# Design of Iterative Message-Passing Receivers with Sparse Channel Estimators

- Master Thesis -

Peter Bjørn Jørgensen

**Aalborg University**
**School of Information and**
**Communication Technology**
pjoerg08@student.aau.dk
June 2013

AALBORG UNIVERSITY
STUDENT REPORT

**Department of Electronic Systems**
Fredrik Bajers Vej 7
DK-9220 Aalborg Øst
http://es.aau.dk

**Title:**
Design of Iterative Message-Passing Receivers with Sparse Channel Estimators

**Master Programme:**
Wireless Communication

**Project Period:**
Spring Semester 2013

**Participant(s):**
Peter Bjørn Jørgensen

**Supervisor(s):**
Bernard H. Fleury*
Mihai-Alin Badiu*
Niels L. Pedersen*
Torben Larsen*
Lars Christensen†

* Department of Electronic Systems

† Wireless Algorithm Design, Renesas Mobile Copenhagen

**Copies:** 6

**Number of Pages:** 100

**Date of Completion:**
June 6, 2013

**Abstract:**

In this thesis we devise a novel iterative orthogonal frequency-division multiplexing receiver algorithm that exploits the assumption that the wireless channel is sparse, i.e. its impulse response consists of a few dominant components. The task of the sparse channel estimation is posed as a sparse signal estimation problem. Using the approach of sparse Bayesian learning with hierarchical prior modeling the channel estimation problem is integrated with the other receiver tasks through a factor graph representation of the whole system. The iterative algorithm for joint channel estimation and decoding is thus analytically derived by applying the combined belief propagation and mean field inference framework as message-passing on this factor graph.
Our numerical results show that the proposed algorithm outperforms, in terms of bit-error-rate, an analogous receiver that uses a robust channel assumption, but does not exploit the sparsity of the channel. As the channel estimation part is of high computational complexity we propose two different methods for reducing the complexity: 1) heuristic modification of the message-computation scheduling such that only part of the soft data information is used in channel estimation 2) grouping channel variables of the factor graph into vectors of a certain size. For the devised receiver algorithm the second method does not degrade the bit-error-rate performance of the receiver, while the complexity is significantly reduced.

AALBORG UNIVERSITET
STUDENTERRAPPORT

**Institut for Elektroniske Systemer**
Fredrik Bajers Vej 7
DK-9220 Aalborg Øst
http://es.aau.dk

**Titel:**
Design af Iterative Modtagere via Approksimativ Bayesiansk Inferens.

**Specialeretning:**
Trådløs Kommunikation

**Projektperiode:**
Forårssemestret 2013

**Deltager(e):**
Peter Bjørn Jørgensen

**Vejleder(e):**
Bernard H. Fleury*
Mihai-Alin Badiu*
Niels L. Pedersen*
Torben Larsen*
Lars Christensen†

* Institut for Elektroniske Systemer

† Wireless Algorithm Design, Renesas Mobile Copenhagen

**Oplagstal:** 6

**Sidetal:** 100

**Afleveringsdato:**
6. juni 2013

**Synopsis:**

I dette speciale udvikles en ny iterativ algoritme til modtagelse af ‘orthogonal frequency division multiplexing’ radiosignaler som udnytter scenariet hvor kanalens impulsrespons består af nogle få dominerende komponenter. Estimering af denne type kanal formuleres som et komprimeret signalbehandlingsproblem. Vha. den bayesianske metode integreres problemet med modtagerens andre opgaver via en faktorgrafrepræsentation af hele systemet. En hierarkisk a priori sandsynlighedsfordeling pålægges kanalens impulsrespons. Den iterative algoritme udledes analytisk ved anvendelse af en kombinationen af ‘belief propagation’ og ‘mean field’ metoderne på denne faktorgraf, hvorved estimering af kanalen og dekodning foretages som en samlet opgave.

Simuleringer viser at den udviklede algoritme opnår lavere bitfejlrate end en tilsvarende modtager baseret på en robust antagelse om kanalen, men som ikke udnytter at kanalens impulsrespons kan beskrives vha. få dominerende komponenter. Derudover undersøges to metoder til reducering af algoritmernes beregningskompleksitet: 1) modifikation af den iterative algoritme således kun en del af data informationen anvendes til estimering af kanalen 2) gruppering af kanalvariablerne i faktorgrafen i vektorer af en bestemt størrelse. Den sidstnævnte metode giver ikke anledning til højere bitfejlrate for udviklede modtager, mens modtagerens beregningskompleksitet reduceres markant.

# Preface

This master thesis is written by me, Peter Bjørn Jørgensen, at the School of Information and Communication Technology, Aalborg University during my 10th semester in project period from February 1st to June 6th 2013. During the project period I was affiliated with the Navigation and Communications (NavCom) section at the Department of Electronics Systems at Aalborg University. The project was conducted in collaboration with the wireless algorithm design group at Renesas Mobile, Copenhagen. The interaction took place via email, online meetings and a physical meeting in May at the premises of Renesas Mobile in Copenhagen.

The inspiration for the topic of the master thesis originates from the NavCom section that already has done remarkable research in the fields of 1) variational Bayesian methods and its application to design of wireless receivers 2) sparse estimation using Bayesian hierarchical prior modeling. I also had the opportunity to work on the latter of the two topics in my 8th semester project. This project is a move in the direction of joining these two vast scientific areas. The involvement of Renesas Mobile emphasizes that the methods are not only of scientific interest, but they have been identified as potential solutions for challenges in the advancement of future wireless communication systems. I would like to thank the people at Renesas Mobile Copenhagen for their engagement in the project.

The report is structured as follows. In Chapter 1 the context of the project and its goals are defined, followed by Chapter 2 in which the scope is further narrowed by the definition of a system model and the problem we want to solve. A brief survey of some of the results from the two aforementioned scientific fields is given in Chapter 3 and in the following chapter a current state-of-the-art iterative receiver is introduced. The main contributions of the thesis is found in Chapter 5, 6 and 7. We propose a novel iterative receiver with sparse channel estimation, benchmark its performance against current state-of-the-art receivers and propose two methods for reducing the computational complexity of the receiver scheme.

The report contains a number of appendices. Appendix A is a list of the notation and symbols used throughout the report and a list of used acronyms is found in Appendix B. The remaining appendices are referenced within the text whenever they become relevant. In addition to this report another outcome of the project is a simulation framework, implemented in Matlab, that includes the system model, channels and receivers discussed in the thesis. All code and simulation results can be found on the accompanying CD together with a digital copy of the report in ‘pdf’ format.

Aalborg University, June 6th, 2013

Peter Bjørn Jørgensen

# Contents

**Appendices**

CHAPTER 1

# Introduction

Mobile communication systems have undergone a rapid evolution during recent years in order to meet the ever-increasing demand for higher data rates. The global trend the last five years has shown an annual increase of 40% in the number of mobile broadband subscriptions [1]. The bandwidth allocated to mobile systems has increased, up to 100 MHz for future systems [2]. On the other hand higher spectral efficiency is achieved with new technologies such as orthogonal frequency division multiplexing (OFDM), mutiple-input multiple-output (MIMO) processing and modern channel coding techniques. One of the challenges in modern wireless digital receiver design is how to implement all the new features and at the same time meet the requirements of low power consumption, low computational complexity and small memory footprint that implementation on handheld devices impose.

Approximate inference on factor graphs [3, 4] in the form of message-passing has shown to be an efficient and flexible tool to design the signal-processing tasks of wireless receivers [5–8]. As opposed to the classical approach in which the different operations (channel estimation, demapping, decoding etc.) of the receiver are optimized separately according to some optimality criteria, this approach allows a unified design of all the receiver tasks.

One of the critical tasks the receiver has to perform is estimation of the wireless channel's response. The accuracy of this estimate has a significant impact on the overall performance of the receiver. To improve the accuracy of the estimates the underlying structure of the channel can be exploited. In a multipath radio channel the transmitted signal arrives at the receiver via multiple propagation paths, each associated with a certain time delay [9]. In some channel models, e.g. the standardized 3GPP channel models [10] there is only a few dominating multipath components. This kind of channel is referred to as sparse channels. Reconstruction of sparse signals from noisy measurements has gained much attention in recent years due to the emerging field of compressed sensing [11, 12] and the reconstruction algorithms have been successfully applied to the channel estimation problem [13–15]. The Bayesian approach for sparse signal estimation, usually denominated sparse Bayesian learning (SBL) [16,17], is also found suitable for the channel estimation task [18,19]. In SBL a (possibly hierarchical) prior distribution is imposed on the weights of the multipath channel's impulse response which leads to algorithms that produce sparse solutions. Some steps have been taken in the direction of integrating sparse channel estimators in iterative receivers, e.g. in [20] where the channel's impulse response is modeled using a two-state Markov process; a channel weight can be in a 'high' or 'low' power state. We see SBL algorithms as natural candidates for integration in message-passing receivers due to their probabilistic formulation and it is still an area to elaborate upon.

### Project Definition

In this project we investigate the integration of sparse channel estimators based on SBL techniques within the framework of message-passing iterative wireless receivers for OFDM-based wireless communication systems. We seek answers to the following questions

- How can SBL-based channel estimation be embedded in an iterative message-passing OFDM receiver architecture?
- In which scenarios and to what extent does the aforementioned receiver benefit from the exploitation of the sparse channel structure?
- Is the proposed receiver architecture feasible for implementation in future wireless digital modems and how can the computational complexity be reduced?

### Outline of Report

The project report is structured in the following way: We start in Chapter 2 by presenting the problem we are solving with the iterative receiver architecture and the assumptions of the underlying system model. In Chapter 3 we briefly introduce the message-passing framework for Bayesian inference as well as the basic ideas of compressed sensing and SBL, which is utilized throughout the rest of the project. Based on the probabilistic description of the system model a current state-of-the-art iterative message-passing receiver is introduced in 4. In Chapter 5 we present how the channel estimation part of the receiver can be modified to exploit the sparse channel assumption using SBL, yielding a novel algorithm for jointly performing sparse channel estimation and decoding. The performance of the two receiver schemes is compared by the use of Monte Carlo simulations. The channel estimation in both receiver algorithms is of high computational complexity and in Chapter 7 simple techniques to lower the computational complexity that applies to both receivers are investigated. Finally, the conclusion of the project and further outlook is presented in Chapter 8.

CHAPTER 2

# System Model and Problem Definition

This chapter defines the problem we are investigating by introducing a probabilistic model of a simple point-to-point communication system. The model and its relation to orthogonal frequency division multiplexing (OFDM) are presented in Section 2.1. In Section 2.2 a simple model for the multipath channel is introduced followed by a brief discussion of optimal receiver design in Section 2.3.

## 2.1. System Model

In the following we describe the system model used in this project and highlight the made assumptions. We consider a generic communication system with one transmitter and one receiver. The baseband representation of the system is shown as a block diagram in Figure 2.1. The vector representation of the digital communication system is employed and the considered channels will be described by their complex baseband models. All signals are thus represented by finite-length column-vectors. The message to be conveyed is a binary vector $\mathbf{u} \in \{0,1\}^K$ sampled from a binary symmetric source. The message is encoded by a channel coder with rate $R$ and then randomly interleaved into a vector of coded bits $\mathbf{c} \in \{0,1\}^{\frac{K}{R}}$. We denote the coding and interleaving function $\mathcal{C}$, i.e. $\mathbf{c} = \mathcal{C}(\mathbf{u})$.

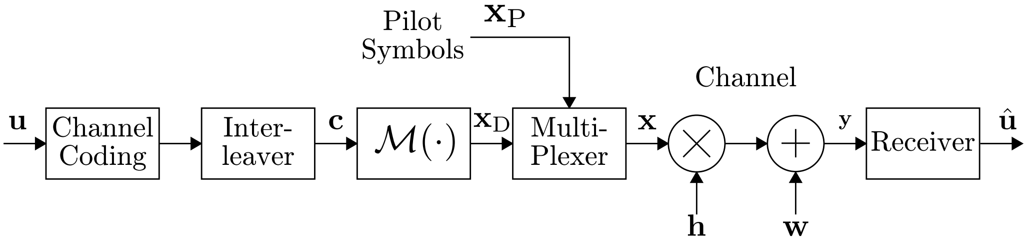


**Figure 2.1:** *Block diagram of the system model.*

The coded bits are divided into segments of $Q$ bits. Each segment is mapped to a complex modulation symbol $x_i \in \mathcal{S}_\mathrm{D}$ where $\mathcal{S}_\mathrm{D}$ denotes the discrete modulation alphabet of size $2^Q$. The mapping function $\mathcal{M}(\cdot)$ is one-to-one and onto. The number of data symbols is $N = \frac{K}{RQ}$. Without loss of generality it is assumed that $K$, $Q$ and $N$ are integers. Similarly $M$ complex pilot symbols are randomly selected from the pilot alphabet $\mathcal{S}_\mathrm{P}$. The data and pilot symbols are multiplexed to form the vector of transmitted symbols $\mathbf{x} \in \mathbb{C}^{N+M}$. The indices of the data and pilot symbols are denoted by the disjoint sets $\mathcal{D}$ and $\mathcal{P}$ respectively. With $\mathcal{D} \cup \mathcal{P} = [1 : M + N]$ the vector of data symbols is $\mathbf{x}_\mathcal{D} = (x_i | i \in \mathcal{D})^\mathrm{T}$ and the vector of pilot symbols is $\mathbf{x}_\mathcal{P} = (x_j | j \in \mathcal{P})^\mathrm{T}$.

The symbols $\mathbf{x}$ are transmitted through the channel with complex channel coefficients $\mathbf{h} \in \mathbb{C}^{N+M}$ and additive noise $\mathbf{w} \in \mathbb{C}^{N+M}$ to yield the observations

$$\mathbf{y} = \mathbf{h} \odot \mathbf{x} + \mathbf{w} \tag{2.1}$$

with $\mathbf{h} \odot \mathbf{x}$ denoting the entry-wise product of the two vectors $\mathbf{h}$ and $\mathbf{x}$. We will also make use of the equivalent expression

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w} \tag{2.2}$$

where $\mathbf{X} \in \mathbb{C}^{(M+N)\times(M+N)}$ is a diagonal matrix with the entries of $\mathbf{x}$ on its diagonal. The additive noise $\mathbf{w}$ is modeled as independent and identically distributed (iid) samples of a complex Gaussian random variable with precision parameter $\lambda$, i.e.

$$p(\mathbf{w}) = \mathrm{CN}\left(\mathbf{w}; 0, \lambda^{-1}\mathbf{I}\right) \tag{2.3}$$

where $\mathbf{I}$ is the $(M + N) \times (M + N)$ identity matrix.

The observation model in (2.1) could for example be used to model a time-varying, frequency-flat channel with each entry of $\mathbf{h}$ being the channel coefficient at the corresponding time instance, but its primary purpose in this project is to model a point-to-point OFDM system with no interference between the subcarriers. In this case $\mathbf{h}$ is a sampled version of the frequency-domain transfer function.

### 2.1.1 Application to OFDM

A block diagram of an OFDM communication system is shown in Figure 2.2. The correspondence with Figure 2.1 is depicted through the naming of the signal vectors.

The pilot and data symbols are multiplexed in frequency by mapping them to orthogonal subcarriers. The time domain signal is obtained through the inverse fast Fourier transform (FFT) of the frequency domain symbols. The cyclic prefix (CP) is inserted by copying the last time-domain samples to the beginning of the OFDM symbol. To avoid inter-symbol-interference

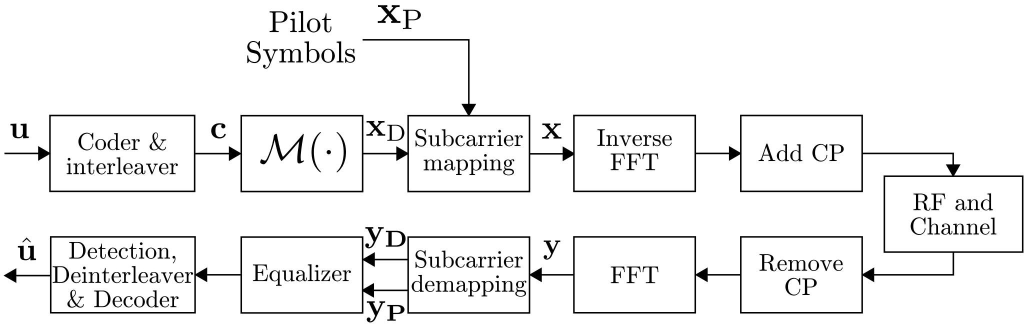


**Figure 2.2:** *Block diagram representation of OFDM communication system with traditional receiver.*

the length of the cyclic prefix must be longer than the impulse response of the channel. The main reason for transmitting the CP instead of using a 'silent' guard time is that, under the assumption that the channel impulse response is static during the transmission of one OFDM symbol, it turns the linear convolution of the channel into circular convolution. This reduces the complexity of the frequency domain equalization as the channel can be modeled as entry-wise multiplication after applying a finite-length Fourier transform on the time-domain samples. See [21, sec. 5.2.1] for this derivation.

## 2.2. Multipath Channel Model

To have a full probabilistic description of the system model we also need to introduce a probabilistic model for the frequency domain channel coefficients $\mathbf{h} = (h_1, \ldots, h_{N+M})^\mathrm{T}$.

The channel impulse response is assumed to be static during the transmission of one OFDM symbol. Employing the specular multipath channel model the impulse response can be written as [9]

$$g(\tau) = \sum_{p=1}^{P} \beta_p \delta(\tau - \tau_p) \tag{2.4}$$

where $P$ is the number of multipath components, $\beta_p$ is the complex gain of the $p$'th multipath component and $\tau_p$ is the corresponding delay. Taking the continuous-time Fourier transform yields the frequency domain representation

$$\mathcal{F}\left\{g(\tau)\right\}(f) = \sum_{p=1}^{P} \beta_p \exp(-j2\pi\tau_p f)\,. \tag{2.5}$$

Denoting the subcarrier spacing $\Delta f$ the elements of the channel coefficient vector are obtained as

$$h_i = \sum_{p=1}^{P} \beta_p \exp(-j2\pi\tau_p i\Delta f)\,,\ i \in [1 : N + M]. \tag{2.6}$$

The multipath gains are usually modelled as independent complex Gaussian random variables [9]

$$p(\beta_p) = \mathrm{CN}\left(\beta_p; 0, \sigma_p^2\right), \ p \in [1:P]. \tag{2.7}$$

where $\sigma_p^2$ is the average power of the $p$'th multipath component. This channel is a special case of the wide-sense-stationary uncorrelated scattering (WSSUS) [22] Rayleigh fading channel. The difference from the general case is that we do not allow $\beta_p$ and $\tau_p$ to evolve with time. It is thus assumed that the channel's impulse response is static during the transmission of one OFDM symbol, hence no Doppler frequency spread is incurred [9]. Given $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_P)^{\mathrm{T}}$ and $\mathbf{s} = (\sigma_1^2, \cdots, \sigma_P^2)^{\mathrm{T}}$ each $h_i$ in (2.6) is a linear combination of the Gaussian random variables $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_P)^{\mathrm{T}}$, hence each $h_i$ is also Gaussian distributed. Therefore we now make the assumption that $h_1, \ldots, h_{N+M}$ are jointly Gaussian, hence we can write the prior distribution as

$$p(\mathbf{h}) = \mathrm{CN}\left(\mathbf{h}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{h}}\right) \tag{2.8}$$

In Chapter 4 we design a receiver that exploits knowledge of the prior (2.8). As part of the performance evaluation in Chapter 6 we further discuss the correctness of the assumption (2.8) and compute the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{h}}$ for different channel models. We also evaluate the performance when the receiver assumes a covariance matrix that is different from the true one. In Chapter 5 a similar receiver is designed that exploit the structure of (2.6) under the assumption that $P$ is small, which we refer to as the sparse channel assumption.

## 2.3. Optimal Receiver Design

Notice that the block diagram in Figure 2.2 also depicts a traditional OFDM receiver as well as the transmitter. In such a receiver each block can be thought of as the reverse operation of a corresponding block in the transmitter. Each block is optimized with respect to some local performance measure, e.g. the equalizer could be the minimum mean-squared error estimator of the channel coefficients given the pilot observations $\mathbf{y}_{\mathcal{P}}$.

However, the ultimate goal of the receiver is not to estimate the channel coefficients but to produce an estimate $\hat{\mathbf{u}} \in \{0,1\}^K$ of the original message, ideally identical to $\mathbf{u}$. To minimize the risk of making an error the maximum a posteriori (MAP) criterion is employed [23]. For minimizing the bit-by-bit error rate the decision rule reads

$$\hat{u}_k = \underset{u_k \in \{0,1\}}{\operatorname{argmax}} \, p(u_k|\mathbf{y}) \tag{2.9}$$

for all $k \in [1:K]$. Alternatively the objective could be to minimize the block-error-rate based on $p(\mathbf{u}|\mathbf{y})$. Computing a closed form expression for the posterior in (2.9) using the joint probability density function (pdf) $p(\mathbf{y}, \mathbf{h}, \lambda, \mathbf{x}_{\mathcal{D}}, \mathbf{c}, \mathbf{u})$ is intractable and we therefore resort to suboptimal methods. In the following chapter different approximate Bayesian inference techniques are introduced, which allow us to approximate the posterior in (2.9) using an iterative algorithm. The framework thus provides a formal method for optimizing all operations of the receiver jointly, aiming to get as close as possible to the MAP criterion in order to minimize the bit-error-rate.

CHAPTER 3

# Theoretical Background

This chapter is a brief survey of the two main theoretical topics this project is built upon: In Section 3.1 we introduce Bayesian inference as message-passing on factor graphs and in Section 3.2 we introduce compressed sensing including its Bayesian approach, namely sparse Bayesian learning.

## 3.1. Bayesian Inference as Message-Passing on Factor Graphs

In Bayesian inference we seek to (approximately) compute the posterior of variables of interest given a set of observed data variables. In this section we first introduce factor graphs [3] as a tool for visualizing how complicated functions, e.g. a joint probability density function (pdf), factorizes as a product of simpler functions. Various Bayesian inference schemes can be formulated as message-passing algorithms on factor graphs. The focus here is on the unified framework [4] that combines belief propagation (BP) [3, 24] and the mean field (MF) [25, 26] approximation. How the expectation maximization (EM) algorithm [27] can be obtained as an instance of the MF algorithm is also mentioned. As we adopt and inference approach to derive message-passing receivers in later chapters, the purpose of the section is to introduce the message-passing framework and the notation that is used throughout the rest of the report. A thorough theoretical analysis of the different algorithms is not within the scope of this project.

### 3.1.1 Factor Graph

When designing inference algorithms that deal with a complicated global function of many variables it can often be exploited how the global function factorizes as a product of 'local' functions that depend on a subset of the variables [3].

In this project our global function is the joint pdf or probability mass function (pmf) of the observations and all the unknown variables of a probabilistic system model. When the observed values are fixed the global function is thus proportional to the posterior of the unobserved values given the observations. For example in the trivial case with a single unobserved variable $x$ and an observed variable $y$ with observed value $y'$, we have:

$$p(x, y = y') = p(x|y = y')p(y = y') \propto p(x|y = y') \tag{3.1}$$

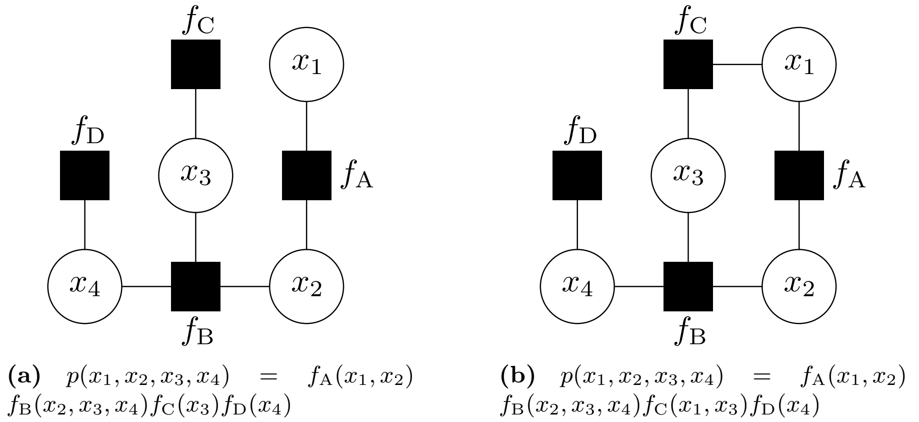
As the observations are fixed they can be considered as a specific parameterization of the local functions. In a more general setting suppose we have a probabilistic model for the set of unobservable random variables $\mathbf{x} = (x_i \,|\, i \in \mathcal{I})^{\mathrm{T}}$ where $\mathcal{I}$ is the set of variable indices and the joint distribution factorizes as

$$p(\mathbf{x}) = \frac{1}{z} \prod_{a \in \mathcal{A}} f_a(\mathbf{x}_a) \tag{3.2}$$

where $\mathcal{A}$ is the set of indices of local functions and $\mathbf{x}_a = (x_i | i \in \mathcal{N}(a))^{\mathrm{T}}$ is the vector of arguments of $f_a$ and $z$ is a normalization constant that ensures that the right-hand side of (3.2) is a proper distribution. The set $\mathcal{N}(a) \subseteq \mathcal{I}$ is the set of indices of the variables that are arguments of $f_a$ for $a \in \mathcal{A}$ and $\mathcal{N}(i) \subseteq \mathcal{A}$ is the set of indices of the local function $f_a$ that have variable $x_i$ for $i \in \mathcal{I}$ as an argument.

This factorization can be visualized using a factor graph. A factor graph is a bipartite, undirected graph that shows which local functions depends on which variables. It has a variable node for each variable $x_i$, a factor node for each local function $f_j$ and an edge connecting variable node $x_i$ to factor node $f_j$ if and only if $x_i$ is an argument of $f_j$ [3]. We label variable nodes with the same name as the corresponding variables and the factor nodes with the same name as the corresponding local functions.

An example of two factor graphs is shown in Figure 3.1, where we represent factor nodes by black squares and variable nodes by circles. A factor graph is a tree (or cycle-free) graph if any two nodes in the factor graph are connected by exactly one simple path, i.e. a path with no repeated nodes.



**(a)** $p(x_1, x_2, x_3, x_4) = f_{\mathrm{A}}(x_1, x_2) f_{\mathrm{B}}(x_2, x_3, x_4) f_{\mathrm{C}}(x_3) f_{\mathrm{D}}(x_4)$

**(b)** $p(x_1, x_2, x_3, x_4) = f_{\mathrm{A}}(x_1, x_2) f_{\mathrm{B}}(x_2, x_3, x_4) f_{\mathrm{C}}(x_1, x_3) f_{\mathrm{D}}(x_4)$

**Figure 3.1:** *Example of factor graph representation of two different global functions. The factor graph in (a) is a tree graph (no cycle) while the one in (b) is not cycle-free.*

### 3.1.2 Message-Passing Inference Framework

The goal of probabilistic inference is to compute the marginal distribution of one or more of the variables $x_i$ for $i \in \mathcal{I}$ in the probabilistic model (3.2). Several inference frameworks can be formulated as message-passing algorithms on factor graphs, in which neighboring nodes exchange information in form of messages.

If the factor graph is cycle-free we can apply the sum-product algorithm to obtain the exact marginal distributions [3]. The belief propagation (BP) [24] algorithm is a special case of the sum-product algorithm [3]. However, if the factor graph contains cycles we may apply an iterative version of the BP algorithm, sometimes referred to as loopy BP [28], but we are no longer guaranteed to obtain the exact marginal distributions [3] and for some models the algorithm will not converge [28]. We refer to the approximation of marginal distributions as 'beliefs'. In some cases applying BP results in very complicated message computations and the resulting algorithm may be infeasible for a practical implementation.

The mean field (MF) approximation is an alternative approximate inference algorithm. The global pdf is approximated by a simpler distribution $q(\mathbf{x})$, which is the product of the beliefs

$$q(\mathbf{x}) = \prod_{i \in \mathcal{I}} q(x_i) \tag{3.3}$$

Note that in this context single variables $x_i$ can be both vectors or scalars even though they are typeset with regular lowercase letters. If all $x_i$ for $i \in \mathcal{I}$ are scalar variables the approximating distribution is fully factorized, which is also known as the naïve mean field approximation [26]. The MF algorithm proceeds by iteratively minimizing the Kullback-Leibler divergence $\mathrm{KL}(q(\mathbf{x})||p(\mathbf{x}|\mathbf{y} = \mathbf{y}'))$ [29] from the approximating distribution to the true one [25].

Some of the pros and cons of the BP and MF according to [4] are as follows

- Mean Field approximation
  - Always admits a convergent implementation.
  - Has simple message-passing update rules, especially for conjugate-exponential models.
  - Is not compatible with hard constraints.
- Belief Propagation
  - Yields good approximation of the marginal distributions if the factor graph has no short cycles.
  - Is compatible with hard constraints.
  - May have high complexity, especially when applied to probabilistic models involving both discrete and continuous random variables.

