \leftarrow 0$;

while $\delta \geq \epsilon$ *and* $i < i_{max}$ **do**

$L_i \leftarrow -\ln \mathcal{L}(H_i, \sigma_i)$;

$\delta \leftarrow (L_{i-1} - L_i)/|L_i|$;

// E-step:

if $\delta < \epsilon$ **then** // The step is too long. Reset it.

$L_i \leftarrow -\ln \mathcal{L}(H_i^{EM}, \sigma_i^{EM})$;

$\delta \leftarrow (L_{i-1} - L_i)/|L_i|$;

$\eta_{i+1} \leftarrow 1$;

$\Omega_{yx} \leftarrow \Omega_{yx}\left(H_i^{EM}, \sigma_i^{EM}\right)$; // Use (4.11) with H_i^{EM}, σ_i^{EM}

$\Omega_{xx} \leftarrow \Omega_{xx}\left(H_i^{EM}, \sigma_i^{EM}\right)$; // Use (4.12) with H_i^{EM}, σ_i^{EM}

else

$\eta_{i+1} \leftarrow \alpha\, \eta_i$; // Increase η

$\Omega_{yx} \leftarrow \Omega_{yx}(H_i, \sigma_i)$; // Use (4.11) with H_i, σ_i

$\Omega_{xx} \leftarrow \Omega_{xx}(H_i, \sigma_i)$; // Use (4.12) with H_i, σ_i

end

// M-step:

$H_{i+1}^{EM} \leftarrow \Omega_{yx}\left(\Omega_{xx}\right)^{-1}$;

$\sigma_{i+1}^{EM} \leftarrow \sqrt{\frac{1}{RK}\left(Y^H Y - 2\,\Re\left(\mathrm{Tr}\left[H_{i+1}^H \Omega_{yx}\right]\right) + \mathrm{Tr}\left[H_{i+1}^H H_{i+1} \Omega_{xx}\right]\right)}$;

$H_{i+1} = H_i + \eta_{i+1}\left(H_{i+1}^{EM} - H_i\right)$;

$\sigma_{i+1} = \sigma_i + \eta_{i+1}\left(\sigma_{i+1}^{EM} - \sigma_i\right)$;

$i \leftarrow i + 1$;

end

Notice that the log-like maximization problem has been formulated for the couple (H, σ) instead of (H, σ^2). In fact, when extending the step size through η, we must make sure that all the parameters stay within their own domain. To avoid iterations with negative variances, we decided to work with standard deviations so that the optimization is unconstrained.

AEM is very appealing for our semi-blind problem; in fact, its cost function has an advantage: when the Gaussian approximation is used, it is possible to compute the log-likelihood just using (3.15), without having to calculate the posterior moments of the E-step. Note that this would not be the case if the general discrete distribution were used. In light of this observation, we slightly modified the algorithm description given by [11]: instead of computing the E-step before the if-clause (see Algorithm 2) and then check for the decrease condition; we moved the E-step inside the if-clause, and then used two different methods for calculating it. This modification does not change the algorithm because the log-likelihood function does not explicitly depend on the posterior moments of the latent variables, i.e. it can be computed without knowledge of the posteriors. On the other hand, we save complexity because, even if the step is too long, we do not need to re-evaluate the E-step. Thus, AEM provides a faster convergence rate without increasing the complexity of the iteration: compared to the standard EM version, the only extra operation consists in re-computing the log-likelihood function in case the step size needs to be reset.

4.6 EM Convergence

A well known result for the EM algorithm is that, at each iteration, the log-likelihood function is non-decreasing:

$$\ln p(Y|\theta_{i+1}) \geq \ln p(Y|\theta_i) \quad \forall i \geq 1 \tag{4.32}$$

A detailed discussion about this result is given in [9]: the proof relies on the fact that the bound function is non-decreasing because it is maximized during the M-step, while the KL divergence decreases or stays constant as a consequence of Jensen's inequality.

In Chapter 5 we will show that, for any optimization algorithm, increasing the cost function (in this case log-likelihood function) is a necessary but not sufficient condition for convergence to a local maximum. Fortunately, EM provides stronger guarantees: Jeff Wu [12] showed that:

Theorem 4.1: (Wu)

If $\mathcal{Q}(q, \theta)$ belongs to the exponential family, then all the limit points of an instance θ_i are stationary points, and $\ln p(Y|\theta_i)$ converges monotonically to $\ln p(Y|\theta^*)$, for some stationary point θ^*.

Since the semi-blind case verifies the hypothesis of this theorem, we can guarantee global convergence to a stationary point (but not to a local maximum).

One may wonder how the presence of a learning rate η (Section 4.5) affects the behaviour of the EM algorithm. I turns out [11] that for AEM convergence to a stationary point is only guaranteed locally.

Chapter 5

Numerical Optimizers

In this chapter we go through a totally different approach for solving the semi-blind estimation problem explained in Chapter 3: instead of using the EM algorithm, several numerical methods are presented. The main advantage coming from these techniques is that they exploit the sufficiency property of Observation 3.2. Unlike EM, the computation of posterior moments is no longer needed and the complexity of the solution does not scale with the length of the data sequence.

In this section we will only consider the Gaussian approximation of the transmitted symbols, and we will try to minimize its negative log-likelihood (3.15), which plays the role of a cost function:

$$\begin{aligned} -\ln \mathcal{L}(H,\sigma^2) = {} & RK_p \ln \sigma^2 + \frac{1}{\sigma^2}\|Y^{(\mathcal{P})} - (I_{K_p} \otimes H)X^{(\mathcal{P})}\|^2 \\ & + K_d \ln|\sigma^2 I_R + H\Sigma_x H^H| + \operatorname{Tr}\left[\left(\sigma^2 I_R + H\Sigma_x H^H\right)^{-1} \Lambda_{yy}^{(\mathcal{D})}\right] \end{aligned} \tag{5.1}$$

where $Y^{(\mathcal{P})}, X^{(\mathcal{P})}, K_p, K_d, \Sigma_x, \Lambda_{yy}^{(\mathcal{D})}$ are known constant quantities. It is convenient to interpret the negative log-likelihood as a real-valued function f of a real vector x:

$$x = \left(\Re \operatorname{row}_1(H), \cdots, \Re \operatorname{row}_R(H), \Im \operatorname{row}_1(H), \cdots, \Im \operatorname{row}_R(H), \sigma\right)^T \in \mathbb{R}^{2RT+1} \tag{5.2}$$

$$f(x) = -\ln \mathcal{L}(H,\sigma^2) \in \mathbb{R} \tag{5.3}$$

where $\operatorname{row}_i(H)$ is the i-th row of H. From our previous discussion, the term $\sigma^2 I_R + H\Sigma_x H^H$ is the covariance matrix of the observations, hence we can assume it is positive definite. Then, the natural logarithm in the third term and the matrix inversion in the last term are always defined. The domain of the cost function is then the convex set $\mathbb{D} = \mathbb{R}^{2RT+1}$.

5.1 Steepest Descent

The simplest numerical optimizer is called steepest descent. At the k-th step, the channel matrix and the noise variance are updated following the direction of their gradient:

$$x_{k+1} = x_k - \alpha_k \nabla f(x) \tag{5.4}$$

where α_k is a scalar determining the length of step to be taken. It is well known that the gradient is the direction along which a function grows the most rapidly, so it makes sense to move to the opposite direction. Notice that moving away from the gradient is not the only possibility: in general a *descent direction* is any direction making an angle θ_k with the opposite gradient that is strictly less than $\pi/2$. In principle, an *infinitesimal* step towards a descent direction causes a reduction of the cost; unfortunately, it would take an infinite number of steps to converge to the optimum. In practice, we are interested in a step producing a significant reduction, but if it is too long we could end up increasing the cost function. For this reason, choosing the correct step size α is crucial and it results in a trade-off between convergence speed and accuracy of the solution. The steepest descent algorithm can be formulated as follows:

Algorithm 3: Steepest Descent

Input: x_0 and a likelihood threshold ϵ

$f_0 \leftarrow f(x_0)$;
$\delta \leftarrow \epsilon$;
$k \leftarrow 0$;
while $\delta \geq \epsilon$ **do**
  $p_k \leftarrow -\nabla f(x_k)$;
  $\alpha_k \leftarrow$ linesearch (c_1, c_0, x_k, p_k);
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
  $f_{k+1} \leftarrow f(x_{k+1})$;
  $\delta \leftarrow (f_k - f_{k+1}) / |f_{k+1}|$;
  $k \leftarrow k + 1$;
end

Since choosing a suitable step size is very important for any numerical optimizer, conditioning whether the optimum is achieved and at which rate, it is worth reviewing the most important results on this topic. A much more thorough analysis can be found in [13] and [14]. We start by presenting an example of poor step length selection and we

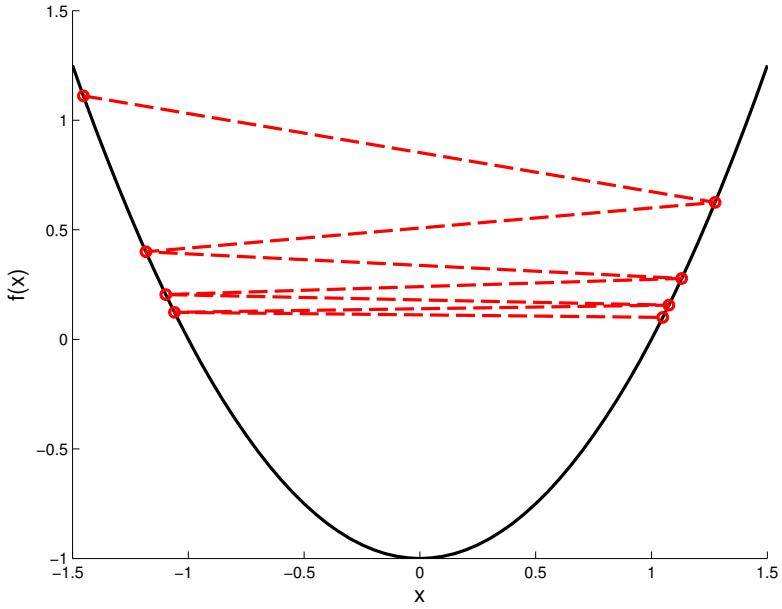

Figure 5.1: Example of sequence of steps decreasing the cost function but not converging to a local minimum.

give some conditions to correctly choose α_k.

The first concern when studying the convergence of an optimizer is making sure that the cost function is reduced at each iteration. This requirement is necessary for convergence to a local minimum, but unfortunately it is not sufficient. In fact, it is easy to demonstrate that, even if a generic algorithm produces a sequence of decreasing steps, it may not reach a local minimum. For example, consider the quadratic function $f(x) = x^2 - 1$ in Figure 5.1, the minimum -1 is reached for $x = 0$; suppose a generic minimization algorithm produces the steps $\{x_k\} = (-1)^k\sqrt{1+k}/k$, then the corresponding sequence of costs is $f(x_k) = 1/k^2$. It is easy to see that, even if each step reduces the cost function, $f_{k+1} < f_k \ \forall k \geq 1$, convergence to the local minimum is not achieved at all.

Let the multivariate real-valued function $f(x)$ be the cost to minimize, x_k is the vector of the variables at k-th iteration and p_k a descent direction, then a commonly accepted strategy for choosing the step size α_k is enforcing the following *Wolfe conditions*:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \tag{5.5}$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k \tag{5.6}$$

with $0 < c_1 < c_2$. The first condition ensures that the cost after the k-th step has significantly reduced; c_1 governs how large this reduction has to be. The second condition

is meant to rule out those steps that are not sufficiently big. The parameter c_2 governs the strictness of the second requirement. Notice that the issue displayed in Figure 5.1 would be solved by enforcing the Wolfe conditions; in fact (5.5) makes sure that the chosen step size produces a significant decrease of the cost function; then, our sequence $\{x_k\}$ would break that rule after some iterations.

One may wonder whether finding step lengths satisfying these requirements is always possible; fortunately it can be proved that, under some non-restrictive conditions:

Lemma 5.1: (Wolfe Conditions Existence)
If $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuous differentiable and bounded below (otherwise the minimization problem is not well-defined) and p_k is a descent direction, then there exist intervals of step lengths satisfying the Wolfe conditions.

The second condition can be modified to ensure that α_k lies closer to its optimal value. The *strong* Wolfe conditions require that α_k satisfies:

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \tag{5.7}$$

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k| \tag{5.8}$$

5.1.1 Steepest Descent Convergence

We give a definition of convergence that takes into account the gradient of the cost function. Since in general it is quite difficult to prove the a sequence tends to a local minimum, this notion of convergence only ensures that a stationary point is approached.

Definition 5.2: (Global Convergence)
A minimization algorithm is said to be globally convergent if, for any starting point $x_0 \in \mathbb{D}$, the norm of the gradient tends to zero:

$$\forall x_0 \in \mathbb{D} \qquad \lim_{k\to\infty} \|\nabla f_k\|^2 = 0 \tag{5.9}$$

Definition 5.3: (Local Convergence)
A minimization algorithm is said to be locally convergent if there exist a neighbourhood δ_{x^*} of the solution x^*, such that, if $x_0 \in \delta_{x^*}$ the norm of the gradient tends to zero:

$$\exists \delta_{x^*} : \forall x_0 \in \delta_{x^*} \qquad \lim_{k\to\infty} \|\nabla f_k\|^2 = 0 \tag{5.10}$$

For the steepest descent method, choosing a direction opposite to the gradient and

fulfilling the Wolfe conditions ensures that a cost reduction is always attained. However, to be sure that a stationary point is finally achieved, we need Zoutendijk's important theorem: it is a quite general result that will come useful in the rest of this thesis. The great advantage of using Wolfe conditions also follows from it.

Let θ_k be the angle between the opposite of the gradient and the search direction p_k. As explained before, in order to p_k to be a discent direction, $-\frac{\pi}{2} < \theta_k < \frac{\pi}{2}$ and thus $0 < \cos\theta_k \leq 1$.

Theorem 5.2: (Zoutendijk's Theorem)
If α_k satisfies the Wolfe conditions and p_k is a descent direction; if f is bounded below in $\mathbb{R}^n$ and it is continuous differentiable in an open set containing the level set $\mathcal{L} = \{x : f(x) < f(x_0)\}$. Moreover assume that the gradient is Lipschitz continuous. Then:

$$\sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty \tag{5.11}$$

We know that a necessary condition for a series to be convergent is that its argument vanishes as the index goes to infinity; then, this theorem has a very important consequence:

$$\lim_{k\to\infty} \cos^2\theta_k \|\nabla f_k\|^2 = 0 \tag{5.12}$$

thus, if we guarantee that at each step the chosen descent direction is bounded away from 90^o, i.e. there is a positive δ such that $0 < \delta \leq \cos\theta_k \leq 1$, then:

$$\lim_{k\to\infty} \|\nabla f_k\|^2 = 0 \tag{5.13}$$

This is the *global convergence* property. Since the gradient descent method follows the direction of the opposite gradient at each step ($\cos\theta_k = 1\ \forall k$), then, under the assumptions of the previous theorem, global convergence is always guaranteed.

5.1.2 Gradient Computation

In this section the gradient of the log-likelihood function is derived: we will make use of complex matrix differentiation theory as presented in [6] and summarized in Appendix A. In order to leave the notation uncluttered and exploit the matrix form of the log-likelihood, we will compute the gradient with respect to the channel matrix in the complex domain, as defined in [6]. Following that notation, the complex gradient is

itself a matrix, and it is given by:

$$\nabla_H \left[-\ln \mathcal{L}(H, \sigma^2) \right] = 2 \, \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial H^*} \in \mathbb{C}^{R \times T} \tag{5.14}$$

Notice that from this expression it is always possible to return to the real domain and handle the real and imaginary parts of the channel coefficients separately; in fact from (A.5) it follows that:

$$\Re \left\{ \nabla_H \left[-\ln \mathcal{L}(H, \sigma^2) \right] \right\} = \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial \Re H} \tag{5.15}$$

$$\Im \left\{ \nabla_H \left[-\ln \mathcal{L}(H, \sigma^2) \right] \right\} = \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial \Im H} \tag{5.16}$$

thus, we can create the real-valued vector gradient taking the real and imaginary parts of the complex matrix gradient and then juxtaposing their rows to form a vector.