This motivates the devisal of a unified message-passing algorithm that exploits the virtues of both algorithms and circumvents their drawbacks. Such an inference framework is proposed in [4]. Within this joint BP-MF inference scheme the factor nodes are split into two disjoint sets, the BP part $\mathcal{A}_{\text{BP}}$ and the MF part $\mathcal{A}_{\text{MF}}$ such that $\mathcal{A}_{\text{BP}} \cup \mathcal{A}_{\text{MF}} = \mathcal{A}$. The global function thus factorizes as

$$p(\mathbf{x}) = \prod_{a \in \mathcal{A}_{\text{MF}}} f_a(\mathbf{x}_a) \prod_{c \in \mathcal{A}_{\text{BP}}} f_c(\mathbf{x}_c) \tag{3.4}$$

While factor nodes belongs to either the BP part or the MF part, variable nodes can be arguments to factor nodes in both parts simultaneously. Provided that the factor graph fulfills certain technical conditions the following iterative update rules converges to a stationary point of the objective function [4]

$$m_{a\to i}^{\text{MF}}(x_i) = \exp\left(\sum_{\mathbf{x}_a \setminus x_i} \log f_a(\mathbf{x}_a) \prod_{j \in \mathcal{N}(a)\setminus i} n_{j\to a}(x_j)\right), \ \forall a \in \mathcal{A}_{\text{MF}}, i \in \mathcal{N}(a). \tag{3.5}$$

$$m_{a\to i}^{\text{BP}}(x_i) = d_a \sum_{\mathbf{x}_a \setminus x_i} f_a(\mathbf{x}_a) \prod_{j \in \mathcal{N}(a)\setminus i} n_{j\to a}(x_j), \ \forall a \in \mathcal{A}_{\text{BP}}, i \in \mathcal{N}(a). \tag{3.6}$$

$$n_{i\to a}(x_i) = e_i \prod_{c \in \mathcal{N}(i)\cap\mathcal{A}_{\text{MF}}} m_{c\to i}^{\text{MF}}(x_i) \prod_{c \in \mathcal{N}(i)\cap\mathcal{A}_{\text{BP}}\setminus a} m_{c\to i}^{\text{BP}}(x_i), \ \forall i \in \mathcal{I}, a \in \mathcal{N}(i) \tag{3.7}$$

where $d_a$ and $e_i$ are positive constants ensuring normalized beliefs. When a local function $f_a(\mathbf{x}_a)$ is a function of continuous random variables, the summations in (3.5) and (3.6) are replaced by integrals over the support of the random variables. The beliefs can be retrieved at any point in the iterative scheme as

$$q_i(x_i) = e_i \prod_{a \in \mathcal{N}(i)\cap\mathcal{A}_{\text{MF}}} m_{a\to i}^{\text{MF}}(x_i) \prod_{a \in \mathcal{N}(i)\cap\mathcal{A}_{\text{BP}}} m_{a\to i}^{\text{BP}}(x_i), \ \forall i \in \mathcal{I} \tag{3.8}$$

Note that messages from factor to variable nodes are computed as in BP if the factor node is in the BP part and as in MF if the factor node is in the MF part. Messages from variable to factor nodes are extrinsic values if the factor node is in the BP part and beliefs (i.e. a posteriori probability values) if the factor node is in the MF part. For the messages in (3.5) we will also use the equivalent notation

$$m_{a\to i}^{\text{MF}}(x_i) = \exp\Big(\langle \log f_a(\mathbf{x}_a)\rangle_{\prod_{j \in \mathcal{N}(a)\setminus i} n_{j\to a}(x_j)}\Big), \ \forall a \in \mathcal{A}_{\text{MF}}, i \in \mathcal{N}(a). \tag{3.9}$$

### Expectation Maximization (EM)

As shown in [4,30] the EM algorithm can be obtained as an instance of MF by restricting the beliefs $q_i(x_i)$ for $i \in \mathcal{I}_{\text{EM}} \subseteq \mathcal{I}$ to a Dirac delta function at the mode of the corresponding belief obtained with the MF algorithm. The message computation rules are the same, except the messages from variable to factor nodes (3.7) are replaced by

$$n^{\mathrm{EM}}_{i\to a}(x_i) = \delta(x_i - \bar{x}_i), \text{ with } \bar{x}_i = \underset{x}{\operatorname{argmax}} \left( \prod_{a\in\mathcal{N}(i)} m_{a\to i}(x_i) \right) \tag{3.10}$$

for all $i \in \mathcal{I}_{\mathrm{EM}}$ and all $a \in \mathcal{N}(i)$. The reason for using EM for some variable nodes is that it may lead to messages that are simpler to compute. However, all information of the 'original' belief except its mode is discarded.

## 3.2. Compressed Sensing

To exploit sparsity of the channel in the design of message-passing receivers we employ the framework of compressed sensing and its Bayesian approach, sparse Bayesian learning. In the following we introduce the notation, concepts and basic results in compressed sensing that are utilized in the project.

In compressed sensing and sparse decomposition it is assumed that a signal of interest $\mathbf{z} \in \mathbb{C}^K$ has a sparse representation $\boldsymbol{\alpha} \in \mathbb{C}^L$ in some (possibly overcomplete) basis $\boldsymbol{\Theta} \in \mathbb{C}^{K\times L}$, i.e. $L \geq K$, $\operatorname{rank}(\boldsymbol{\Theta}) = K$. We say that $\mathbf{z}$ is $P$-sparse in $\boldsymbol{\Theta}$, if there exists a vector $\boldsymbol{\alpha}$ with only $P \ll L$ or fewer non-zero components such that

$$\mathbf{z} = \boldsymbol{\Theta}\boldsymbol{\alpha} \tag{3.11}$$

In many practical cases the vector $\boldsymbol{\alpha}$ is only approximately sparse, meaning that it is well approximated by a sparse vector. The compressibility of a signal can be quantified by the error incurred by approximating it by a sparse vector [31]

$$\min_{\mathbf{x}'\in\left\{\mathbf{x}' \,|\, ||\mathbf{x}'||_0 \leq P\right\}} ||\mathbf{x} - \mathbf{x}'||_2 \tag{3.12}$$

where the $||\cdot||_0$ denotes the number of nonzero components of a vector. For example $\boldsymbol{\Theta}$ could be a Fourier or wavelet basis [12] with $(K = L)$, in which case the weights $\boldsymbol{\alpha}$ can easily be determined by $\boldsymbol{\alpha} = \boldsymbol{\Theta}^{-1}\mathbf{z}$. However, one of the ramifications of compressed sensing is that in this case we do not need to observe $\mathbf{z}$ to be able to find its sparse representation $\boldsymbol{\alpha}$. Instead we observe a vector $\mathbf{v}$ containing $N < K$ linear combinations of the entries of $\mathbf{z}$:

$$\mathbf{v} = \mathbf{B}\mathbf{z} = \mathbf{B}\boldsymbol{\Theta}\boldsymbol{\alpha} \tag{3.13}$$

If the matrix $\mathbf{A} = \mathbf{B}\boldsymbol{\Theta} \in \mathbb{C}^{N\times L}$ has certain properties, it is possible to pose guarantees on the ability to recover any $P$-sparse $\boldsymbol{\alpha}$ from observing $\mathbf{v}$ [31]. In the literature $\mathbf{A}$ is often referred to as the measurement or sensing matrix [31,32] and $\boldsymbol{\Theta}$ may be referred to as the dictionary matrix [31]. In this project the distinction between $\boldsymbol{\Theta}$ and $\mathbf{A}$ is not important and throughout this report we therefore refer to $\mathbf{A}$ as the dictionary, i.e. the dictionary is the linear mapping from the sparse representation $\boldsymbol{\alpha}$ to the observations $\mathbf{v}$ as in [17, 33]. In the standard Compressed Sensing (CS) framework the measurements are non-adaptive, meaning that the matrix $\mathbf{A}$ is fixed in advance [31].

### 3.2.1 Conditions for Reconstruction

Since $N \ll L$, (3.13) represents the compression of signal the signal $\boldsymbol{\alpha}$ into $\mathbf{v}$, which may indicate loss of information. In order to uniquely reconstruct a $P$-sparse signal $\boldsymbol{\alpha}$ from $\mathbf{v}$, we must be able to distinguish between any $P$-sparse signals obtained by applying the mapping $\mathbf{A}$. Guarantees on the ability to recover a sparse signal are often based on the restricted isometry property (RIP) [34].

**Restricted Isometry Property** [34] The matrix $\mathbf{A}$ satisfies RIP of order $P$ if there exists a restricted isometry constant $\delta_P \in (0, 1)$ such that

$$(1 - \delta_P) \, ||\boldsymbol{\alpha}||_2^2 \leq ||\mathbf{A}\boldsymbol{\alpha}||_2^2 \leq (1 + \delta_P) \, ||\boldsymbol{\alpha}||_2^2 \tag{3.14}$$

for all $P$-sparse $\boldsymbol{\alpha}$.

When $\mathbf{A}$ satisfies the RIP of order $2P$, one interpretation of (3.14) is that the distance between any pair of $P$-sparse vectors (note that their difference is thus $2P$-sparse) is approximately preserved under the linear map $\mathbf{A}$. The RIP guarantees reconstruction up to a certain sparsity level, but it is in general difficult to verify if $\mathbf{A}$ satisfies RIP [31]. An easier computable property to provide reconstruction guarantees is the coherence of $\mathbf{A}$ [32].

**Coherence** [31] The coherence of a $N \times L$ matrix $\mathbf{A}$, $\mu(\mathbf{A})$ is the largest absolute inner product between two columns $\mathbf{a}_i$, $\mathbf{a}_j$ of $\mathbf{A}$

$$\mu(\mathbf{A}) = \max_{1 \leq i < j \leq L} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{||\mathbf{a}_i||_2 \, ||\mathbf{a}_j||_2} \tag{3.15}$$

It can be shown $\mu(\mathbf{A}) \in \left[\sqrt{\frac{L-N}{N(L-1)}}, 1\right]$ [32]. In some cases there is a direct relation between the coherence and RIP.

**RIP and Coherence** [35] If $\mathbf{A}$ has unit-norm columns and coherence $\mu$, then $\mathbf{A}$ satisfies RIP of order $P$ with $\delta_P = (P - 1)\mu$ for all $P < \frac{1}{\mu}$.

The coherence can also be used to guarantee the uniqueness of the sparse representation of the signal.

**Uniqueness of $\boldsymbol{\alpha}$** [36] If $P < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right)$, then for each measurement vector $\mathbf{v} \in \mathbb{R}^N$ there exists at most one $P$-sparse signal $\boldsymbol{\alpha}$ such that $\mathbf{v} = \mathbf{A}\boldsymbol{\alpha}$.

We do not apply the above results directly to the design of the dictionary, but they aid the understanding of the CS problem and provide some intuition on the desired properties of the dictionary.

### 3.2.2 Reconstruction Algorithms

We now focus on the different algorithms for solving the CS reconstruction problem: given the dictionary matrix $\mathbf{A} \in \mathbb{C}^{N \times L}$ and the measurement vector $\mathbf{v} \in \mathbb{C}^N$ find a sparse vector $\boldsymbol{\alpha} \in \mathbb{C}^L$ such that $\mathbf{v} = \mathbf{A}\boldsymbol{\alpha}$.

The simplest way of posing a recovery algorithm is to find the sparsest solution that explains the observations, i.e.

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ ||\boldsymbol{\alpha}||_0 \ \text{subject to } \mathbf{v} = \mathbf{A}\boldsymbol{\alpha} \tag{3.16}$$

However, this algorithm requires an exhaustive search of $\boldsymbol{\alpha}$, i.e. for every $P = [1 : L]$ we need to check if $\mathbf{v}$ is in the span of all combinations of $P$ columns of $\mathbf{A}$ [32].

We can relax the $\ell_0$ 'norm' assumption and replace it with the $\ell_1$ norm and thus obtain a convex optimization problem. Furthermore, when the observations are impaired by additive noise, i.e., $\mathbf{y} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{w}$ the equality constraint is relaxed with an inequality.

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ ||\boldsymbol{\alpha}||_1 \ \text{subject to } ||\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}||_2^2 \leq \epsilon \tag{3.17}$$

where $\epsilon \geq ||\mathbf{w}||_2^2$ is a bound on the noise. Now, using Lagrangian relaxation of the constraints we can obtain the basis pursuit denoising method [37]

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ ||\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}||_2^2 + \mu \, ||\boldsymbol{\alpha}||_1 \tag{3.18}$$

Alternative algorithms are greedy algorithms such as CoSaMP [38] and OMP [39]. The general approach of these algorithms is to use the correlation between the columns of the dictionary $\mathbf{A}$ and the observation vector $\mathbf{y}$ to (iteratively) find the columns of $\mathbf{A}$ that contribute most to $\mathbf{y}$ [32].

In this project we focus on the Bayesian approach to the compressed sensing reconstruction problem also known as Sparse Bayesian Learning (SBL) [16, 17, 40], because, besides its good performance, it is suitable for deriving a sparse channel estimator that is naturally embedded in an iterative receiver obtained using Bayesian inference. In contrast, applying a greedy pursuit algorithm to estimate the sparse channel would require a heuristic way of embedding it into the receiver scheme with the risk of tainting the convergent properties of the inference schemes.

### 3.2.3 SBL and Hierarchical Prior Modeling

In SBL we generally deal with the signal model $\mathbf{y} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{w}$, where $\mathbf{w} \sim \text{CN}(\mathbf{w}; \mathbf{0}, \lambda^{-1}\mathbf{I})$. Notice that we consider the case where $\mathbf{y}, \mathbf{A}, \boldsymbol{\alpha}$ and $\mathbf{w}$ are complex-valued although the original literature [16,17,40] considers only the real-valued case. The complex case is investigated in [33]. For simplicity of the following discussion the noise precision $\lambda$ is assumed known, although it is modeled as an unknown parameter when deriving the novel receiver in Section 5.2. Thus $p(\mathbf{y}|\boldsymbol{\alpha}) = \mathrm{CN}\left(\mathbf{y}; \mathbf{A}\boldsymbol{\alpha}, \lambda^{-1}\mathbf{I}\right)$ and, by selecting a prior distribution $p(\boldsymbol{\alpha})$, the maximum a posteriori probability (MAP) estimate of $\boldsymbol{\alpha}$ is obtained as

$$\hat{\boldsymbol{\alpha}}_{\mathrm{MAP}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\, p(\boldsymbol{\alpha}|\mathbf{y}) = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}}\, \frac{p(\mathbf{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\mathbf{y})} \tag{3.19}$$

or equivalently using the negative log-probability

$$\hat{\boldsymbol{\alpha}}_{\mathrm{MAP}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\ ||\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}||_2^2 + \lambda^{-1}Q(\boldsymbol{\alpha}) \tag{3.20}$$

where the term $Q(\boldsymbol{\alpha}) = -\log(p(\boldsymbol{\alpha})) + \text{const.}$ is a penalization term that enforces sparsity of $\hat{\boldsymbol{\alpha}}_{\mathrm{MAP}}$ [33]. We can select different prior distributions for $\boldsymbol{\alpha}$ to induce sparse solutions. For example choosing $p(\boldsymbol{\alpha})$ as the product of $L$ Laplace pdfs yields the $\ell_1$ norm penalty term as in basis pursuit denoising (3.18) [41]. Instead of working directly with the prior $p(\boldsymbol{\alpha})$ a hierarchical representation of the prior is often employed in order to yield more computationally tractable inference algorithms. In SBL the prior of the weights $\boldsymbol{\alpha}$ conditioned on the hyperparameters $\boldsymbol{\gamma}$, $p(\boldsymbol{\alpha}|\boldsymbol{\gamma})$, is usually governed by a Gaussian pdf, e.g. [16, 17, 33, 41–43].

In the following we present the three-layer hierarchical prior model proposed in [33]. This model is able to encompass several other models proposed in the literature through specific choices of parameters and by treating some of the random variables as known quantities. Here the joint pdf of the weights $\boldsymbol{\alpha}$, and the hyperparameters $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ factorizes as

$$p(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\eta}) = p(\boldsymbol{\alpha}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}|\boldsymbol{\eta})p(\boldsymbol{\eta}) \tag{3.21}$$

where

$$p(\boldsymbol{\alpha}|\boldsymbol{\gamma}) = \mathrm{CN}\left(\boldsymbol{\alpha}; \mathbf{0}, \boldsymbol{\Gamma}\right) = \prod_{\ell=1}^{L} \mathrm{CN}\left(\alpha_\ell; 0, \gamma_\ell\right) \tag{3.22}$$

$$p(\boldsymbol{\gamma}|\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\gamma_\ell; \varepsilon, \eta_\ell\right) \tag{3.23}$$

$$p(\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\gamma_\ell; c, d\right) \tag{3.24}$$

where $\boldsymbol{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma})$ is the diagonal matrix with the entries of $\boldsymbol{\gamma}$ on the diagonal. We can also obtain two-layer hierarchical models from this description by treating $\boldsymbol{\eta}$ as a fixed parameter vector. For example, using the two-layer model, we can obtain the $\ell_1$ penalty term for complex $\boldsymbol{\alpha}$ by setting $\eta_\ell = \eta$, $\forall\, \ell$, and choosing $\varepsilon = \frac{3}{2}$ [33]. The well-known Relevance Vector Machine (RVM) can also be derived from the two-layer model by using a 'flat' prior on $\boldsymbol{\gamma}$ as shown in [17].

For the two-layer model it can be shown that the marginal prior of $\boldsymbol{\alpha}$ is given by [33]

$$p(\boldsymbol{\alpha}; \varepsilon, \boldsymbol{\eta}) = \prod_{\ell=1}^{L} p(\alpha_\ell; \varepsilon, \eta_\ell) \tag{3.25}$$

with

$$p(\alpha_\ell; \varepsilon, \eta_\ell) = \frac{2}{\pi\Gamma(\varepsilon)} \eta_\ell^{\frac{\varepsilon+1}{2}} |\alpha_\ell|^{\varepsilon-1} K_{\varepsilon-1}(2\sqrt{\eta_\ell}|\alpha_\ell|) \tag{3.26}$$

where $K_v(\cdot)$ is the modified Bessel function of the second kind and order $v$ and $\Gamma(\cdot)$ is the gamma function.

Similarly, for the three-layer model, marginalizing over $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ yields [33]

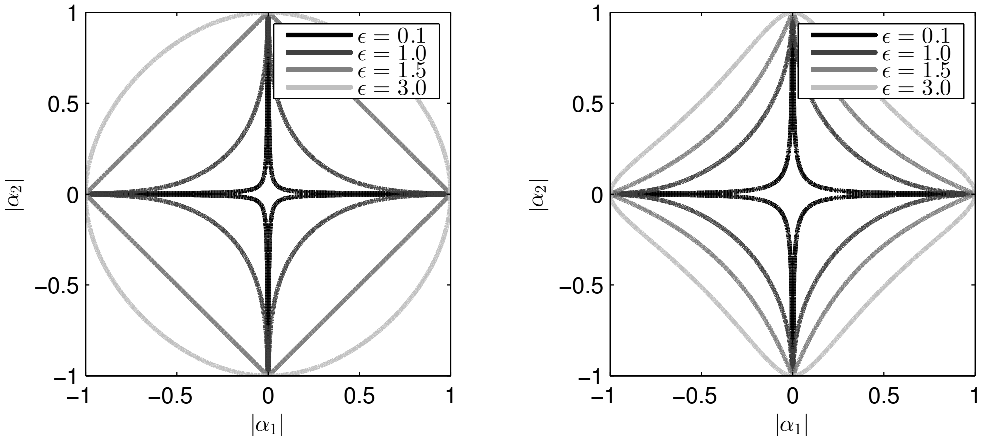
$$p(\boldsymbol{\alpha}; \varepsilon, c, d) = \prod_{\ell=1}^{L} p(\alpha_\ell; \varepsilon, c, d) \tag{3.27}$$

with

$$p(\alpha_\ell; \varepsilon, c, d) = \left(\frac{1}{\pi d}\right) \frac{\Gamma(\varepsilon + c)\Gamma(c+1)}{\Gamma(\varepsilon)\Gamma(c)} \left(\frac{|\alpha_\ell|^2}{d}\right)^{\varepsilon-1} U(\varepsilon + c; \varepsilon; \frac{|\alpha_\ell|^2}{d}) \tag{3.28}$$

where $U(\cdot;\cdot;\cdot)$ is the confluent hypergeometric function [44].

To get an idea of how the choice of parameters affects the penalty term of the weights, we plot in Figure 3.2 contour lines for both the two-layer and three-layer prior distributions for $L = 2$.



**(a)** **(b)**

**Figure 3.2:** *Contour plot of (a) the marginalized two-layer prior distribution $p(\alpha_1, \alpha_2; \varepsilon, \eta_1, \eta_2)$ with $\eta_1 = \eta_2 = 1$ and (b) the marginalized three-layer prior distribution $p(\alpha_1, \alpha_2; \varepsilon, c, d)$ with $c = 1$ and $d = 0.1$.*

In all cases the prior pdf is monotonically increasing for $\alpha_1, \alpha_2 \to 0$ and the contour plots show how the probability mass gets more centered around the axes as the value of $\varepsilon$ decreases. In Figure 3.2a, the contour for the $\ell_1$ penalty term ($\varepsilon = 1.5$) is also recognizable. By selecting $\varepsilon < 1.5$ we obtain prior distributions that are more sparsity-inducing than the Bayesian version of basis pursuit denoising (3.18) and it is thus expected that the algorithms derived from this model will produce sparser solutions. This hypothesis is supported by the numerical results in [33].

### 3.2.4 Iterative Inference Scheme

With the chosen prior distribution we could in principle find the MAP estimate of $\boldsymbol{\alpha}$ by solving (3.20). This is in general not straightforward when the choice of prior distribution leads to a non-convex penalty term [45]. Therefore, an iterative Bayesian inference scheme is often applied. Popular choices are the EM algorithm [27] used in [17, 33, 41, 43] and the 'fast inference scheme' originally devised for Fast RVM [46] but later used in, e.g., [33, 43, 47]. The MF inference scheme is another method that has also been applied for sparse estimation using hierarchical prior modelling [18].

In the derivation of the receiver algorithm with sparse channel estimation in Chapter 5 we employ the three-layer hierarchical prior model for the delay domain weights and apply the MF inference scheme formulated as message-passing on a factor graph. The three-layer model is used here due to its generality. As the same prior model is used for pilot-assisted sparse channel estimation in [18], the devised receiver can be considered as a data-aided extension of this one. Since we are targeting low-complexity algorithms and previous work [33, 48] indicates that the two-layer version exhibits faster convergence than its three-layer counterpart, we use a two-layer version in the numerical results in Chapter 6.

CHAPTER 4

# Message-Passing Receiver Design

Based on the system model introduced in Chapter 2, a message-passing iterative receiver structure is introduced in the following. The receiver reflects current state-of-the-art within the field of message-passing algorithms for channel estimation and decoding. The derivations and discussions this chapter rely on contemporary literature, especially the work in [4, 7]. The purpose of introducing this receiver algorithm is to provide a reference and a basis for the development of receiver structures that exploit the channel sparsity. The receiver algorithm presented here is thus included as a reference in the simulations in Chapter 6.

The factorization of the probabilistic system model and its factor graph representation are presented in Section 4.1 and 4.2, respectively. After discussing the choice of inference framework in Section 4.3, we derive the messages of the iterative receiver algorithm in Section 4.4. The scheduling of message computations defining the iterative algorithm is presented in Section 4.5.

## 4.1. Probabilistic Model

Using the system model described in Chapter 2 we can express the probabilistic system function as the joint pdf of the observations and all the unknown variables. The inherent conditional independences of the system model allow the following factorization of the system function

$$p(\mathbf{y}, \mathbf{x}_{\mathcal{D}}, \mathbf{h}, \lambda, \mathbf{c}, \mathbf{u}) = p(\mathbf{y}|\mathbf{x}_{\mathcal{D}}, \mathbf{h}, \lambda)p(\mathbf{h})p(\lambda)p(\mathbf{x}_{\mathcal{D}}|\mathbf{c})p(\mathbf{c}|\mathbf{u})p(\mathbf{u}) \tag{4.1}$$

Notice that the pilot symbols $\mathbf{x}_{\mathcal{P}}$ are not included in the probabilistic model as they are known a priori. The following observations have been used.

- Since $\mathbf{x}_{\mathcal{D}}$ is a deterministic function of $\mathbf{c}$ which in turn is a deterministic function of $\mathbf{u}$, $\mathbf{x}$ is independent of $\mathbf{u}$ given $\mathbf{c}$.
- The channel weights $\mathbf{h}$, the data symbols $\mathbf{x}_{\mathcal{D}}$ and the noise process $\mathbf{w}$ are independent.

- Given $\mathbf{x}_\mathcal{D}$ and $\mathbf{h}$ the uncertainty of the observations $\mathbf{y}$ in (2.2) is due to the noise $\mathbf{w}$ whose statistics are parameterized by the noise precision $\lambda$.

Further factorization is possible as the noise samples are iid the information bits are iid and the data symbols are independent given the coded bits:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}_\mathcal{D}, \mathbf{h}, \lambda) &= \mathrm{CN}\left(\mathbf{y}; \mathbf{X}\mathbf{h}, \lambda^{-1}\mathbf{I}\right) && (4.2)\\ &= \prod_{i\in\mathcal{D}} p(y_i|x_i, h_i, \lambda) \prod_{j\in\mathcal{P}} p(y_j|h_j, \lambda) && (4.3)\\ &= \prod_{i\in\mathcal{D}\cup\mathcal{P}} \mathrm{CN}(y_i; x_i h_i, \lambda^{-1}) && (4.4)\\ p(\mathbf{u}) &= \prod_{k=1}^{K} p(u_k) && (4.5)\\ p(\mathbf{x}_\mathcal{D}|\mathbf{c}) &= \prod_{n=1}^{N} p(x_{i_n}|\mathbf{c}^{(n)}), && (4.6) \end{aligned}$$

where $\mathbf{c}^{(n)}$ is the $n$th length Q segment of $\mathbf{c}$ and $i_n \in \mathcal{D}$ is the corresponding index into $\mathbf{x}$. The functions $p(x_{i_n}|\mathbf{c}^{(n)})$ and $p(\mathbf{c}|\mathbf{u})$ mimic hard constraints. The functions are thus given by degenerate distributions, i.e.

$$p(x_{i_n}|\mathbf{c}^{(n)}) = \begin{cases} 1 & \text{if } x_{i_n} = \mathcal{M}(\mathbf{c}^{(n)}) \\ 0 & \text{otherwise} \end{cases}, \; n \in [1:N] \qquad (4.7)$$

$$p(\mathbf{c}|\mathbf{u}) = \begin{cases} 1 & \text{if } \mathbf{c} = \mathcal{C}(\mathbf{u}) \\ 0 & \text{otherwise} \end{cases} \qquad (4.8)$$

The prior distribution $p(u_k)$ will in most cases be the uniform pdf, i.e. $p(u_k) = \frac{1}{2}$ on its domain. The noise precision prior $p(\lambda)$ can conveniently be chosen as the gamma pdf

$$p(\lambda) = \mathrm{Ga}(\lambda; a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} \exp(-b\lambda) \qquad (4.9)$$

as the gamma pdf is conjugate prior for a Gaussian distribution with known mean and unknown precision. The channel weight prior $p(\mathbf{h})$ is obtained from the knowledge or assumptions on the channel. As discussed in section 2.2, choosing a Gaussian pdf is appropriate, i.e.

$$p(\mathbf{h}) = \mathrm{CN}(\mathbf{h}; \mathbf{0}, \mathbf{\Sigma}_\mathbf{h}) \qquad (4.10)$$

## 4.2. Factor Graph Representation

Using the factorisation of the probabilistic model we can now visualize the dependencies between the variables in a factor graph. To ease the notation in the following derivations, we define the following local functions pertaining to three categories.

**Observation Model**

$$\begin{aligned} f_{\mathrm{D}_i}(h_i, x_i, \lambda) &= p(y_i|x_i, h_i, \lambda) && (4.11)\\ &= \mathrm{CN}(y; h_i x_i, \lambda^{-1}),\ i \in \mathcal{D} && (4.12)\\ f_{\mathrm{P}_j}(h_j, \lambda) &= p(y_j|x_j, h_j, \lambda) && (4.13)\\ &= \mathrm{CN}(y; h_j x_j, \lambda^{-1}),\ j \in \mathcal{P} && (4.14) \end{aligned}$$

**Channel and Noise Prior**

$$\begin{aligned} f_{\mathrm{H}}(\mathbf{h}) &= p(\mathbf{h}) && (4.15)\\ &= \mathrm{CN}(\mathbf{h}; \mathbf{0}, \mathbf{\Sigma}_{\mathbf{h}}) && (4.16)\\ f_{\lambda}(\lambda) &= p(\lambda) = \mathrm{Ga}(\lambda; a, b) && (4.17) \end{aligned}$$

**Coding and Modulation Constraints**

$$\begin{aligned} f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)}) &= p(x_{i_n}|\mathbf{c}^{(n)}),\ n \in [1:N] && (4.18)\\ f_{\mathrm{C}}(\mathbf{c}, \mathbf{u}) &= p(\mathbf{c}|\mathbf{u}) && (4.19)\\ f_{\mathrm{U}}(u_k) &= p(u_k),\ k \in [1:K] && (4.20) \end{aligned}$$

We thus have the following factorization of the system function

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}_{\mathcal{D}}, \mathbf{h}, \lambda, \mathbf{c}, \mathbf{u}) = &\prod_{i\in\mathcal{D}} f_{\mathrm{D}_i}(h_i, x_i, \lambda) \prod_{j\in\mathcal{D}} f_{\mathrm{P}_j}(h_j, \lambda)\\ &\cdot f_{\mathrm{H}}(\mathbf{h}) f_{\lambda}(\lambda) f_{\mathrm{C}}(\mathbf{c}, \mathbf{u}) \cdot \prod_{n=1}^{N} f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)}) \prod_{k=1}^{K} f_{\mathrm{U}_k}(u_k) \quad (4.21) \end{aligned}$$

The observation vector $\mathbf{y}$ is not an input of any of the local functions and is therefore not present on the right-hand side. One can think of the observations as creating an instance of a family of graphical models parameterized by the observation vector $\mathbf{y}$. The factor graph representation of (4.21) is shown in Figure 4.1. The channel coding and interleaving subgraph is represented by the box labeled $f_{\mathrm{C}}$. In Subection 4.4.4 we show the factor graph representation of a convolutional code and outline how decoding can be implemented by the application of the BP algorithm.

Given an instance (parameterized by the observations $\mathbf{y}$) of the factor graph we apply an inference framework to find the beliefs:

$$q(u_k) \approx p(u_k|\mathbf{y})\ \forall\, k \in [1:K]. \tag{4.22}$$

Notice that the factor graph representation of the system function is not unique. For example we could choose some variable nodes as vectors instead of using the individual entries as shown here. Furthermore, instead of using the fully factorized form in (4.21) in some cases it can be beneficial to eliminate factor nodes by joining several nodes into one, i.e. using the joint

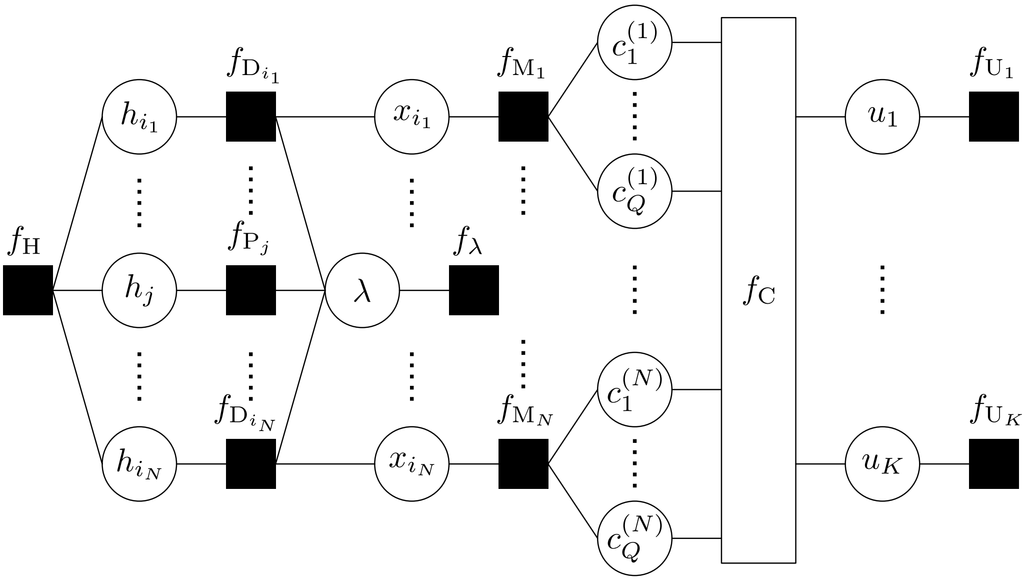


**Figure 4.1:** *Factor graph representation of the factorized probabilistic system model.*

pdf instead of the factorized form. This can be done to eliminate cycles in the graph, but it may also lead to intractable or computationally heavy algorithms. The outcome depends on the choice of inference framework, thus there are many degrees of freedom in choosing a combination of factor graph representation and inference framework.

## 4.3. Choice of Inference Framework

The choice of inference framework highly affects the performance of the resulting receiver algorithm. A study of the feasibility and bit error rate (BER) performance of different inference schemes applied to the presented system model and factor graph is presented in [7]. The system model is the same except that in [7] the noise precision $\lambda$ is assumed known and is thus not included in the estimation. The basis of the study is the combined BP-MF framework from which the BP and MF frameworks can be obtained as special cases. We make use of the following conclusions [7]. Using BP for all nodes in the factor graph becomes computationally intractable because the message $m_{f_{\mathrm{H}} \to h_i}(h_i)$ becomes a Gaussian mixture with $2^{Q(N-1)}$ and $2^{QN}$ for $i \in \mathcal{D}$ and $i \in \mathcal{P}$ respectively [7]. Therefore different approximations are introduced. An approximation that exploits the correlation between the channel coefficients $\mathbf{h}$ is desired. The following approaches are assessed in [7]:

*a)* BP with Gaussian approximation: Each message $m_{f_{\mathrm{D}_i} \to h_i}(h_i)$ for $i \in \mathcal{D}$ is approximated by a single Gaussian pdf instead of a Gaussian mixture with $2^Q$ components.

*b)* Expectation propagation (EP): The belief of each channel coefficient

$h_i$ for $i \in \mathcal{D}$ is restricted to a Gaussian pdf.

*c*) BP-MF: The factor graph is split into the MF and BP part. The factor nodes $f_{\mathrm{D}_i}$ for $i \in \mathcal{D}$ are in the MF part and the remaining nodes are in the BP part.

*d*) BP-EM: The splitting of the graph is the same as in BP-MF, but the beliefs of $h_i$ for $i \in [1 : M + N]$ are restricted to be Dirac delta functions.

In the considered scenario all algorithms except BP with Gaussian approximation shows BER performance within 0.5 dB compared to a reference estimator based on BP with known channel coefficients. The scheme using BP with Gaussian approximation performs about 2 dB worse. The EP-based scheme becomes unstable and needs the use of an heuristic approach to dampen the updates of the beliefs of $h_i$. The BP-MF and BP-EM based approaches shows the same performance. However, there is no reduction in computational complexity by choosing the BP-EM approach even though only point estimates of the channel coefficients are used. Therefore we will focus on the BP-MF based scheme in the following.

## 4.4. Computation of Messages

Unlike [7] we also consider a scenario in which the noise precision is unknown. Therefore we also include the factor nodes corresponding to pilot symbols $f_{\mathrm{P}_j}$ in the MF part of factor graph. Denoting the set of all factor nodes $\mathcal{A}$, we thus have

$$\mathcal{A}_{\mathrm{MF}} = \{f_{\mathrm{D}_i} | i \in \mathcal{D}\} \cup \left\{f_{\mathrm{P}_j} | j \in \mathcal{P}\right\} \tag{4.23}$$

$$\mathcal{A}_{\mathrm{BP}} = \{\mathcal{A} \backslash \mathcal{A}_{\mathrm{MF}}\} \tag{4.24}$$

$$= \{f_{\mathrm{H}}\} \cup \{f_{\lambda}\} \cup \{f_{\mathrm{C}}\} \cup \{f_{\mathrm{M}_n} | n \in [1 : N]\} \cup \{f_{\mathrm{U}_k} | k \in [1 : K]\} \tag{4.25}$$

### 4.4.1 Channel Coefficients

In the following we compute the beliefs of the channel coefficients $\mathbf{h}$. Given the beliefs $q(\lambda)$ and $q(x_i)$, for all $i \in \mathcal{D}$, we can compute

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i} \to h_i}(h_i) \propto \exp\left( \int_0^\infty \sum_{x_i \in \mathcal{S}_\mathcal{D}} \log f_{\mathrm{D}_i}(h_i, x_i, \lambda) n_{x_i \to f_{\mathrm{D}_i}}(x_i) n_{\lambda \to f_{\mathrm{D}_i}}(\lambda) d\lambda \right) \tag{4.26}$$

$$= \exp\Big( \langle \log f_{\mathrm{D}_i}(h_i, x_i, \lambda) \rangle_{q(x_i)\, q(\lambda)} \Big) \tag{4.27}$$

$$= \exp\left( \left\langle \log\left(\frac{\lambda}{\pi}\right) - \lambda |y_i - h_i x_i|^2 \right\rangle_{q(x_i)\, q(\lambda)} \right) \tag{4.28}$$

$$\propto \exp\Big( \big\langle -\lambda \left( |y_i|^2 + |h_i|^2 |x_i|^2 - 2\,\mathrm{Re}\left( y_i^* h_i x_i \right) \right) \big\rangle_{q(x_i)\, q(\lambda)} \Big) \tag{4.29}$$

$$\propto \mathrm{CN}\left( h_i; \mu_{h_i^o}, \sigma^2_{h_i^o} \right), \tag{4.30}$$

for all $i \in \mathcal{D}$, where

$$\mu_{h_i^o} = \frac{y_i \langle x_i \rangle^*_{q(x_i)}}{\langle |x_i|^2 \rangle_{q(x_i)}}, \; \sigma^2_{h_i^o} = \frac{1}{\langle \lambda \rangle_{q(\lambda)} \langle |x_i|^2 \rangle_{q(x_i)}} \tag{4.31}$$

Similarly for the factor nodes corresponding to pilot symbols

$$m^{\mathrm{MF}}_{f_{\mathrm{P}_j} \to h_j}(h_j) \propto \exp\Big( \langle \log f_{\mathrm{P}_j}(h_j, \lambda) \rangle_{q(\lambda)} \Big) \tag{4.32}$$

$$= \exp\left( \left\langle \log\left(\frac{\lambda}{\pi}\right) - \lambda |y_j - h_j x_j|^2 \right\rangle_{q(\lambda)} \right) \tag{4.33}$$

$$\propto \exp\Big( \big\langle -\lambda \left( |y_j|^2 + |h_j|^2 |x_j|^2 - 2\,\mathrm{Re}\left( y_j^* h_j x_j \right) \right) \big\rangle_{q(\lambda)} \Big) \tag{4.34}$$

$$\propto \mathrm{CN}\left( h_j; \mu_{h_j^o}, \sigma^2_{h_j^o} \right) \tag{4.35}$$

for all $j \in \mathcal{P}$, where

$$\mu_{h_j^o} = \frac{y_j x_j^*}{|x_j|^2}, \; \sigma^2_{h_j^o} = \frac{1}{\langle \lambda \rangle_{q(\lambda)} |x_j|^2} \tag{4.36}$$

The messages from the channel prior node $f_{\mathrm{H}}$ to each variable node $h_i$ are given by

$$m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) \propto \int f_{\mathrm{H}}(\mathbf{h}) \prod_{j \in \mathcal{P} \setminus i} m^{\mathrm{MF}}_{f_{\mathrm{P}_j} \to h_j}(h_j) dh_j \prod_{k \in \mathcal{D} \setminus i} m^{\mathrm{MF}}_{f_{\mathrm{D}_k} \to h_k}(h_k) dh_k \tag{4.37}$$

for all $i \in [1 : N + M]$. Note that

$$\prod_{j \in \mathcal{P}} \prod_{k \in \mathcal{D}} m^{\mathrm{MF}}_{f_{\mathrm{P}_j} \to h_j}(h_j)\, m^{\mathrm{MF}}_{f_{\mathrm{D}_k} \to h_k}(h_k) \propto \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}^o_{\mathbf{h}}, \boldsymbol{\Sigma}^o_{\mathbf{h}}) \tag{4.38}$$

where the entries of

$$\boldsymbol{\mu}^o_{\mathbf{h}} = (\mu_{h^o_i} | i \in [1 : M + N])^{\mathrm{T}} \tag{4.39}$$

$$\boldsymbol{\Sigma}^o_{\mathbf{h}} = \mathrm{diag}\left[\left(\sigma^2_{h^o_i} | i \in [1 : N + M]\right)\right] \tag{4.40}$$

are computed as in (4.31) and (4.36). Now denote the vector $\boldsymbol{\mu}^o_{\mathbf{h}_{\bar{i}}}$ as the vector $\boldsymbol{\mu}^o_{\mathbf{h}}$ with the $i$th entry removed and $[\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}$ as the matrix $\boldsymbol{\Sigma}^o_{\mathbf{h}}$ with the $i$th row and $i$th column removed. In the following we also make use of the notation $[\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{i,\bar{i}}$, which is the $i$th row of $\boldsymbol{\Sigma}^o_{\mathbf{h}}$ with the $i$th column removed and similarly $[\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},i}$ is the $i$th column with the $i$th row removed. Using this notation (4.37) can be written as

$$m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) \propto \int f_{\mathrm{H}}(\mathbf{h}) \, \mathrm{CN}(\mathbf{h}_{\bar{i}}; \boldsymbol{\mu}^o_{\mathbf{h}_{\bar{i}}}, [\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}) d\mathbf{h}_{\bar{i}} \tag{4.41}$$

Considering $f_{\mathrm{H}}(\mathbf{h})$ as a product of two pdfs $p(h_i|\mathbf{h}_{\bar{i}})p(\mathbf{h}_{\bar{i}})$ and using the identity of Section C.3 we obtain

$$\begin{aligned} m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) \propto \int \mathrm{CN}(h_i; \mu_{h_i} + [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,\bar{i}} \, [\boldsymbol{\Sigma}]^{-1}_{\bar{i},\bar{i}} \, \mathbf{h}_{\bar{i}}, [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,i} - [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,\bar{i}} \, [\boldsymbol{\Sigma}_{\mathbf{h}}]^{-1}_{\bar{i},\bar{i}} \, [\boldsymbol{\Sigma}_{\mathbf{h}}]_{\bar{i},i}) \\ \cdot \, \mathrm{CN}(\mathbf{h}_{\bar{i}}; \boldsymbol{\mu}_{\mathbf{h}_{\bar{i}}}, [\boldsymbol{\Sigma}_{\mathbf{h}}]_{\bar{i},\bar{i}}) \, \mathrm{CN}(\mathbf{h}_{\bar{i}}; \boldsymbol{\mu}^o_{\mathbf{h}_{\bar{i}}}, [\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}) d\mathbf{h}_{\bar{i}} \end{aligned} \tag{4.42}$$

By rewriting the last two factors of (4.42) using the identity of Section C.5 and dropping the normalization constant, we get a function of the form $\int p(h_i|\mathbf{h}_{\bar{i}})p(\mathbf{h}_{\bar{i}})d\mathbf{h}_{\bar{i}}$. This corresponds to computing the marginal distribution $p(h_i)$ and we can thus apply the identity in Section C.4 to obtain

$$m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) \propto \mathrm{CN}\left(h_i; \mu_{h^c_i}, \sigma^2_{h^c_i}\right) \tag{4.43}$$

with

$$\mu_{h^c_i} = [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,\bar{i}} \left([\boldsymbol{\Sigma}_{\mathbf{h}}]_{\bar{i},\bar{i}} + [\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}\right)^{-1} \boldsymbol{\mu}^o_{h_{\bar{i}}} \tag{4.44}$$

$$\sigma^2_{h^c_i} = [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,i} + [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,\bar{i}} \left([\boldsymbol{\Sigma}_{\mathbf{h}}]_{\bar{i},\bar{i}} + [\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}\right)^{-1} \boldsymbol{\mu}^o_{h_{\bar{i}}} \tag{4.45}$$

The computation of each message requires the computation of the inverse of a matrix of dimension $(M + N - 1) \times (M + N - 1)$, which is an operation of complexity[1] $\mathcal{O}\left((M + N - 1)^3\right)$. Fortunately, computation of these messages is not required in an implementation of the algorithm. To see why, notice that the messages are used to compute

$$n_{h_i \to f_{\mathrm{D}_i}}(h_i) \propto m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) \, m^{\mathrm{MF}}_{f_{\mathrm{D}_i} \to h_i}(h_i), \ \forall i \in \mathcal{D} \tag{4.46}$$

$$n_{h_j \to f_{\mathrm{P}_j}}(h_j) \propto m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_j}(h_j) \, m^{\mathrm{MF}}_{f_{\mathrm{P}_j} \to h_j}(h_j), \ \forall j \in \mathcal{P} \tag{4.47}$$

[1] As common practice we treat computing the inverse of the $L \times L$ matrix as an operation of $\mathcal{O}(L^3)$ though an algorithm of complexity $\mathcal{O}(L^{2.373})$ exists [49].

Inserting (4.41), (4.30) and (4.35) into (4.46) and (4.47) we obtain

$$\begin{aligned}
n_{h_i \to f_{\mathrm{D}_i} \mathrm{or} f_{\mathrm{P}_i}}(h_i) &\propto \int \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}_{\mathbf{h}}, \boldsymbol{\Sigma}_{\mathbf{h}}) \, \mathrm{CN}\Big(\mathbf{h}_{\bar{i}}; \boldsymbol{\mu}_{\mathbf{h}^o_{\bar{i}}}, [\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{\bar{i},\bar{i}}\Big) \\
&\qquad \cdot \mathrm{CN}\Big(h_i; \mu_{h^o_i}, \sigma^2_{h^o_i}\Big) \, d\mathbf{h}_{\bar{i}} && (4.48) \\
&= \int \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}_{\mathbf{h}}, \boldsymbol{\Sigma}_{\mathbf{h}}) \, \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}^o_{\mathbf{h}}, \boldsymbol{\Sigma}^o_{\mathbf{h}}) \, d\mathbf{h}_{\bar{i}} && (4.49) \\
&\propto \int \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}^q_{\mathbf{h}}, \boldsymbol{\Sigma}^q_{\mathbf{h}}) \, d\mathbf{h}_{\bar{i}} && (4.50) \\
&= \mathrm{CN}(h_i; [\boldsymbol{\mu}^q_{\mathbf{h}}]_i, [\boldsymbol{\Sigma}^q_{\mathbf{h}}]_{i,i}) && (4.51) \\
&= q(h_i) && (4.52)
\end{aligned}$$

for all $i \in [1 : M + N]$, with

$$\begin{aligned}
\boldsymbol{\Sigma}^q_{\mathbf{h}} &= \big((\boldsymbol{\Sigma}_{\mathbf{h}})^{-1} + (\boldsymbol{\Sigma}^o_{\mathbf{h}})^{-1}\big)^{-1} && (4.53) \\
\boldsymbol{\mu}^q_{\mathbf{h}} &= \boldsymbol{\Sigma}^q_{\mathbf{h}} \, (\boldsymbol{\Sigma}^o_{\mathbf{h}})^{-1} \, \boldsymbol{\mu}^o_{\mathbf{h}} && (4.54)
\end{aligned}$$

All messages in (4.51) can thus be computed[2] by (4.53) and (4.54) involving a single matrix inversion of dimension $(M + N) \times (M + N)$, without explicitly using the messages $\{m^{\mathrm{BP}}_{f_{\mathrm{H}} \to h_i}(h_i) | i \in [1 : N + M]\}$, since $\boldsymbol{\Sigma}_{\mathbf{h}}$ is given in the definition of the prior of channel coefficients $f_{\mathrm{H}}(\mathbf{h})$ and the quantities $(\boldsymbol{\mu}^o_{\mathbf{h}}, \boldsymbol{\Sigma}^o_{\mathbf{h}})$ are given by the set of messages $\{m^{\mathrm{MF}}_{f_{\mathrm{D}_i} \to h_i}(h_i) | i \in \mathcal{D}\} \cup \{m^{\mathrm{MF}}_{f_{\mathrm{P}_j} \to h_j}(h_j) | j \in \mathcal{P}\}$.

Notice that if we gather the channel coefficients into a vector variable node $\mathbf{h} = (h_i | i \in [1 : N + M])^{\mathrm{T}}$ and ‘move’ this node to the MF part, the messages to the factor nodes $f_{\mathrm{D}_i}$ and $f_{\mathrm{P}_j}$ are identical to the expression in (4.51). In this case the belief $q(\mathbf{h}) = \mathrm{CN}(\mathbf{h}; \boldsymbol{\mu}^q_{\mathbf{h}}, \boldsymbol{\Sigma}^q_{\mathbf{h}})$. This approach is used in [4] and is an example of how two different choices of factor graph representation and inference scheme may lead to the exact same algorithm. We generalize the approach in Chapter 7, where the channel coefficients are moved to the MF part and gathered into smaller groups in order to lower the computational complexity.

[2] As we show in Chapter 6, $\boldsymbol{\Sigma}_{\mathbf{h}}$ may be singular. Applying the Woodbury formula (Appendix C.6) we can instead compute the equivalent expression $\boldsymbol{\Sigma}^q_{\mathbf{h}} = \boldsymbol{\Sigma}_{\mathbf{h}} - \left(\boldsymbol{\Sigma}_{\mathbf{h}} + \boldsymbol{\Sigma}^o_{\mathbf{h}}\right)^{-1} \boldsymbol{\Sigma}_{\mathbf{h}}$.

### 4.4.2 Noise Precision

Given the current beliefs of the data symbols $q(x_i)$ and channel coefficients $q(h_i)$, the messages to the noise precision variable node $\lambda$ are given by

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}(\lambda) \propto \exp\Big(\langle \log f_{\mathrm{D}_i}(h_i, x_i, \lambda)\rangle_{q(x_i)q(h_i)}\Big) \tag{4.55}$$

$$\propto \lambda \exp\Big(-\lambda \left\langle |y_i - h_i x_i|^2\right\rangle_{q(x_i)q(h_i)}\Big)\,,\ \forall\, i \in \mathcal{D} \tag{4.56}$$

$$m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\lambda}(\lambda) \propto \exp\Big(\langle \log f_{\mathrm{P}_j}(h_j, \lambda)\rangle_{q(h_j)}\Big) \tag{4.57}$$

$$\propto \lambda \exp\Big(-\lambda \left\langle |y_j - h_j x_j|^2\right\rangle_{q(h_j)}\Big)\,,\ \forall\, j \in \mathcal{P} \tag{4.58}$$

$$m^{\mathrm{BP}}_{f_\lambda\to\lambda}(\lambda) \propto \lambda^{a-1}\exp(-b\lambda) \tag{4.59}$$

We thus have

$$q(\lambda) \propto m^{\mathrm{BP}}_{f_\lambda\to\lambda}(\lambda) \prod_{i\in\mathcal{D}} m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}(\lambda) \prod_{j\in\mathcal{P}} m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\lambda}(\lambda) \tag{4.60}$$

$$\propto \lambda^{a+M+N-1}\exp(-\lambda b)\exp\left(\sum_{i\in\mathcal{D}} \left\langle |y_i - x_i h_i|^2\right\rangle_{q(x_i)q(h_i)}\right)$$

$$\cdot \exp\left(\sum_{j\in\mathcal{P}} \left\langle |y_j - x_j h_j|^2\right\rangle_{q(h_j)}\right) \tag{4.61}$$

$$= \lambda^{a+M+N-1}\exp\Big(-\lambda\,\Big(b + \left\langle ||\mathbf{y}-\mathbf{X}\mathbf{h}||_2^2\right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)q(h_i)\prod_{j\in\mathcal{P}} q(h_j)}\Big)\Big) \tag{4.62}$$

Thus the belief of the noise precision reads

$$q(\lambda) = \mathrm{Ga}\left(\lambda; a+M+N, b+\left\langle ||\mathbf{y}-\mathbf{X}\mathbf{h}||_2^2\right\rangle\right) \tag{4.63}$$

The belief is used in (4.31), (4.36) and (4.65) through its mean

$$\langle\lambda\rangle_{q(\lambda)} = \frac{a+M+N}{b+\left\langle ||\mathbf{y}-\mathbf{X}\mathbf{h}||_2^2\right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)q(h_i)\prod_{j\in\mathcal{P}} q(h_j)}} \tag{4.64}$$

Algorithms for scenarios in which the noise precision $\lambda$ is known can be obtained by replacing every occurrence of $\langle\lambda\rangle_{q(\lambda)}$ in (4.31), (4.36) and (4.65) by the true value. Obviously the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}(\lambda)$, $m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\lambda}(\lambda)$ and $m^{\mathrm{BP}}_{f_\lambda\to\lambda}(\lambda)$ should not be computed in these cases.

### 4.4.3 Data Symbols

The derivation of the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i)$ is very similar to $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to h_i}(h_i)$. Given $q(h_i)$ and $q(\lambda)$ we thus get

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i) \propto \mathrm{CN}\left(x_i; \frac{y_i\,\langle h_i\rangle^*_{q(h_i)}}{\langle |h_i|^2\rangle_{q(h_i)}}, \frac{1}{\langle\lambda\rangle_{q(\lambda)}\,\langle |h_i|^2\rangle_{q(h_i)}}\right) \tag{4.65}$$

for all $i \in \mathcal{D}$. Note that even though (4.65) has the functional form of a Gaussian pdf, the message is a pmf since $x_i \in \mathcal{S}_{\mathrm{D}}$ is discrete-valued. Given the messages $n_{c_q^{(n)} \to f_{\mathrm{M}_n}}(c_q^{(n)})$ from the coding and interleaving subgraph we can compute

$$m^{\mathrm{BP}}_{f_{\mathrm{M}_n} \to x_{i_n}}(x_{i_n}) \propto \sum_{\mathbf{c}^{(n)} \in \{0,1\}^Q} f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)}) \prod_{q \in [1:Q]} n_{c_q^{(n)} \to f_{\mathrm{M}_n}}(c_q^{(n)}) \qquad (4.66)$$

for all $n \in [1:N]$. The belief $q(x_i)$ are thus given by

$$q(x_{i_n}) \propto m^{\mathrm{BP}}_{f_{\mathrm{M}_n} \to x_{i_n}}(x_{i_n}) \cdot m^{\mathrm{MF}}_{f_{\mathrm{D}_{i_n}} \to x_{i_n}}(x_{i_n}) \; \forall \, n \in [1:N] \qquad (4.67)$$

### 4.4.4 Channel Coding and Interleaving

In the following we outline how the channel coding and interleaving part is implemented by applying BP to a factor graph representation of convolutional channel codes. The resulting algorithm is identical to the BCJR [50] algorithm [3].

The factor graph representation of the channel coding and interleaving subgraph is shown in Figure 4.2. In this representation we have exploited how the trellis representation of the code translates into a factor graph as described in [3]. Again the factor nodes $f_{\mathrm{U}_k}$ for all $k \in [1:K]$ represent the prior distribution of the information bits. The factor nodes $f_{\mathrm{S}_k}$ for all $k \in [0:K]$ represent indicator functions equal 1 if the information bit $u_k$ produces the vector of coded bits $\mathbf{w}_k$ and changes the trellis state from state $s_{k-1}$ to state $s_k$, and 0 otherwise. The initial state $s_0$ is known a priori. Using a rate $R$ code the vector of coded bits is $\mathbf{w}_k \in \{0,1\}^{\frac{1}{R}}$ for all $k \in [1:K]$ and we thus assume that $\frac{1}{R}$ is an integer. To do bit-level interleaving, each vector of coded bits is mapped to $\frac{1}{R}$ individual bits by the mapping function $f_{\mathrm{w}_k}(\mathbf{c}, \mathbf{w}_k)$. Notice that the edges shown on 4.2 are just an example, i.e. the node $w_1$ may have an edge to any of the variables $\{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(N)}\}$ because of the interleaving. The coded and interleaved bits $\mathbf{c} \in \{0,1\}^{\frac{K}{R}}$ are divided into $N$ segments of size $Q$ and each segment is mapped to one data subcarrier.

Since all variables are discrete and $f_{\mathrm{S}_k}$ and $f_{\mathrm{W}_k}$ represents hard constraints, the BP algorithm is trivially applied and therefore the expressions for any of the messages are not shown here. In the implementation we are dealing with pmfs with very small values. For better numerical stability we

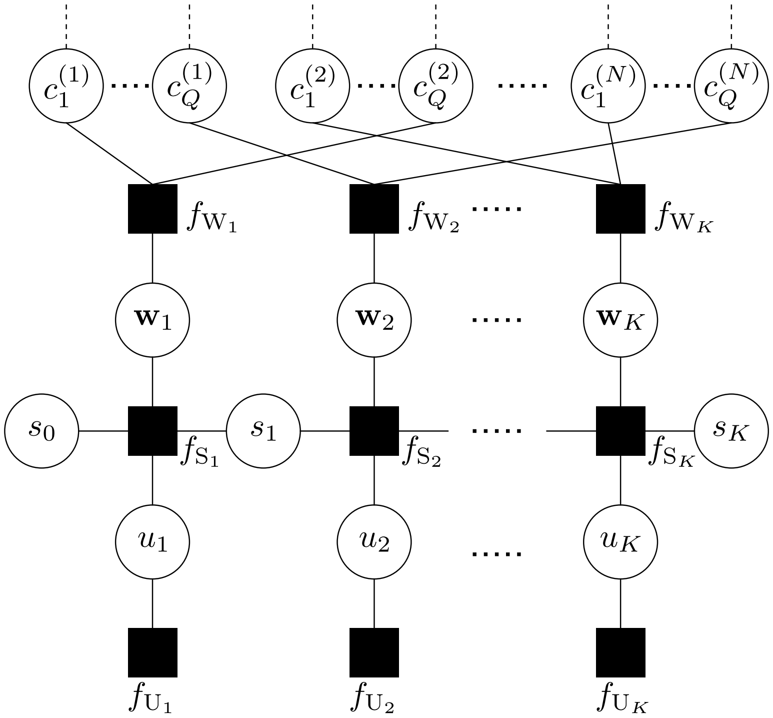


**Figure 4.2:** *Factor graph representation of the channel coding and interleaving part.*

use log-domain versions of the BP messages (3.6) (3.7)

$$\tilde{m}^{\mathrm{BP}}_{a\to i}(x_i) = \log m^{\mathrm{BP}}_{a\to i}(x_i) \tag{4.68}$$

$$= \log d_a + \log \sum_{\mathbf{x}_a \setminus x_i} f_a(\mathbf{x}_a) \exp\left( \sum_{j\in\mathcal{N}(a)\setminus i} \tilde{n}_{j\to a}(x_j) \right) \tag{4.69}$$

$$\tilde{n}_{i\to a}(x_i) = \log n_{i\to a}(x_i) \tag{4.70}$$

$$= \log e_i + \sum_{c\in\mathcal{N}(i)\cap\mathcal{A}_{\mathrm{MF}}} \tilde{m}^{\mathrm{MF}}_{c\to i}(x_i) \sum_{c\in\mathcal{N}(i)\cap\mathcal{A}_{\mathrm{BP}}\setminus a} \tilde{m}^{\mathrm{BP}}_{c\to i}(x_i) \tag{4.71}$$

To avoid numerical underflows when evaluating the sum (4.69) the following identity is also used

$$\log \sum_i \exp(x_i) = M + \log \sum_i \exp(x_i - M) \tag{4.72}$$

with $M = \max_i x_i$. The algorithm is implemented in Matlab and is available on the accompanying CD. The implementation relies on lookup tables to avoid evaluating all configurations of $\mathbf{x}_a \setminus x_i$ in the left-most sum (4.69) for which the indicator functions $f_a(\mathbf{x}_a)$ evaluate to zero. However, to run the large-scale simulations a faster implementation of the decoder is needed. Therefore our implementation of the decoder can be replaced with the Iterative Solutions Coded Modulation Library [51]. See Appendix H for a description of the implemented simulation framework and running time measurements for the algorithms with and without the Coded Modulation Library.

## 4.5. Iterative Algorithm and Scheduling

The iterative inference scheme produces a local minimizer of the objective function. Which local minimizer is found depends on the initialization of the algorithm as well as the order in which messages are computed. The initialization of beliefs and scheduling of messages are therefore important design parameters and the optimal choice depends on the model. As pilot symbols are transmitted in order to estimate $\mathbf{h}$ it makes intuitively sense to first compute the beliefs $q(h_i)$ for all $i \in [1 : N+M]$ using the pilot symbols only. Thus the algorithm proceeds as follows

1) If $\lambda$ is unknown initialize $\langle\lambda\rangle_{q(\lambda)}$ using a heuristic scheme.

2) Initialize the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to h_i}(h_i) \propto \mathrm{CN}(h_i; 0, \infty)$ for all $i \in \mathcal{D}$ and compute $m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to h_j}(h_j)$ for all $j \in \mathcal{P}$ using (4.35).

3) Update $q(h_i)$ for all $i \in [1 : N + M]$ using (4.51) with $\boldsymbol{\Sigma}^q_{\mathbf{h}}$ and $\boldsymbol{\mu}^q_{\mathbf{h}}$ defined in (4.53) and (4.54) respectively.

4) If $\lambda$ is unknown update $\langle q(\lambda)\rangle_{q(\lambda)}$ using (4.64).

5) Compute the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i)$ for all $i \in \mathcal{D}$ using (4.65).

6) Compute the messages in the modulation, interleaving and coding subgraph yielding the belief of the information bits $q(u_k)$ for all $k \in [1 : K]$. Terminate the algorithm if the stopping criterion is fulfilled.