The partial complex derivative with respect to H^* is given by:

$$\begin{aligned} \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial H^*} = \frac{1}{\sigma^2} \Bigg[& -\frac{\partial}{\partial H^*} \mathrm{Tr} \left\{ H \Lambda_{yx}^{(\mathcal{P})^H} \right\} - \frac{\partial}{\partial H^*} \mathrm{Tr} \left\{ H^H \Lambda_{yx}^{(\mathcal{P})} \right\} + \frac{\partial}{\partial H^*} \mathrm{Tr} \left\{ H \Lambda_{xx}^{(\mathcal{P})} H^H \right\} \Bigg] \\ & + K_d \, \frac{\partial}{\partial H^*} \ln \left| \sigma^2 I_R + H \Sigma_x H^H \right| + \frac{\partial}{\partial H^*} \mathrm{Tr} \left\{ \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \right\} \end{aligned} \tag{5.17}$$

After some calculations all the terms can be computed using the properties of complex differentials [6] and the definition of formal complex derivative (A.2):

$$\begin{aligned} \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial H^*} = & \frac{1}{\sigma^2} \left(-\Lambda_{yx}^{(\mathcal{P})} + H \Lambda_{xx}^{(\mathcal{P})} \right) + K_d \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} H \Sigma_x \\ & - \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} H \, \Sigma_x \end{aligned} \tag{5.18}$$

Given that the term $\sigma^2 I_R + H \Sigma_x H^H$ is positive definite, both the real and the imaginary parts of this matrix appear to be continuous and differentiable in their domain.

For the standard deviation σ, the regular derivative for real functions of real variables is used, producing:

$$\begin{aligned} \frac{\partial \left(-\ln \mathcal{L}(H, \sigma^2) \right)}{\partial \sigma} = & \frac{2RK_p}{\sigma} - \frac{2}{\sigma^3} \| Y^{(\mathcal{P})} - (I_{K_p} \otimes H) X^{(\mathcal{P})}) \|^2 \\ & + 2\sigma K_d \mathrm{Tr} \left\{ \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-1} \right\} - 2\sigma \mathrm{Tr} \left\{ \left(\sigma^2 I_R + H \Sigma_x H^H \right)^{-2} \Lambda_{yy}^{(\mathcal{P})} \right\} \end{aligned} \tag{5.19}$$

So, the global vector gradient $\nabla f(x) \in \mathbb{R}^{2RT+1}$ is computed taking the real and imaginary parts of the complex matrix gradient $\nabla_H \left[-\ln \mathcal{L}(H, \sigma^2)\right]$, juxtaposing their rows and the noise derivative to form a vector of length $2RT + 1$.

Notice that the cost of computing these derivatives is dominated by the matrix inversion $\left(\sigma^2 I_R + H\Sigma_x H^H\right)^{-1}$, which requires $\mathcal{O}(R^3)$ operations. The number of pilots is also relevant because (5.19) is $\mathcal{O}(K_p)$. On the other hand, the number of data symbols does not influence the complexity of the gradient; in fact $\Lambda_{yy}^{(\mathcal{D})}$ can be computed before the execution of the algorithm, while the samples are being collected.

5.2 Newton-Raphson

Another important optimization strategy is the *Newton-Raphson* method. The main idea is approximating the cost function by means of a second-order Taylor expansion around the current iteration point:

$$f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \triangleq m_k(p) \tag{5.20}$$

where $\nabla^2 f_k$ is the Hessian matrix at x_k. With the (very restrictive) assumption of positive definite Hessian, it is possible to find the vector p that directly minimizes $m_k(p)$. In other words, it is possible to locally approximate the cost function with a convex paraboloid that touches its surface in x_k and has the same curvature; then, minimizing the paraboloid instead of f, the optimal step consists in moving from x_k to the vertex. More formally, setting the derivative of $m_k(p)$ to zero, the direction to follow is given by:

$$p_k = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k \tag{5.21}$$

This direction is also called *Newton direction*. The Newton-Raphson method consists of the steps shown in the description below.

This method has at least two important advantages:

- Convergence rate is quadratic, i.e. faster than gradient descent and then any other method not employing the Hessian.
- The line search algorithm can be greatly simplified to become a *backtracking line search* [13]: in fact, in those regions where the cost function is close to be quadratic, then $\alpha_k = 1$ is the optimal choice. This is due to the fact that taking a unitary step means moving directly to the vertex of the paraboloid, which is the optimal

Algorithm 4: Newton-Raphson

Input: x_0 and a likelihood threshold ϵ

$f_0 \leftarrow f(x_0)$ and $f_{-1} \leftarrow +\infty$;

$\delta \leftarrow \epsilon$;

$k \leftarrow 0$;

while $\delta \geq \epsilon$ **do**

$p_k \leftarrow -\left(\nabla^2 f_k\right)^{-1} \nabla f_k$;

$\alpha_k \leftarrow$ linesearch (c_1, c_0, x_k, p_k);

$x_{k+1} \leftarrow x_k + \alpha_k\, p_k$;

$f_{k+1} \leftarrow f(x_{k+1})$;

$\delta \leftarrow (f_k - f_{k+1}) / |f_{k+1}|$;

$k \leftarrow k + 1$;

end

choice if f is almost quadratic. In case the quadratic approximation is too loose and the full step does not satisfy the Wolfe conditions, then we can compensate trying smaller values of α. This technique is called backtracking like search.

On the other hand, supposing the Hessian to be positive definite is a really restrictive, nevertheless essential, assumption for convergence: failing this, we have no guarantees that p_k is a descending direction. This means that Newton-Raphson method is ensured to be globally convergent only for those functions where the Hessian is always positive definite, i.e. for convex functions. Alternatively, it can be proved that for any twice differentiable function with continuous Hessian and local maximizer x^*, there exists a neighbourhood of x^* where the algorithm converges. Then, according to this result, Newton-Raphson attains local convergence only if started inside this local convergence region.

Even if the cost function is not globally convex, it is possible to come up with methods for dealing with a non-positive definite Hessian: the simplest solution would be taking a steepest descent step in case the Hessian is not positive definite; alternatively, it is possible to adjust its negative eigenvalues to make them positive without wasting the curvature information conveyed by the Hessian matrix. However, these procedures add additional complexity and typically involve a Cholesky or LU decomposition of the Hessian [13, Appendix D]. Anyway, there is no agreement on which modification strategy is the best and certainly no theoretical results ensuring that a modified step would preserve the convergence speed of the original Newton direction.

In our case, we cannot assume the cost function to be convex; moreover, simulation results have shown that initializing the optimization with a pilot-only ML estimate is not sufficient to let the staring point sit within a convergent neighbourhood of the minimum. In addition to these convergence issues, computing the Newton step is a cumbersome operation that requires to build a $\mathbb{R}^{(2RD+1)\times(2RD+1)}$ Hessian matrix and invert it. For these two reasons we decided to focus on numerical methods not involving the computation of the Hessian.

5.3 Conjugate Gradient Methods

Conjugate Gradient (CG) methods are a class of algorithms that were first proposed for solving linear systems with positive definite coefficient matrices. As we shall see, CG copes with these problems in an elegant and inexpensive fashion. Afterwards, these techniques have been extended to deal with large non-linear optimization problems; the advantage of such methods is that they attain a fast convergence rate without requiring the computation of the Hessian matrix. Our discussion will start by presenting the linear conjugate gradient; then the non-linear version is derived.

5.3.1 Linear Methods

Let A be a symmetrical positive definite matrix and b a known vector, the linear conjugate gradient method aims to solve a linear system for x:

$$Ax = b; \tag{5.22}$$

which is equivalent to the following minimization problem:

$$\min \phi(x) \triangleq \frac{1}{2}x^T Ax - b^T x. \tag{5.23}$$

This can be seen by noticing that $\phi(x)$ is a convex quadratic function with gradient

$$\nabla\phi(x) = Ax - b \triangleq r(x), \tag{5.24}$$

and the stationary point x^* such that $Ax^* = b$ is also a global minimizer, on account of the convexity of $\phi(x)$. Hence, (5.22) and (5.23) are equivalent as they share the same unique solution x^* .

The main characteristic of CG algorithms is that search directions p_k must satisfy

the *conjugacy property*:

$$p_i^T A p_j = 0 \qquad \forall i \neq j. \tag{5.25}$$

Apart from this distinctive feature, the sequence of solutions is then updated according to the usual rule

$$x_{k+1} = x_k + \alpha_k p_k, \tag{5.26}$$

where α is a suitable step size; in this case α_k can be computed in closed form minimizing the function $\phi(x)$ along the direction $x_k + \alpha_k p_k$:

$$\alpha_k = \arg\min_{\alpha} \{\phi(x_k + \alpha p_k)\} = -\frac{r_k^T p_k}{p_k^T A p_k}. \tag{5.27}$$

Choosing conjugate search directions results in some interesting features [14]:

- For any starting point $x_0 \in \mathbb{R}^n$, the sequence $\{x_k\}$ converges to the solution x^* in at most n steps.
- The gradient r_k at each step is orthogonal to all previous directions:

$$\forall k \quad r_k^T p_i = 0 \text{ with } i = 1, \ldots, k-1 \tag{5.28}$$

Of course these properties only hold if the conjugacy requirement is met. Thus, the main challenge is finding a set of directions fulfilling (5.25); for instance, a orthogonal basis of eigenvectors $v_1, \ldots, v_n$ of A does the job:

$$v_i^T A v_j = v_i^T v_j = 0 \qquad \forall i \neq j \tag{5.29}$$

Unfortunately finding $v_1, \ldots, v_n$ using the Gram-Schmidt procedure is too expansive for a large number of parameters. Instead, we want any conjugate direction p_k to be computed from the previous one; in particular, p_k is written as a linear combination of the gradient r_k and p_{k-1}:

$$p_k = -r_k + \beta_k p_{k-1}, \tag{5.30}$$

where β_k is chosen to satisfy $p_i^T A p_j = 0$. Multiplying each term by $p_{k-1}^T A$ yields:

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}} \tag{5.31}$$

It can be shown that the gradient vectors resulting from these equations have the property of being mutually orthogonal. Summarizing, the linear conjugate gradient method consists of the steps listed in Algorithm 5.

Algorithm 5: Linear Conjugate Gradient

Input: x_0

$r_0 \leftarrow Ax_0 - b$ and $p_0 \leftarrow -k_0$;

$k \leftarrow 0$;

while $r_k \neq 0$ **do**

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k}; \tag{5.32}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \tag{5.33}$$

$$r_{k+1} \leftarrow A x_{k+1} - b; \tag{5.34}$$

$$\beta_k \leftarrow \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}; \tag{5.35}$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \tag{5.36}$$

$k \leftarrow k + 1$;

end

5.3.2 Non-Linear Methods

It has been shown by Fletcher and Reeves that this simple algorithm for linear systems can be extended to optimize general non-linear functions $f(x)$, only with some small changes. Firstly, the appropriate step size has to be chosen by a line search algorithm enforcing the Wolfe conditions (5.5) and (5.6); secondly r_k is now the gradient of the cost function $f(x)$. With some slight modifications to the linear version, the non-linear Fletcher-Reeves (FR) conjugate gradient can be written as in Algorithm 6.

Comparing to steepest descent, the additional complexity needed for computing β_{k+1} and p_{k+1} consists of two inner products and a sum. Each of these operations is $\mathcal{O}(n)$; in addition, the cost of computing the gradient must be taken into account.

Another version of non-linear conjugate gradient was proposed by Polak and Ribière

Algorithm 6: Fletcher-Reeves Conjugate Gradient

Input: x_0 and c_1, c_0 conveniently chosen

$f_0 \leftarrow f(x_0)$ and $f_{-1} \leftarrow +\infty$;

$\nabla f_0 \leftarrow \nabla f(x_0)$;

$p_0 \leftarrow -\nabla f_0$;

$\delta \leftarrow \epsilon$ and $k \leftarrow 0$;

while $\delta \geq \epsilon$ **do**

$$\alpha_k \leftarrow \text{linesearch}\,(c_1, c_0, x_k, p_k); \tag{5.37}$$

$$x_{k+1} \leftarrow x_k + \alpha_k\, p_k; \tag{5.38}$$

$$\nabla f_{k+1} \leftarrow \nabla f(x_{k+1}); \tag{5.39}$$

$$\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}; \tag{5.40}$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k; \tag{5.41}$$

$\delta \leftarrow (f_k - f_{k+1}) / |f_{k+1}|$

$k \leftarrow k + 1$;

end

(PR); the only difference is the way β is computed:

$$\beta_{k+1}^{PR} \leftarrow \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k} \tag{5.42}$$

Even though PR and FR coincide if the function $f(x)$ is strongly convex, in general they differ significantly. Even their convergence properties are quite different, as we shall discuss.

5.3.3 CG Convergence

Non-linear CG methods have unusual convergence properties [14]: while linear CG is proven to converge at most in r steps, with r being the number of eigenvalues of the coefficient matrix A; non-linear CG convergence depends on a number of factors, and it is strongly influenced by the choice of parameter β_k. In general, Fletcher-Reeves algorithm has better theoretical convergence guarantees than Polak-Ribière; this behaviour, however, is quite unexpected because in practice PR performs better than F-R [14, p.130]. In Chapter 6 we will show that this is the case also for our specific problem. For the Fletcher-Reeves algorithm this result holds:

Theorem 5.3: (Al-Baali)
Suppose that Fletcher-Reeves algorithm is implemented with a line search satisfiying the strong Wolfe conditions (5.7) *and* (5.8) *with* $0 < c_1 < c_2 < \frac{1}{2}$*, moreover the cost function* f *has these properties:*

- *the level set associated to the starting point* $\mathcal{L} \triangleq \{x | f(x) \leq f(x_0)\}$ *is bounded*
- *In some open neighbourhood* $\mathcal{N} \subseteq \mathcal{L}$*, the function* $f(x)$ *is Lipschitz continuously differentiable.*

Then:

$$\liminf_{k \to \infty} \|\nabla f_k\| = 0 \tag{5.43}$$

Notice that the assumptions of this theorem are quite permissive, the first of which is certainly met by the semi-blind cost function. Although we could not prove formally that the second one is met, in light of the simulation results we conjecture that the second assumption should also be true. This convergence result is not as strong as for gradient descent, in fact, only the infimum of the sequence of gradients tends to zero. Unfortunately there is no similar result for the Polak-Ribière algorithm: it seems that its properties have not been fully understood yet [14, p.130].

5.4 Alternative Methods

In this section we propose some unconventional methods for optimization that have been studied in this work. The reason for studying these methods is to try to reduce complexity even further. We will start by describing an estimator based on a second-order constraint for SISO systems; then an approximate MIMO estimator based on LQ decomposition of the channel matrix is presented.

5.4.1 Constrained Optimization

One of the first attempts we made at solving the semi-blind estimation problem consisted of a constrained maximization of the log-likelihood function. As we shall see, this way of solving the problem has a limited applicability to SISO (single input single output) channels, which are a specific case of MIMO systems. In this section, channel, symbols and observations are respectively denoted with lower-case letters h, x, y; however, this notation is equivalent to the general MIMO model when $R = T = 1$. Then the channel $h \in \mathbb{C}$ is just a complex scalar, while $y \in \mathbb{C}^K$ $x \in \mathbb{C}^K$ are vectors, and as usual

$K = K_p + K_d$. We will start by illustrating how this SISO estimator is built, why it is attractive and finally what drawbacks are encountered.

The idea behind this SISO estimator is to make use of a sensible constraint to simplify the maximization of the log-likelihood function. Even though it will still be impossible to formulate a closed-form solution for h and σ^2, we can try to simplify each iteration through a *coordinate descent* approach. In fact, if h can be expressed as a function of σ^2 and vice-versa, each step of the iterative algorithm will alternate the optimization of one of the variables while keeping the other fixed.

A good constraint to apply to the maximum likelihood function can be derived noticing that, according to the Gaussian approximation of Section 3.2, observations are normally distributed with zero mean and covariance matrix:

$$\mathbb{E}\left[y^{(\mathcal{D})}{y^{(\mathcal{D})}}^H\right] = I_{K_d} \otimes \left(\sigma^2 + \rho_x|h|^2\right) \tag{5.44}$$

where $\rho_x \triangleq \mathbb{E}\left[x[k]\,x[k]^*\right]$ is the average power of the transmitted symbols. Then:

$$\mathbb{E}\left[\|y^{(\mathcal{D})}\|^2\right] = \mathbb{E}\left[{y^{(\mathcal{D})}}^H y^{(\mathcal{D})}\right] = \mathrm{Tr}\left\{\mathbb{E}\left[y^{(\mathcal{D})}{y^{(\mathcal{D})}}^H\right]\right\} = K_d\left(\sigma^2 + \rho_x|h|^2\right) \tag{5.45}$$

But, since the elements $y[k]$ of the vector y are i.i.d circular Gaussian, we can approximate this second-order moment with the corresponding sample moment:

$$\mathbb{E}\left[\|y^{(\mathcal{D})}\|^2\right] \approx \sum_{k=0}^{K_d} |y[k]|^2 = \|y^{(\mathcal{D})}\|^2 \tag{5.46}$$

Hence, combining these two expressions, the resulting constraint will be:

$$\sigma^2 + \rho_x|h|^2 \approx \frac{1}{K_d}\|y^{(\mathcal{D})}\|^2 \tag{5.47}$$

Equation (5.47) will be used as an exact maximization constraint for the log-likelihood function, despite its nature is intrinsically inaccurate; in fact, according to the law of large numbers, the equality only holds when $K_d \to \infty$.

We can incorporate (5.47) inside the maximization process using the Lagrange multipliers method; then, introducing a real multiplier λ, the cost function becomes:

$$-\ln\mathcal{L}(h,\sigma^2) = K_p \ln\sigma^2 + \frac{1}{\sigma^2}\|y^{(\mathcal{P})} - hx^{(\mathcal{P})}\|^2 + K_d \ln\left(\sigma^2 + \rho_x|h|^2\right) \tag{5.48}$$

$$+ \frac{\|y^{(\mathcal{D})}\|^2}{\sigma^2 + \rho_x|h|^2} + \lambda\left(\sigma^2 + \rho_x|h|^2 - \frac{1}{K_d}\|y^{(\mathcal{D})}\|^2\right) \tag{5.49}$$

After differentiating with respect to h, σ^2, λ and manipulating these derivatives, we have two closed-form update rules:

$$h = -\sqrt{\frac{\|y^{(\mathcal{D})}\|^2/K_d - \sigma^2}{\rho_x \|x^{(\mathcal{P})^H} y^{(\mathcal{P})}\|^2}}\; x^{(\mathcal{P})^H} y^{(\mathcal{P})} \tag{5.50}$$

$$\sigma^2 = \begin{cases} \frac{1}{K_d}\|y^{(\mathcal{D})}\|^2 - |h|^2 & \text{if } \|y^{(\mathcal{D})}\|^2 \geq K_d|h|^2 \\ \|y^{(\mathcal{P})} - hx^{(\mathcal{P})}\|^2 \left(K_p - \frac{\|y^{(\mathcal{P})} - hx^{(\mathcal{P})}\|^2}{\|y^{(\mathcal{D})}\|^2/K_d - \rho_x|h|^2}\right)^{-1} & \text{otherwise} \end{cases} \tag{5.51}$$

Notice that if we did not set this constraint, it would not be possible to write any closed-form expression for channel and noise: in general a fourth-degree system of two equations has to be solved. On the other hand, the presence of the multiplier reduces the degree from fourth to second. Note also that in (5.50), data observations only influence the magnitude of h, while its phase is completely determined by pilot-related observations.

The quality of the estimate given by this method depends on the precision of approximation (5.47): the closer to the expected value, the more accurate the resulting estimate. We shall prove that the approximation is much closer to the expected value if symbols are drawn from a constant modulus constellation such as M-PSK. In fact, in case symbols have different magnitudes, an extra degree of uncertainty is added. Remembering that $y[k] \sim \mathcal{CN}(y[k] \mid 0,\ \sigma^2 + \rho_x|h|^2)$, we can write

$$\frac{1}{K_d}\|y^{(\mathcal{D})}\|^2 = \frac{\sigma^2 + \rho_x|h|^2}{2K_d} \sum_{k=0}^{K_d} \frac{2\,|y[k]|^2}{\sigma^2 + \rho_x|h|^2} \sim \frac{\sigma^2 + \rho_x|h|^2}{2K_d} \chi^2\,(2K_d) \tag{5.52}$$

Since our constraint is distributed as a χ^2, then its mean is $\sigma^2 + \rho_x|h|^2$; however, we are more interested in the variance:

$$\mathrm{Var}\left(\frac{1}{K_d}\|y^{(\mathcal{D})}\|^2\right) = \left(\frac{\sigma^2 + \rho_x|h|^2}{2K_d}\right)^2 \mathrm{Var}\left(\chi^2\,(2K_d)\right) = \frac{2}{K_d}\left(\sigma^2 + \rho_x|h|^2\right) \tag{5.53}$$

It is easy to see that, for M-PSK constellations, symbol power ρ_x is constant for all the symbols and often set equal to 1. Conversely, for the rest of modulation schemes that power is only unitary on average, while its instantaneous value can change from one symbol to another. For instance, any normalized 16-QAM symbol has power 1/5 with probability 1/4; power 9/5 with probability 1/4 and power 1 with probability 1/2. Hence, taking the expected value of (5.53) with respect to this distribution yields the actual variance of the constraint. In Figure 5.2, the variance of $\|y^{(\mathcal{D})}\|^2/K_d$ is

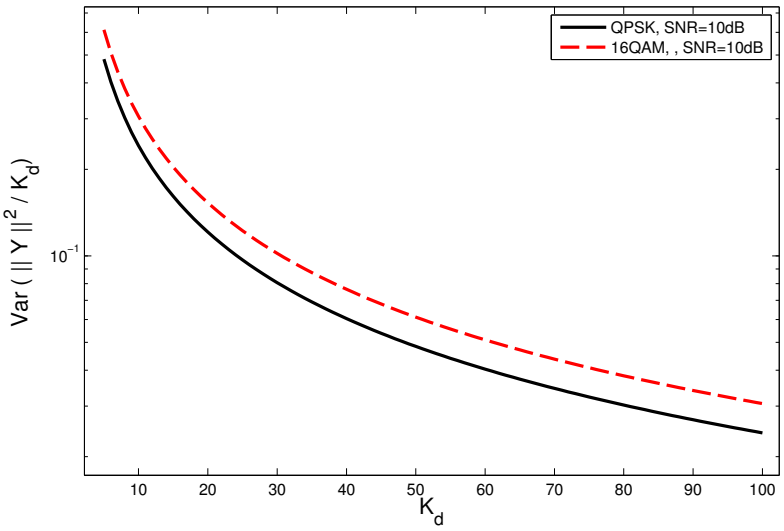

Figure 5.2: $\mathrm{Var}(\|y^{(\mathcal{D})}\|^2/K_d)$ against number of data symbols K_d for QPSK and 16-QAM modulation.

plotted against the number of data symbols for two different modulation schemes. It is evident that non-constant distribution of powers worsens the accuracy of the estimate. From this observation it follows that this constrained optimization technique is especially suited for constant modulus modulations such as M-PSK; as we shall see in Chapter 6, performances are quite disappointing otherwise.

5.4.2 LQ Approximation

In this section a suboptimal approximation for the Gaussian distribution is proposed. In particular, the aim of this approximation is reducing the space of parameters to be optimized. Consider the channel matrix $H \in \mathbb{C}^{R\times T}$, it is possible to break it in two factors using the LQ decomposition:

$$H = LQ \qquad \text{with } Q \in \mathbb{C}^{T\times T}, \quad L \in \mathbb{C}^{R\times T}, \tag{5.54}$$

where Q is a unitary square matrix ($QQ^H = Q^HQ = I_R$) and L is lower triangular matrix ($L_{ij} = 0$ if $i > j$). So far we considered a general symbol covariance matrix $\Sigma_x = \mathbb{E}\left[X[k]X[k]^H\right]$; in this section we assume that symbols have equal power for all antennas and they are uncorrelated over different antennas, $\Sigma_x = \rho_x I_T$. Inserting the LQ decomposition inside the negative log-likelihood function and using the fact that Q

is unitary, we obtain:

$$-\ln \mathcal{L}(Q, L, \sigma^2) = RK_p \ln \sigma^2 + \frac{1}{\sigma^2} \|Y^{(\mathcal{P})} - (I_{K_p} \otimes LQ) X^{(\mathcal{P})}\|^2 + K_d \ln |\sigma^2 I_R + \rho_x L L^H| + \mathrm{Tr}\left[\left(\sigma^2 I_R + \rho_x L L^H\right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \right] \tag{5.55}$$

This equation shows that the data part of the likelihood does not depend on the unitary matrix Q; hence, similarly to the pilot-only case, it iseasy to derive a closed form solution for Q.

$$\frac{-\ln \mathcal{L}(Q, L, \sigma^2)}{\partial Q^*} = \frac{1}{\sigma^2} \left(-L^H \Lambda_{yx}^{(\mathcal{P})} + L^H L Q \Lambda_{xx}^{(\mathcal{P})} \right) \tag{5.56}$$

The value of Q is given by:

$$Q = \left(L^H L\right)^{-1} L^H \Lambda_{yx}^{(\mathcal{P})} \Lambda_{xx}^{(\mathcal{P})^{-1}} = L^\dagger \Lambda_{yx}^{(\mathcal{P})} \Lambda_{xx}^{(\mathcal{P})^{-1}} \tag{5.57}$$

where $L^\dagger$ indicates the Moore-Penrose matrix inverse: $\left(L^H L\right)^{-1} L^H \triangleq L^\dagger$. Conversely, it is not possible to extract a closed-form solution for L, so the best we can do is writing its partial derivative as:

$$\begin{aligned} \frac{-\ln \mathcal{L}(Q, L, \sigma^2)}{\partial L^*} = {} & \frac{1}{\sigma^2} \left(-\Lambda_{yx}^{(\mathcal{P})} Q^H + L Q \Lambda_{xx}^{(\mathcal{P})} Q^H \right) + \rho_x K_d \left(\sigma^2 I_R + \rho_x L L^H\right)^{-1} L \\ & - \rho_x \left(\sigma^2 I_R + \rho_x L L^H\right)^{-1} \Lambda_{yy}^{(\mathcal{D})} \left(\sigma^2 I_R + \rho_x L L^H\right)^{-1} L \end{aligned} \tag{5.58}$$

Since the expression for the parameter Q is linked to L through (5.57), the formally correct way for solving this system is using a *coordinate descent* approach: at each iteration, only one parameter is optimized at the time, while keeping the other fixed

[13]. A hypothetical algorithm would require the following steps:

Algorithm 7: LQ Optimization

Input: an initial pilot-only guess for H_{LS}

LQ-Decompose $H_{LS} = L_{LS}Q_{LS}$;

$L_0 \leftarrow L_{LS}$ and $Q_0 \leftarrow Q_{LS}$;

$k \leftarrow 0$;

while *convergence not achieved* **do**

$$\left(L_{k+1}, \sigma^2\right) \leftarrow \arg\min_{L,\sigma^2} \left\{-\ln \mathcal{L}(L, Q_k, \sigma^2)\right\}; \tag{5.59}$$

$$Q_{k+1} \leftarrow L_{k+1}^{\dagger}\ \Lambda_{yx}^{(\mathcal{P})} {\Lambda_{xx}^{(\mathcal{P})}}^{-1}; \tag{5.60}$$

$k \leftarrow k + 1$;

end

This way of solving the problem does not introduce any losses in performance and it is expected to reach the same optimal value as the other iterative optimizers. Of course, optimizing L several times is quite cumbersome even if the space of parameters has reduced.

Simulations have shown that the value of Q changes only slightly during the iterations; starting from this observation a suboptimal optimizer is proposed. This suboptimal solution consists in neglecting the update of Q and applying one of the iterative optimizers studied in this chapter on L only.

Algorithm 8: Suboptimal LQ optimization

Input: an initial pilot-only guess for H_{LS}

LQ-Decompose $H_{LS} = L_{LS}Q_{LS}$;

$\left(L, \sigma^2\right) \leftarrow \arg\min_{L,\sigma^2} \left\{-\ln \mathcal{L}(L, Q_{LS}, \sigma^2)\right\}$;

The advantage resulting from LQ decomposition is that in some cases the dimensionality of the optimization decreases considerably. Let us quantify this reduction: if $R \neq T$, then the matrix H can take two different structures, as shown in Table 5.1; in particular, L_{sq} is a $T \times T$ square lower triangular matrix and, as its number of non-zero entries in each row increases by one unit from 1 to T, it is equal to $\sum_{n=1}^{T} n = T(T-1)/2$; moreover L_{full} is a $(T-R) \times R$ matrix whose number of non-zero elements is $(T-R)R$.

	H decomposition	Non-zero elements of L
$R = T$	$[L_{sq}]\,[Q]$	$T(T-1)/2$
$R > T$	$\left[\begin{array}{c} L_{sq} \\ \hline L_{full} \end{array}\right] [\ Q\ ]$	$T(T-1)/2 + (T-R)R$
$R < T$	$\left[\begin{array}{c\|c} L_{sq} & 0 \end{array}\right] [\ Q\ ]$	$T(T-1)/2$

Table 5.1: H decomposition for different values of R and T

From this table it turns out that the maximum reduction in terms of number of parameters is achieved when $R < T$. On the other hand, it is difficult to quantify which is the experienced loss due to suboptimality in terms of MSE; in the next chapter simulations are performed to answer this question.

Chapter 6

Simulations

The first objective of this work was investigating low-complexity solutions to the problem of semi-blind estimation. Unfortunately, evaluating the complexity of an iterative algorithm is not always easy. Defining complexity as the average number of computations it takes to execute a complete optimization process, it is clear that it depends on two factors: the average number of calculations per iteration and the average number of iterations. In general, we are interested in estimating how these two factors scale with respect to MIMO dimensions R and T and the size of the data sequence K_d. While the $\mathcal{O}$-notation is a popular way of expressing the asymptotic behaviour of each iteration, the rate of convergence presented in Chapter 5 gives a rough idea of how fast the gradient of the cost function tends to zero. In practice, simulations are always performed to test the theoretical results and provide an effective comparison among different solutions.

Not only the computational complexity requires simulations, but also the performance of an iterative estimator can be hardly predicted theoretically. The best we can do is writing a CRLB for semi-blind estimation, but its formulation would just provide a lower bound for the MSE of any estimator based on the same likelihood function. Then, CRLB does not help to relate different methods.

This chapter will start by comparing semi-blind and pilot-only algorithms in terms of achieved estimation accuracy, i.e. mean square error. The goal is also assessing to what extent assumptions and approximations influence the MSE. As we shall see, the number of transmitting and receiving antennas plays an important role and in Section 6.1 several different settings will be tested. Afterwards, in Section 6.2 the convergence rate is discussed analysing the average number of iterations needed to reach a certain threshold. Finally, in Section 6.3 another comparison consists in measuring the global execution time of a large number of optimization problems. This type of experiments

gives an overall figure of merit about the complexity of different solutions.

There are several different setups and pictures worth to be shown, however too many images might be confusing to the reader; for the sake of clarity, only the most important figures are displayed in this chapter, while the rest of them can be found in Appendix B.

6.1 Estimation Accuracy

In this section our primary interest is evaluating the empirical MSE for different channel estimators: the expected value in Equation (1.1) is approximated by taking the average over a substantial number of independent estimates. The MSE is plotted as a function of the SNR to observe how performances are influenced by the noise level and especially to find out which is the SNR-range where the Gaussian approximation is most beneficial. In order to obtain reliable results, estimation is carried out 2000 times for each SNR level. In each simulation the MSE for pilot-only estimation is also given, in fact it is a useful reference to assess what relative advantage we are gaining from taking into account the data observations.