7) Using the messages from the subgraph $\left\{n_{c_q^{(n)}\to f_{\mathrm{M}_n}}(c_q^{(n)}) |\, q \in [1 : Q], n \in [1 : N]\right\}$ as extrinsic values compute $m_{f_{\mathrm{M}_n}\to x_{i_n}}(x_{i_n})$ for all $n \in [1 : N]$ using (4.66).

8) Update $q(x_i)$ for all $i \in \mathcal{D}$ using (4.67)

9) Compute $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to h_i}(h_i)$ for all $i \in \mathcal{D}$ and $m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to h_j}(h_j)$ for all $j \in \mathcal{P}$ using (4.30) and (4.35) respectively. Repeat from step (3).

The algorithm runs until a stopping criterion is fulfilled. Selecting a good stopping criteria is not an important objective of this project, so we choose the simplest stopping criterion possible: The algorithm is terminated after a fixed number of iterations have been completed. After completion hard decisions are made on the beliefs of the information bits.

CHAPTER 5

# Message-Passing Receiver with Sparse Channel Estimation

In this chapter a novel message-passing iterative receiver algorithm with sparse channel estimation is introduced. The key difference between this algorithm and the one presented in the previous chapter is that it exploits the sparsity of the channel. We pose the task of sparse channel estimation as a compressed sensing problem and apply the methods of sparse Bayesian learning (SBL) with hierarchical prior modeling to embed the problem into the message-passing receiver framework. This enables a unified scheme in which sparse channel estimation, detection and decoding is jointly performed as message-passing on a factor graph representation of the system.

In Section 5.1 we define the dictionary required for the compressed sensing problem and discuss the problem that arises from the fact that the propagation delays of the multipath components are continuous valued while employing a dictionary in which the delays are discretized. In Section 5.2, 5.3 and 5.4 we present the probabilistic model, its factor graph representation and the derivation of the messages respectively. Considerations on the scheduling of the iterative algorithm are presented in Section 5.5. In Section 5.6 we show how increasing the spacing between the pilots of the OFDM system affects the dictionary of the compressed sensing problem.

## 5.1. Dictionary for Sparse Signal Representation

In the iterative receiver algorithm devised in the previous chapter we obtain estimates (in the form of beliefs) of the frequency domain channel coefficients $\mathbf{h}$ as an integrated part of the inference scheme. We now investigate an alternative approach for estimating $\mathbf{h}$ by exploiting the sparse structure of the multipath channel model. To apply the sparse Bayesian learning framework as described in Chapter 3 we need to define a dictionary matrix $\mathbf{\Phi}$ in which $\mathbf{h}$ has a sparse representation $\boldsymbol{\alpha}$.

Sparse channel estimation has previously been applied to estimate the multipath OFDM-channel in [13, 18]. In [13] the optimization problem is solved using greedy pursuit algorithms (Basis Pursuit and Orthogonal Matching Pursuit) and [18] follows a Bayesian approach with hierarchical prior modelling. In both cases only the pilot symbols are used for channel estimation and therefore only the pilot symbol observations are used

$$\mathbf{y}_{\mathcal{P}} = \mathbf{X}_{\mathcal{P}}\mathbf{h}_{\mathcal{P}} + \mathbf{w}_{\mathcal{P}} \tag{5.1}$$

where $\mathbf{y}_{\mathcal{P}}$ denotes the vector formed from the entries of $\mathbf{y}$ indexed by the pilot indices $\mathcal{P}$, and the remaining vectors are similarly defined. In [13] the pilot symbols $\mathbf{X}_{\mathcal{P}}$ are included in the dictionary, whereas in [18] they are divided out to obtain the 'modified' observation model

$$\mathbf{r}_{\mathcal{P}} = (\mathbf{X}_{\mathcal{P}})^{-1}\mathbf{y}_{\mathcal{P}} = \mathbf{h}_{\mathcal{P}} + (\mathbf{X}_{\mathcal{P}})^{-1}\mathbf{w}_{\mathcal{P}} \tag{5.2}$$

After obtaining the estimate of the pilot symbol channel coefficients $\mathbf{h}_{\mathcal{P}}$ the estimate is interpolated to yield estimates of the data symbol channel coefficients $\mathbf{h}_{\mathcal{D}}$. However, as we want to investigate data-aided sparse channel estimation this approach does not simplify the observation model. At the receiver it is not possible to compute $\mathbf{r} = \mathbf{X}^{-1}\mathbf{y}$ since the entries of $\mathbf{X}$ corresponding to data symbols are unknown and needs to be estimated. We thus stick to the observation model in (2.2).

Following the notation in Section 2.2 we can represent the channel coefficients $\mathbf{h}$ in (2.6) using a sum of parameterized vectors

$$\mathbf{h} = \sum_{p=1}^{P} \boldsymbol{\psi}(\tau_p)\beta_p \tag{5.3}$$

where the column vectors $\boldsymbol{\psi} \in \mathbb{C}^{(N+M)\times 1}$ are given by

$$\boldsymbol{\psi}(\tau_p) = \begin{pmatrix} \exp(-j2\pi\Delta f\tau_p) \\ \exp(-j2\pi 2\Delta f\tau_p) \\ \vdots \\ \exp(-j2\pi(N+M)\Delta f\tau_p) \end{pmatrix} \tag{5.4}$$

We can then define the parameterized dictionary $\boldsymbol{\Psi}(\boldsymbol{\tau})$ as

$$\boldsymbol{\Psi}(\boldsymbol{\tau}) = (\boldsymbol{\psi}(\tau_1), \boldsymbol{\psi}(\tau_2), \ldots, \boldsymbol{\psi}(\tau_P)) \tag{5.5}$$

and the channel coefficients are compactly written in matrix notation

$$\mathbf{h} = \boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\beta} \tag{5.6}$$

Each column in $\boldsymbol{\Psi}(\boldsymbol{\tau})$ thus corresponds to a specific tap delay and each row corresponds to one subcarrier frequency.

However, the multipath delays $\boldsymbol{\tau}$ are unknown and $\boldsymbol{\Psi}(\boldsymbol{\tau})$ can therefore not be used as dictionary, because the standard CS framework presumes a fixed dictionary [31]. Following the same approach as in [13, 18, 52, 53] we employ a uniformly spaced grid of delays in the interval $[0, \ldots, \tau_{\max}]$

$$\mathbf{t} = (t_1, t_2, t_3, \ldots, t_L)^{\mathrm{T}} = (0, T_{\mathrm{d}}, 2T_{\mathrm{d}}, \ldots, \tau_{\max})^{\mathrm{T}} \tag{5.7}$$

where $\frac{\tau_{\max}}{T_{\mathrm{d}}}$ is an integer and $T_{\mathrm{d}}$ is the delay resolution. The delay vector $\mathbf{t}$ is thus of dimension $L = \frac{\tau_{\max}}{T_{\mathrm{d}}} + 1$. The dictionary is then defined as $\boldsymbol{\Phi} \in \mathbb{C}^{(N+M)\times L} = \boldsymbol{\Psi}(\mathbf{t})$. Using this dictionary we obtain an estimate $\hat{\mathbf{h}} = \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$, where $\hat{\boldsymbol{\alpha}} \in \mathbb{C}^{L\times 1}$. Notice that since $\boldsymbol{\Phi}$ is defined using discretized delays it may not be possible to represent $\mathbf{h}$ as $\boldsymbol{\Phi}\boldsymbol{\alpha}$ for any $P$ sparse $\boldsymbol{\alpha} \in \mathbb{C}^{L\times 1}$. In other words, the observation model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha} + \mathbf{w} \tag{5.8}$$

is an approximation of the true observation model (2.2).

### 5.1.1 Basis Mismatch

In this section we dwell on the problem of reconstructing a signal using one dictionary ($\boldsymbol{\Phi} = \boldsymbol{\Psi}(\mathbf{t})$) when the true signal has a sparse representation in a different dictionary ($\boldsymbol{\Psi}(\boldsymbol{\tau})$). The purpose is to gain insight on how the choice of delay resolution $T_{\mathrm{d}}$ affects the ability to do reconstruction.

Using a dictionary for reconstruction that is different from the one that has generated the signal can be considered as basis mismatch and is analyzed in [54] with a focus on the discrete Fourier basis. The authors derive an upper bound for the best $P$-term approximation error and a lower bound for the worst-case best $P$-term approximation error in terms of the $\ell_1$-norm, i.e. bounds on

$$\min_{\mathbf{h}'_P \in \left\{ \boldsymbol{\Phi}\boldsymbol{\alpha}_P \,|\, ||\boldsymbol{\alpha}_P||_0 \leq P \right\}} ||\mathbf{h} - \mathbf{h}'_P||_1 \tag{5.9}$$

where $\mathbf{h}'_P$ is a linear combination of $P$ columns of $\boldsymbol{\Phi}$. However, the results are only applicable to dictionaries of finite dimension and since our generating dictionary $\boldsymbol{\Psi}(\boldsymbol{\tau})$ is parameterized by a continuous-valued vector it is not of finite dimension.

Different algorithms for estimating continuous-valued parameters within the compressed sensing framework have been proposed in the literature. These includes convex optimization methods in [53, 55], greedy pursuit algorithms [52] and the Bayesian approach [56]. In [52,55,56] new algorithms are developed for signals that are sparse in frequency domain. These results are also useful in this project, where we have sparsity in delay domain, because of the duality of the Fourier transform.

A Bayesian approach to handle the problem of basis mismatch is to model the difference between the true delays and the corresponding delays in the dictionary as a uniform random variable, as done for frequency domain mismatch in [56]. The dictionary is then conditioned on the displacement $\mathbf{d}$, i.e. $\boldsymbol{\Phi}|\mathbf{d} = \boldsymbol{\Psi}(\mathbf{t} + \mathbf{d})$ where $\mathbf{d} = (d_1, \ldots, d_L)^{\mathrm{T}}$ is uniformly distributed $p(\mathbf{d}) = \prod_{\ell=1}^{L} \mathrm{U}(d_\ell; -\frac{1}{2}T_{\mathrm{d}}, \frac{1}{2}T_{\mathrm{d}})$. However this approach does not lead to analytically tractable pdfs and computationally heavy numerical methods are applied to circumvent this [56]. In this project we therefore utilize the fixed dictionary approach.

### 5.1.2 Delay Resolution

As shown in [52] the reconstruction accuracy using standard greedy pursuit algorithms degrades significantly when the sinusoidal frequencies are located outside the sampling grid and the performance does not increase by increasing the resolution of the dictionary. This is caused by the fact that a finer resolution also increases the coherence of the dictionary. As shown in Appendix E the absolute inner product between two columns of the dictionary is a function of the absolute difference in delay between the columns. We have

$$\frac{|\boldsymbol{\psi}(t_k)^{\mathrm{H}}\boldsymbol{\psi}(t_\ell)|}{||\boldsymbol{\psi}(t_k)||_2\,||\boldsymbol{\psi}(t_\ell)||_2} = \frac{|D_{N+M}(2\pi\Delta f(t_k - t_\ell))|}{N+M} \tag{5.10}$$

where

$$D_K(x) = \sum_{k=0}^{K-1} \exp(jkx) = \frac{\sin(\frac{1}{2}Kx)}{\sin(\frac{1}{2}x)} \exp\big(jx\tfrac{K-1}{2}\big) \tag{5.11}$$

is the Dirichlet kernel [57]. The normalized inner product in (5.10) is plotted as a function of the absolute delay difference $|t_k - t_\ell|$ in Figure 5.1 for two different numbers of subcarriers $N+M$. Notice how the rate of decay of the envelope depends on the number of subcarriers, i.e. the system bandwidth. To minimize the coherence of the dictionary one can choose $T_{\mathrm{d}} = \frac{1}{\Delta f(N+M)}$, which makes the columns orthogonal to each other and $\boldsymbol{\Phi}$ is thus formed from the columns of a discrete inverse Fourier matrix. However, in this case $\mathbf{h}$ may not have a sparse representation in $\boldsymbol{\Phi}$. Furthermore, the approach in [55] can be considered as using a fully coherent dictionary ($\mu(\boldsymbol{\Phi}) \approx 1$) for reconstruction; however this fact was shown to not be an obstacle to

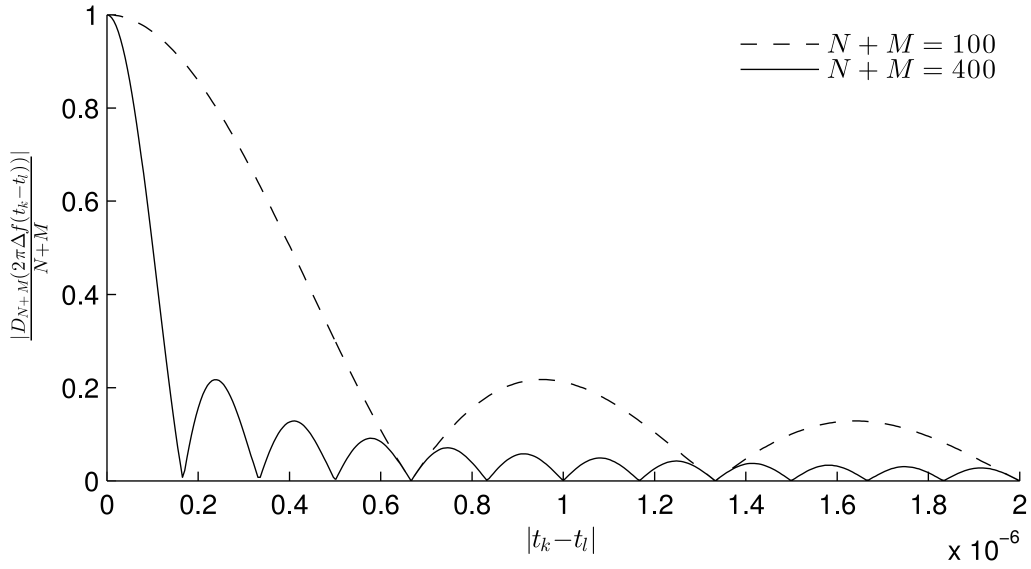


**Figure 5.1:** *Normalized correlation between two columns of the dictionary* $\frac{|\boldsymbol{\psi}(t_k)^{\mathrm{H}}\boldsymbol{\psi}(t_\ell)|}{||\boldsymbol{\psi}t_k||_2||\boldsymbol{\psi}(t_\ell)||_2}$ *as a function of their relative delay* $|t_k - t_\ell|$.

recovery. In that case it is more important that the parameters of the true signal are sufficiently separated [55]. Whether we can use coherent dictionaries for reconstruction thus depends on the algorithm that is applied. From (5.10), we see that for $\tau_{\max} \leq \frac{1}{2\Delta f}$ and $T_{\mathrm{d}} \leq \frac{1}{2\Delta f(N+M)}$ the coherence of the dictionary is

$$\mu(\mathbf{\Phi}) = \frac{|D_{N+M}(2\pi\Delta f T_{\mathrm{d}})\,|}{N+M} \tag{5.12}$$

This result will be used for comparison in the following.

We cannot hope to successfully apply a sparse reconstruction scheme if the signal of interest $\mathbf{h}$ does not have a sparse representation in the dictionary $\mathbf{\Phi}$. With $\mathbf{h}$ given in (5.6), let $\mathbf{h}_P$ denote the best approximation of $\mathbf{h}$ one can obtain from a linear combination of $P$ columns of $\mathbf{\Phi}$, i.e.

$$\mathbf{h}_P = \underset{\mathbf{h}' \in \{\mathbf{\Phi}\boldsymbol{\alpha}\,|\,||\boldsymbol{\alpha}||_0 \leq P\}}{\operatorname{argmin}} ||\mathbf{h} - \mathbf{h}'||_2 \tag{5.13}$$

As shown in Appendix F we can write an upper bound for the best $P$-term approximation error as

$$\frac{||\mathbf{h} - \mathbf{h}_P||_2^2}{N+M} \leq \left(1 - \left|\frac{D_{N+M}(\pi\Delta f T_{\mathrm{d}})}{N+M}\right|^2\right)\left(\sum_{p=1}^{P} |\beta_p|\right)^2 \tag{5.14}$$

Even if the upper bound is loose we can still draw some conclusions from it. Notice how the Dirichlet kernel of order $N+M$ shows up here as in the expression for coherence of the dictionary in (5.12). We can drive the upper bound to zero by letting $T_{\mathrm{d}} \to 0$ and thus get an arbitrarily low approximation error. It means that a sparse representation of $\mathbf{h}$ in $\mathbf{\Phi}$ does exist if the delay resolution is sufficiently fine. However, this leads to increased coherence (5.12) and maybe more importantly it increases the computational complexity of the sparse reconstruction algorithm as the number of columns $L = 1 + \frac{\tau_{\max}}{T_{\mathrm{d}}}$ increases. The bound thus suggests that choosing the delay resolution $T_{\mathrm{d}}$ is a trade-off between the approximation error of the sparse representation versus the coherence of the dictionary and the computational complexity of the algorithm.

It is not required that a perfect sparse approximation exists, meaning an approximation error of 0. Using the oracle estimator as a lower bound (see Appendix G), we can not expect to obtain perfect estimates of $\mathbf{h}$ anyway, because our observations are impaired by noise. The delay resolution $T_{\mathrm{d}}$ should thus be chosen such that the approximation error is small compared to the lower bound of the estimation error. We can not derive a perfect choice of $T_{\mathrm{d}}$ using the bound (5.14), but it suggests that the choice of $T_{\mathrm{d}}$ should depend on the system bandwidth as the decay rate of the Dirichlet kernel depends on the order as shown in Figure 5.1. When looking at $|D_{N+M}(\pi\Delta f T_{\mathrm{d}})\,|$ as function of $T_{\mathrm{d}}$ the width of the main lobe is $\frac{1}{\Delta f(N+M)}$. For a certain trade-off between approximation error and computational complexity we can thus expect that $T_{\mathrm{d}} \propto \frac{1}{\Delta f(N+M)}$, which implies that the number of columns $L$ of the dictionary should be proportional to the system bandwidth, i.e. $L \propto \Delta f(N+M)$. This hypothesis is supported by the numerical results in Chapter 6. If we consider the dictionary as an oversampled discrete Fourier transform matrix this is equivalent to keeping a constant oversampling factor.

## 5.2. Probabilistic Model

With the approximated observation model (5.8) we can now devise an iterative receiver with data-aided sparse channel estimation. The iterative receiver presented in Chapter 4 is used as a base, as all the functionality except the channel estimation part can be reused. As $\mathbf{h}$ is not directly represented in the observation model (5.8) it is dropped from the system function. Similar to (4.1) the system function factorizes as

$$p(\mathbf{y}, \mathbf{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \lambda, \mathbf{c}, \mathbf{u}) = p(\mathbf{y}|\mathbf{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \lambda)p(\boldsymbol{\alpha})p(\lambda)p(\mathbf{x}_{\mathcal{D}}|\mathbf{c})p(\mathbf{c}|\mathbf{u})p(\mathbf{u}) \tag{5.15}$$

This resembles the factorization in (4.1) with $\mathbf{h}$ replaced by $\boldsymbol{\alpha}$. However, to enforce sparse values of $\boldsymbol{\alpha}$, we model $\boldsymbol{\alpha}$ using hierarchical priors and thus include the hyper-parameters in the system function. To consider a very general case we employ the three-layer hierarchical prior model introduced in Chapter 3, hence the hyper-parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ must be included in the system function

$$\begin{aligned} &p(\mathbf{y}, \mathbf{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\eta}, \lambda, \mathbf{c}, \mathbf{u}) = \\ &\qquad p(\mathbf{y}|\mathbf{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \lambda)p(\boldsymbol{\alpha}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}|\boldsymbol{\eta})p(\boldsymbol{\eta})p(\lambda)p(\mathbf{x}_{\mathcal{D}}|\mathbf{c})p(\mathbf{c}|\mathbf{u})p(\mathbf{u}) \end{aligned} \tag{5.16}$$

The observations can be factorized as

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}_{\mathcal{D}}, \boldsymbol{\alpha}, \lambda) &= \mathrm{CN}\left(\mathbf{y}; \mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha}, \lambda^{-1}\mathbf{I}\right) && (5.17)\\ &= \prod_{i\in\mathcal{D}} p(y_i|x_i, \boldsymbol{\alpha}, \lambda) \prod_{j\in\mathcal{P}} p(y_j|\boldsymbol{\alpha}, \lambda) && (5.18)\\ &= \prod_{i\in\mathcal{D}\cup\mathcal{P}} \mathrm{CN}(y_i; x_i \left[\boldsymbol{\Phi}\boldsymbol{\alpha}\right]_i, \lambda^{-1}) && (5.19) \end{aligned}$$

The channel and noise prior pdfs are given by

$$p(\lambda) = \mathrm{Ga}(\lambda; a, b) = \frac{b^a}{\Gamma(a)}\lambda^{a-1}\exp(-b\lambda) \tag{5.20}$$

$$p(\boldsymbol{\alpha}|\boldsymbol{\gamma}) = \mathrm{CN}\left(\boldsymbol{\alpha}; \mathbf{0}, \boldsymbol{\Gamma}\right) = \prod_{\ell=1}^{L} \mathrm{CN}\left(\alpha_\ell; 0, \gamma_\ell\right) \tag{5.21}$$

$$p(\boldsymbol{\gamma}|\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\gamma_\ell; \varepsilon, \eta_\ell\right) \tag{5.22}$$

$$p(\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\gamma_\ell; c, d\right) \tag{5.23}$$

Where $\boldsymbol{\Gamma} = \operatorname{diag}(\boldsymbol{\gamma})$ is the diagonal matrix with the entries of $\boldsymbol{\gamma}$ on the diagonal. The pmfs governing the modulation, interleaving and coding constraints $p(\mathbf{x}_{\mathcal{D}}|\mathbf{c})$, $p(\mathbf{c}|\mathbf{u})$ and $p(\mathbf{u})$ are the same as in Section 4.1.

## 5.3. Factor Graph Representation

As in Section 4.2 we define local functions within three categories:

**Observation Model**

$$f_{\mathrm{D}_i}(\boldsymbol{\alpha}, x_i, \lambda) = p(y_i|x_i, \boldsymbol{\alpha}, \lambda) \tag{5.24}$$

$$= \mathrm{CN}(y; [\boldsymbol{\Phi\alpha}]_i x_i, \lambda^{-1}),\ i \in \mathcal{D} \tag{5.25}$$

$$f_{\mathrm{P}_j}(\boldsymbol{\alpha}, \lambda) = p(y_j|x_j, \boldsymbol{\alpha}, \lambda) \tag{5.26}$$

$$= \mathrm{CN}(y; [\boldsymbol{\Phi\alpha}]_j x_j, \lambda^{-1}),\ j \in \mathcal{P} \tag{5.27}$$

**Channel and Noise Prior**

$$f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = p(\boldsymbol{\alpha}|\boldsymbol{\gamma}) = \mathrm{CN}(\boldsymbol{\alpha}; \mathbf{0}, \boldsymbol{\Gamma}) \tag{5.28}$$

$$f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}, \boldsymbol{\eta}) = p(\boldsymbol{\gamma}|\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\gamma_\ell; \varepsilon, \eta_\ell\right) \tag{5.29}$$

$$f_{\boldsymbol{\eta}}(\boldsymbol{\eta}) = p(\boldsymbol{\eta}) = \prod_{\ell=1}^{L} \mathrm{Ga}\left(\eta_\ell; c, d\right) \tag{5.30}$$

$$f_{\lambda}(\lambda) = p(\lambda) = \mathrm{Ga}(\lambda; a, b) \tag{5.31}$$

**Coding and Modulation Constraints**

$$f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)}) = p(x_{i_n}|\mathbf{c}^{(n)}),\ n \in [1:N] \tag{5.32}$$

$$f_{\mathrm{C}}(\mathbf{c}, \mathbf{u}) = p(\mathbf{c}|\mathbf{u}) \tag{5.33}$$

$$f_{\mathrm{U}}(u_k) = p(u_k),\ k \in [1:K] \tag{5.34}$$

Note that the local functions (5.25) and (4.12) as well as (5.27) and (4.14) are denoted by the same name even though they are different functions. Their functional form is very similar and they play the same role in the probabilistic model. The factorization of the system function is thus

$$p(\mathbf{y}, \mathbf{x}_{\mathcal{D}}, \mathbf{h}, \lambda, \mathbf{c}, \mathbf{u}) = \prod_{i\in\mathcal{D}} f_{\mathrm{D}_i}(\boldsymbol{\alpha}, x_i, \lambda) \prod_{j\in\mathcal{D}} f_{\mathrm{P}_j}(\boldsymbol{\alpha}, \lambda)$$

$$\cdot f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}, \boldsymbol{\eta}) f_{\boldsymbol{\eta}}(\boldsymbol{\eta}) f_{\lambda}(\lambda) f_{\mathrm{C}}(\mathbf{c}, \mathbf{u}) \cdot \prod_{n=1}^{N} f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)}) \prod_{k=1}^{K} f_{\mathrm{U}_k}(u_k) \tag{5.35}$$

The factor graph representation of (5.35) is shown in Figure 5.2. Since the observation functions $f_{\mathrm{D}_i}(\boldsymbol{\alpha}, x_i, \lambda)$ and $f_{\mathrm{D}_j}(\boldsymbol{\alpha}, \lambda)$ depend on all $\boldsymbol{\alpha}$, it is convenient to represent $\boldsymbol{\alpha}$ as a single vector node. As in [18] we do not factorize the pdfs of the hierarchical prior into several local functions. The nodes $f_{\boldsymbol{\gamma}}$, $f_{\boldsymbol{\gamma}}$ and $f_{\boldsymbol{\eta}}$ are included in the MF part of the factor graph. In this case we get the same algorithm if we choose to represent every entry of $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ as single variable nodes and fully factorize $f_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}, \boldsymbol{\eta})$ and $f_{\boldsymbol{\eta}}(\boldsymbol{\eta})$. Using the structured approach eases the notation in the following derivations. Denoting the set of all factor nodes $\mathcal{A}$, we thus have

$$\mathcal{A}_{\text{MF}} = \{f_{\text{D}_i} | i \in \mathcal{D}\} \cup \left\{f_{\text{P}_j} | j \in \mathcal{P}\right\} \cup \{f_{\boldsymbol{\alpha}}\} \cup \{f_{\boldsymbol{\gamma}}\} \cup \{f_{\boldsymbol{\eta}}\} \tag{5.36}$$

$$\begin{aligned} \mathcal{A}_{\text{BP}} &= \{\mathcal{A} \backslash \mathcal{A}_{\text{MF}}\} \\ &= \{f_{\lambda}\} \cup \{f_{\text{C}}\} \cup \{f_{\text{M}_n} | n \in [1:N]\} \cup \{f_{\text{U}_k} | k \in [1:K]\} \end{aligned} \tag{5.37}$$

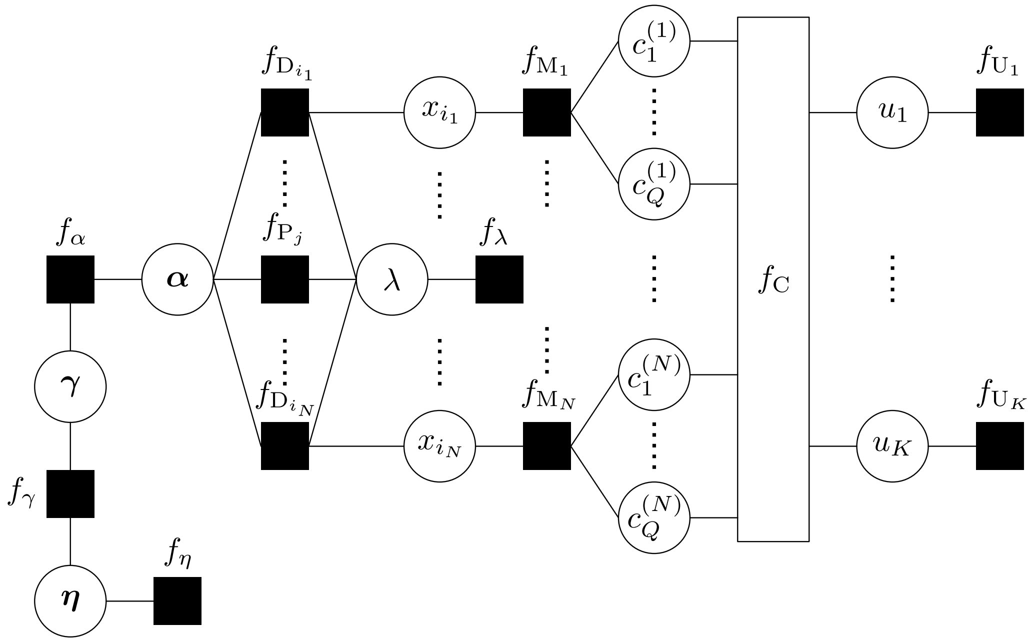


**Figure 5.2:** *Factor graph representation of the factorized system model with hierarchical channel prior.*

## 5.4. Computation of Messages

### 5.4.1 Channel Weights

Given the beliefs $q(\lambda)$ and $q(x_i)$ for all $i \in \mathcal{D}$ we can compute

$$m^{\text{MF}}_{f_{\text{D}_i} \to \boldsymbol{\alpha}}(\boldsymbol{\alpha}) \propto \exp\Big( \langle \log f_{\text{D}_i}(\boldsymbol{\alpha}, x_i, \lambda) \rangle_{q(x_i)\, q(\lambda)} \Big) \tag{5.38}$$

$$\propto \exp\Big( \big\langle -\lambda |y_i - [\boldsymbol{\Phi}\boldsymbol{\alpha}]_i x_i|^2 \big\rangle_{q(x_i)\, q(\lambda)} \Big) \tag{5.39}$$

$$\propto \exp\Big( - \langle \lambda \rangle_{q(\lambda)} \Big( |[\boldsymbol{\Phi}\boldsymbol{\alpha}]_i|^2 \left\langle |x_i|^2 \right\rangle_{q(x_i)} - 2 \operatorname{Re}\Big( y_i^* \langle x_i \rangle_{q(x_i)} [\boldsymbol{\Phi}\boldsymbol{\alpha}]_i \Big) \Big) \Big) \tag{5.40}$$

$$= \exp\Big( - \langle \lambda \rangle_{q(\lambda)} \Big( |\mathbf{r}_i \boldsymbol{\alpha}|^2 \left\langle |x_i|^2 \right\rangle_{q(x_i)} - 2 \operatorname{Re}\Big( y_i^* \langle x_i \rangle_{q(x_i)} \mathbf{r}_i \boldsymbol{\alpha} \Big) \Big) \Big) \tag{5.41}$$

for all $i \in \mathcal{D}$, where $\mathbf{r}_i \in \mathbb{C}^{1\times(N+M)}$ is the $i$th row of $\boldsymbol{\Phi}$. Similarly we get

$$m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \propto \exp\Big(\left\langle \log f_{\mathrm{P}_j}(\boldsymbol{\alpha},\lambda)\right\rangle_{q(\lambda)}\Big) \tag{5.42}$$

$$\propto \exp\Big(-\left\langle\lambda\right\rangle_{q(\lambda)}\left(|\mathbf{r}_j\boldsymbol{\alpha}|^2|x_j|^2 - 2\,\mathrm{Re}\left(y_j^* x_j \mathbf{r}_j\boldsymbol{\alpha}\right)\right)\Big) \tag{5.43}$$

for all $j \in \mathcal{P}$. Given the belief $q(\boldsymbol{\gamma})$ the message from the factor node $f_{\boldsymbol{\alpha}}$ reads