The simulation environment reflects the assumption we made in Section 1.4 and channel and noise are jointly estimated. Observations are created using the i.i.d. Rayleigh fading model, then, each element in H is drawn from an independent complex Gaussian distribution with:

$$\mathbb{E}\left[H_{ij}\right] = 0 \qquad\qquad \mathbb{E}\left[|H_{ij}|^2\right] = 1 \tag{6.1}$$

for $i = 1, \ldots, R$ and $j = 1, \ldots, T$. The transmitted symbols are coded independently across transmitting antennas and the overall power is normalized to one:

$$\Sigma_x \triangleq \mathbb{E}\left[X[k]X[k]^H\right] = \frac{1}{T} I_T \tag{6.2}$$

From the normalization made on the channel and the choice of the symbol covariance matrix, it follows that the SNR at each receiving antenna is simply:

$$\begin{aligned} \mathrm{SNR} &\triangleq \frac{\mathbb{E}\left[X[k]^H H^H H X[k]\right]}{\mathbb{E}\left[W[k]^H W[k]\right]} = \frac{\mathrm{Tr}\left\{\mathbb{E}\left[H^H H X[k]X[k]^H\right]\right\}}{\mathrm{Tr}\left\{\mathbb{E}\left[W[k]W[k]^H\right]\right\}} = \frac{\frac{1}{T}\mathbb{E}\left[\mathrm{Tr}\left\{HH^H\right\}\right]}{R\,\sigma^2} \\ &= \frac{RT}{RT\sigma^2} = \frac{1}{\sigma^2} \end{aligned} \tag{6.3}$$

When evaluating an estimator, it is interesting to understand how its accuracy impacts

on the global communication reliability; unfortunately, there is no simple rule relating estimation MSE and bit error probability. In some cases, full knowledge about the channel H does not produce a consistent improvement in terms of BER; for instance, if the SNR is very poor, errors are mostly due to the AWGN term W more than the fading nature of the channel H. Hence, under these circumstances even an estimator with very low MSE would not be able to improve the overall bit error probability. We can conclude that an estimator boosting performance in that region is less valuable than another one enhancing the estimation where it really influences the error rate. From this standpoint, the capacity C [bit/s/Hz] associated to the channel H is a good gauge because it gives an upper bound on the highest reliable transmission rate. For MIMO systems with uniform transmitting power allocation, C it is given by [2]:

$$C \triangleq \mathbb{E}\left[\log\det\left(I_R + \frac{1}{\sigma^2} H \Sigma_x H^H\right)\right] = \mathbb{E}\left[\log\det\left(I_R + \frac{\mathrm{SNR}}{T} H H^H\right)\right] \tag{6.4}$$

As before, the expectation over channel realizations is approximated by averaging over a large number of repetitions. In order to insert channel uncertainty inside (6.4), we can think of the estimation error as a noise source other than AWGN:

$$Y[k] = HX[k] + \underbrace{W[k] + (\hat{H}(y) - H)X[k]}_{\triangleq N[k]} \tag{6.5}$$

where $N[k]$ is the total noise due to AWGN and channel uncertainty. Notice that, when we assume that $N[k]$ is an independent noise source, we are making an approximation: $N[k]$ cannot be independent of the signal as it contains the transmitted symbols in its expression. Then, we define a different approximated $\widetilde{SNR}$ that takes into account the estimation error:

$$\begin{aligned}\widetilde{SNR} &\triangleq \frac{\mathrm{Tr}\left\{\mathbb{E}\left[H^H H X[k] X[k]^H\right]\right\}}{\mathrm{Tr}\left\{\mathbb{E}\left[N[k]N[k]^H\right]\right\}} = \frac{R}{\mathrm{Tr}\left\{\sigma^2 I_R\right\} + \frac{1}{T}\mathrm{Tr}\left\{\mathbb{E}\left[\left(\hat{H} - H\right)\left(\hat{H} - H\right)^H\right]\right\}} \\ &= \frac{1}{\sigma^2 + \cdot MSE(\hat{H})}\end{aligned} \tag{6.6}$$

Now we are able to judge the impact of our estimators on the global communication system: capacity (6.4) is approximated on a large number of channel realizations making use of the modified SNR expression given by (6.6).

6.1.1 1R x 1T SISO

Let us start by the simplest case: $1R \times 1T$ (SISO) channel. In Figure 6.1 we plot the MSE for the channel matrix H and in Figure B.1 for the noise variance σ^2. Channel and noise are jointly estimated and the chosen modulation is QPSK. Several curves are displayed in the plots, for simplicity here we give an extended legend:

Pilot-only ML ML estimation as in Chapter 2 based on pilot-only. Solid blue line.

Pilot+Known data ML estimation as in Chapter 2 based on pilot and data (assumed known). Dashed blue line.

Discrete EM EM algorithm with discrete distribution of the transmitted symbols. Dashed green line.

Gaussian EM EM algorithm with Gaussian approximation of symbols. Continuous green line.

AEM Adaptive Overrelaxed EM algorithm with Gaussian approximation of symbols. Dashed dark green line.

Steepest Descend Steepest descend algorithm. Solid black line.

Conjugate Gradient Polak-Ribiere version of the conjugate gradient method, solid purple line. The Fletcher-Reeves version where present is a dashed grey line.

LQ Conjugate Gradient Conjugate gradient with LQ approximation as in Section 5.4.2. Solid red line.

Constrained Coordinate Descent Constrained coordinate descent as in Section 5.4.1. Dashed grey line.

Observing Figure 6.1 and taking the two blue lines as references, the first remark is that for low SNRs the discrete version of EM (dashed green line) gives only a little improvement to the pilot-only estimation accuracy. In fact, it is clear that discrete EM, Gaussian EM, AEM, steepest descent and CG yield the same average error in the low SNR region. This behaviour was expected: in that region the noise component dominates the signal component, then the observations are only weakly informative about the transmitted symbols. On the contrary, for high SNRs discrete EM attains the same performance as if the transmitted symbols were known by the receiver. As explained in Chapter 2, least square estimator (dashed blue line) achieves the unbiased

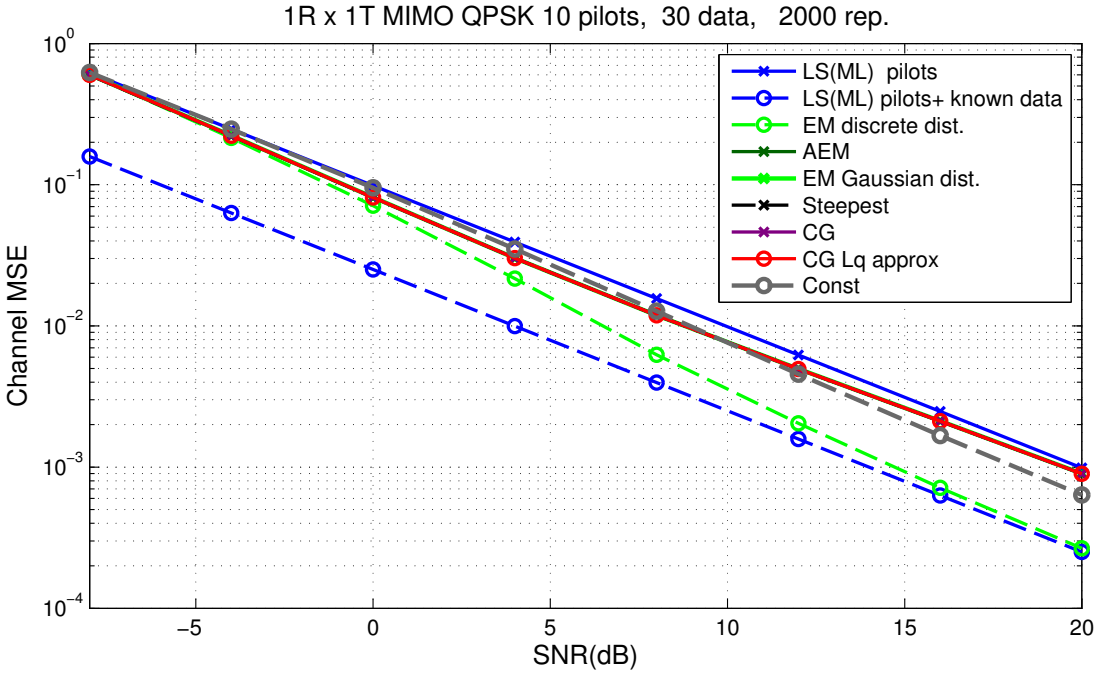

Figure 6.1: MSE versus channel SNR for SISO QPSK channel. 10 pilots and 30 data symbols.

CRLB; thus, we can say that in a high-SNR regime discrete EM is the optimal unbiased estimator.

We can also notice that the MSE curves relative to Gaussian EM, AEM, Steepest descent and CG overlap for the whole SNR range; these methods in fact optimize the same Gaussian semi-blind log-likelihood and seem to converge to the same optimum. This is a positive result since these algorithms can only be proved to converge to local minima, but in this case they all reach the same point. Moreover, we can see that the Gaussian approximation produces a real benefit in terms of MSE only in a middle-range SNR region: in fact for extreme SNR values these curves return to the pilot-only level. We already discussed the low SNR regime, while at high SNRs this behaviour is explained considering the expression (3.15) of the log-likelihood: it can be seen that if σ^2 shrinks (high SNR), then the contribution of the data part vanishes. Intuitively, this means that for high SNRs the approximation in use is too loose and does not capture the complexity of the problem: as noise decreases, approximating a mixture of Gaussians with a single bell is no longer permitted.

Considering the purple line, we observe that the CG optimizer with LQ approximation does not introduce any loss in performance, which is due to the fact that in the SISO 1×1 case there is no parameter reduction (see Table 5.1) with respect to the full

problem; however, as the diagonal of L is real, the optimization is performed in the real domain.

Let us describe the behaviour of the constrained optimization method developed in 5.4.1 (dashed grey line): for low and middle range SNRs its MSE is higher than the non-constrained methods; conversely for high SNRs it considerably outruns the accuracy of the other methods based on the same Gaussian assumptions. In some sense it seems that the presence of a maximization constraint drags the algorithm away from the natural optimum and produces a sub-optimal solution that unexpectedly boasts a better MSE. Although this may look like a contradiction, we have to consider the fact that the non-constrained likelihood function comes from an approximation we made on the symbol distribution; then, enforcing the constraint may produce solutions that are sub-optimal for the approximated Gaussian log-likelihood, but mimic in a better way the actual discrete likelihood.

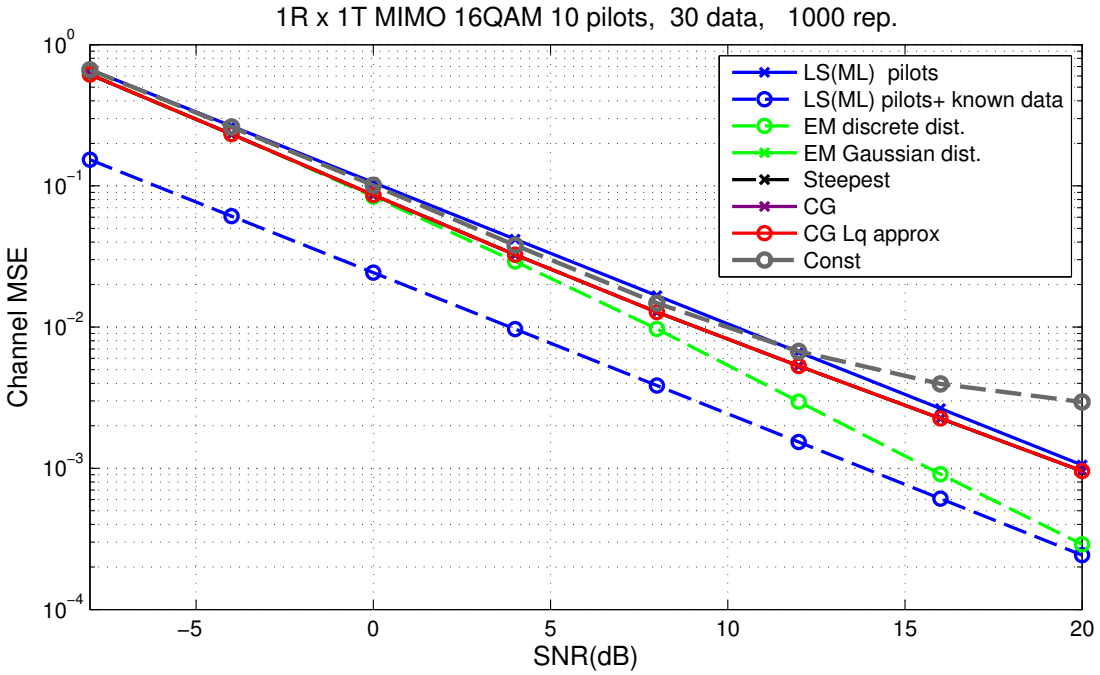

Figure 6.2: MSE versus channel SNR for SISO 16QAM channel. 10 pilots and 30 data symbols.

In Figure 6.2 channel MSE is given for a 16-QAM constellation. Notice that from (2.19) the pilot-only MSE is unchanged, since the power of the constellation has been normalized to one. Comparing this picture with Figure 6.1 for QPSK, it is easy to see a worsening in MSE for the grey line associated to constrained optimization. This behaviour was explained in Section 5.4.1: when the constellation does not have constant

modulus, approximating the symbol variance with its sample variance produces poorer results. Furthermore, as already seen in Figure 5.2, the approximation becomes relatively coarser as the SNR increases. Hence, for high SNRs the maximization constraint is too rough and does not give any advantage. Apart from this, the other curves based on Gaussian approximation are consistent to what we pointed out for QPSK. Conversely, it is easy to see that discrete EM (dashed green line) approaches the optimum pilot-data LSE much more slowly than for the QPSK case. We can interpret this observation noting that the number of possible sequences of transmitted data grows exponentially with the constellation order; this results in a larger degree of uncertainty when computing the posterior moments in the E step (see Section 4.3).

Analysing how accurately noise variance is estimated (Figure B.1 on page 85), we note that its behaviour is in line with the results obtained for channel MSE. The only exception is constrained optimization (dashed gray line) that performs poorly for the central range of SNRs.

6.1.2 2R x 1T MIMO

Let us consider the case of $2R \times 1T$ MIMO transmission. Figure 6.3 displays the channel MSE. First of all, notice that from (2.19) it follows that the pilot-only accuracy does not depend on the number of receiving antennas R; then the blue lines used as references are unchanged. While pilot-only estimation is unaffected by the presence of an extra receiving antenna, both discrete and Gaussian semi-blind estimation improve considerably. For instance, we can see that discrete EM reaches the pilot-data level at 8dB instead of 12dB. Similarly, in the central region Gaussian EM achieves a 3dB improvement compared to the pilot-only case, while for SISO it only attains 1.5dB. This improvement is due to the fact that, for a given number of data symbols K_d, having two receiving antennas provides twice as many observations as the SISO channel.

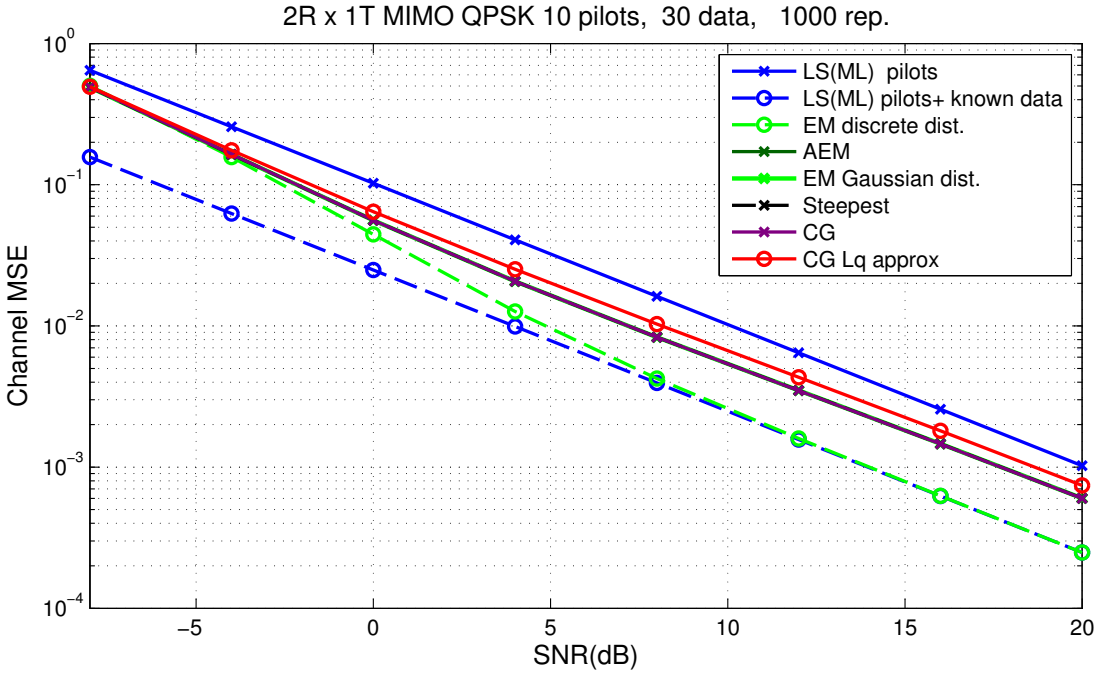

Figure 6.3: MSE versus channel SNR for $2R \times 1T$ MIMO QPSK channel. 10 pilots and 30 data symbols.

As expected, LQ approximation experiences a performance loss that in the central range is approximately equal to 1dB.

Noise estimators (Figure B.2 at page 86) also produce lower errors when semi-blind estimation is employed; in particular, methods based on Gaussian approximation are close to discrete EM: they need almost 3dB less to achieve the same MSE as pilot-based LS. We omit plots for 16-QAM since they are very similar to QPSK, with the only exception of discrete EM converging slower.

6.1.3 1R x 2T MIMO

Simulations for $1R \times 2T$ QPSK are presented in Figure 6.4 and Figure B.3; unlike the previous $2R \times 1T$ MIMO case, here it is not possible to record a significant improvement for semi-blind estimation compared to SISO channel. This can be explained by the fact that for the same number of observations we have a double amount of transmitted symbols; the uncertainty is then increased.

Pilot-only estimation also appears to augment the error variance with the number of transmitted antennas, this can be noted comparing Figure 6.4 and 6.3. The explicit relation between transmitting antennas and pilot-only MSE is provided by equation

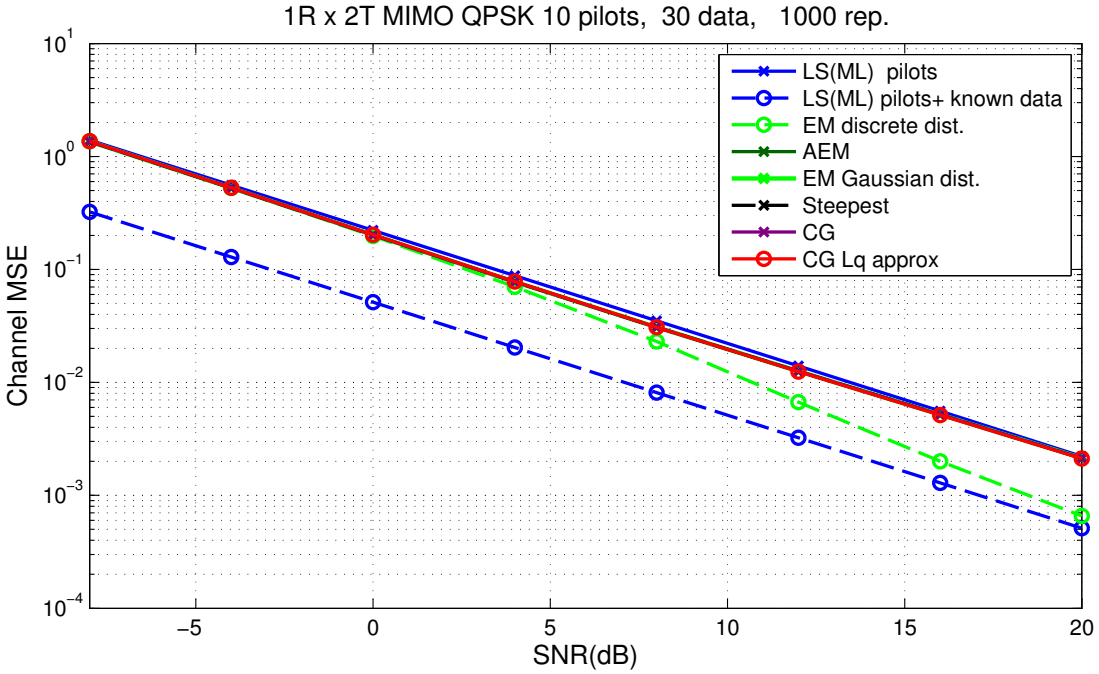

Figure 6.4: MSE versus channel SNR for $1R \times 2T$ MIMO QPSK channel. 10 pilots and 30 data symbols.

(2.19), and depends the matrix Λ_{xx} given by the actual pilot allocation strategy. For instance, it can be shown [15] that if only one antenna at the time transmits pilots when the others are silent, then the MSE grows linearly with T. Even if MSE is not improved in this scenario, having two transmitting antennas is beneficial from another point of view: it can either provide diversity, in fact transmitting the same symbol over the two antennas produces two independently fading channels; or it can alternatively double the symbol rate. Simulation results are similar for noise variance estimation, displayed in Figure B.3 at page 86. Note that for both channel and noise estimation it is not possible to spot any difference in performance between regular Gaussian approximation and LQ approximation of Section 5.4.2.

6.1.4 2R x 2T MIMO

Let us consider the $2R \times 2T$ MIMO scenario, channel and noise MSE are shown in Figure 6.5 and B.4. As the number of receiving antennas does not influence the pilot-only error, the pilot-only reference lines are placed at the same level as for $1R \times 2T$ MIMO. Compared to that case, however, semi-blind estimation is now more accurate, because of the higher number of observations available. In particular, around the 5dB

region those methods based on Gaussian approximation offer a 1.5dB advantage over pilot-only LS, while LQ approximation only achieves a 1dB improvement.

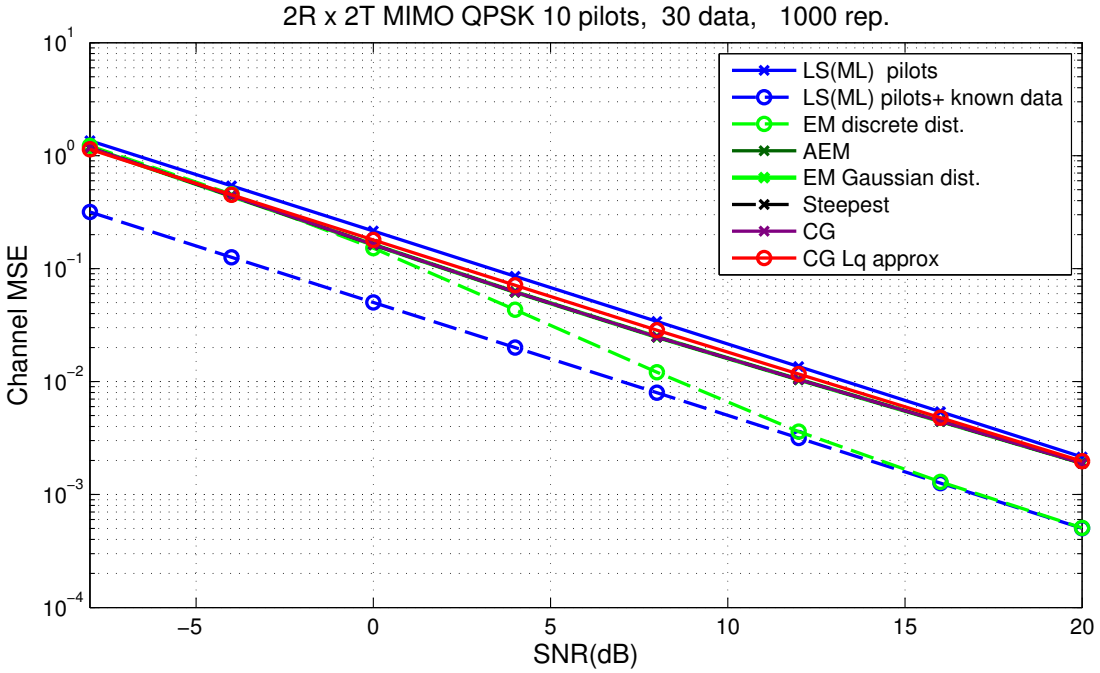

Figure 6.5: MSE versus channel SNR for $2R \times 2T$ MIMO QPSK channel. 10 pilots and 30 data symbols.

Figure 6.6 shows how capacity is affected by the estimation accuracy achieved by different methods; as expected, different estimators do not differ at low SNR levels, as AWGN dominates the effects of channel uncertainty. On the other hand, sensibility to estimation errors is maximum for low noise setups. At middle-range SNR levels, where Gaussian semi-blind methods produce some MSE gain over pilot-only ML, we also record a capacity boost: considering $2R \times 2T$ MIMO with $K_p = 10$ and $K_d = 30$, there is a 0.25 dB enhancement for Gaussian semi-blind estimators, and 0.6dB for discrete EM.

Summing up our remarks about MIMO estimation accuracy, we can conclude that:

- more transmitting antennas provide higher rates or additional diversity but make the estimation process less accurate (for both pilot-based and semi-blind methods).
- more receiving antennas increase diversity and do not affect the error of pilot-only estimation, conversely, semi-blind estimation gains precision.
- all the estimators based on Gaussian approximation reach the same error level and the same capacity.

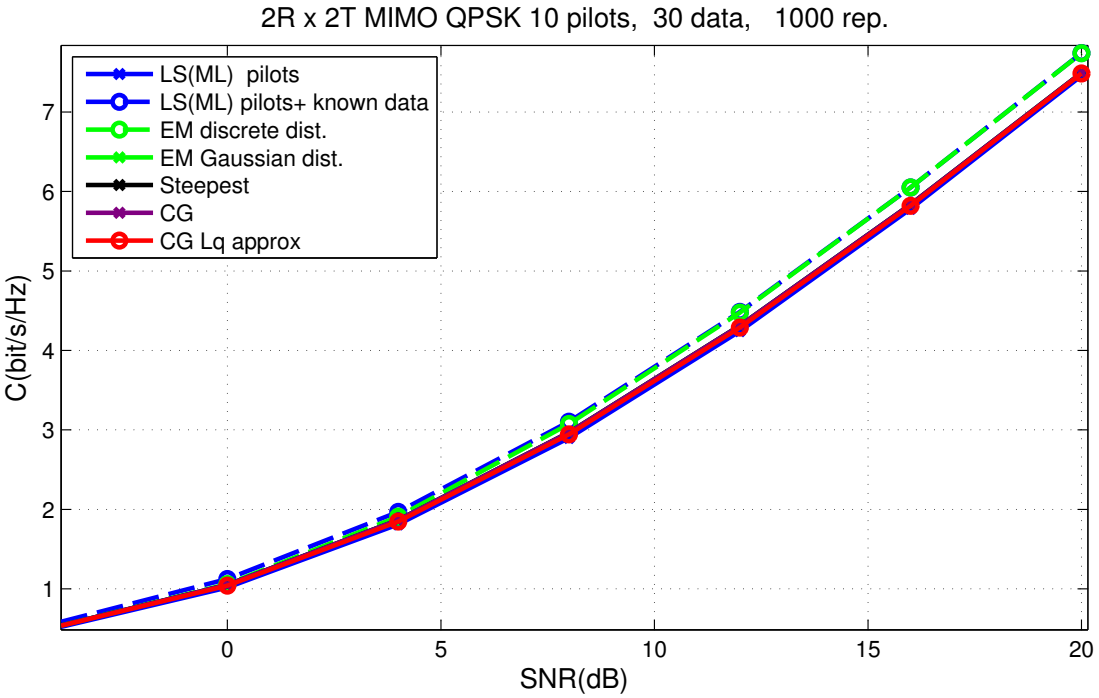

Figure 6.6: Capacity versus channel SNR for $2R \times 2T$ MIMO QPSK channel. 10 pilots and 30 data symbols.

- LQ approximation adds an additional error compared to regular Gaussian approximation, this is approximately 1db extra for $2R \times 2T$ MIMO.

6.2 Convergence Rate

In this section we evaluate the convergence rate of the estimators studied in this thesis, in particular, we will focus on those methods based on Gaussian approximation. Convergence rate is determined by two main factors: data/pilot ratio and SNR level. As we shall see, some methods are slower in specific SNR regions while they are faster in others. For simplicity, we will only consider the $2R \times 2T$ setup, which is the most representative. Figure 6.7 displays the evolution of MSE for a low-noise setup (15 dB) while Figure 6.8 refers to the 0 dB case. In general we observe that for high SNRs numerical

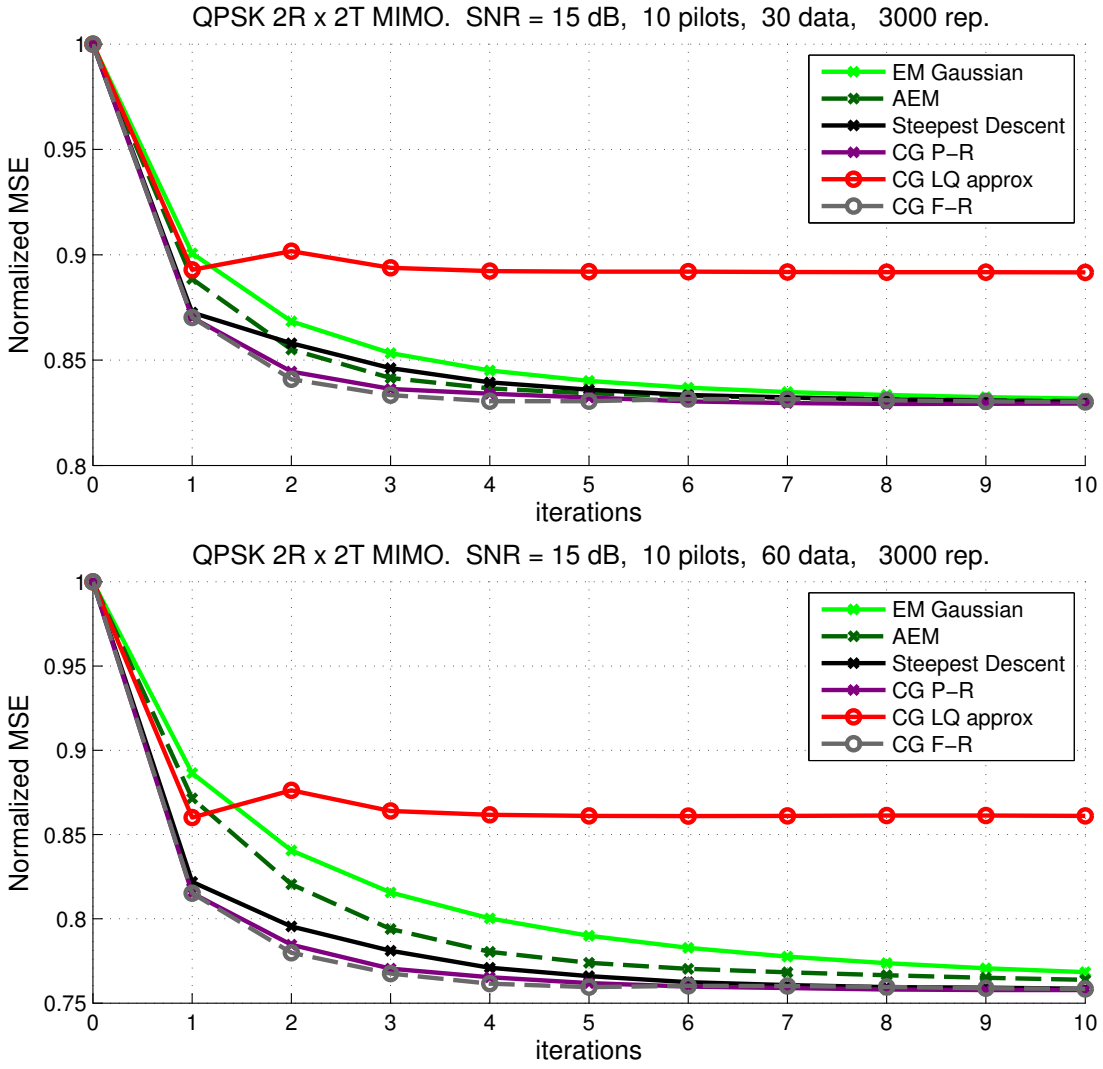

Figure 6.7: Evolution of MSE over the iterations for 3:1 and 6:1 data/pilot ratio, SNR equal to 15dB.

methods are faster then EM, especially as the data/pilot ratio increases; conversely, at lower SNRs EM dominates. As a general remark, given an SNR level, we notice an opposite behaviour for EM and numerical optimizers: if the K_d/K_p ratio increases, then EM becomes slower while numerical optimizers tend to speed up convergence. This issue is partially solved by Adaptive overrelaxed EM that, especially after few iterations, converges to the solution significantly faster than regular EM.