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\alpha}}\to\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \propto \exp\Big(\left\langle \log f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha},\boldsymbol{\gamma})\right\rangle_{q(\boldsymbol{\gamma})}\Big) \tag{5.44}$$

$$\propto \exp\Big(-\boldsymbol{\alpha}^{\mathrm{H}}\left\langle\boldsymbol{\Gamma}^{-1}\right\rangle_{q(\boldsymbol{\gamma})}\boldsymbol{\alpha}\Big) \tag{5.45}$$

The belief $q(\boldsymbol{\alpha})$ is proportional to the product of the messages (5.41), (5.43) and (5.45)

$$\begin{aligned} q(\boldsymbol{\alpha}) &\propto \exp\left(-\boldsymbol{\alpha}^{\mathrm{H}}\left(\left\langle\sum_{i=1}^{N+M}\left[\mathbf{r}_i^{\mathrm{H}}\mathbf{r}_i|x_i|^2\lambda\right] + \boldsymbol{\Gamma}^{-1}\right\rangle\right)\boldsymbol{\alpha}\right) \\ &\quad\cdot \exp\left(\left\langle\lambda\right\rangle 2\left\langle\sum_{i=1}^{N+M}\mathrm{Re}(y_i^* x_i \mathbf{r}_i\boldsymbol{\alpha})\right\rangle\right) \qquad (5.46) \\ &= \exp\Big(-\boldsymbol{\alpha}^{\mathrm{H}}\left(\boldsymbol{\Phi}^{\mathrm{H}}\left\langle\mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle\boldsymbol{\Phi}\left\langle\lambda\right\rangle + \left\langle\boldsymbol{\Gamma}^{-1}\right\rangle\right)\boldsymbol{\alpha} + \left\langle\lambda\right\rangle 2\,\mathrm{Re}(\mathbf{y}^{\mathrm{H}}\left\langle\mathbf{X}\right\rangle\boldsymbol{\Phi}\boldsymbol{\alpha})\Big) \qquad (5.47) \\ &\propto \mathrm{CN}\left(\boldsymbol{\alpha};\boldsymbol{\mu}_{\boldsymbol{\alpha}},\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\right) \qquad (5.48) \end{aligned}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathrm{H}}\left\langle\mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\prod_{i\in\mathcal{D}}q(x_i)}\boldsymbol{\Phi}\left\langle\lambda\right\rangle_{q(\lambda)} + \left\langle\boldsymbol{\Gamma}^{-1}\right\rangle_{q(\boldsymbol{\gamma})}\right)^{-1} \tag{5.49}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \left\langle\lambda\right\rangle_{q(\lambda)}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}^{\mathrm{H}}\left\langle\mathbf{X}\right\rangle^{\mathrm{H}}_{\prod_{i\in\mathcal{D}}q(x_i)}\mathbf{y} \tag{5.50}$$

### 5.4.2 Noise Precision

The messages to the noise precision variable node are given by

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}(\lambda) \propto \lambda\exp\Big(-\lambda\left\langle|y_i - \mathbf{r}_i\boldsymbol{\alpha}x_i|^2\right\rangle_{q(x_i)q(\boldsymbol{\alpha})}\Big),\ \forall\, i\in\mathcal{D} \tag{5.51}$$

$$m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\lambda}(\lambda) \propto \lambda\exp\Big(-\lambda\left\langle|y_j - \mathbf{r}_j\boldsymbol{\alpha}x_j|^2\right\rangle_{q(\boldsymbol{\alpha})}\Big),\ \forall\, j\in\mathcal{P} \tag{5.52}$$

$$m^{\mathrm{BP}}_{f_\lambda\to\lambda}(\lambda) \propto \lambda^{a-1}\exp(-b\lambda) \tag{5.53}$$

The belief $q(\lambda)$ is the product of the messages (5.51), (5.52) and (5.53)

$$q(\lambda) \propto \lambda^{a+M+N-1}\exp\left(-\lambda\left(b + \sum_{i=1}^{N+M}\left\langle|y_i - x_i\mathbf{r}_i\boldsymbol{\alpha}|^2\right\rangle_{q(\boldsymbol{\alpha})\prod_{i\in\mathcal{D}}q(x_i)}\right)\right) \tag{5.54}$$

$$\propto \mathrm{Ga}\left(\lambda; a+M+N, b+\left\langle||\mathbf{y}-\mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha}||_2^2\right\rangle_{q(\boldsymbol{\alpha})\prod_{i\in\mathcal{D}}q(x_i)}\right) \tag{5.55}$$

And the first moment is thus

$$\left\langle\lambda\right\rangle_{q(\lambda)} = \frac{a+M+N}{b+\left\langle||\mathbf{y}-\mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha}||_2^2\right\rangle_{q(\boldsymbol{\alpha})\prod_{i\in\mathcal{D}}q(x_i)}} \tag{5.56}$$

### 5.4.3 Data Symbols

Given $q(\boldsymbol{\alpha})$ and $q(\lambda)$ we can compute the messages

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i) \propto \exp\Big(\langle \log f_{\mathrm{D}_i}(\boldsymbol{\alpha}, x_i, \lambda)\rangle_{q(\alpha),q(\lambda)}\Big) \tag{5.57}$$

$$\propto \exp\Big(-\langle\lambda\rangle_{q(\lambda)}\left\langle |y_i - x_i\mathbf{r}_i\boldsymbol{\alpha}|^2\right\rangle_{q(\boldsymbol{\alpha})}\Big) \tag{5.58}$$

$$\propto \mathrm{CN}\left(x_i; y_i\frac{(\mathbf{r}_i\langle\boldsymbol{\alpha}\rangle_{q(\boldsymbol{\alpha})})^*}{\langle|\mathbf{r}_i\boldsymbol{\alpha}|^2\rangle_{q(\boldsymbol{\alpha})}}, \frac{1}{\langle\lambda\rangle_{q(\lambda)}\langle|\mathbf{r}_i\boldsymbol{\alpha}|^2\rangle_{q(\boldsymbol{\alpha})}}\right) \tag{5.59}$$

where $\langle|\mathbf{r}_i\boldsymbol{\alpha}|^2\rangle_{q(\boldsymbol{\alpha})} = \mathbf{r}_i\left(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} + \boldsymbol{\mu}_{\boldsymbol{\alpha}}\boldsymbol{\mu}_{\boldsymbol{\alpha}}^{\mathrm{H}}\right)\mathbf{r}_i^{\mathrm{H}}$ for all $i \in \mathcal{D}$. As in Chapter 4 the beliefs $q(x_i)$ are thus given by

$$q(x_{i_n}) \propto m^{\mathrm{BP}}_{f_{\mathrm{M}_n}\to x_{i_n}}(x_{i_n}) \cdot m^{\mathrm{MF}}_{f_{\mathrm{D}_{i_n}}\to x_{i_n}}(x_{i_n})\ \forall n \in [1:N] \tag{5.60}$$

### 5.4.4 Hyper-Parameters

The messages to variable node $\boldsymbol{\gamma}$ are

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\alpha}}\to\boldsymbol{\gamma}}(\boldsymbol{\gamma}) \propto \exp\Big(\langle\log f_{\boldsymbol{\alpha}}(\boldsymbol{\alpha},\boldsymbol{\gamma})\rangle_{q(\boldsymbol{\alpha})}\Big) \tag{5.61}$$

$$\propto \prod_{\ell=1}^{L}\gamma_\ell^{-1}\exp\Big(-\gamma_\ell^{-1}\left\langle|\alpha_\ell|^2\right\rangle_{q(\boldsymbol{\alpha})}\Big) \tag{5.62}$$

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\gamma}}\to\boldsymbol{\gamma}}(\boldsymbol{\gamma}) \propto \prod_{\ell=1}^{L}\gamma_\ell^{\varepsilon-1}\exp\Big(-\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}\gamma_\ell\Big) \tag{5.63}$$

Computing the product of the messages yields

$$q(\boldsymbol{\gamma}) \propto \prod_{\ell=1}^{L}\gamma_\ell^{\varepsilon-2}\exp\Big(-\gamma_\ell^{-1}\left\langle|\alpha_\ell|^2\right\rangle_{q(\boldsymbol{\alpha})} - \gamma_\ell\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}\Big) \tag{5.64}$$

The belief $q(\boldsymbol{\gamma})$ is thus a product of generalized inverse Gaussian pdfs [58] of order $p = \varepsilon - 1$. The moments for any $n \in \mathbb{R}$ are given by [58]

$$\langle\gamma_\ell^n\rangle_{q(\boldsymbol{\gamma})} = \left(\frac{\langle|\alpha_\ell|^2\rangle_{q(\boldsymbol{\alpha})}}{\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}}\right)^{\frac{n}{2}}\frac{K_{p+n}\left(2\sqrt{\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}\langle|\alpha_\ell|^2\rangle_{q(\boldsymbol{\alpha})}}\right)}{K_p\left(2\sqrt{\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}\langle|\alpha_\ell|^2\rangle_{q(\boldsymbol{\alpha})}}\right)} \tag{5.65}$$

where $K_\nu(\cdot)$ is the modified Bessel function of the second kind with order $\nu \in \mathbb{R}$. In a practical implementation it may be inconvenient to evaluate the modified Bessel function. To avoid this we can restrict the belief of $\boldsymbol{\gamma}$ to a Dirac delta function at the mode of (5.64), hence obtain EM type updates for $q(\boldsymbol{\gamma})$

$$q_{\mathrm{EM}}(\boldsymbol{\gamma}) = \prod_{\ell=1}^{L}\delta(\gamma_\ell - D_\ell) \tag{5.66}$$

where

$$D_\ell = \frac{(\varepsilon - 2) + \sqrt{(\varepsilon-2)^2 + 4\left\langle \eta_\ell \right\rangle_{q(\boldsymbol{\eta})} \left\langle |\alpha_\ell|^2 \right\rangle_{q(\boldsymbol{\alpha})}}}{2\left\langle \eta_\ell \right\rangle_{q(\boldsymbol{\eta})}} \tag{5.67}$$

Finally the messages to the variable node $\boldsymbol{\eta}$ are given by

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\gamma}}\to\boldsymbol{\eta}}(\boldsymbol{\eta}) \propto \prod_{\ell=1}^{L} \eta_\ell^{\varepsilon} \exp\Big(-\eta_\ell (\left\langle \gamma_\ell \right\rangle_{q(\boldsymbol{\gamma})})\Big) \tag{5.68}$$

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\eta}}\to\boldsymbol{\eta}}(\boldsymbol{\eta}) \propto \prod_{\ell=1}^{L} \eta_\ell^{c-1} \exp(-\eta_\ell d) \tag{5.69}$$

The belief $q(\boldsymbol{\eta})$ is obtained by computing the product of the messages (5.68) and (5.69)

$$q(\boldsymbol{\eta}) \propto \prod_{\ell}^{L} \eta_i^{\varepsilon+c-1} \exp\Big(-\eta_\ell\left(\left\langle \gamma_\ell \right\rangle_{q(\boldsymbol{\gamma})} + d\right)\Big) \tag{5.70}$$

The belief $q(\boldsymbol{\eta})$ is thus a product of $L$ gamma pdfs. The first moment is computed as

$$\left\langle \eta_\ell \right\rangle_{q(\boldsymbol{\eta})} = \frac{\varepsilon + c}{\left\langle \gamma_\ell \right\rangle_{q(\boldsymbol{\gamma})} + d} \tag{5.71}$$

The messages for the modulation, interleaving and coding subgraphs are computed as in Section 4.4.

### 5.4.5 Using Pilots Only

For comparison we want to investigate a similar algorithm in which only the pilot symbols are used for sparse channel estimation as in [13, 18]. The algorithm in [18] uses the same hierarchical prior model (three-layer version) as presented here and we thus obtain a very similar algorithm. It is obtained from the graphical model presented in this chapter, simply by replacing the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\boldsymbol{\alpha}}$ and $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}$ for all $i \in \mathcal{D}$ with constant messages. The update expressions for the parameters $(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}, \boldsymbol{\mu}_{\boldsymbol{\alpha}})$ of $q(\boldsymbol{\alpha})$ can be obtained from (5.49) and (5.50) by setting $\left\langle x_i \right\rangle_{q(x_i)} = \left\langle |x_i|^2 \right\rangle_{q(x_i)} = 0$ for all $i \in \mathcal{D}$, which is equivalent to

$$\boldsymbol{\Sigma}^{\mathcal{P}}_{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathrm{H}}_{\mathcal{P}}\mathbf{X}^{\mathrm{H}}_{\mathcal{P}}\mathbf{X}_{\mathcal{P}}\boldsymbol{\Phi}_{\mathcal{P}}\left\langle \lambda \right\rangle_{q(\lambda)} + \left\langle \boldsymbol{\Gamma}^{-1} \right\rangle_{q(\boldsymbol{\gamma})}\right)^{-1} \tag{5.72}$$

$$\boldsymbol{\mu}^{\mathcal{P}}_{\boldsymbol{\alpha}} = \left\langle \lambda \right\rangle_{q(\lambda)} \boldsymbol{\Sigma}^{\mathcal{P}}_{\boldsymbol{\alpha}}\boldsymbol{\Phi}^{\mathrm{H}}_{\mathcal{P}}\mathbf{X}^{\mathrm{H}}_{\mathcal{P}}\mathbf{y}_{\mathcal{P}} \tag{5.73}$$

where $\boldsymbol{\Phi}_{\mathcal{P}}$ is the matrix consisting of the rows of $\boldsymbol{\Phi}$ indexed by $\mathcal{P}$, $\mathbf{y}_{\mathcal{P}}$ is the column vector with the entries of $\mathbf{y}$ that corresponds to $\mathcal{P}$ and $\mathbf{X}_{\mathcal{P}}$ is the diagonal matrix consisting of the diagonal entries of $\mathbf{X}$ indexed by $\mathcal{P}$. In addition the belief $q(\lambda)$ becomes

$$q_{\mathcal{P}}(\lambda) = \mathrm{Ga}\left(\lambda; M + a, b + \left\langle ||\mathbf{y}_{\mathcal{P}} - \mathbf{X}_{\mathcal{P}}\boldsymbol{\Phi}_{\mathcal{P}}\boldsymbol{\alpha}||_2^2 \right\rangle_{q(\boldsymbol{\alpha})}\right) \tag{5.74}$$

We also employ channel estimation using only the pilot symbols in the first iteration of the data-aided receiver algorithm, because no feedback from the modulation and coding part is available in this iteration.

## 5.5. Iterative Algorithm and Scheduling

The initialization and scheduling of messages are also important design parameters when using a hierarchical prior model. We propose the following scheme

1) If $\lambda$ is unknown initalize $\langle\lambda\rangle_{q(\lambda)}$ using a heuristic scheme.
2) Initialize $\langle\mathbf{\Gamma}^{-1}\rangle_{q(\boldsymbol{\gamma})}$ and $\langle\eta_\ell\rangle_{q(\boldsymbol{\eta})}$ for all $\ell \in [1:L]$.
3) Set $\langle x_i\rangle_{q(x_i)} = \langle|x_i|^2\rangle_{q(x_i)} = 0$ for all $i \in \mathcal{D}$.
4) Perform several sub-iterations to update beliefs of the hierarchical prior model:
   a) Update belief $q(\boldsymbol{\alpha})$ using (5.49) and (5.50).
   b) Update belief $q(\boldsymbol{\gamma})$ and compute $\langle\boldsymbol{\gamma}\rangle_{q(\boldsymbol{\gamma})}$ and $\langle\boldsymbol{\gamma}^{-1}\rangle_{q(\boldsymbol{\gamma})}$ using either (5.65) or (5.66) (EM update).
   c) Update belief $q(\boldsymbol{\eta})$ and compute $\langle\boldsymbol{\eta}\rangle_{q(\boldsymbol{\eta})}$ using (5.71).
   d) If $\lambda$ is unknown, update $q(\lambda)$ and compute $\langle\lambda\rangle_{q(\lambda)}$ using (5.56).
5) Compute the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i)$ for all $i \in \mathcal{D}$ using (5.59).
6) Compute one iteration of the modulation, interleaving and coding subgraph yielding the belief of the information bits $q(u_k)$ for all $k \in [1:K]$. Terminate the algorithm if the stopping criterion is fulfilled.
7) Using the messages from the subgraph $\left\{n_{c_l^{(n)}\to f_{\mathrm{M}_n}}(c_l^{(n)})\,|\, l \in [1:Q], n \in [1:N]\right\}$ as extrinsic values compute $m_{f_{\mathrm{M}_n}\to x_{i_n}}(x_{i_n})$ for all $n \in [1:N]$ using (4.66).
8) Update $q(x_i)$ for all $i \in \mathcal{D}$ using (5.60).
9) Proceed from step (4).

Again, the first step is to update the channel estimate, which ultimately means the belief $q(\boldsymbol{\alpha})$. Setting $\langle x_i\rangle_{q(x_i)} = \langle|x_i|^2\rangle_{q(x_i)} = 0$ for all $i \in \mathcal{D}$ corresponds to only using the pilot symbols for channel estimation in the first iteration. The hierarchical model implies that it may be beneficial to do several iterations within the channel estimation subgraph (step (4)) in between updating the beliefs of the modulation, interleaving and coding subgraph. In the simulations in Section 6.3 we use 20 iterations in the channel estimation sub-graph within the first outer iteration and 5 iterations for every following outer iteration. The idea is to first obtain a fairly accurate estimate of the channel using pilots only before including the data to refine this estimate. The number of iterations using pilots only should not be so high that the estimate can not be refined, for example we want to avoid that several of the weights $\boldsymbol{\alpha}$ and the hyper-parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ are being 'pruned' from the model before including the data in the channel estimation.

Pruning is a detail in the implementation in which columns of the dictionary $\boldsymbol{\Phi}$ as well as the corresponding entries of $\boldsymbol{\gamma}$, $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$ are 'erased'. This is necessary in order to avoid ill-conditioning of the matrix inversion in (5.49), see for example [16, appendix B]. In our implementation we prune the parameters corresponding to index $i$ when $\left\langle \gamma_i^{-1} \right\rangle > \frac{1}{100\epsilon}$ where $\epsilon$ is the machine precision ($\epsilon \approx 2.22 \cdot 10^{-16}$ in the Matlab implementation). In addition to avoiding numerical problems, pruning reduces the computational complexity of the later iterations as the number of columns in $\boldsymbol{\Phi}$ decreases.

Because we are using a hierarchical prior model, we also need initial settings of $\langle \boldsymbol{\Gamma}^{-1} \rangle_{q(\boldsymbol{\gamma})}$ and $\langle \eta_\ell \rangle_{q(\boldsymbol{\eta})}$. Note, that when using the two-layer model the belief of $\boldsymbol{\eta}$ is not updated, but fixed to an initial value. When using pilots only in all iterations we do not utilize the messages from the modulation, interleaving and coding subgraph and there is thus no need to iterate between this subgraph and the channel estimation subgraph.

## 5.6. Pilot Spacing

In Subsection 5.4.5 we have particularized our proposed receiver to a receiver algorithm using only the pilot symbols for channel estimation. We now investigate how the pilot spacing affects the dictionary of the compressed sensing problem.

Similar to the analysis of the correlation between the columns of the dictionary in the discussion of delay resolution in Section 5.1.2 we can derive an expression for the correlation between two columns of the reduced dictionary as shown in Appendix E. By reduced dictionary we mean the matrix $\boldsymbol{\Phi}_A$ that is the concatenation of every $A$th row from the original dictionary $\boldsymbol{\Phi}$. This corresponds to the dictionary used when using evenly spaced pilots, i.e. $\mathcal{P} = [1, 1{+}A, 1{+}2A \ldots, \lceil \frac{N+M}{A} \rceil]$. From (E.12) we see that for $|t_k - t_\ell| = [\frac{1}{\Delta f A}, \frac{2}{\Delta f A}, \ldots]$ the normalized absolute correlation between the two columns corresponding to $t_k$ and $t_\ell$ will be 1. This is illustrated in Figure 5.3 for different values of the pilot spacing $A$.

When increasing $A$ we obtain that columns of the dictionary corresponding to completely different path delays become fully correlated. The number of observations is insufficient to distinguish between the channel's multipath components with different path delays and in this case we do not expect the compressed sensing recovery algorithms to be successful. If the maximum multipath delay $\tau_{\max}$ is small, a larger pilot spacing $A$ can be used before this problem of ambiguity arises. For example using $A = 10$ with

$\tau_{\max} = 5.2\,\mu s$ we do not see this problem, as illustrated with the solid line on Figure 5.3.

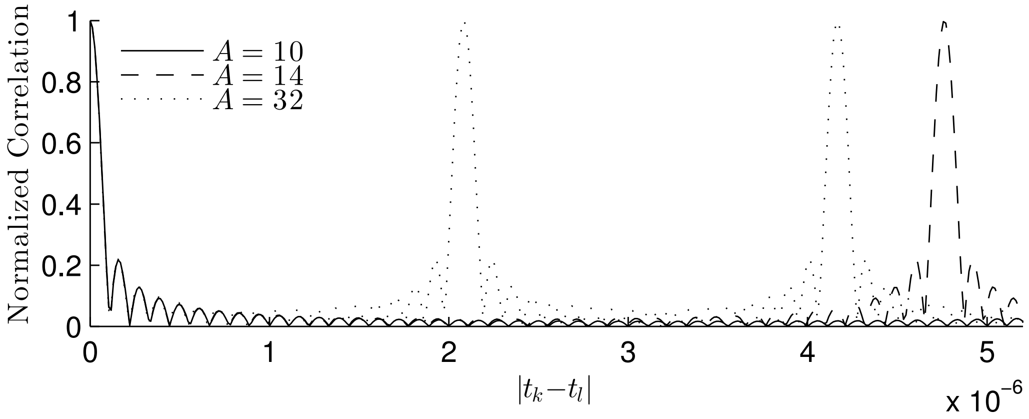


**Figure 5.3:** *Normalized correlation between two columns of the reduced dictionary* $\frac{|\boldsymbol{\psi}_A(t_k)^{\mathrm{H}}\boldsymbol{\psi}_A(t_\ell)|}{||\boldsymbol{\psi}_A t_k||_2 ||\boldsymbol{\psi}_A(t_\ell)||_2}$ *as a function of their relative delay* $|t_k - t_l|$. *The reduced dictionary consists of every* $A$*th row of the original dictionary. The number of subcarriers* $N + M = 600$ *and* $|t_k - t_l| \in (0, \tau_{\max} = 5.2\,\mu s)$.

CHAPTER 6

# Performance Evaluation

To evaluate the performance of the proposed receiver with sparse channel estimation we perform a number of simulations using the developed OFDM simulation framework described in Appendix H. Direct comparison is made with the more conventional iterative receiver described in Chapter 4. To distinguish between the two receiver schemes we (with abuse of notation) refer to the receiver exploiting the sparse channel assumption as the ‘sparsity-aware receiver’ and to the receiver devised in Chapter 4 as the ‘frequency-domain receiver’.

In Section 6.1 we discuss the channel models used for the simulations and their relation to the choice of covariance matrix for the prior of the channel coefficients $\mathbf{h}$ used by the receiver in Chapter 4. The details of the algorithms we evaluate are given in Section 6.2 followed by simulation results for different scenarios in Section 6.3.

## 6.1. Channel Model

To perform simulations of the devised receiver algorithms we need an appropriately defined channel model. We consider two different methods for specifying the characteristics of the multipath channel model introduced in Section 2.2. The first method suits the standardized 3GPP LTE reference channel models [10] and the second method is the exponentially decaying channel model used in e.g. [18, 59].

### 6.1.1 3GPP Channel Models

Returning to the multipath channel (2.6) and following the notation of Section 2.2, the 3GPP LTE reference channels directly specifies the delays $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_P)^{\mathrm{T}}$ and the average powers $\mathbf{s} = (\sigma_1^2, \cdots, \sigma_P^2)^{\mathrm{T}}$ of each multipath component, thus $p(\boldsymbol{\beta}) = \mathrm{CN}(\boldsymbol{\beta}; \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ with $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \mathrm{diag}(\mathbf{s})$. The power delay profiles for the channel models are shown in Table 6.1. Given $\boldsymbol{\tau}$ and $\mathbf{s}$ each $h_i$ is a sum of independent Gaussian random variables, hence $h_i$ is also Gaussian distributed. In the following we compute the mean and covariance

| $p$ | $\tau_p$ [ns] | $\sigma_p^2$ [dB] |
|---|---|---|
| 1 | 0 | 0.0 |
| 2 | 30 | -1.0 |
| 3 | 70 | -2.0 |
| 4 | 90 | -3.0 |
| 5 | 110 | -8.0 |
| 6 | 190 | -17.2 |
| 7 | 410 | -20.8 |

**(a)** *EPA*

| $p$ | $\tau_p$ [ns] | $\sigma_p^2$ [dB] |
|---|---|---|
| 1 | 0 | -0.0 |
| 2 | 30 | -1.5 |
| 3 | 150 | -1.4 |
| 4 | 310 | -3.6 |
| 5 | 370 | -0.6 |
| 6 | 710 | -9.1 |
| 7 | 1090 | -7.0 |
| 8 | 1730 | -12.0 |
| 9 | 2510 | -16.9 |

**(b)** *EVA*

| $p$ | $\tau_p$ [ns] | $\sigma_p^2$ [dB] |
|---|---|---|
| 1 | 0 | -1 |
| 2 | 50 | -1 |
| 3 | 120 | -1 |
| 4 | 200 | 0 |
| 5 | 230 | 0 |
| 6 | 500 | 0 |
| 7 | 1600 | -3 |
| 8 | 2300 | -5 |
| 9 | 5000 | -7 |

**(c)** *ETU*

**Table 6.1:** *Power delay profiles of the 3GPP channel models Extended Pedestrian A (EPA), Extended Vehicular A (EVA) and Extended Typical Urban (ETU) [10].*

of $\mathbf{h}$. Each $\beta_p$ is independent of the others and has zero mean, therefore

$$\langle h_i \rangle = \sum_{p=1}^{P} \langle \beta_p \rangle \exp(-j2\pi\Delta f \tau_p i) = 0, \ i \in [1 : M + N] \tag{6.1}$$

The entries of the covariance matrix of $\mathbf{h}$ are thus given by

$$\begin{aligned}[\mathbf{\Sigma_h}]_{i,k} &= \langle h_i h_k^* \rangle &(6.2)\\ &= \sum_{p=1}^{P}\sum_{q=1}^{P} \left\langle \beta_p \beta_q^* \right\rangle \exp(-j2\pi\Delta f(\tau_p i - \tau_q k)) &(6.3)\\ &= \sum_{p=1}^{P} \sigma_p^2 \exp(-j2\pi\Delta f \tau_p (i-k)) &(6.4)\end{aligned}$$

for all $i, k \in [1 : M + N]$. Using the parametric matrix description in (5.6) ($\mathbf{h} = \mathbf{\Psi}(\boldsymbol{\tau})\boldsymbol{\beta}$) and by using that a linear transformation of a complex Gaussian random vector is also complex Gaussian [60, appendix 15B], we obtain

$$\mathbf{h} \sim \mathrm{CN}\left(\mathbf{h}; \mathbf{0}, \mathbf{\Psi}(\boldsymbol{\tau})\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}\right) \tag{6.5}$$

Since $\mathrm{rank}(\mathbf{\Sigma}_{\boldsymbol{\beta}}) = P$ we have that $\mathrm{rank}(\mathbf{\Psi}(\boldsymbol{\tau})\mathbf{\Sigma}_{\boldsymbol{\beta}}\mathbf{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}) \leq P$. For the sparse channel models with fixed delays we have $N + M > P$, and $\mathbf{h}$ is thus a degenerate multi-variate complex Gaussian random variable.

The 3GPP channel models are used as reference to ease comparison with other receivers in the literature. However, it is not a realistic assumption that the delays of the multipath components are fixed. If the implemented algorithms were only tested against these channel models, one could unintentionally exploit the specific configuration of the parameters. This motivates us to additionally consider other multipath channel models with a higher degree of uncertainty in the parameters.