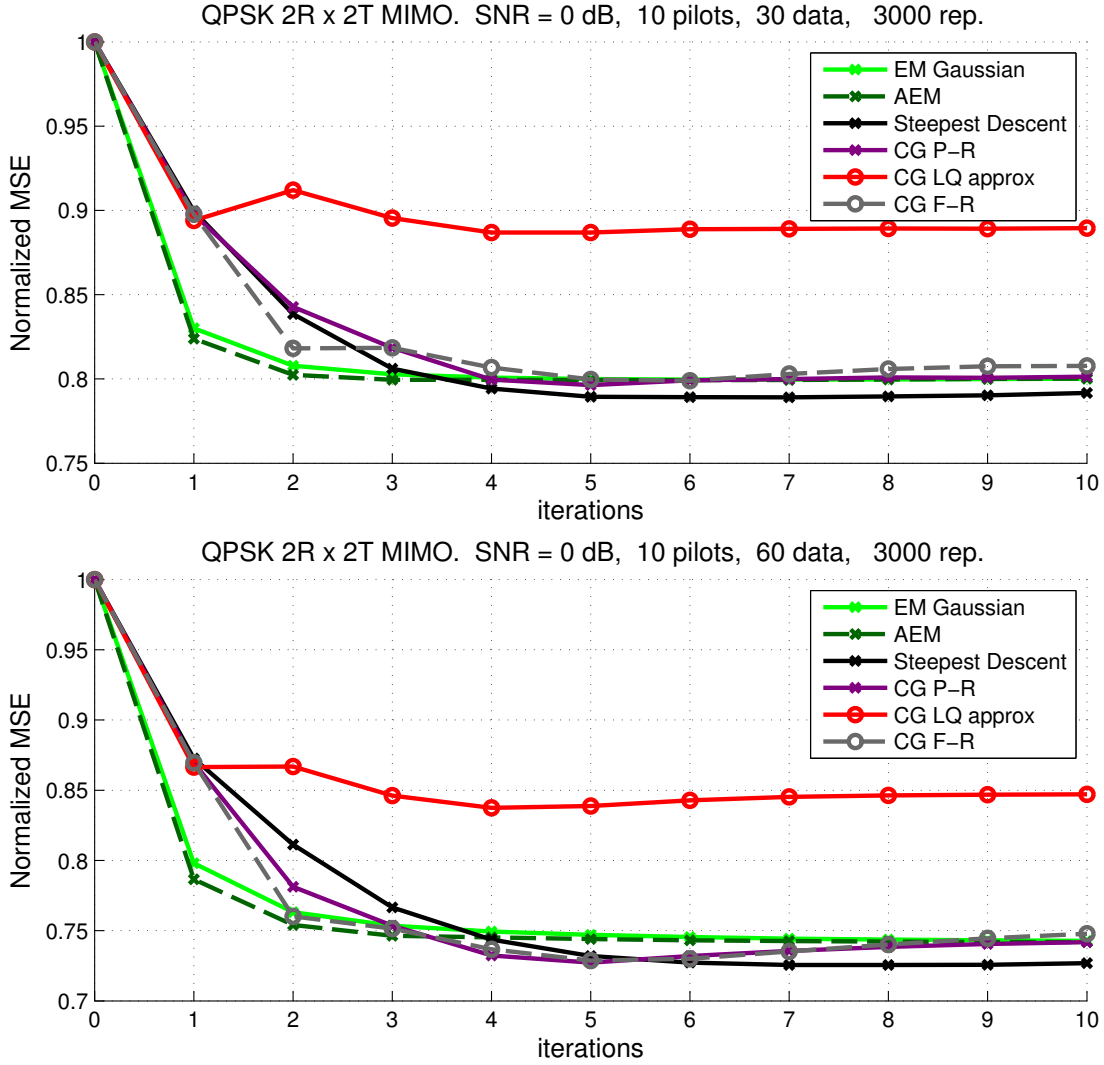

Figure 6.8: Evolution of MSE over the iterations for 3:1 and 6:1 data/pilot ratio, SNR equal to 15dB.

6.3 Computational Load

In this section we will try to evaluate the overall complexity of semi-blind estimators. As explained at the beginning of this chapter, it is not only important to define how many iterations are needed for convergence, but also what computational cost is involved. It is clear that there exists a sort of trade-off between the cost of each iteration and the total number of iterations. As before, convergence is dependent on the SNR and the data/pilot ratio; for this reason, algorithms are simulated varying both the noise level and the number of data symbols. As seen in Chapter 3, semi-blind estimators based on numerical optimization have the great advantage of being unaffected by an increase of the number of data symbols: in fact, according to Observation 1, sample covariance is a sufficient statistic for the observations and can be computed "off-line", before the actual optimization. Conversely, EM needs to iterate between data posterior moment evaluation and joint estimation; as already noted, this makes the execution time of EM grow linearly with the size K_d of the data array.

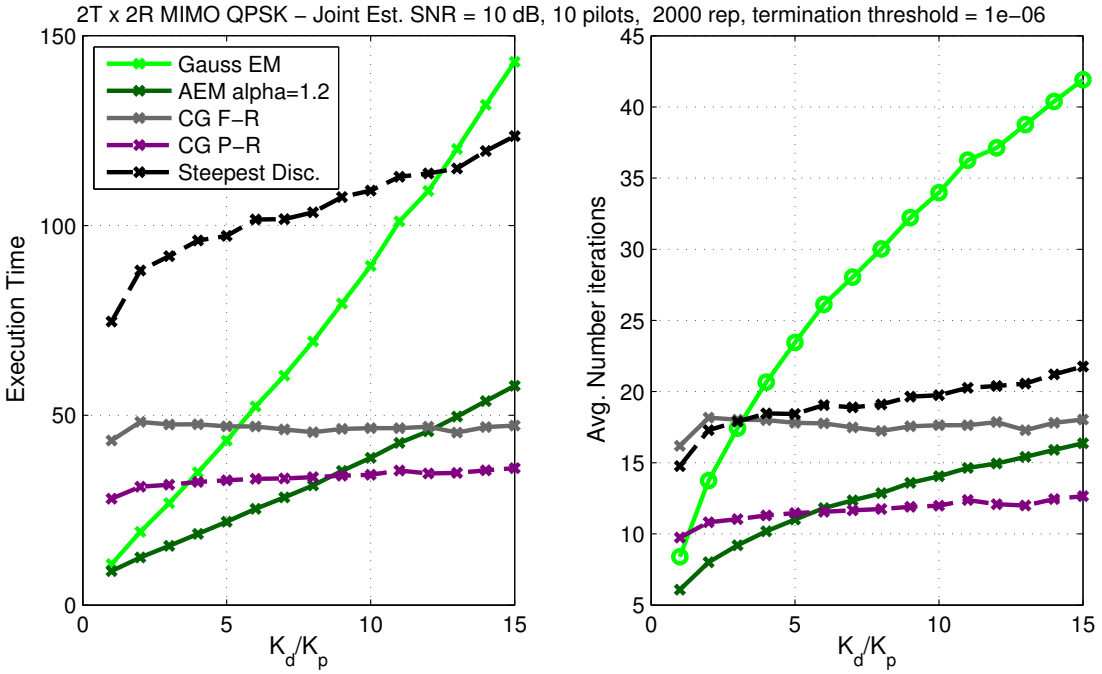

Figure 6.9: Execution time and average number of iterations as functions of the data/pilot ratio for $2R \times 2T$ MIMO QPSK channel. SNR = 10 dB

From Figure 6.9 it is easy to observe that, for conjugate gradient methods, the execution time remains almost constant with K_d, while it grows for EM. It is interesting to spot at which data/pilot ratio CG-PR becomes more advantageous than EM: this

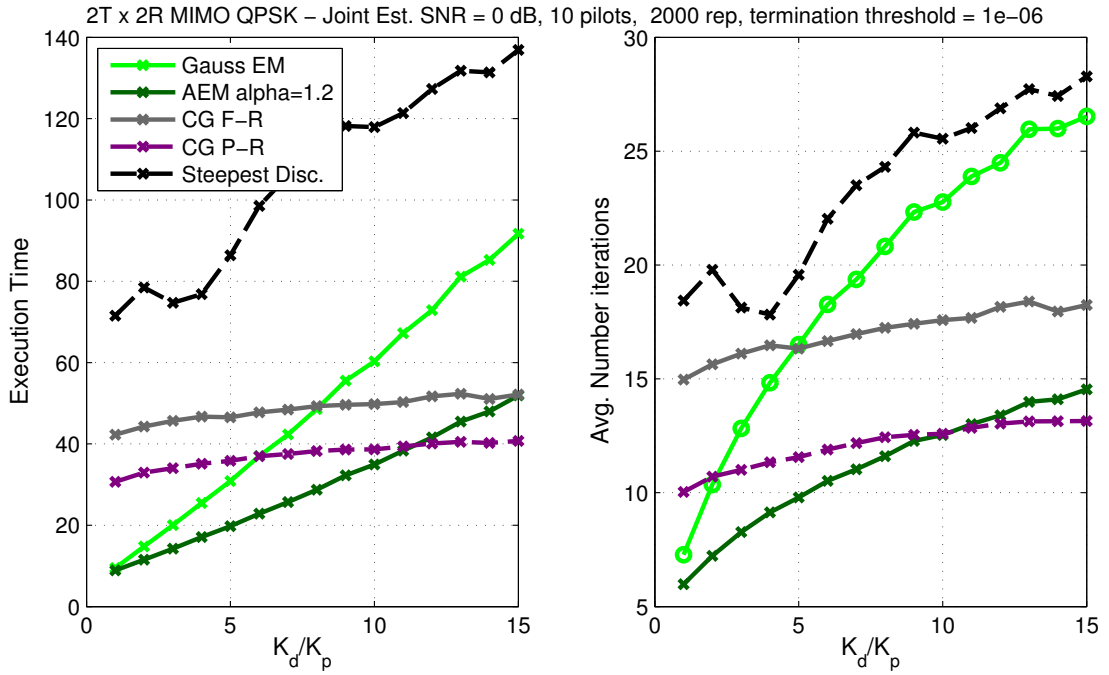

Figure 6.10: Execution time and average number of iterations as functions of the data/pilot ratio for $2R \times 2T$ MIMO QPSK channel. SNR = 0 dB

happens at 4:1 for 10 dB SNR level and at 6:1 if the SNR is 0 dB dB (see Figure 6.10).

From Figure 6.11 and 6.12 we notice that, varying the SNR, EM behaves unexpectedly: for mid-range SNRs, the number of iterations raises, while it decreases for high and low SNR levels. This fact has a negative impact on application purposes; in fact, looking at Figure 6.5, we notice that the range where EM is the slowest is also the range where it provides an accuracy gain over pilot-only LSE. In other words, it is only beneficial to use EM in the region where it converges slowly. Conversely, CG methods seem to be more stable for both the number of iterations and the total execution time.

As discussed in Section 4.5, AEM is expected to outperform standard EM in terms of number of iterations; this is confirmed by our simulations and, most notably, Figures 6.9 and 6.10 show that this acceleration comes with a small computational overhead. Notice that, according to Figure 6.11, this advantage is more evident in the central SNR region, where EM proved to be slower: unlike EM, AEM curve is flat across the SNR span. We can conclude that AEM solves the slowing-down convergence issue experienced by EM, and it is complexity-wise superior to it for any SNR and K_d/K_p ratio.

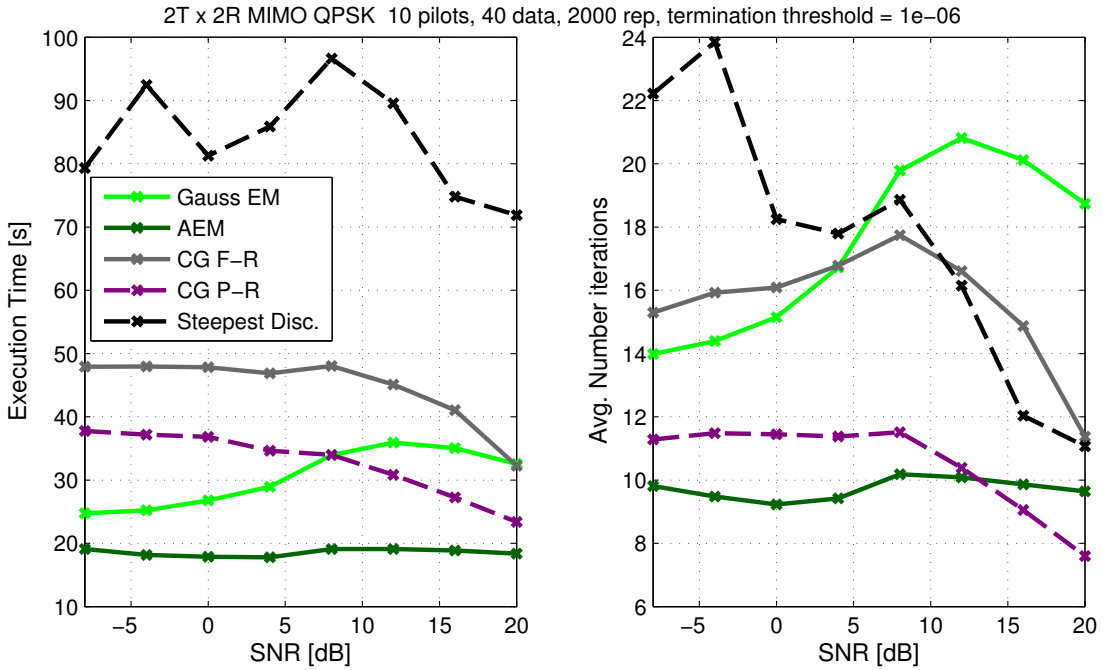

Figure 6.11: Execution time and average number of iterations as functions of the SNR for $2R \times 2T$ MIMO QPSK channel. $K_d/K_p = 4:1$

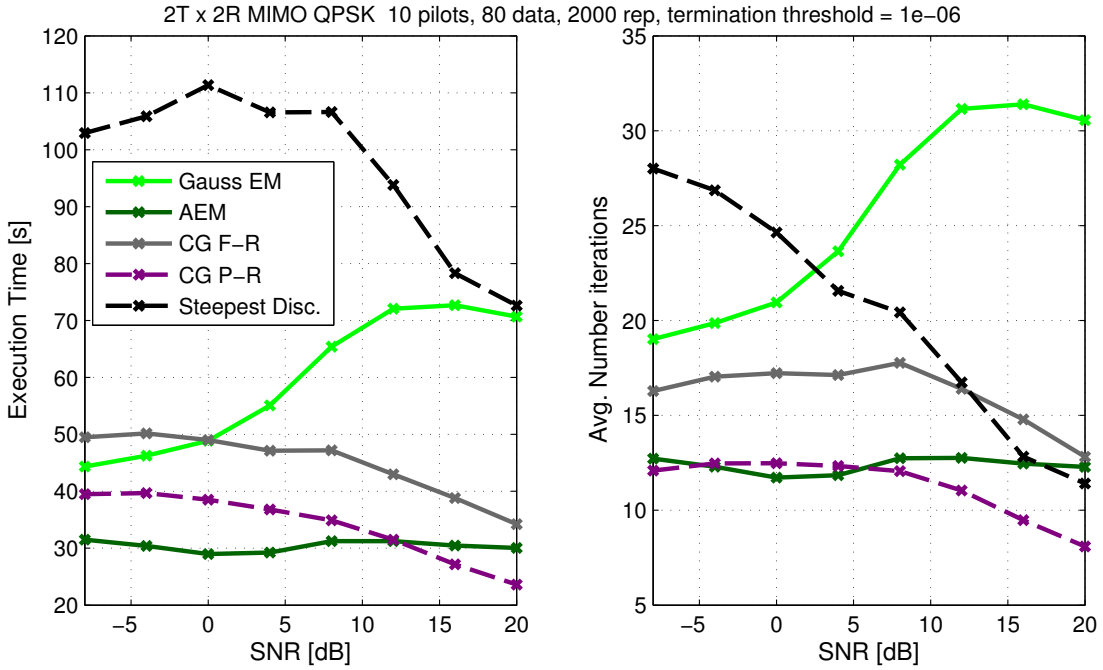

Figure 6.12: Execution time and average number of iterations as functions of the SNR for $2R \times 2T$ MIMO QPSK channel. $K_d/K_p = 8:1$

Regarding the relation between modulation scheme and computational cost of the estimators, Figure 6.13 shows that results for 16QAM are very similar to QPSK. Since all these methods assume a Gaussian distribution of symbols, they are unaffected by the constellation chosen. On the other hand, as CG complexity is given by the cost of computing the gradient matrix, conjugate gradient performs better if the number of receiving antennas is reduced; this can be seen in Figure B.7 on page 88 for a $1R \times 2T$ setup and Figure B.8 for a $2R \times 1T$ setup.

Summarizing, the choice between AEM and CG depends on the SNR, the data/pilot ratio and the MIMO setup; in light of the simulations we can deduce that:

- AEM outperforms EM for the whole SNR span and K_d/K_p range.
- CG is preferable to AEM at middle-high SNR levels (more than 10dB for $2R \times 2T$) and for large data/pilot ratios (the exact value depending on the specific MIMO setup).
- CG degrades its performance with respect to AEM as the number of antennas increases.

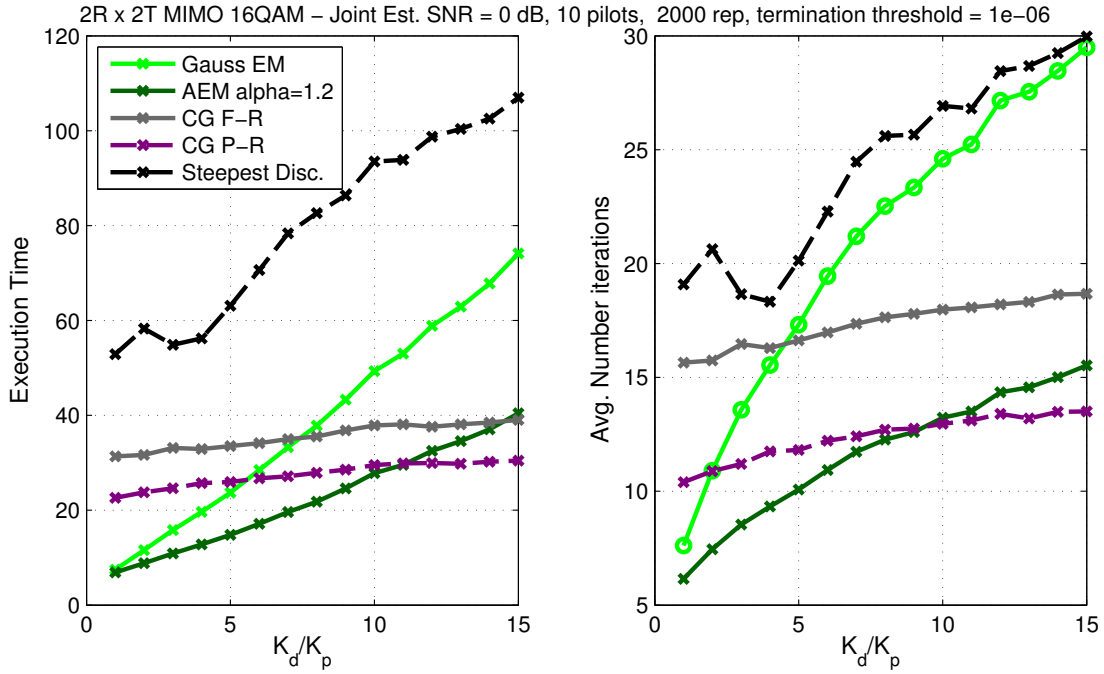

Figure 6.13: Execution time and average number of iterations as functions of the data/pilot ratio for $2R \times 2T$ MIMO 16QAM channel. SNR = 0 dB

Chapter 7

Conclusions

In this thesis we described and addressed the problem of semi-blind channel estimation from a complexity-aware prospective: our aim was to investigate and test different approaches leading to a reduction in the number of computations required by the estimation process.

We started by defining a model for OFDM-MIMO, assuming a flat-fading Rayleigh channel with white Gaussian additive noise; then the semi-blind problem was formulated. In Chapter 2, pilot-only estimation was reviewed: this method consisted in a robust one-shot maximization of the likelihood function of the channel given the observations and the pilot-sequence. As it only involves one $R \times T$ matrix inversion, pilot-only ML is the simplest way of inferring channel and noise variance. Afterwards, we derived a general expression for the mean squared error of this estimator; additionally, it can be shown that, if one non-zero pilot per antenna is allocated at each OFDM symbol, the MSE increases with the number of transmitting antennas, while it drops as the inverse of the number of pilots. This implies that, in order to achieve the same error, if the number of transmitting antennas grows, the length of the pilot sequence has to increase as well. Of course, longer pilot sequences reduce the overall efficiency of the medium.

Instead of sending more pilots, we can enhance the estimation error exploiting the additional information provided by data observations. In Chapter 3 we introduced the semi-blind approach and presented two different ways to deal with the uncertainty concerning transmitted symbols. Firstly, the true discrete symbol distribution was used to marginalize the likelihood function: this yielded a complex combinatorial expression. Then, we approximated the unknown symbols by means of a zero-mean Gaussian distribution. In the latter case, the expression was easier and had an interesting feature: the observation sample covariance matrix is a sufficient statistic for the observation vector.

However, in both of the cases it was not possible to deduce a closed-form expression for the channel and the noise maximizing the likelihood function. This means that, unlike pilot-only ML, semi-blind estimators will require several iterations to converge to a local solution. It is clear that there is a trade-off between the complexity of an iterative semi-blind solution and the spectral inefficiency of the pilot-only ML estimator.

Chapter 4 presented the EM framework as an elegant solution to the semi-blind problem. It alternates between two steps: the computation of first and second-order posterior moments of the unknown symbols (E-step) and the estimation of H and σ^2 (M-step). While the M-step only requires one $R \times T$ matrix inversion, the complexity of the E-step depends on the unknown symbols. In case of discrete distribution, the number of operations grows linearly with the number of data symbols and exponentially with the constellation order. Conversely, if the Gaussian approximation is utilised, there is no dependence on the actual modulation scheme, but complexity still expands with the size of the data. This situation is undesirable because we are typically interested in using as much data as possible to refine the estimate.

Preliminary tests showed that EM convergence rate was unsatisfactory, in particular we recorded an evident slow-down at high-SNR regime. To correct this situation, an adaptive overrelaxed version of EM (AEM) has been adapted to our case to accelerate the traditional EM algorithm. Employing a quality of the Gaussian log-likelihood, we showed that, for each iteration, it is possible to enhance the step size without compromising the convergence guarantees and adding only a small computational overhead.

In order to make use of the sufficiency property 1, in Chapter 5 we reviewed several numerical optimization techniques that could solve the problem without computing posteriors. Starting from the basic steepest descent, we analysed the Newton-Raphson method and two non-linear conjugate gradient techniques. Convergence properties have been explained and related to our case of interest. Moreover, we gave an operative expression for the complex matrix-valued gradient of the log-likelihood. Finally, two additional solutions were proposed: a reduced-complexity estimator relying on the LQ decomposition that can be used to reduce the space of parameters and a constraint-based optimizer applicable to constant-modulus SISO channels.

Finally, in Chapter 6 all these algorithms were tested for accuracy, convergence speed and computational cost. We observed that, for low SNRs, semi-blind methods do not enhance pilot-only estimation because noise dominates the contribution given by the data. At high SNR levels, discrete EM approaches CRLB with perfect symbol knowledge; however, this result is achieved at the expense of a great computational load. On the other hand, methods based on the Gaussian approximation produce an estimation gain

only for middle-high (0-15 dB) SNR levels. Furthermore, we noted that employing numerical optimizers in place of EM is advantageous for small MIMO setups, higher SNRs (over 10-15 dB) and large data/pilot ratios (over 5-8:1). Moreover, AEM proved to outperform EM in terms of average number of iterations and computations across all the SNR range. As it attains the same mean squared error, AEM is to prefer to the standard version of EM.

Appendix A

Complex Derivatives

Let $f(z)$ be a complex-valued function of a complex variable $z \in R \subseteq \mathbb{C}$, before differentiating we should make sure that f is *analytical* in R, i.e. that the Cauchy-Riemann equations are satisfied:

$$\frac{\partial f(z)}{\partial \Im z} = i\frac{\partial f(z)}{\partial \Re z} \quad \forall z \in R \tag{A.1}$$

This is equivalent to require that $f(z)$ is independent of z^*. Unfortunately, for most of the practical functions in use this requirement is not met. To overcome this limitation, complex derivatives can be generalized as *formal complex derivatives*:

$$\frac{df(z)}{dz} \triangleq \frac{1}{2}\left(\frac{\partial f(z)}{\partial \Re z} - i\frac{\partial f(z)}{\partial \Im z}\right) \qquad \frac{df(z)}{dz^*} \triangleq \frac{1}{2}\left(\frac{\partial f(z)}{\partial \Re z} + i\frac{\partial f(z)}{\partial \Im z}\right) \tag{A.2}$$

If f is analytical, the generalized derivative coincides with the standard one and $\frac{df(z)}{dz^*} = 0$.

If we assume that $f(Z)$ is a real-valued function of the $n \times m$ complex matrix Z, then, the *matrix complex derivatives* $\frac{df(Z)}{dZ}$ and $\frac{df(Z)}{dZ^*}$ are $n \times m$ matrices defined as follows:

$$\left(\frac{df(Z)}{dZ}\right)_{ij} \triangleq \frac{df(Z)}{dZ_{ij}} \quad i = 1, \ldots, n \quad j = 1, \ldots, m \tag{A.3}$$

$$\left(\frac{df(Z)}{dZ^*}\right)_{ij} \triangleq \frac{df(Z)}{dZ^*_{ij}} \quad i = 1, \ldots, n \quad j = 1, \ldots, m \tag{A.4}$$

where $Z_{ij} \in \mathbb{C}$ is the (i, j) element in Z. This type of derivative is the most used in this thesis.

The gradient of $f(Z)$ can be defined as the $n \times m$ matrix:

$$\nabla f(Z) \triangleq 2\frac{df(Z)}{dZ^*} = \frac{\partial f(z)}{\partial \Re Z} + i\frac{\partial f(Z)}{\partial \Im Z} \tag{A.5}$$

Appendix B

Additional Figures

Noise Estimation MSE

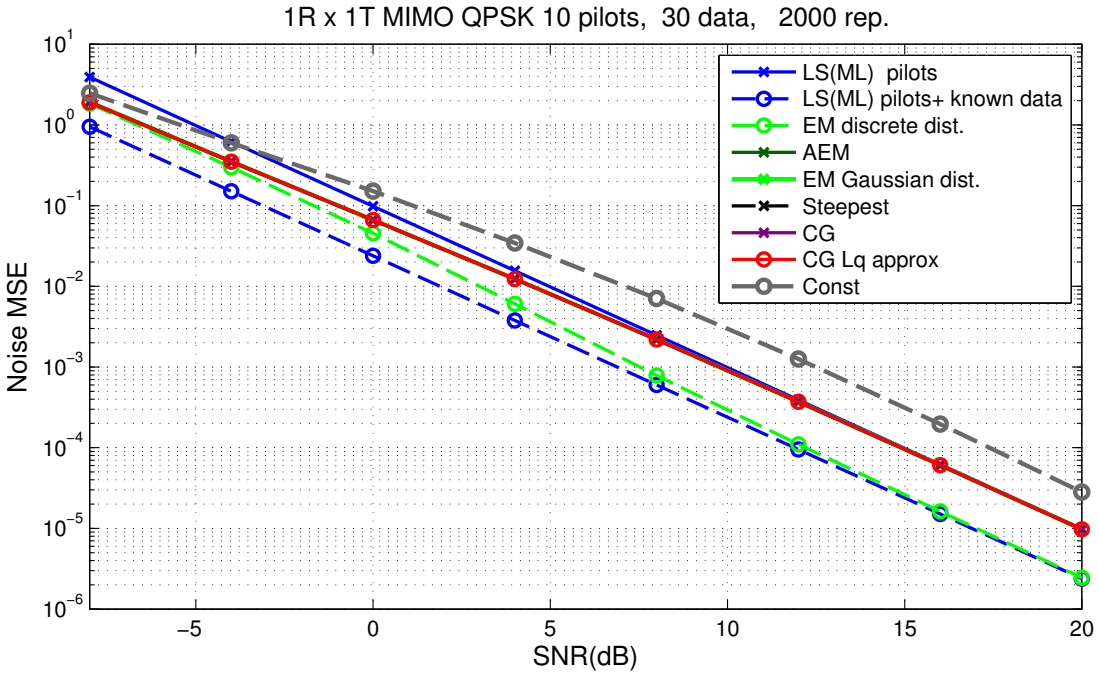

Figure B.1: MSE versus noise SNR for SISO QPSK noise variance. 10 pilots and 30 data symbols.

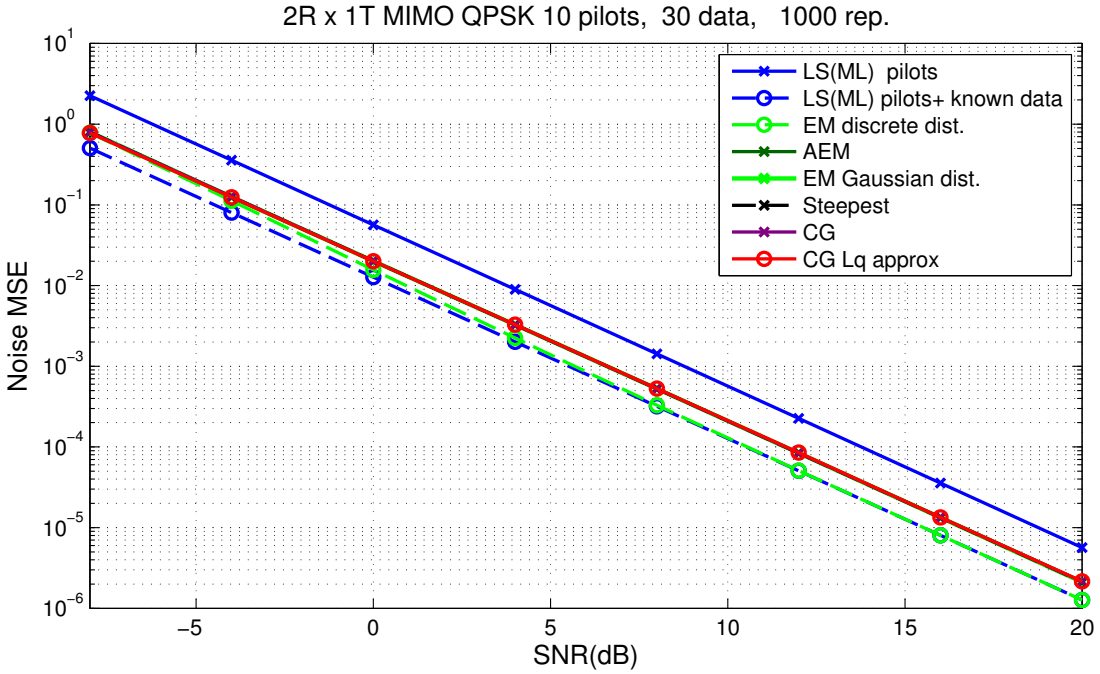

Figure B.2: MSE versus noise SNR for $2R \times 1T$ QPSK noise variance. 10 pilots and 30 data symbols.