### 6.1.2 Exponentially Decaying Power Delay Profile

As opposed to the 3GPP channel models, we will now also model the number of multipath components and the delays as random variables. Similar to the approach in [18,59] we employ an exponentially decaying power delay profile, where the number of taps and the tap delays are unknown. It is assumed that we know the maximum tap delay $\tau_{\text{max}}$. This assumption is justified as the cyclic prefix needs to be dimensioned such that the multipath components extending beyond the cyclic prefix length are insignificant to avoid inter-symbol-interference. We write the pdf of the unknown parameters as

$$p(\boldsymbol{\beta}, \boldsymbol{\tau}, P) = p(\boldsymbol{\beta}|\boldsymbol{\tau})p(\boldsymbol{\tau}|P)p(P) \tag{6.6}$$

where $P$ is a discrete, positive random variable with mean $\mu_P$, $p(\boldsymbol{\tau}|P)$ is the product of $P$ continuous uniform pdfs over the interval $(0; \tau_{\text{max}})$ and

$$p(\boldsymbol{\beta}|\boldsymbol{\tau}) = \prod_{p=1}^{P} \text{CN}\left(\beta_p; 0, u \cdot \exp\left(-\frac{\tau_p}{v}\right)\right). \tag{6.7}$$

The parameters $u$ and $v$ respectively govern the scale and decay rate of the power delay profile. In a simulation setup realizations of $\mathbf{h}$ can be obtained by drawing samples of the parameter $P$, then $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_P)^{\text{T}}$ and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_P)^{\text{T}}$ and finally applying equation (2.6). By letting $v \to \infty$ a flat power delay profile is obtained. Because of the more complicated joint distribution (6.6) it is no longer straight-forward to determine the pdf of the channel weights $\mathbf{h}$. However, we can still determine the first and second moments exactly. The mean of the channel weights is given by

$$\begin{aligned}
\langle h_i \rangle &= \left\langle \sum_{p=1}^{P} \left\langle \langle \beta_p \rangle_{p(\boldsymbol{\beta}|\boldsymbol{\tau})} \exp(-j2\pi\Delta f \tau_p i) \right\rangle_{p(\boldsymbol{\tau}|P)} \right\rangle_{p(P)} && (6.8)\\
&= \left\langle \sum_{p=1}^{P} \langle 0 \cdot \exp(-j2\pi\Delta f \tau_p i) \rangle_{p(\boldsymbol{\tau}|P)} \right\rangle_{p(P)} && (6.9)\\
&= 0 && (6.10)
\end{aligned}$$

for all $i \in [1 : M + N]$. The derivation of the covariance matrix for both finite $v$ and $v \to \infty$ is given in Appendix D. For finite $v$ we get

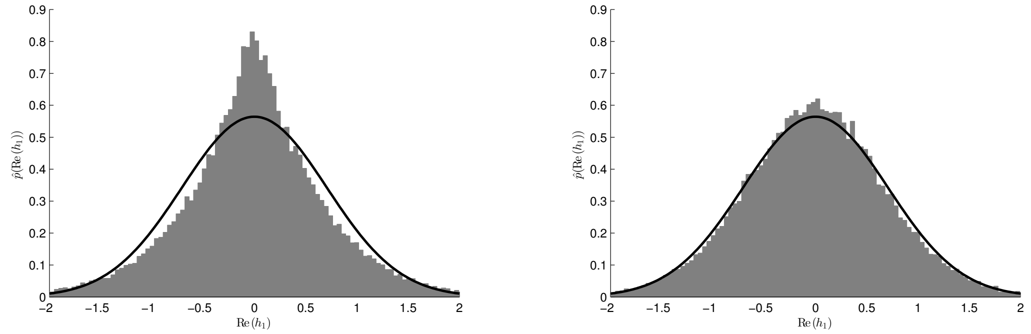
$$[\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,k} = \frac{1 - \exp\left(-\tau_{\text{max}}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)\right)}{v\left(1 - \exp\left(-\frac{\tau_{\text{max}}}{v}\right)\right)\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)}. \tag{6.11}$$

and for $v \to \infty$

$$\lim_{v\to\infty} [\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,k} = \frac{1 - \exp(-\tau_{\text{max}} j2\pi\Delta f(i-k))}{\tau_{\text{max}} j2\pi\Delta f(i-k)} \tag{6.12}$$

The receiver from Chapter 4 requires the prior of the channel coefficients given as $p(\mathbf{h}) = \text{CN}(\mathbf{h}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{h}})$. Since we are using this receiver for comparison with the sparsity-aware receiver we employ the same channel assumption (flat power delay profile and knowledge of $\tau_{\max}$) [1], even though a different channel model is used for the actual simulation. In this case (6.12) is used as the covariance of the prior distribution. The flat power delay profile can be considered as a 'robust' channel assumption when more detailed knowledge of the channel statistics is missing [59, 61].

In some scenarios we also employ the true covariance matrix (6.12) for the prior distribution. However, even in this case the prior does not match true distribution of $\mathbf{h}$ as the true distribution is not Gaussian when using the exponential channel model. The mean and covariance of the prior matches that of the true distribution, and the prior is thus a Gaussian approximation of the true distribution. A histogram of a single frequency domain channel coefficient $h_1$ for two parameterizations of the exponential channel model is shown in Figure 6.1. This numerical example shows that when the average number of multipath components $\mu_P$ grows the Gaussian approximation is more accurate.



**(a)** $\mu_P = 5$

**(b)** $\mu_P = 15$

**Figure 6.1:** *Normalized histograms for the real part of 50000 realizations of the channel coefficient $h_1$ using the exponentially decaying channel model where the number of multipath components $P$ is Poisson distributed with mean $\mu_P$, $v = 1\,\mu\text{s}$, $\tau_{\max} = 5.2\,\mu\text{s}$ and $u$ selected such that $\langle |h_1|^2 \rangle = 1$. The same result can be obtained for the imaginary part of $h_1$ and therefore we do not show this result. The solid black line shows the marginal distribution of the coefficient $h_1$ assumed by using the Gaussian prior.*

In Figure 6.2 we plot complex magnitude of the frequency correlation function $|\langle h_i h_k^* \rangle|$ for the three different 3GPP power delay profiles as well as the frequency correlation function for the exponentially decaying channel model including its parameterization to the robust channel assumptions (flat power delay profile). The assumption of a flat power delay profile in delay domain translates into a frequency correlation function that is decaying rapidly with frequency, i.e. it is assumed that the correlation

[1] In the sparse estimation algorithm the flat power delay assumption appears in the selection of the parameters of the two-layer hierarchical prior distribution. For the two-layer model, selecting $\eta_\ell = \eta$ for all $\ell \in [1 : L]$ gives the same prior distribution $p(\alpha_\ell; \varepsilon, \eta_\ell)$ for all the multipath components, which corresponds to a flat power delay profile. In addition we also employ the same initialization of the beliefs $\left\langle \gamma_\ell^{-1} \right\rangle_{q(\boldsymbol{\gamma})}$ for all $\ell \in [1 : L]$.

between subcarriers is much lower than it is according to the other channel models.

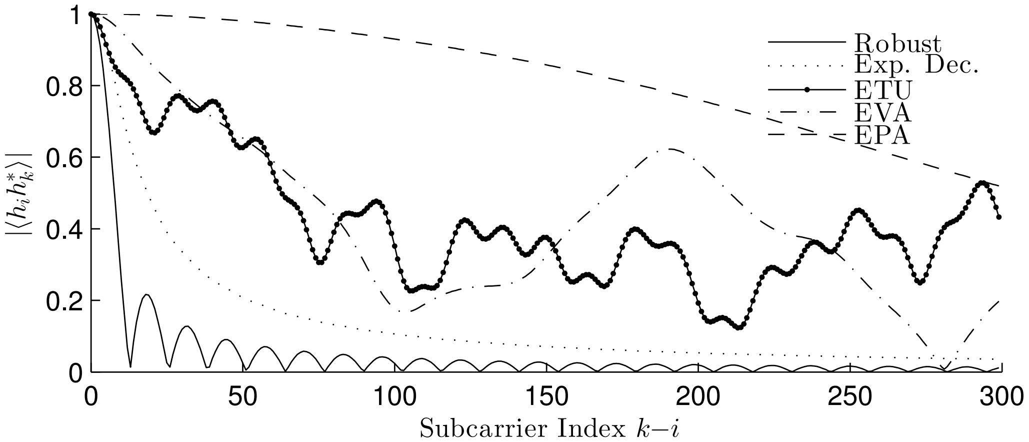


**Figure 6.2:** *Frequency correlation function for the different 3GPP channels and the exponential model (with* $v = 1\,\mu\text{s}$*) and the robust channel assumption (*$v \to \infty$*), in both cases* $\tau_{\max} = 5.2\,\mu\text{s}$.

### 6.1.3 Signal-to-noise Ratio

For simulation purposes it is important to be able to control the signal-to-noise ratio (SNR) the receiver is operating in. From the observation model (2.2) we write the average SNR as

$$
\begin{aligned}
\text{SNR} &= \frac{\left\langle ||\mathbf{Xh}||_2^2 \right\rangle}{\left\langle ||\mathbf{w}||_2^2 \right\rangle} && (6.13)\\
&= \frac{\left\langle \mathbf{h}^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{Xh} \right\rangle}{\lambda^{-1}(M+N)} && (6.14)\\
&= \frac{\operatorname{trace}\left(\left\langle \mathbf{h}^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{Xh} \right\rangle\right)}{\lambda^{-1}(M+N)} && (6.15)\\
&= \frac{\operatorname{trace}\left(\left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\mathbf{h}\mathbf{h}^{\mathrm{H}} \right\rangle\right)}{\lambda^{-1}(M+N)} && (6.16)\\
&= \frac{\operatorname{trace}\left(\left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \right\rangle \left\langle \mathbf{h}\mathbf{h}^{\mathrm{H}} \right\rangle\right)}{\lambda^{-1}(M+N)} && (6.17)\\
&= \frac{\sum_{i=1}^{M+N} \left\langle |x_i|^2 \right\rangle [\mathbf{\Sigma_h}]_{i,i}}{\lambda^{-1}(M+N)} && (6.18)
\end{aligned}
$$

Without loss of generality we can choose to normalize the diagonal elements of $\mathbf{\Sigma_h}$. In case the channel is modelled by a fixed number of multipath components with known average power as in (6.4) the normalization is obtained with $\sum_{p=1}^{P} \sigma_p^2 = 1$. For the exponential channel model the normalization is shown in Appendix D. If we further choose the modulation alphabets $\mathcal{S}_{\mathrm{D}}$ and $\mathcal{S}_{\mathrm{P}}$ such that the modulation symbols have unit average power the SNR expression simplifies to

$$\mathrm{SNR} = \lambda. \tag{6.19}$$

Choosing the modulation alphabets to achieve this can in general be complicated. However, if $\mathbf{u}$ is generated from a binary symmetric source (a good assumption when source coding is employed) we can expect the codewords $\mathbf{c}$ to be equiprobable, hence also the modulation symbols due to the one-to-one and onto mapping $\mathcal{M}(\cdot)$.

## 6.2. Evaluated Algorithms

For the sparsity-aware receiver we use the two-layer version of the hierarchical prior model in all simulations as previous results [33, 48] indicates that the two-layer version has faster convergence and often equivalent estimation accuracy as its three-layer counterparts. Optimal settings for the parameters $\varepsilon$ and $\boldsymbol{\eta}$ of the hierarchical prior depend on the scenario. Some initial simulations show that setting $\varepsilon = 1$ and $\eta_\ell = 1 \, \forall \, \ell \in [1 : L]$ gives good results for the scenarios considered here and this setting of the parameters is thus used throughout all the simulations. The parameters have thus been coarsely tuned to the scenario. Furthermore we use the EM type update (5.66) for the belief $q(\boldsymbol{\gamma})$ to avoid the evaluation of the Bessel function. The algorithm proceeds as described in Section 5.5 with the initialization $\left\langle \gamma_\ell^{-1} \right\rangle_{q(\boldsymbol{\gamma})} = 1$. The first time channel estimation is performed 20 iterations of the channel estimation subgraph are computed (step (4) using pilots only), while in every following iteration 5 iterations are performed in the subgraph.

We include the oracle estimator as described in Appendix G as a reference, as it is often used as a lower bound for the estimation error of sparse estimators [62]. Using the oracle estimator we obtain a point estimate $\hat{\mathbf{h}}$, compute $m^{\mathrm{MF}}_{f_{\mathrm{D}_i} \to x_i}(x_i)$ for all $i \in \mathcal{D}$ and perform 5 iterations of the modulation, interleaving and coding part of the receiver.

For comparison we include the receiver devised in Chapter 4, both with pilots only and data-aided channel estimation. This receiver needs the prior covariance matrix $\boldsymbol{\Sigma}_{\mathbf{h}}$ of the channel weights $\mathbf{h}$. Using a flat power delay profile and knowledge of the maximum propagation delay $\tau_{\max}$ we get the covariance matrix in (6.12). The receiver based on these assumptions is in the following referred to as the 'robust' receiver. In addition we include a reference receiver with perfect knowledge of the channel's second-order statistics. That is, it uses the exact covariance matrix for the prior pdf of $\mathbf{h}$, given in (6.11) for the exponential channel model and in (6.4) for the 3GPP channel models.

In all simulation scenarios it is assumed that the noise precision $\lambda$ is unknown and it is initialized as the inverse of the sample variance as in [18], i.e. $\langle \lambda \rangle_{q(\lambda)} = \frac{N+M-1}{||\mathbf{y}-\bar{\mathbf{y}}||^2}$. In all algorithms the parameters of the prior

$p(\lambda; a, b) = \mathrm{Ga}(\lambda; a, b)$ are set to $a = b = 0$, which corresponds to the (improper) Jeffrey's prior $p(\lambda) \propto \frac{1}{\lambda}$ [63].

## 6.3. Numerical Results

Some parameters of the OFDM simulation scenario are fixed throughout all of the following simulations. These parameters are summarized in Table 6.2. In the simulations of the exponential channel model the number of multipath components $P$ is sampled from a Poisson distribution with mean $\mu_P = 10$ if not otherwise specified.

| | |
|---|---|
| CP length ($\tau_{\max}$) | 5.2 µs |
| Subcarrier spacing ($\Delta f$) | 15 kHz |
| Pilot pattern | Equally spaced, QPSK |
| Data modulation order | 16 QAM |
| Code rate ($R$) | $\frac{1}{2}$, (Polynomial: $[13, 15]_8$) |
| Noise precision ($\lambda$) | Estimated |

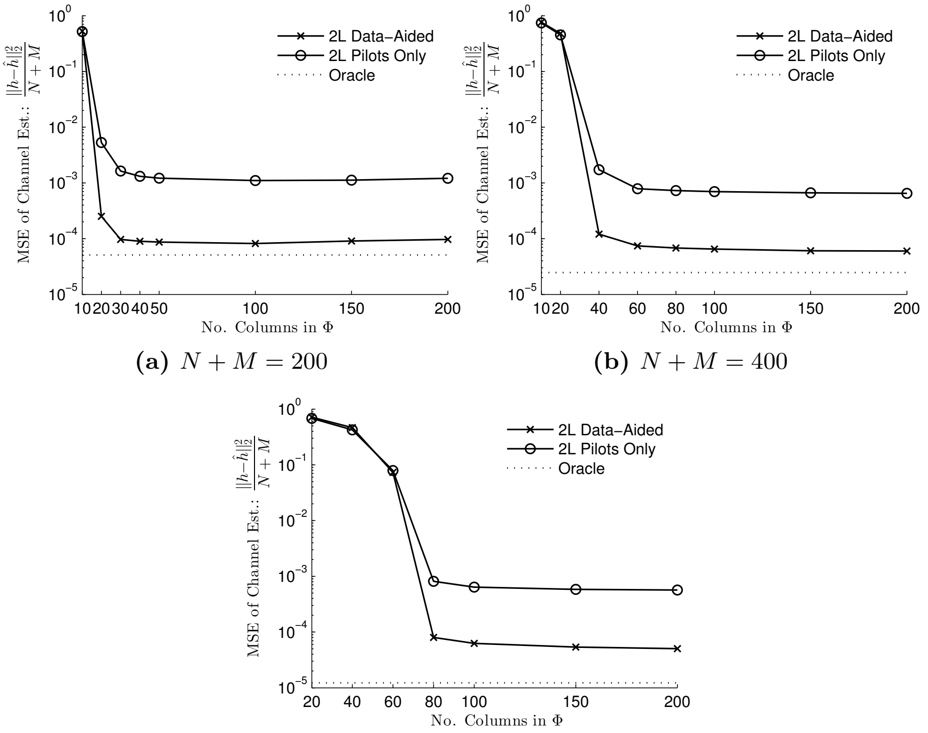
**Table 6.2:** *Common simulation parameters for all simulated scenarios.*

### 6.3.1 Sufficient Delay Resolution

The analysis of the design of the dictionary in Section 5.1, suggested that the number of columns $L$ of the dictionary $\mathbf{\Phi}$ should be proportional to the system bandwidth $\Delta f(N + M)$. Using the parameters in Table 6.3 the mean squared error (MSE) of the channel estimate $\hat{\mathbf{h}} = \mathbf{\Phi} \langle \boldsymbol{\alpha} \rangle_{q(\boldsymbol{\alpha})}$ is plotted versus number of columns for different number of subcarriers in Figure 6.3. In all three plots there is a clear transition after which the MSE of the channel estimate no longer decreases by increasing the number of columns (decreasing $T_{\mathrm{d}}$). The point of the transition is independent of whether we use pilots only or include data in the channel estimation, i.e. the required delay resolution does not depend on the number of observations available to the sparse estimator.

| | |
|---|---|
| Pilot Spacing | 10 |
| Average SNR | 30 dB |
| Channel model | Exp. decay, $\mu_P = 10, v = 1$ µs |

**Table 6.3:** *Simulation parameters for the scenario in which the MSE of the channel estimate is plotted versus the number of columns of the dictionary for different number of subcarriers* $(N + M)$.



**(a)** $N + M = 200$

**(b)** $N + M = 400$

**(c)** $N + M = 800$

**Figure 6.3:** *MSE versus number of columns $L$ of the dictionary $\mathbf{\Phi}$ for different number of subcarriers $N + M$.*

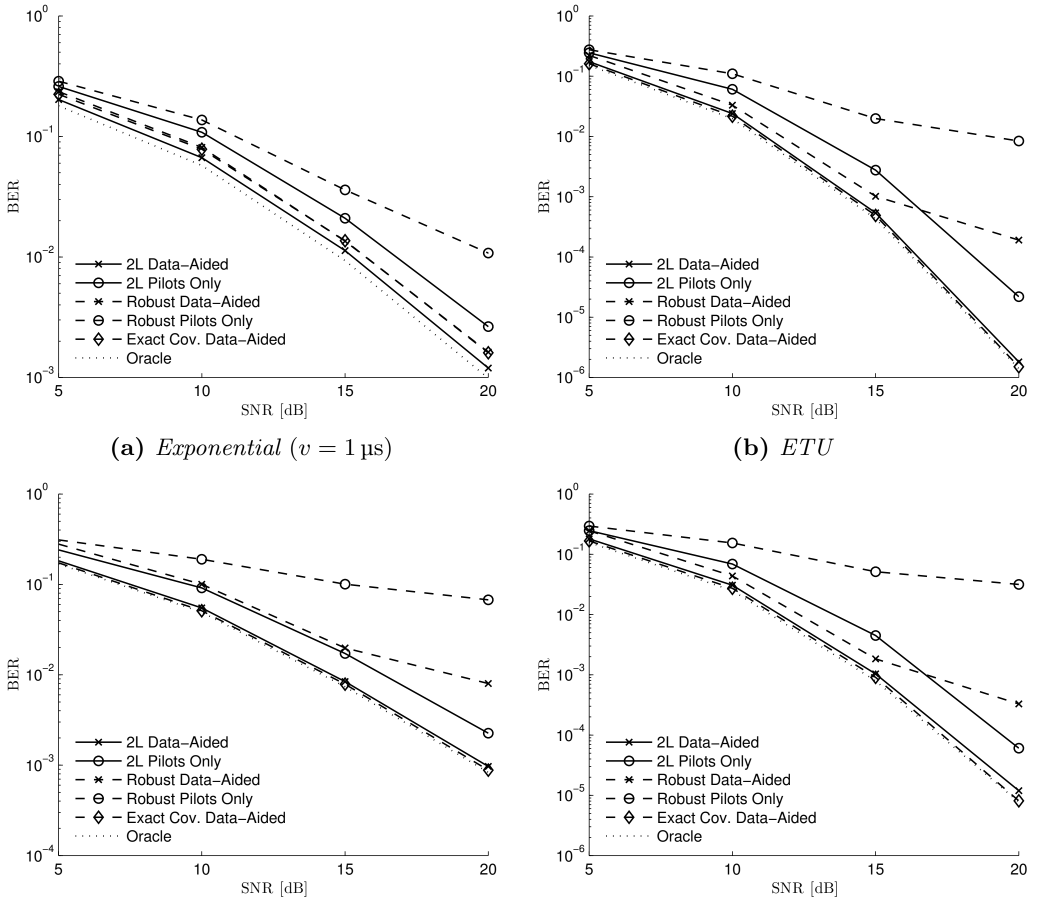
### 6.3.2 Performance versus SNR with Different Channel Models

An important figure of merit for any communication system is the BER versus the SNR. Using the simulation parameters in Table 6.4 and four different channel models (exponential, ETU, EPA, EVA) the BER versus SNR is plotted in Figure 6.4.

For all the 3GPP channel models the BER obtained using the oracle estimator and the estimator of the frequency-domain coefficients $\mathbf{h}$ using the exact covariance matrix coincide. For the exponential channel model the oracle estimator outperforms the other algorithms, but the BER of the data-aided sparse channel estimation algorithm follows that of the oracle estimator closely. It even outperforms the receiver with frequency-domain estimation using the exact covariance matrix by a small margin (less than 1 dB). In general we notice a significant decrease in BER when including

| | |
|---|---|
| Pilot Spacing | 10 |
| No. subcarriers $(N + M)$ | 600 |
| No. columns in $\mathbf{\Phi}$ $(L)$ | 300 |

**Table 6.4:** *Simulation parameters used in the evaluation of the performance of the different receiver schemes versus SNR.*



**(a)** *Exponential* ($v = 1\,\mu s$)

**(b)** *ETU*

**(c)** *EPA*

**(d)** *EVA*

**Figure 6.4:** *BER of the different receiver schemes versus SNR in different 3GPP channel models and the exponential channel model.*

the soft data information in the channel estimation. For the sparsity-aware algorithm the improvements is approximately 2.5 dB, while for the robust receiver inclusion of the data is crucial for acceptable BER, especially at high SNR.

When using the exponential channel model, the data-aided robust receiver shows the same performance as the receiver with knowledge of the exact covariance matrix. By including soft data estimation in the channel estimation the robust receiver is able to compensate for the mismatch between the assumed and true covariance[2]. To limit the number of simulation scenarios we only use the exponential channel model in all of the following simulation scenarios.

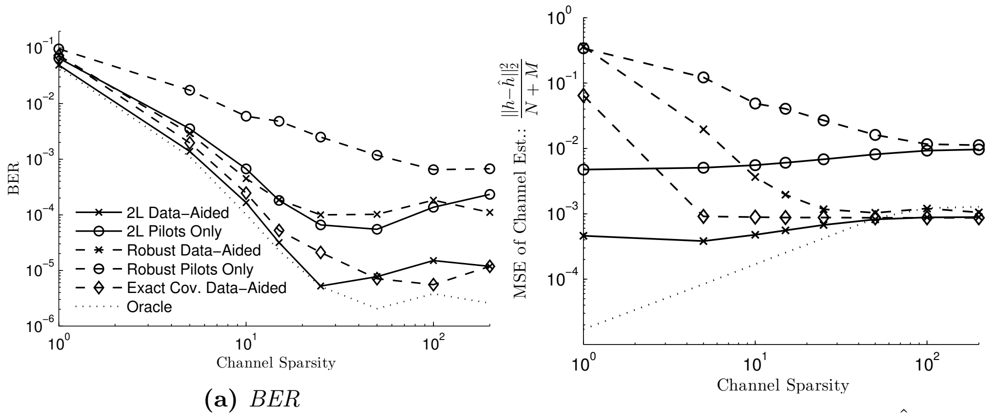
[2] for all the 3GPP channel models it is noticed in Figure 6.4 that the BER curve for the robust data-aided receiver indicates an 'error floor', i.e. the curve flattens for high SNR. Further investigations have shown that this algorithm has slow convergence and needs more iterations in high SNR. For an average SNR of 20 dB it has not converged within the 20 iterations after which the algorithm is terminated. We expect that the BER curves of this algorithm would follow the same trend as in Figure6.4a if it had run until convergence. When using the exponential channel model or lower average SNR the algorithm reaches convergence.

| | |
|---|---|
| Pilot Spacing | 10 |
| No. subcarriers $(N + M)$ | 600 |
| No. columns in $\mathbf{\Phi}$ $(L)$ | 300 |
| Average SNR | 20 dB |
| Channel model | Exp. decay, $P$ fixed, $v = 1\,\mu\text{s}$ |

**Table 6.5:** *Simulation parameters used in the evaluation of the performance of the different receiver schemes versus the sparsity of the channel.*

### 6.3.3 Robustness to Non-sparse Channels

As shown above the sparsity-aware receiver successfully exploits the sparsity of the channel and outperforms the robust receiver, that does not exploit the sparsity of the channel. A question that naturally arises is how sparse the channel should be in order to exploit the sparsity and will the receiver algorithm that assumes sparsity fail if the channel is not sparse? To answer these questions we evaluate the performance of the different receiver algorithms in channels with different levels of sparsity. We employ the exponentially decaying channel but now fix the number of multipath components $P = [1, 5, 10, 15, 25, 50, 100, 200]$. The remaining simulation parameters are found in Table 6.5.



**(a)** *BER*

**(b)** *MSE of channel estimate* $\hat{\mathbf{h}}$

**Figure 6.5:** *BER and MSE performance of the different receivers versus number of multipath components in the channel's impulse response. Note that the legend on (a) is also valid for (b).*

The BER and MSE of the channel estimate are plotted versus the number of multipath components in Figure 6.5. Looking at the MSE of the channel estimate it is noticed that when the channel has only one multipath component the sparsity-aware algorithms outperform the receiver algorithms that do not exploit the sparsity, but the difference in BER is small. With less components the SNR $\frac{\|\mathbf{Xh}\|_2^2}{\|\mathbf{w}\|_2^2}$ fluctuates more between each channel realization, which may lead to higher BER. When the number of multipath components increases the performance of the data-aided sparsity-aware receiver is similar to that of the receiver with knowledge of the exact covariance matrix. It outperforms the data-aided robust algorithm in terms of BER. This result shows that even for channels with many multipath components there is not a significant penalty in terms of BER by assuming that the channel is sparse. It is expected however, that this result will depend on the choice of prior parameters $\varepsilon, \boldsymbol{\eta}$ that controls the sparsity of the estimates of $\boldsymbol{\alpha}$ as discussed in Section 3.2.

### 6.3.4 Reducing the Number of Pilot Symbols

When performing data-aided channel estimation we can consider the data symbols as 'pseudo-pilots', as they play the same role as the pilots symbols in channel estimation, but there is a higher degree of uncertainty associated with the data symbols. This interpretation of data-aided channel estimation suggests that the number of true pilot symbols can be reduced while retaining the same performance. To test this hypothesis we perform a number of simulations with different pilot spacings. The remaining simulation parameters are fixed to the values shown in Table 6.6.The BER curves for this scenario are shown in Figure 6.6. There is a transition in the BER by increasing the pilot spacing from 10 to 14 subcarriers. This is the range in which columns of the reduced dictionary, corresponding to completely different multipath delays, start becoming fully correlated as discussed in Section 5.6. For the sparse channel estimation scheme based on pilots only the transition is sharper than for the data-aided channel estimation scheme as the interpretation of the data symbols as 'pseudo-pilots' suggested.

| | |
|---|---|
| No. subcarriers $(N+M)$ | 600 |
| No. columns in $\mathbf{\Phi}$ $(L)$ | 150 |
| Average SNR | 20 dB |
| Channel model | Exp. decay, $\mu_P = 10, v = 1\,\mu\text{s}$ |

**Table 6.6:** *Simulation parameters used in the evaluation of the performance of the different receiver schemes versus the number of pilot symbols.*

With high spacing between pilots the channel estimation scheme with knowledge of the exact covariance matrix outperforms the other schemes. The algorithms based on the robust channel assumption assume that the correlation between the frequency domain channel coefficients $\mathbf{h}$ is much lower than it actually is as depicted in Figure 6.2.

There is a big gap in the BER performance between the receiver algorithm based on the robust assumption and the one using the exact covariance matrix in scenarios with high pilot spacing. This motivates an investigation of whether incorporating knowledge of the channel's second-order statistics into the sparse channel estimation scheme will lead to an improvement of the same magnitude. One approach would be to adjust the parameters of the hierarchical prior model such that the second moment of

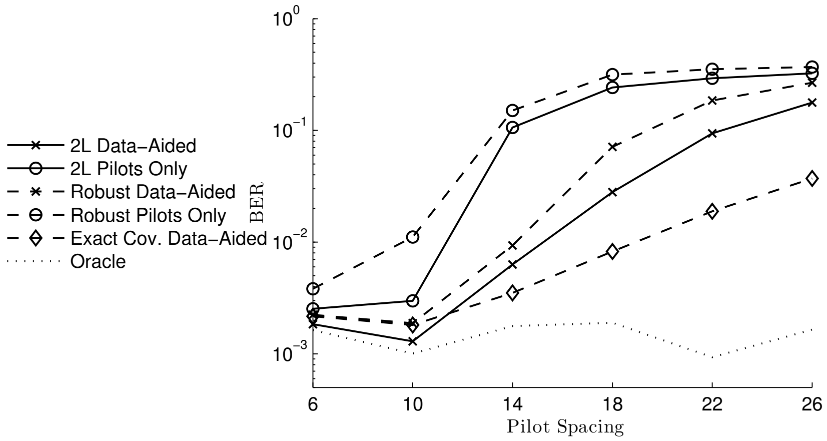


**Figure 6.6:** *BER for the different receiver schemes versus the pilot spacing.*

the marginal prior distribution ($p(\boldsymbol{\alpha}; \varepsilon, \boldsymbol{\eta})$ in the two-layer case) matches the power delay profile of the channel, instead of employing the flat power delay profile assumption ($\eta_\ell = \eta \, \forall \, \ell$).

CHAPTER 7

# Reducing Complexity of Channel Estimation

In this chapter we propose two methods for reducing the computational complexity of the channel estimation task of the iterative receivers presented in Chapter 4 and 5 respectively. We first identify the dominating complexity of both receivers in Section 7.1. Then we propose partial data-aided receivers in Section 7.2, i.e. receivers that do not utilize the soft estimates of all data symbols for channel estimation. A different method is presented in Section 7.3 in which the generalized mean field approximation is employed to reduce the computational complexity. This method also applies to both receiver schemes. The performance of the low-complexity receivers is assessed through Monte Carlo simulations in Section 7.4.

## 7.1. Computing the Matrix Inverse

The computational complexity of both receiver algorithms is dominated by computation of the belief of the channel weights ($q(\mathbf{h})$ and $q(\boldsymbol{\alpha})$ respectively). Therefore the methods we propose seek to reduce the complexity of these particular computations. Each section in the following has a subsection dealing with the update of $q(\mathbf{h})$ for the receiver in Chapter 4 and then a subsection dealing with the update of $q(\boldsymbol{\alpha})$ for the receiver in Chapter 5.

### 7.1.1 Frequency Domain Channel Coefficients

For the receiver in Chapter 4 the operation with highest complexity is the computation of the beliefs $q(h_i)$ for $i \in [1 : N+M]$. As explained in Section 4.4 it requires inversion of the matrix (4.53) of dimension $(N+M)\times(N+M)$ with complexity $\mathcal{O}\left((N+M)^3\right)$.