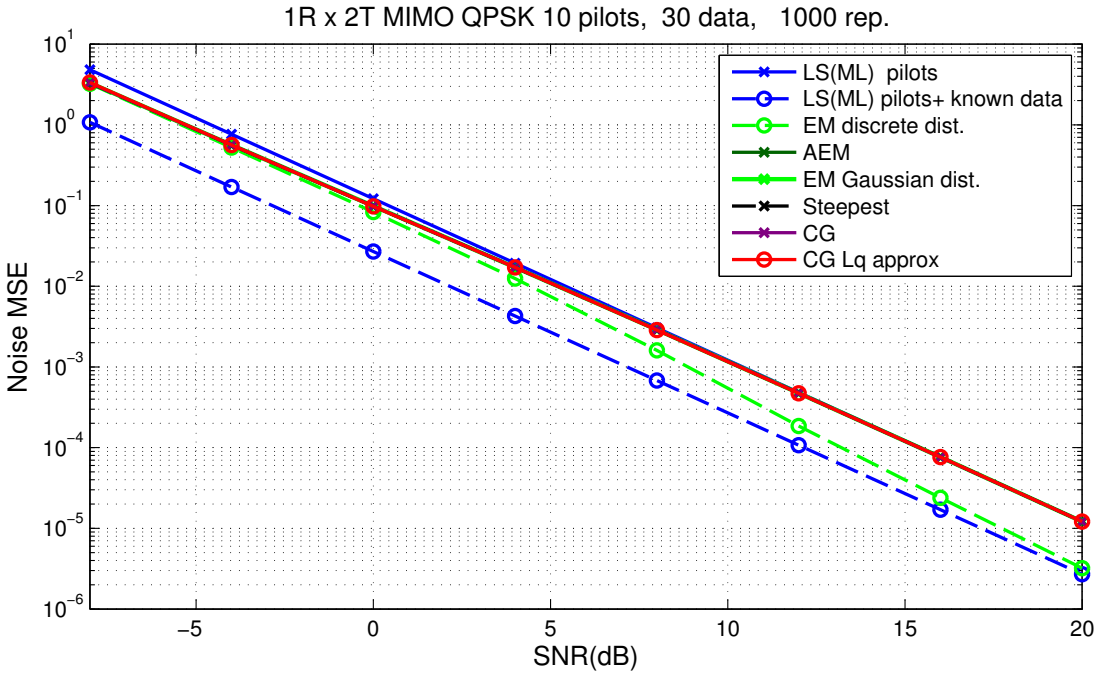

Figure B.3: MSE versus noise SNR for $1R \times 2T$ QPSK noise variance. 10 pilots and 30 data symbols.

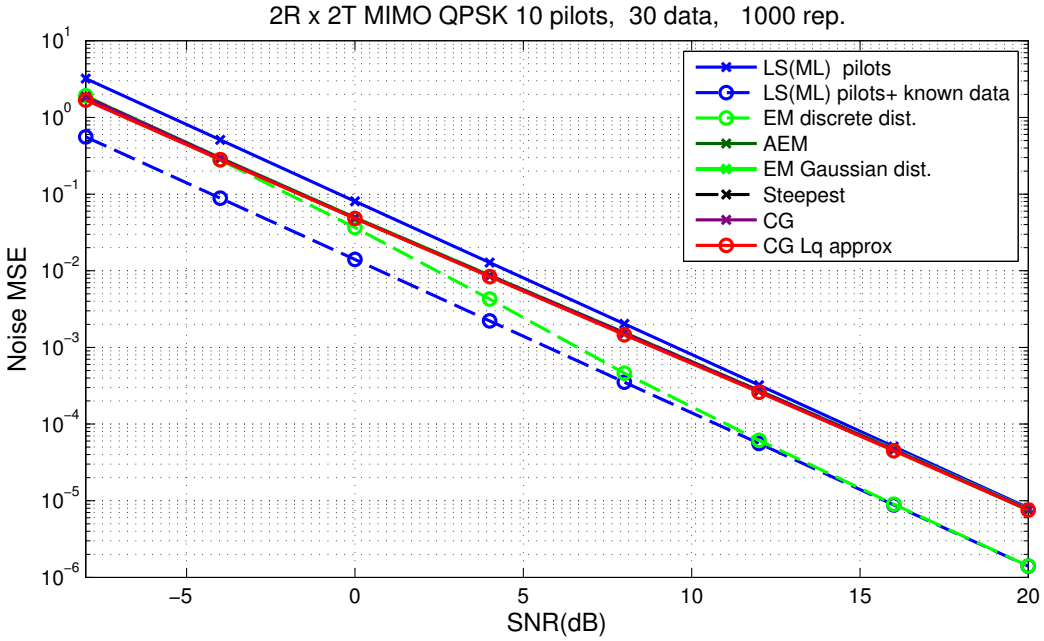

Figure B.4: MSE versus noise SNR for $2R \times 2T$ QPSK noise variance. 10 pilots and 30 data symbols.

Capacity for a Different Data/Pilot Ratio

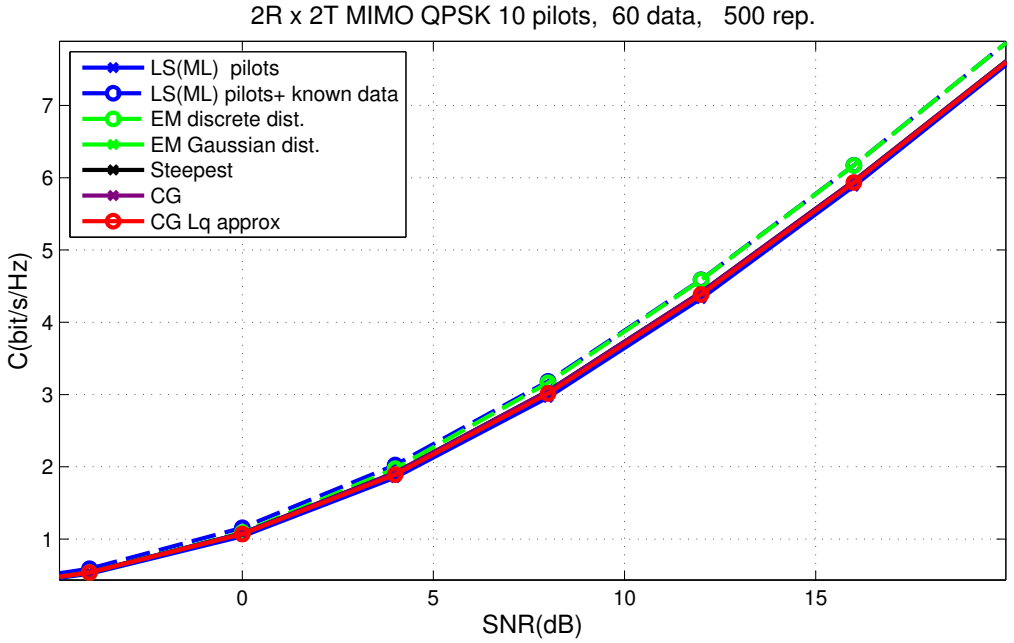

Figure B.5: Capacity versus channel SNR for $2R \times 2T$ MIMO QPSK channel. 10 pilots and 60 data symbols.

Computational Load for Different MIMO Setups

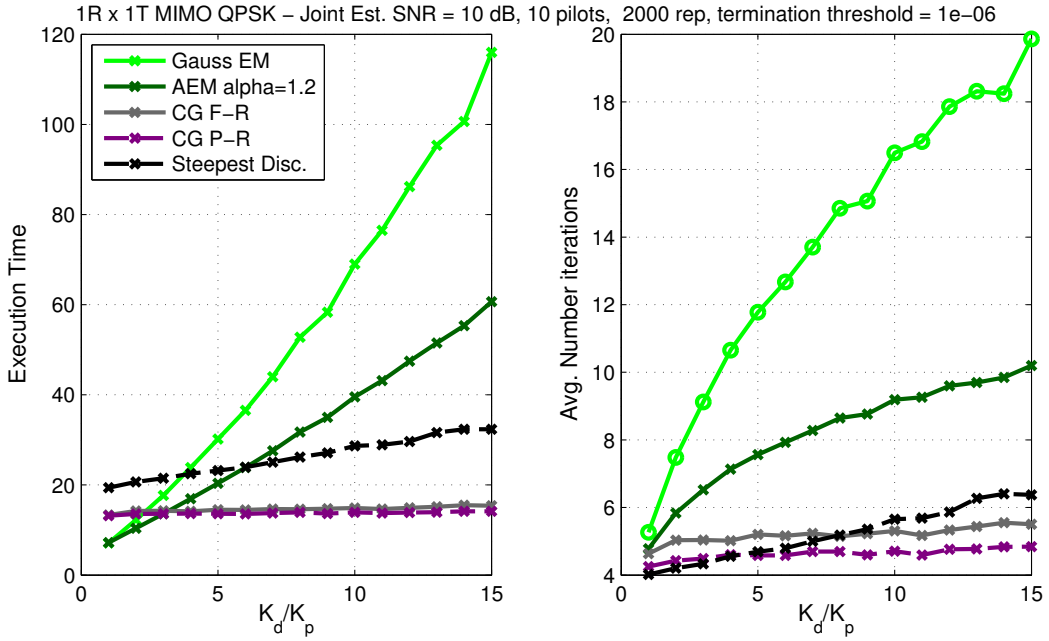

Figure B.6: Execution time and average number of iterations as functions of the data/pilot ratio for $1R \times 1T$ MIMO QPSK channel. SNR = 10 dB

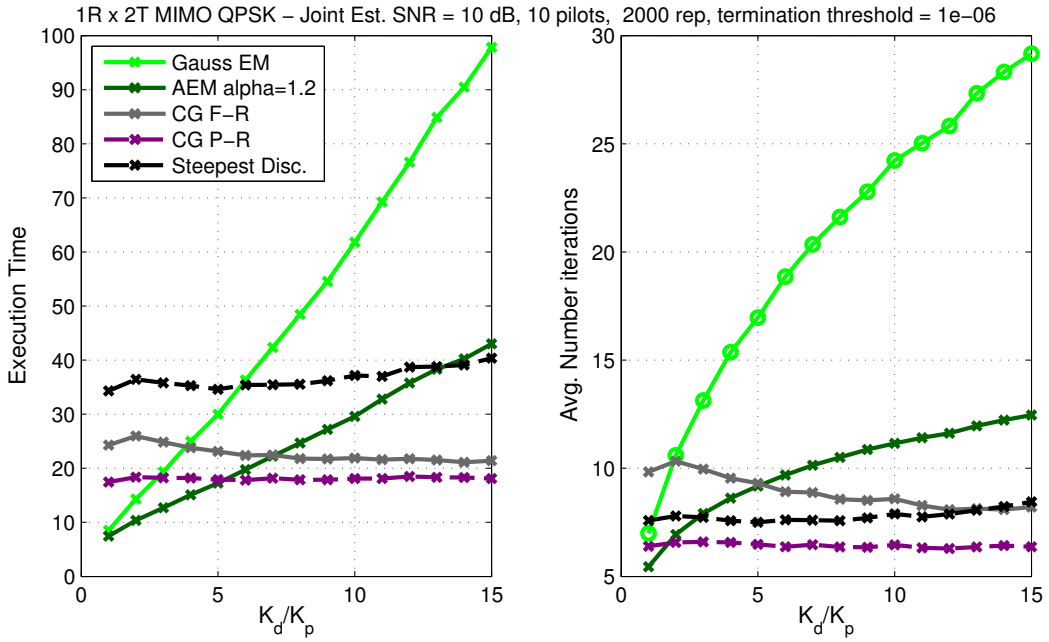

Figure B.7: Execution time and average number of iterations as functions of the data/pilot ratio for $1R \times 2T$ MIMO QPSK channel. SNR = 10 dB

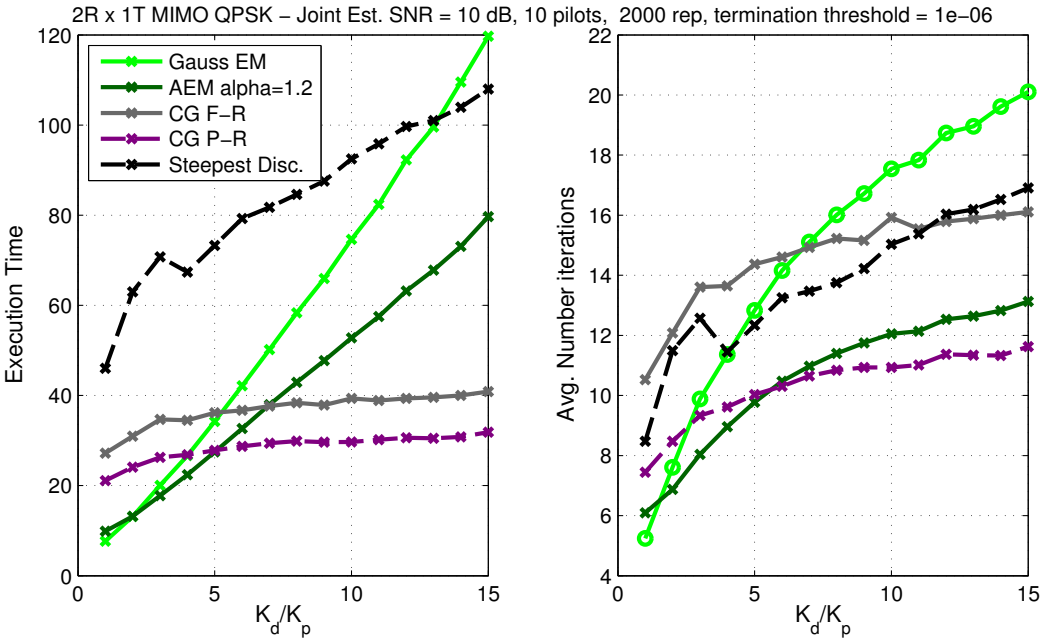

Figure B.8: Execution time and average number of iterations as functions of the data/pilot ratio for $2R \times 1T$ MIMO QPSK channel. SNR $= 10$ dB

Bibliography

[1] James R. Beniger. *The Control Revolution: Technological and Economic Origins of the Information Society.* Harvard University Press, 1986.

[2] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, Cambridge, 2005.

[3] Adriaan van den Bos. A Cramer-Rao Lower Bound for Complex Parameters. *IEEE Transactions on Signal Processing*, 42(10):–, 1994.

[4] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing.* Prentice Hall, 4 edition, April 2006.

[5] Yong Soo Cho, Jaekwon Kim, Won Young Yang, and Chung G. Kang. *MIMO-OFDM wireless communications with MATLAB.* Wiley & Sons, Singapore, 2010.

[6] A. Hjorungnes and D. Gesbert. Complex-Valued Matrix Differentiation: Techniques and Key Results. *IEEE Transactions on Signal Processing*, 55(6):2740 –2746, june 2007.

[7] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, New York, NY, 2006.

[8] Nitis Mukhopadhyay. *Probability and Statistical Inference.* Marcel Dekker, Inc., 2000.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[10] K.B. Petersen, O. Winther, and L.K. Hansen. On the slow convergence of EM and VBEM in low-noise linear models. *Neural computation*, 17(9):1921–1926, 2005.

[11] Ruslan Salakhutdinov and Sam Roweis. Adaptive Overrelaxed Bound Optimization Methods. In *Proceedings of International Conference on Machine Learning, ICML. International Conference on Machine Learning, ICML*, pages 664–671, 2003.

[12] C. F. Jeff Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[13] Dimitri P. Bertsekas and Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999.

[14] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, August 2000.

[15] Xiaoli Ma, Liuqing Yang, and Georgios B. Giannakis. Optimal Training for MIMO Frequency-Selective Fading Channels. *IEEE Transactions on Wireless Communications*, 4:453–466, 2005.