### 7.1.2 Delay Domain Channel Weights

For the belief $q(\boldsymbol{\alpha})$ the critical computations (5.49), (5.50) are

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathrm{H}} \left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)} \boldsymbol{\Phi} \left\langle \lambda \right\rangle_{q(\lambda)} + \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle_{q(\boldsymbol{\gamma})}\right)^{-1} \tag{7.1}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \left\langle \lambda \right\rangle_{q(\lambda)} \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \boldsymbol{\Phi}^{\mathrm{H}} \left\langle \mathbf{X}\right\rangle^{\mathrm{H}}_{\prod_{i\in\mathcal{D}} q(x_i)} \mathbf{y} \tag{7.2}$$

As common practice we consider computing the matrix inverse of the $L \times L$ matrix in (7.1) as an operation of complexity $\mathcal{O}(L^3)$, even though an algorithm of complexity $\mathcal{O}(L^{2.373})$ exists [49]. Likewise computing $\boldsymbol{\Phi}^{\mathrm{H}} \left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)} \boldsymbol{\Phi}$ is an operation of complexity $\mathcal{O}(L^2(N+M))$. If $(N+M) < L$ it is beneficial to apply the Woodbury matrix inverse formula in appendix C.6 to obtain

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle^{-1} - \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle^{-1} \boldsymbol{\Phi}^{\mathrm{H}} \left( \left\langle \lambda \right\rangle^{-1} \left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X}\right\rangle^{-1} + \boldsymbol{\Phi} \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle^{-1} \boldsymbol{\Phi}^{\mathrm{H}} \right)^{-1} \boldsymbol{\Phi} \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle^{-1} \tag{7.3}$$

which reduces the size of the matrix to be inverted to dimension $(N+M)\times(N+M)$. In turn we need to compute $\boldsymbol{\Phi} \left\langle \boldsymbol{\Gamma}^{-1}\right\rangle^{-1} \boldsymbol{\Phi}^{\mathrm{H}}$ with complexity $\mathcal{O}\left((N+M)^2 L\right)$. Thus if $L > (N+M)$ we use (7.3) and the complexity is $\mathcal{O}\left((N+M)^2 L\right)$ and if $L \leq (N+M)$ we use (7.1) and the complexity is $\mathcal{O}\left(L^2(N+M)\right)$. The operation of updating the belief $q(\boldsymbol{\alpha})$ is thus $\min(\mathcal{O}(L^2(N+M)), \mathcal{O}(L(N+M)^2))$.

Computing $\left\langle \lambda \right\rangle_{q(\lambda)}$ in (5.56) is also of high complexity for the sparse estimation algorithm, because it requires evaluation of the term

$$\left\langle ||\mathbf{y} - \mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha}||_2^2 \right\rangle_{q(\boldsymbol{\alpha})\prod_{i\in\mathcal{D}} q(x_i)} \tag{7.4}$$

It can be written as

$$\begin{aligned}
\left\langle ||\mathbf{y} - \mathbf{X}\boldsymbol{\Phi}\boldsymbol{\alpha}||_2^2 \right\rangle &= \sum_{i=1}^{N+M} |y_i|^2 + \left\langle |x_i|^2 \right\rangle \mathbf{r}_i \left\langle \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{H}} \right\rangle \mathbf{r}_i^{\mathrm{H}} - 2 \left\langle \mathrm{Re}(y_i^* x_i \mathbf{r}_i \boldsymbol{\alpha}) \right\rangle \\
&= \sum_{i=1}^{N+M} |y_i|^2 + \left\langle |x_i|^2 \right\rangle \mathbf{r}_i \left( \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} + \boldsymbol{\mu}_{\boldsymbol{\alpha}} \boldsymbol{\mu}_{\boldsymbol{\alpha}}^{\mathrm{H}} \right) \mathbf{r}_i^{\mathrm{H}} - 2\,\mathrm{Re}(y_i^* \left\langle x_i \right\rangle \mathbf{r}_i \boldsymbol{\mu}_{\boldsymbol{\alpha}}) \\
&= \sum_{i=1}^{N+M} |y_i|^2 + \left\langle |x_i|^2 \right\rangle \left( \mathbf{r}_i \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \mathbf{r}_i^{\mathrm{H}} + ||\mathbf{r}_i \boldsymbol{\mu}_{\boldsymbol{\alpha}}||_2^2 \right) - 2\,\mathrm{Re}(y_i^* \left\langle x_i \right\rangle \mathbf{r}_i \boldsymbol{\mu}_{\boldsymbol{\alpha}})
\end{aligned} \tag{7.5}$$

where $\mathbf{r}_i$ is the $i$th row of $\boldsymbol{\Phi}$ and the total complexity is thus $\mathcal{O}(L^2(N+M))$. A similar computation is needed for computing the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i)$ for all $i \in \mathcal{D}$, which is of complexity $\mathcal{O}(L^2 N)$. The per-iteration complexity of the sparse channel estimation algorithm is thus $\mathcal{O}(L^2(N+M))$.

## 7.2. Partial Data-Aided Channel Estimation

In a conventional receiver often only the pilot symbols are utilized for estimating the channel. The receivers proposed in Chapter 4 and 5 are fully data-aided, i.e. the beliefs of all data symbols ($q(x_i)$ for all $i \in \mathcal{D}$) are used in the channel estimation of the receiver. It is easy to imagine a hybrid scheme between pilot-only channel estimation as in [13, 18] and fully data-aided channel estimation. In the following we investigate how such a hybrid scheme influences the computational complexity of the two receivers.

Denote the set of indices of data symbols we want to use for channel estimation $\mathcal{F} \subseteq \mathcal{D}$. The special case $\mathcal{F} = \emptyset$ is the pilot-only scheme and $\mathcal{F} = \mathcal{D}$ is the fully data-aided scheme presented in the previous chapters.

### 7.2.1 Frequency Domain Channel Coefficients

Referring to the factor graph representation of the receiver in Figure 4.1 we want to eliminate the contribution of the messages $\{m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to h_i}(h_i) \mid i \in \{\mathcal{D}\backslash\mathcal{F}\}\}$ and $\{m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\lambda}(\lambda) \mid i \in \{\mathcal{D}\backslash\mathcal{F}\}\}$ to channel and noise precision estimation, respectively. This is achieved heuristically by forcing these messages to be constant. To keep the same functional form as the original messages, we formulate these constant messages as complex normal with zero mean and infinite variance, i.e.

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to h_i}(h_i) \propto \mathrm{CN}\left(h_i; \mu_{h_i^o}, \sigma^2_{h_i^o}\right) \tag{7.6}$$

where

$$\mu_{h_i^o} = \frac{y_i \left\langle x_i \right\rangle^*_{q(x_i)}}{\left\langle |x_i|^2 \right\rangle_{q(x_i)}}, \qquad \sigma^2_{h_i^o} = \frac{1}{\left\langle \lambda \right\rangle_{q(\lambda)} \left\langle |x_i|^2 \right\rangle_{q(x_i)}}, \qquad \forall\, i \in \{\mathcal{F}\} \tag{7.7}$$

$$\mu_{h_j^o} = 0, \qquad \sigma^2_{h_j^o} = \infty, \qquad \forall\, j \in \{\mathcal{D}\backslash\mathcal{F}\} \tag{7.8}$$

As in Chapter 4 $\mathbf{\Sigma}^o_{\mathbf{h}} = \mathrm{diag}\left[\left(\sigma^2_{h_i^o} | i \in [1 : N + M]\right)\right]$ and the required computation is

$$\mathbf{\Sigma}^q_{\mathbf{h}} = \left((\mathbf{\Sigma}_{\mathbf{h}})^{-1} + (\mathbf{\Sigma}^o_{\mathbf{h}})^{-1}\right)^{-1} \tag{7.9}$$

$$\boldsymbol{\mu}^q_{\mathbf{h}} = \mathbf{\Sigma}^q_{\mathbf{h}} \left(\mathbf{\Sigma}^o_{\mathbf{h}}\right)^{-1} \boldsymbol{\mu}^o_{\mathbf{h}} \tag{7.10}$$

where $\mathbf{\Sigma}_{\mathbf{h}}$ is the prior covariance matrix of the channel coefficients $\mathbf{h}$. The following toy example demonstrates how this ‘elimination’ of messages reduces the computational complexity.

Say $\mathcal{D} = \{1, 3, 4\}$, $\mathcal{P} = \{2\}$ and $\mathcal{F} = \{4\}$. We can write $(\mathbf{\Sigma}^o_{\mathbf{h}})^{-1}$ using its compact singular value decomposition (SVD)

$$(\boldsymbol{\Sigma}_{\mathbf{h}}^{o})^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma^2_{h^o_2}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma^2_{h^o_4}} \end{pmatrix} \tag{7.11}$$

$$= \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma^2_{h^o_2}} & 0 \\ 0 & \frac{1}{\sigma^2_{h^o_4}} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7.12}$$

$$= \mathbf{UCV} \tag{7.13}$$

Without loss of generality we have in this example assumed that $\frac{1}{\sigma^2_{h^o_2}} > \frac{1}{\sigma^2_{h^o_2}}$. Inserting (7.13) into (7.9) and applying the Woodbury matrix inverse formula (Appendix C.6) gives

$$\boldsymbol{\Sigma}_{\mathbf{h}}^{q} = \boldsymbol{\Sigma}_{\mathbf{h}} - \boldsymbol{\Sigma}_{\mathbf{h}}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\boldsymbol{\Sigma}_{\mathbf{h}}\mathbf{U})^{-1}\mathbf{V}\boldsymbol{\Sigma}_{\mathbf{h}} \tag{7.14}$$

Due to the structure of $\mathbf{U}$ and $\mathbf{V}$ multiplication by these matrices can be computed by indexing, i.e. in this example $\boldsymbol{\Sigma}_{\mathbf{h}}\mathbf{U}$ is the concatenation of the second and fourth columns of $\boldsymbol{\Sigma}_{\mathbf{h}}$, likewise $\mathbf{V}\boldsymbol{\Sigma}_{\mathbf{h}}$ is the concatenation of the second and fourth rows of $\boldsymbol{\Sigma}_{\mathbf{h}}$.

The dimension of the matrix inverse is thus reduced from $(N + M)$ to $(|\mathcal{F}|+M)$. Computing $\boldsymbol{\Sigma}_{\mathbf{h}}^{q}$ and $\boldsymbol{\mu}_{\mathbf{h}}^{q}$ is thus of complexity $\mathcal{O}\left((N + M)^2 + (|\mathcal{F}| + M)^3\right)$.

To completely eliminate the influence of the data symbols that are not in $\mathcal{F}$, we also replace the messages $m_{f_{\mathrm{D}_j}\to\lambda}(\lambda)$ for all $j \in \{\mathcal{D}\backslash\mathcal{F}\}$ by constant messages. Denote $\mathcal{C} = \mathcal{P} \cup \mathcal{F}$ and let $\mathbf{y}_{\mathcal{C}}$ be the column vector obtained from the entries of $\mathbf{y}$ indexed by $\mathcal{C}$ and $\mathbf{X}_{\mathcal{C}}$ the diagonal matrix with the entries of the diagonal of $\mathbf{X}$ indexed by $\mathcal{C}$, the belief $q(\lambda)$ becomes

$$q_{\mathcal{C}}(\lambda) = \mathrm{Ga}\left(\lambda; |\mathcal{C}| + a, b + \langle ||\mathbf{y}_{\mathcal{C}} - \mathbf{X}_{\mathcal{C}}\mathbf{h}_{\mathcal{C}}||_2^2\rangle_{\prod_{i\in\mathcal{C}} q(h_i)}\right) \tag{7.15}$$

and the computation of the parameters is of linear complexity in $|\mathcal{C}|$ and is therefore not dominating.

As a side-note if the noise precision $\lambda$ is known the entries of $\boldsymbol{\Sigma}_{\mathbf{h}}^{o}$ corresponding to pilot symbols are constant with iterations and this can be exploited to further reduce the complexity. If we write $(\boldsymbol{\Sigma}_{\mathbf{h}}^{o})^{-1}$ as the sum of two diagonal matrices $(\boldsymbol{\Sigma}_{\mathbf{h}}^{o})^{-1} = (\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{op}})^{-1} + (\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{od}})^{-1}$, where $(\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{op}})^{-1}$ contains the entries corresponding to pilot symbols and $(\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{od}})^{-1}$ the entries corresponding to data symbols, using an SVD approach as shown above with $(\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{od}})^{-1} = \mathbf{ADB}$, the computation in each iteration becomes

$$\boldsymbol{\Sigma}_{\mathbf{h}}^{q} = \left((\boldsymbol{\Sigma}_{\mathbf{h}})^{-1} + (\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{op}})^{-1} + (\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{od}})^{-1}\right)^{-1} \tag{7.16}$$

$$= \left((\boldsymbol{\Sigma}_{\mathbf{h}})^{-1} + (\boldsymbol{\Sigma}_{\mathbf{h}}^{\mathrm{op}})^{-1} + \mathbf{ADB}\right)^{-1} \tag{7.17}$$

$$= \mathbf{G} - \mathbf{GA}(\mathbf{D}^{-1} + \mathbf{BGA})^{-1}\mathbf{BG} \tag{7.18}$$

where $\mathbf{G} = ((\mathbf{\Sigma_h})^{-1} + (\mathbf{\Sigma_h^{\mathrm{op}}})^{-1})^{-1}$ can be computed once, requiring a matrix inversion of dimension $M$, and all subsequent updates requires a matrix inversion (7.18) of dimension $|\mathcal{F}|$.

### 7.2.2 Delay Domain Channel Weights

We now make a similar modification of the receiver that exploits the sparsity of the channel. Referring to the factor graph representation in Figure 5.2 the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_j}\to\boldsymbol{\alpha}}(\boldsymbol{\alpha})$ for all $j \in \{\mathcal{D}\backslash\mathcal{F}\}$ are replaced by constant messages. This corresponds to setting $\langle x_i \rangle_{q(x_i)} = \langle |x_i|^2 \rangle_{q(x_i)} = 0$ for all $i \in \{\mathcal{D}\backslash\mathcal{F}\}$ in (7.1) and (7.2). Still denoting $\mathcal{C} = \{\mathcal{P} \cup \mathcal{F}\}$ and let $\mathbf{\Phi}_\mathcal{C}$ be the matrix consisting of the rows of $\mathbf{\Phi}$ indexed by $\mathcal{C}$, the parameters of the belief $q(\boldsymbol{\alpha}) = \mathrm{CN}\left(\boldsymbol{\alpha}; \boldsymbol{\mu}_{\boldsymbol{\alpha}}, \mathbf{\Sigma}_{\boldsymbol{\alpha}}\right)$ are computed by

$$\mathbf{\Sigma}_{\boldsymbol{\alpha}} = \left(\mathbf{\Phi}_\mathcal{C}^{\mathrm{H}} \left\langle \mathbf{X}_\mathcal{C}^{\mathrm{H}} \mathbf{X}_\mathcal{C} \right\rangle_{\prod_{i\in\mathcal{F}} q(x_i)} \mathbf{\Phi}_\mathcal{C} \left\langle \lambda \right\rangle_{q(\lambda)} + \left\langle \mathbf{\Gamma}^{-1} \right\rangle_{q(\boldsymbol{\gamma})}\right)^{-1} \tag{7.19}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}} = \left\langle \lambda \right\rangle_{q(\lambda)} \mathbf{\Sigma}_{\boldsymbol{\alpha}} \mathbf{\Phi}_\mathcal{C}^{\mathrm{H}} \left\langle \mathbf{X}_\mathcal{C} \right\rangle_{\prod_{i\in\mathcal{F}} q(x_i)}^{\mathrm{H}} \mathbf{y}_\mathcal{C} \tag{7.20}$$

Again we replace the messages $m_{f_{\mathrm{D}_j}\to\lambda}(\lambda)$ for all $j \in \{\mathcal{D}\backslash\mathcal{F}\}$ by constant messages. The belief $q(\lambda)$ thus becomes

$$q_\mathcal{C}(\lambda) = \mathrm{Ga}\left(\lambda; |\mathcal{C}| + a, b + \left\langle ||\mathbf{y}_\mathcal{C} - \mathbf{X}_\mathcal{C} \mathbf{\Phi}_\mathcal{C} \boldsymbol{\alpha}||_2^2 \right\rangle\right) \tag{7.21}$$

and the computational complexity of (7.19), (7.20) and (7.21) is $\mathcal{O}(L^2(|\mathcal{F}| + M))$. This implies that computing the messages $m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to x_i}(x_i)$ for all $i \in \mathcal{D}$ may become the dominating complexity with $\mathcal{O}(L^2 N)$. The total complexity is thus $\max(\mathcal{O}(L^2(|\mathcal{F}| + M)), \mathcal{O}(L^2 N))$.

## 7.3. Generalized Mean Field Approximation

In this section we present the second method for reducing the computational complexity of the channel estimation part of both receivers. We utilize the generalized mean field (GMF) approximation [26] in the same way as done for pilot-only sparse channel estimation with hierarchical priors in [64]. In GMF the posterior pdf of a set of variables is approximated by an approximating function that is constrained to factorize over groups of the variables. The idea here is to divide the channel weights ($\mathbf{h}$ in frequency domain and $\boldsymbol{\alpha}$ in delay domain) into disjoint groups in order to avoid the inversion of a matrix of large dimension.

### 7.3.1 Frequency Domain Channel Coefficients

As opposed to the approach in Chapter 4 we include the factor node $f_\mathrm{H}$, corresponding to the prior of the channel coefficients, to the MF part of the factor graph. We thus have

$$\mathcal{A}_{\mathrm{MF}} = \{f_{\mathrm{D}_i} | i \in \mathcal{D}\} \cup \left\{f_{\mathrm{P}_j} | j \in \mathcal{P}\right\} \cup \{f_{\mathrm{H}}\} \tag{7.22}$$

$$\mathcal{A}_{\mathrm{BP}} = \{\mathcal{A} \backslash \mathcal{A}_{\mathrm{MF}}\} \tag{7.23}$$

$$= \{f_\lambda\} \cup \{f_{\mathrm{C}}\} \cup \{f_{\mathrm{M}_n} | n \in [1:N]\} \cup \{f_{\mathrm{U}_k} | k \in [1:K]\} \tag{7.24}$$

where the local functions are defined as in Chapter 4. The channel coefficients $\mathbf{h}$ are divided into groups of size $G$ by defining vector $\mathbf{h}_z = (h_i \,|\, i \in [(z-1)G+1 : zG])^{\mathrm{T}}$ for all $z \in [1:Z]$, where $Z$ is the number of groups ($Z = \frac{N+M}{G}$). The belief $q(\mathbf{h})$ is constrained to factorize as:

$$q(\mathbf{h}) = \prod_{z=1}^{Z} q(\mathbf{h}_z) \tag{7.25}$$

The factor graph representation of the chosen factorization for the whole receiver is shown in Figure 7.1. Choosing $Z = 1$ yields the algorithm in Chapter 4 as $\mathbf{h}$ has been moved to the MF part as in [4]. It can easily be shown that the belief of $\mathbf{h}_z$ for all $z \in [1:Z]$ is given by

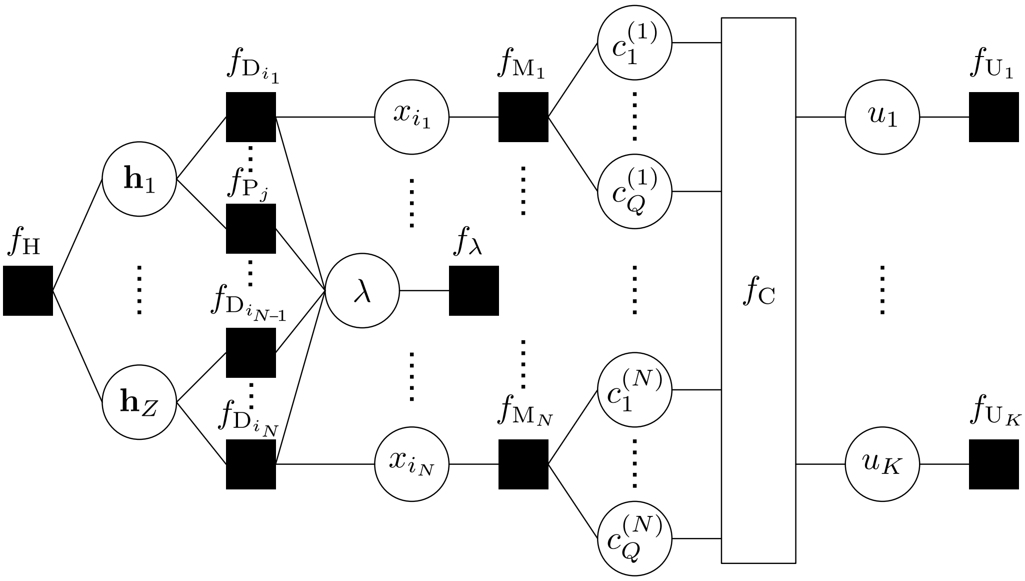


**Figure 7.1:** *Factor graph representation of the system function, in which the channel coefficients* $\mathbf{h}$ *are divided into disjoint groups.*

$$q(\mathbf{h}_z) = \mathrm{CN}\left(\mathbf{h}_z; \boldsymbol{\mu}^q_{\mathbf{h}_z}, \boldsymbol{\Sigma}^q_{\mathbf{h}_z}\right) \tag{7.26}$$

where

$$\boldsymbol{\Sigma}^q_{\mathbf{h}_z} = ((\boldsymbol{\Sigma}_{\mathbf{h}_z})^{-1} + (\boldsymbol{\Sigma}^o_{\mathbf{h}_z})^{-1})^{-1} \tag{7.27}$$

$$\boldsymbol{\mu}^q_{\mathbf{h}_z} = \boldsymbol{\Sigma}^q_{\mathbf{h}_z} (\boldsymbol{\Sigma}^o_{\mathbf{h}_z})^{-1} \boldsymbol{\mu}^o_{\mathbf{h}_z} \tag{7.28}$$

where

$$\boldsymbol{\Sigma}_{\mathbf{h}_z} = ([\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,j} \,|\, i,j \in [(z-1)G+1 : zG]) \tag{7.29}$$

$$\boldsymbol{\Sigma}^o_{\mathbf{h}_z} = ([\boldsymbol{\Sigma}^o_{\mathbf{h}}]_{i,j} \,|\, i,j \in [(z-1)G+1 : zG]) \tag{7.30}$$

$$\boldsymbol{\mu}^o_{\mathbf{h}_z} = (\mu^o_i \,|\, i \in [(z-1)G+1 : zG])^{\mathrm{T}} \tag{7.31}$$

for all $z \in [1 : Z]$ are partitions of $\boldsymbol{\Sigma}_{\mathbf{h}}, \boldsymbol{\Sigma}^{o}_{\mathbf{h}}, \boldsymbol{\mu}^{o}_{\mathbf{h}}$ respectively. By redefining $\boldsymbol{\Sigma}^{q}_{\mathbf{h}}$ as the block diagonal matrix $\boldsymbol{\Sigma}^{q}_{\mathbf{h}} = \operatorname{diag}(\boldsymbol{\Sigma}^{q}_{\mathbf{h}_1}, \ldots, \boldsymbol{\Sigma}^{q}_{\mathbf{h}_Z})$ and $\boldsymbol{\mu}^{q}_{\mathbf{h}} = ((\boldsymbol{\mu}^{q}_{\mathbf{h}_1})^{\mathrm{T}}, \ldots, (\boldsymbol{\mu}^{q}_{\mathbf{h}_Z})^{\mathrm{T}})^{\mathrm{T}}$, the remaining messages and update of the beliefs are as shown in Chapter 4.

By choosing the approximating distribution to factorize over groups of variables we thus need to compute the inverse of a $G \times G$ matrix $Z = \frac{N+M}{G}$ times, which gives a complexity of $\mathcal{O}(G^2(N+M))$.

### 7.3.2 Delay Domain Channel Weights

Similarly the delay domain channel weights $\boldsymbol{\alpha}$ are also divided into groups of size $G$. Be aware that we reuse notation from the previous section and also use $Z = \frac{L}{G}$ as the number of groups, though it may be different from $Z$ in the previous section. All local functions and the partitioning into MF and BP part are as in Chapter 5. The factor graph representation of the factorized system function is shown in Figure 7.2. Similarly to previous section we define $\boldsymbol{\alpha}_z = (\alpha_\ell \,|\, \ell \in [(z-1)G+1 : zG])^{\mathrm{T}}$ and obtain the messages

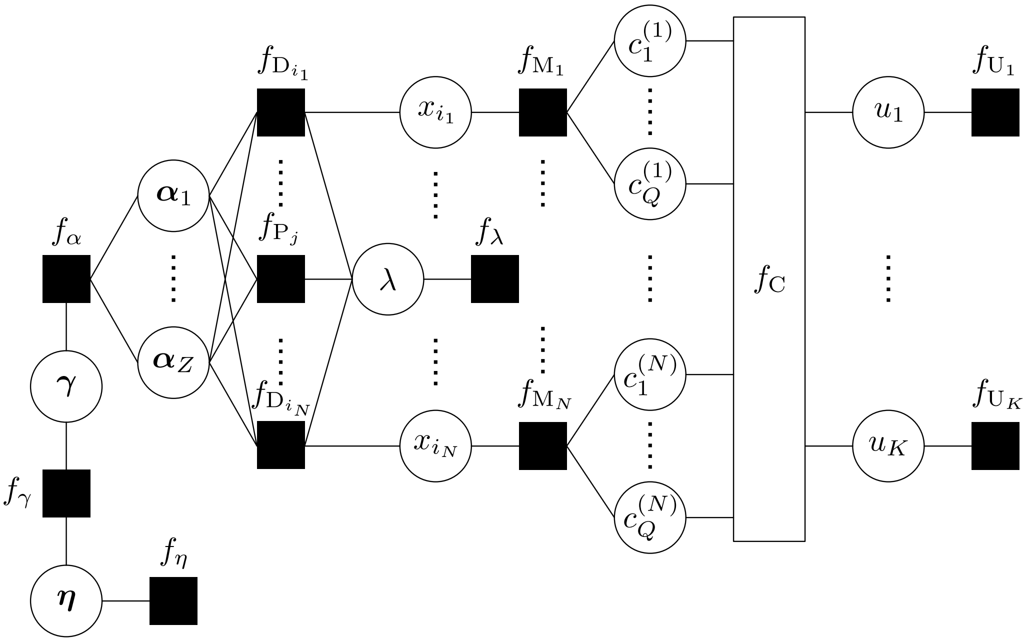


**Figure 7.2:** *Factor graph representation of the system function, in which the delay domain channel weights $\boldsymbol{\alpha}$ have been divided into smaller groups.*

$$m^{\mathrm{MF}}_{f_{\boldsymbol{\alpha}}\to\boldsymbol{\alpha}_z}(\boldsymbol{\alpha}_z) \propto \exp\Big(\boldsymbol{\alpha}_z \left\langle \boldsymbol{\Gamma}_z \right\rangle_{q(\boldsymbol{\gamma})} \boldsymbol{\alpha}_z\Big) \tag{7.32}$$

$$m^{\mathrm{MF}}_{f_{\mathrm{D}_i}\to\boldsymbol{\alpha}_z}(\boldsymbol{\alpha}_z) \propto \exp\Big(-\left\langle \lambda \right\rangle_{q(\lambda)} \Big(\left\langle |\mathbf{r}_i\boldsymbol{\alpha}|^2 \right\rangle_{\prod_{z'\neq z} q(\boldsymbol{\alpha}_{z'})} \left\langle |x_i|^2 \right\rangle_{q(x_i)} \\ -2\operatorname{Re}\Big(y_i^* \left\langle x_i \right\rangle_{q(x_i)} (\mathbf{r}_i)_z \boldsymbol{\alpha}_z\Big)\Big) \tag{7.33}$$

$$m^{\mathrm{MF}}_{f_{\mathrm{P}_j}\to\boldsymbol{\alpha}_z}(\boldsymbol{\alpha}_z) \propto \exp\Big(-\left\langle \lambda \right\rangle_{q(\lambda)} \Big(\left\langle |\mathbf{r}_j\boldsymbol{\alpha}|^2 \right\rangle_{\prod_{z'\neq z} q(\boldsymbol{\alpha}_{z'})} |x_j|^2 - 2\operatorname{Re}\Big(y_j^* x_j (\mathbf{r}_j)_z \boldsymbol{\alpha}_z\Big)\Big)\Big) \tag{7.34}$$

for all $i \in \mathcal{D}$, all $j \in \mathcal{P}$ and all $z \in [1:Z]$. The belief $q(\boldsymbol{\alpha}_z)$ for all $z \in [1:Z]$ is thus given by

$$q(\boldsymbol{\alpha}_z) = \mathrm{CN}\left(\boldsymbol{\alpha}_z; \boldsymbol{\mu}_{\boldsymbol{\alpha}_z}, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_z}\right) \tag{7.35}$$

where

$$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_z} = \left(\boldsymbol{\Phi}_z^{\mathrm{H}} \left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)} \boldsymbol{\Phi}_z \left\langle \lambda \right\rangle_{q(\lambda)} + \left\langle \boldsymbol{\Gamma}_z^{-1} \right\rangle_{q(\boldsymbol{\gamma})}\right)^{-1} \tag{7.36}$$

$$\boldsymbol{\mu}_{\boldsymbol{\alpha}_z} = \left\langle \lambda \right\rangle_{q(\lambda)} \boldsymbol{\Sigma}_z \boldsymbol{\Phi}_z^{\mathrm{H}} \left( \left\langle \mathbf{X} \right\rangle^{\mathrm{H}}_{\prod_{i\in\mathcal{D}} q(x_i)} \mathbf{y} - \left\langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \right\rangle_{\prod_{i\in\mathcal{D}} q(x_i)} \sum_{z' \neq z} \boldsymbol{\Phi}_{z'} \boldsymbol{\mu}_{\boldsymbol{\alpha}_{z'}} \right) \tag{7.37}$$

Similarly to previous section we redefine $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ as the block diagonal matrix $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} = \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_1}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_Z})$ and $\boldsymbol{\mu}_{\boldsymbol{\alpha}} = ((\boldsymbol{\mu}_{\boldsymbol{\alpha}_1})^{\mathrm{T}}, \ldots, (\boldsymbol{\mu}_{\boldsymbol{\alpha}_Z})^{\mathrm{T}})^{\mathrm{T}}$, such that the remaining messages and beliefs can be computed using the expressions in Chapter 5.

Notice how $\boldsymbol{\mu}_{\boldsymbol{\alpha}_z}$ depends on all $\boldsymbol{\mu}_{\boldsymbol{\alpha}_{z'}}$ for $z' \neq z$ because of the sum in (7.37). In order to not evaluate the full sum for each $z$, the update of the beliefs are computed recursively as depicted in Algorithm 1. When $G = 1$ the recursive update of $\boldsymbol{\mu}_z$ resembles the Gauss-Seidel method for solving systems of linear equations [65]. It is thus expected that smaller group sizes requires a larger number of iterations to converge, i.e. it may be necessary to perform Algorithm 1 multiple times when updating the beliefs. The complexity of each step in the algorithm (assuming $G \leq N + M$) is shown in the right column of Algorithm 1. Notice that for the steps within the 'for'-loop the increase in complexity that comes from performing the step $Z = \frac{L}{G}$ times is included in the expressions. The complexity of updating all the beliefs $q(\boldsymbol{\alpha}_z)$ for all $z \in [1:Z]$ is thus $\mathcal{O}\left(G(N+M)L\right)$.

**Algorithm 1** Update of beliefs $q(\boldsymbol{\alpha}_z)\, \forall\, z \in [1:Z]$

```
Input: $\mathbf{y}$, $\langle \mathbf{X} \rangle$, $\langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \rangle$, $\langle \lambda \rangle$, $\boldsymbol{\Phi}$ and mean of current belief $\boldsymbol{\mu}'_{\boldsymbol{\alpha}}$
Output: Mean $\boldsymbol{\mu}_{\boldsymbol{\alpha}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ of updated belief.
$\mathbf{a} \leftarrow \boldsymbol{\Phi}\boldsymbol{\mu}'_{\boldsymbol{\alpha}}$                                                    $\mathcal{O}\left((N+M)L\right)$
for $z = [1:Z]$ do
    $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_z} \leftarrow (\boldsymbol{\Phi}_z^{\mathrm{H}} \langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \rangle \boldsymbol{\Phi}_z \langle \lambda \rangle + \langle \boldsymbol{\Gamma}_z^{-1} \rangle)^{-1}$          $\mathcal{O}\left(G(N+M)L\right)$
    $\boldsymbol{\mu}_{\boldsymbol{\alpha}_z} \leftarrow \langle \lambda \rangle \boldsymbol{\Sigma}_z \boldsymbol{\Phi}_z^{\mathrm{H}} \left( \langle \mathbf{X} \rangle^{\mathrm{H}} \mathbf{y} - \langle \mathbf{X}^{\mathrm{H}}\mathbf{X} \rangle \left(\mathbf{a} - \boldsymbol{\Phi}_z \boldsymbol{\mu}'_{\boldsymbol{\alpha}_z}\right)\right)$   $\mathcal{O}\left((N+M)L\right)$
    $\mathbf{a} \leftarrow \mathbf{a} + \boldsymbol{\Phi}_z(\boldsymbol{\mu}_{\boldsymbol{\alpha}_z} - \boldsymbol{\mu}'_{\boldsymbol{\alpha}_z})$                              $\mathcal{O}\left((N+M)L\right)$
end for
$\boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \leftarrow \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{\alpha}_1}, \ldots, \boldsymbol{\Sigma}_{\boldsymbol{\alpha}_Z})$
$\boldsymbol{\mu}_{\boldsymbol{\alpha}} \leftarrow ((\boldsymbol{\mu}_{\boldsymbol{\alpha}_1})^{\mathrm{T}}, \ldots, (\boldsymbol{\mu}_{\boldsymbol{\alpha}_Z})^{\mathrm{T}})^{\mathrm{T}}$
```

When computing $\left\langle \lambda \right\rangle_{q_\lambda}$ using (7.5) we need to exploit that $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}$ is now a block diagonal matrix. Computing $\mathbf{r}_i \boldsymbol{\Sigma}_{\boldsymbol{\alpha}} \mathbf{r}_i^{\mathrm{H}}$ for all $i \in [1:N+M]$ is then also of complexity $\mathcal{O}(G(N+M)L)$.

The complexity of one iteration of the channel estimation part of the two receiver schemes and their modified versions is summarized in Table 7.1.

| Modification | Frequency Domain | Delay Domain |
|---|---|---|
| Original | $(N+M)^3$ | $L^2(N+M)$ |
| Partial Feedback | $(\lvert\mathcal{F}\rvert+M)^3+(N+M)^2$ | $\max\left(L^2(\lvert\mathcal{F}\rvert+M), L^2N\right)$ |
| GMF Approximation | $G^2(N+M)$ | $G(N+M)L$ |

**Table 7.1:** *Computational complexity in Big-$\mathcal{O}$ notation for one iteration of the two investigated channel estimation schemes and the versions modified for reduced computational complexity. It is assumed $G \leq (N+M)$.*

## 7.4. Numerical Results

In this section the impact of the proposed complexity-reducing methods on the performance of the receivers is evaluated through Monte Carlo simulations. We therefore investigate the same receivers as in Chapter 6, but with the channel estimation part modified with the proposed methods. The initialization of the algorithms is also the same as in Chapter 6 and the parameters shown in Table 6.2 are also fixed throughout all simulations. Notice that the methods of partial feedback and the GMF approximation can easily be combined to yield receiver algorithms using both methods for reducing the complexity. However, to limit the number of simulation scenarios we consider the two methods separately.

### 7.4.1 Partial Feedback

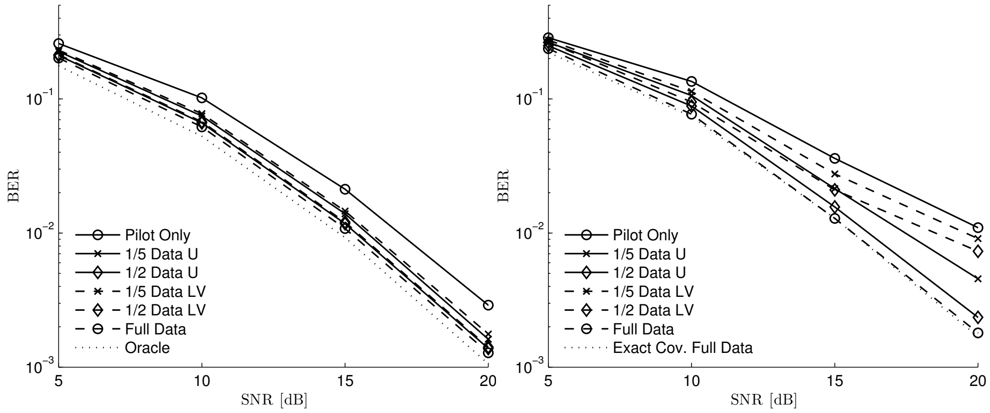
When using partial feedback, i.e. when $0 < |\mathcal{F}| < |\mathcal{D}|$ different choices of the index set $\mathcal{F}$ of data symbols to be used for channel estimation can be considered. In the following numerical experiment we employ two different approaches for selecting $\mathcal{F}$. In the first method we use uniform feedback. Let $\mathbf{d} = (d_1, d_2, \dots d_N)^{\mathrm{T}}$ be the vector with the elements of $\mathcal{D}$ ordered in ascending order. With $|\mathcal{F}|$ chosen such that $S = \frac{N}{|\mathcal{F}|}$ is an integer we use the indices $\mathcal{F} = \{d_S, d_{2S}, \dots, d_N\}$.

In the second method $\mathcal{F}$ is allowed to change between iterations. Just after updating $q(x_i)$ for all $i \in \mathcal{D}$ in the iterative algorithm we select $\mathcal{F}$ to be the set of the $|\mathcal{F}|$ indices of the beliefs $q(x_i)$ with the smallest variances. The motivation for this approach is that in each iteration we thus use soft information of the data symbols with least amount of uncertainty.

The results for both methods applied to the two receivers are shown in Figure 7.3. For the sparsity-aware receiver scheme we see no difference

|  |  |
|---|---|
| Pilot Spacing | 10 |
| No. subcarriers $(N+M)$ | 600 |
| No. columns in $\mathbf{\Phi}$ (L) | 150 |
| Channel model | Exp. decay, $\mu_P = 10, v = 1\,\mu\mathrm{s}$ |

**Table 7.2:** *Simulation parameters for the scenario in which the performance of the receiver schemes is evaluated versus SNR for different levels of feedback.*



**(a)** *Sparsity-aware receiver*

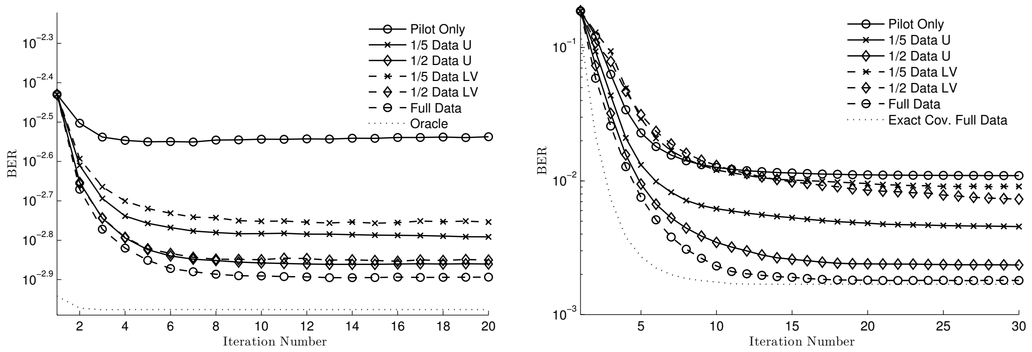
**(b)** *Robust receiver*

**Figure 7.3:** *BER for the (a) sparsity-aware receiver and (b) the robust receiver schemes using different levels of data-aided feedback. The fraction in the legend refers to the relative size of* $|\mathcal{F}|$ *compared to* $|\mathcal{D}|$*. 'U' denotes the uniform selection approach and 'LV' is the lowest variance method. Note that the 'dotted' curve in (b) corresponds to the receiver using the exact covariance matrix for the prior distribution of* $\mathbf{h}$ *and is thus not using the robust channel assumption.*

between using uniform feedback or the lowest variance method, the curves coincide on Figure 7.3a. Furthermore the difference between using all the data symbols for channel estimation or only half of them is very small.

For the robust receiver the picture is different. For this scheme the uniformly spaced feedback outperforms the lowest variance method. Furthermore the penalty for not using all the data for the channel estimation is larger. By using all the data symbols there is a gain of approximately 1 dB compared to '$\frac{1}{2}$ Data U' and 3 dB compared to '$\frac{1}{5}$ Data U' in high SNR.

To see how the use of soft-information changes the convergence properties of receiver algorithms we plot the BER versus iteration number for both receivers in Figure 7.4 for the scenario with 15 dB average SNR. Notice that the iteration number refers to the outer iterations of the algorithm, which means that for the sparsity-aware receiver the number of iterations used in the channel estimation subgraph is $20 + 5(i - 1)$ where $i$ is the iteration number. The first 20 of these iterations uses pilot symbols only and are therefore of lower complexity. As shown in Figure 7.4a the sparsity-aware algorithm in reaches convergence in approximately 10 outer iterations. The number of iterations in the channel estimation part (65 in total) is thus larger than for the robust receiver in Figure 7.4b for which there are only minor improvements in BER after approximately 20 iterations. However, for the fully data-aided case the complexity per iteration is lower for the sparse channel estimation part as the size of the required matrix inversion is smaller (150 vs 600 in this case). This is also indicated by the running time measurements in Appendix H, that shows that in the Matlab implementation it is faster to compute 20 outer iterations for the sparsity-aware receiver than for the robust receiver with the same parameters as used here.



**(a)** *Sparsity-aware receiver*

**(b)** *Robust receiver*

**Figure 7.4:** *BER versus iteration number for the (a) sparsity-aware receiver and (b) the robust receiver using different levels of data-aided feedback. The average SNR is* 20 dB*. The fraction in the legend refers to the relative size of* $|\mathcal{F}|$ *compared to* $|\mathcal{D}|$*. 'U' denotes the uniform selection approach and 'LV' is the lowest variance method. The iteration number refers to the outer iterations.*

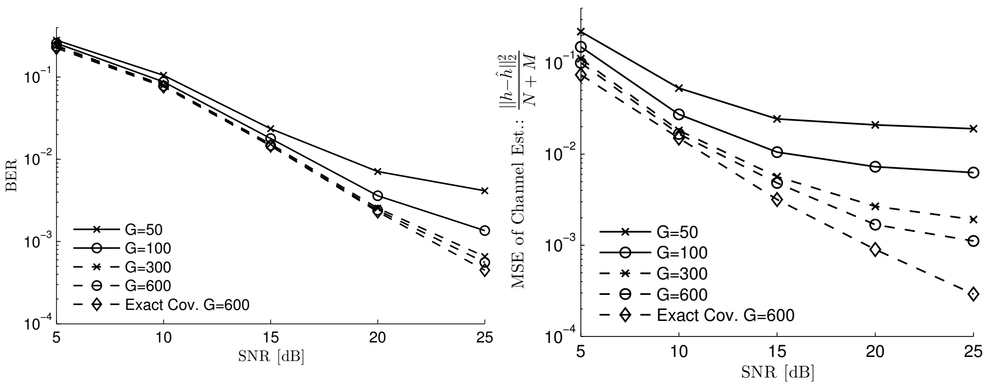
It is noted that this result is not general as it is highly dependent on the implementation of the algorithm and the hardware platform it is running on. For both receivers in Figure 7.4b using only the pilots symbols for the channel estimation requires fewer iterations to reach convergence than for data-aided channel estimation. The algorithm is iterative because the noise precision $\lambda$ is estimated.

### 7.4.2 Performance for Different Group Sizes

Using the simulation parameters in Table 7.3 we plot the BER and the MSE of the channel estimate for the robust receiver in Figure 7.5. As the number of subcarriers is 600 a group size $G = 600$ is the same receiver as in previous chapters. Using two groups ($G = 300$) has no visible impact on the BER performance, but already with $G = 100$ there is a significant increase in both BER and MSE of the channel estimate, especially at high SNR. Therefore we do not consider this simple method as a viable solution for reducing the computational complexity of this receiver scheme. A more sophisticated solution with the same computational complexity is proposed

| | |
|---|---|
| Pilot Spacing | 10 |
| No. subcarriers ($N + M$) | 600 |
| No. columns in $\mathbf{\Phi}$ (L) | 150 |
| Channel model | Exp. decay, $\mu_P = 10, v = 1\,\mu\text{s}$ |
| Noise precision ($\lambda$) | Known |

**Table 7.3:** *Simulation parameters for the scenario in which the performance of the receiver schemes is evaluated versus SNR for different group sizes.*



**(a)** *BER*

**(b)** *MSE of channel estimate* $\hat{\mathbf{h}}$

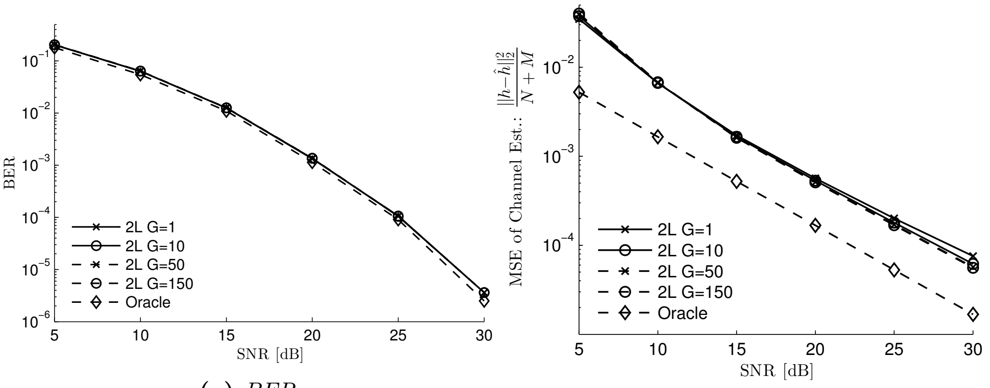
**Figure 7.5:** *BER and MSE performance of the robust receiver with different group sizes G versus SNR. All curves are for the receiver using the robust channel assumption, except the 'dashed' curve with diamond-shaped markers.*

in [66] [1].

The results for the receiver with sparse channel estimation in the same scenario are shown in Figure 7.6. Since the number of columns of the dictionary is $L = 150$ the receiver with $G = 150$ is the same as in previous chapters. Here there is no increase in BER when decreasing the group size, even with a group size $G = 1$ the BER performance is the same and the per-iteration complexity of the receiver scheme is significantly reduced. The same observation was made for pilot-only channel estimation in [64].

As discussed in Section 7.3.2 the algorithm may have slower convergence with smaller group sizes. At low to medium SNR we do not observe any difference in convergence, but in the high SNR region there is a noticeable difference in the convergence of the MSE of the channel estimate as illustrated in Figure 7.7. However, the difference is so small that it is not noticeable in BER. Using small group sizes is thus considered as a viable option for reducing the computational complexity. With $G = 1$ the complexity per iteration is $\mathcal{O}((N+M)L)$. That is the same complexity per iteration as if we used the fast inference scheme [46] for solving the compressed sensing problem [48]. In the fast inference scheme only one of the entries of $\boldsymbol{\alpha}$ is updated in each iteration, while the method shown here updates all entries with the same complexity.

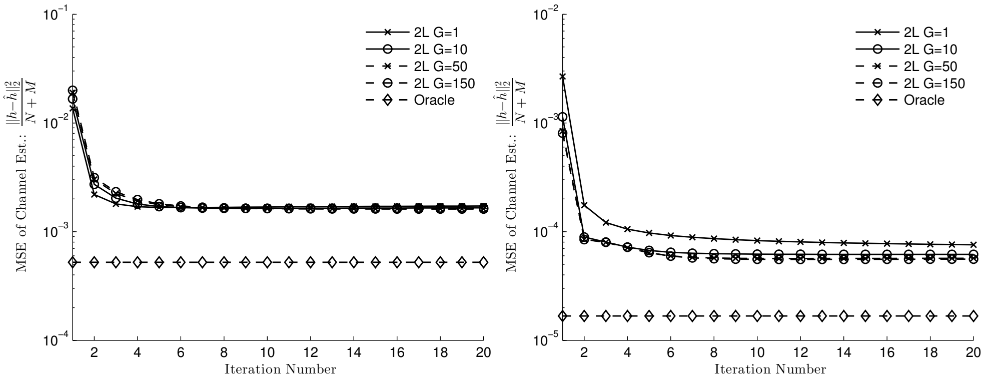
[1] In [66] $p(\mathbf{h}) = \mathrm{CN}(\mathbf{h}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{h}})$ is approximated by a Markov model $p(\mathbf{h}) = p(\mathbf{h}_1) \prod_{z=2}^{Z} p(\mathbf{h}_z | \mathbf{h}_{z-1})$. The resulting algorithm is also of complexity $\mathcal{O}(G^2(N+M))$. In a simulation scenario using channels with different frequency correlation functions the BER performance using small group sizes ($G = 6$) is close (approximately 0.5 dB) to the performance when using only one group ($G = 300$).



**(a)** *BER*

**(b)** *MSE of channel estimate* $\hat{\mathbf{h}}$

**Figure 7.6:** *BER and MSE performance of the sparse channel estimation receiver with different group sizes G versus SNR.*



**(a)** 15 dB *SNR*

**(b)** 30 dB *SNR*

**Figure 7.7:** *MSE of channel estimate versus iteration number. The iteration number refers to the outer iteration.*

CHAPTER 8

# Conclusion

In this thesis we have investigated the problem of embedding sparse channel estimation in iterative message-passing receivers. We model a simple point-to-point OFDM communication system operating in a noisy, specular multipath channel with static impulse response.

The goal of the receiver is to minimize the bit-error-rate (BER) which can be achieved by the maximum a posteriori criterion. However computing the posterior probability density function of the information bits required for exact implementation of this criterion is analytically intractable. We therefore resort to iterative inference techniques to approximate the posterior. By employing a factor graph representation of the probabilistic system model the different tasks of the receiver (channel estimation, equalization, demapping and decoding) can be identified as local computations of an inference scheme formulated as message-passing on the factor graph. In previous work the combined belief propagation (BP) and mean field (MF) algorithm (BP-MF) was proven effective for performing the inference in this kind of communication system. Based on this work we studied and implemented a state-of-the-art iterative receiver that was utilized as an important reference throughout the thesis.

### Sparse Channel Estimation

The main contribution of the thesis is a novel iterative receiver that exploits the sparse structure of a specular multipath channel having a few significant components. We have posed the problem of estimating the sparse channel as a sparse estimation problem. Formulating this problem involves the design of a proper dictionary matrix. The multipath delays are continuous valued parameters, but we employ a dictionary with discretized delays, which causes mismatch between the 'true dictionary' and the one used for signal recovery. The analysis shows that if the discretization is fine enough the signal has an accurate sparse representation in the dictionary, but a finer discretization increases the coherence property of the dictionary. More importantly the computational complexity increases with finer resolution as the number of columns of the dictionary increases. Using the approach of sparse Bayesian learning (SBL) with hierarchical prior modelling the sparse estimation problem was integrated with the other receiver tasks in the factor graph representation of the whole system. This enables a unified design of the sparse channel estimation and decoding.

The performance of the receiver with sparse channel estimation was evaluated through Monte Carlo simulations using a multipath channel model with exponentially decaying power delay profile and different 3GPP power delay profiles. The results show that for a certain level of discretization a further increase in the delay resolution does not lead to better accuracy of the channel estimates. The proposed receiver uses a sparsity-inducing prior that models the channel weights, each corresponding to a certain multipath delay within the length of the cyclic prefix, as independent and identically distributed. Therefore the proposed receiver scheme was benchmarked against the reference receiver with the same 'robust' channel assumption (flat power delay profile and knowledge of the maximum multipath delay). The proposed receiver outperforms the reference receiver with the robust channel assumption in all considered scenarios and the performance is on par or better with the reference receiver using exact knowledge of the channel covariance. However, this is only the case if the number of pilot subcarriers is sufficiently high. With a high pilot spacing, knowing the exact covariance matrix shows a significant gain over the sparse channel estimation scheme as designed here.

We also investigated the performance of the receivers using the exponential model with different levels of sparsity (number of multipath components). Even in scenarios with a high number of multipath components the BER and MSE of the channel estimate using the sparsity-aware receiver is still similar to that of the receiver with knowledge of the exact covariance matrix.

### Complexity Reduction

The channel estimation task of both the sparsity-aware and the reference receiver was shown to be of high computational complexity. To alleviate this problem we proposed two methods for reducing the computational complexity of both receivers.

The first method is a heuristic scheme in which some of the messages corresponding to feedback from the soft decoder are replaced by constant messages. In each iteration the reference receiver computes the inverse of a square matrix of dimension $(|\mathcal{F}| + M)$ where $|\mathcal{F}|$ is the number of data subcarriers used for channel estimation and $M$ is the number of pilot symbols, hence the size of the matrix to be inverted is linear with $|\mathcal{F}|$. For the sparse channel estimation scheme other operations of the receiver can become dominant and the complexity is $\max\left(L^2(|\mathcal{F}| + M), L^2N\right)$, where $L$ is the number of multipath components in the dictionary and $N$ is the number of data subcarriers.

The second method is based on the generalized mean field approximation in which the channel coefficients (in frequency or delay domain) are divided into disjoint groups of a certain size. With this method the complexity of the reference receiver is reduced from $\mathcal{O}((N+M)^3)$ to $\mathcal{O}(G^2(N+M))$ and for the sparsity-aware receiver it is reduced from $\mathcal{O}(L^2(N+M))$ to $\mathcal{O}(G(N+M)L)$ where $N+M$ is the total number of subcarriers, $L$ is the number of multipath delays in the dictionary matrix and $G$ is the group size. Numerical results show that using smaller groups for the reference receiver is not a viable method as it leads to significantly higher BER, especially in high SNR. For the sparsity-aware receiver, using smaller group sizes has little to no impact on the BER performance of the receiver whereas the computational complexity is significantly decreased. However, a small penalty in the rate of convergence was noticed in the high SNR scenario.

In the considered scenarios the sparsity-aware receiver requires more iterations to converge, but the complexity of each iteration is lower because the number of multipath delays in the dictionary $L$ could be chosen to be small compared to total number of subcarriers $(N+M)$. In general it is cumbersome to accurately assess and compare the computational complexity between the two receivers. The complexities mentioned here are per iteration and in addition the big-$\mathcal{O}$ notation only capture how the complexity scales with the size of the problem. For a practical implementation we would for example be more interested in the total number of floating point operations which in turn depends on the number of iterations required to reach convergence. Depending on the hardware platform memory consumption and the suitability for parallel execution may also be important properties of the algorithms.

## Outlook

We have successfully shown how sparse channel estimation can be naturally embedded in an iterative receiver scheme, but some open research problems have been identified in the process.

As mentioned above we solved the sparse signal estimation problem by using a dictionary with discretized multipath delays, though the true multipath delays are continuous valued. It is of interest to investigate whether other approaches to this problem such as [55, 56] could improve the estimation accuracy or be beneficial in another aspect, for example yield receiver algorithms with lower complexity.

The iterative receiver using the proposed sparse channel estimation scheme was found to be less robust to large spacings between the pilot subcarriers compared to the reference receiver with perfect knowledge of the covariance of the frequency domain channel coefficients. The covariance matrix used by the reference receiver is derived from the knowledge of the power delay profile of the channel's impulse response. As mentioned above the sparse channel estimation scheme assumes a flat power delay profile. If the knowledge of the channel's power delay profile is incorporated into the sparse channel estimation scheme, it may be possible to achieve the same robustness to large pilot spacings as exhibited by the reference receiver. One approach would be to shape the parameters of the prior pdf of the delay-domain weights to match the power delay profile of the channel, instead of using the same setting for all channel weights. However, it is neither a realistic assumption that we know the power delay profile of the channel exactly. If the aforementioned technique is implemented it would also become relevant to investigate how to track a time-varying power delay profile in the sparse channel setting. If the power-delay profile is slowly varying, one could for example initialize the beliefs of the unknown variables with the beliefs obtained in the processing of the previous OFDM symbol. In such a setting it would also be more realistic compare with a reference receiver scheme that estimates the covariance matrix of the time-varying channel, e.g. [67, 68].

Another natural extension is embedding the sparse channel estimation in an iterative message-passing receiver within a multiple-input multiple-output (MIMO) communication system such as [69].

# Appendices

# APPENDIX A

# Notation and Symbols

Throughout the report vectors are typeset with a bold lowercase font (e.g. $\mathbf{x}$) and matrices are bold uppercase (e.g. $\mathbf{X}$). Scalar variables and functions are written in italics (e.g. $x$). The following list clarifies the notation and symbols used throughout the report.

## A.1 Notation

| | |
|---|---|
| $(\cdot)^{\mathrm{T}}$ | Transpose of a vector or matrix. |
| $(\cdot)^{\mathrm{H}}$ | Hermitian transpose of a vector or matrix. |
| $(\cdot)^{*}$ | Complex-conjugate (entry-wise for vector or matrix). |
| $\langle f(x) \rangle_{p(x)}$ | Expected value of the function $f(x)$ with respect to the pdf or pmf $p(x)$. The pdf/pmf may be omitted if it is clear from the context. |
| $\lvert\lvert\cdot\rvert\rvert_p$ | The $l_p$-norm of a vector. For $p = 0$ it is the number of non-zero components of a vector. |
| $[\cdot]_{i,j}$ | The entry on the $i$th row and $j$th column of a matrix. |
| $[\cdot]_i$ | The $i$th entry of a vector. |
| $\lvert \cdot \rvert$ | The absolute value of a scalar (real or complex), the determinant of a matrix or the cardinality of a set. |

## A.2 Symbols

| | |
|---|---|
| $\mathbf{\Phi} \in \mathbb{C}^{(M+N)\times L}$ | Dictionary matrix. In the OFDM model each row corresponds to one subcarrier and each column to one multipath delay. |
| $\mathbf{h} \in \mathbb{C}^{(M+N)\times 1}$ | Frequency domain channel coefficient vector, can be approximated by $\mathbf{h} = \mathbf{\Phi}\boldsymbol{\alpha}$. |
| $\boldsymbol{\alpha} \in \mathbb{C}^{L\times 1}$ | Complex weight of each multipath component. |
| $\mathbf{y} \in \mathbb{C}^{(M+N)\times 1}$ | Observation vector, $\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}$. |
| $\mathbf{X} \in \mathbb{C}^{(M+N)\times(M+N)}$ | Diagonal matrix with $N$ data and $M$ pilot symbols. |
| $\mathbf{u} \in \{0,1\}^{K\times 1}$ | Uncoded information bits. |
| $\mathbf{c} \in \{0,1\}^{\frac{K}{R}\times 1}$ | Coded and interleaved bits. $R$ is the code rate. |
| $\mathbf{w} \in \mathbb{C}^{(M+N)\times 1}$ | Noise vector, circular symmetric zero-mean complex Gaussian with covariance matrix $\lambda^{-1}\mathbf{I}$. |
| $\lambda$ | Noise precision. |
| $a$ and $b$ | Shape and rate parameter for the gamma prior on the noise precision. |

APPENDIX B

# Acronyms

| | |
|---|---|
| **BER** | bit error rate |
| **BP** | belief propagation |
| **CP** | cyclic prefix |
| **CS** | Compressed Sensing |
| **EM** | expectation maximization |
| **EP** | expectation propagation |
| **EPA** | 3GPP Extended Pedestrian A |
| **ETU** | 3GPP Extended Typical Urban |
| **EVA** | 3GPP Extended Vehicular A |
| **FFT** | fast Fourier transform |
| **iid** | independent and identically distributed |
| **MAP** | maximum a posteriori |
| **MF** | mean field |
| **MSE** | mean squared error |
| **OFDM** | orthogonal frequency division multiplexing |
| **pdf** | probability density function |
| **pmf** | probability mass function |
| **RVM** | Relevance Vector Machine |
| **SBL** | Sparse Bayesian Learning |
| **SNR** | signal-to-noise ratio |

APPENDIX C

# Identities and Definitions

### C.1 Complex Gaussian pdf

The pdf of a circular symmetric complex random variable $\mathbf{x} \in \mathbb{C}^N$ is defined as

$$p(\mathbf{x}) = \frac{1}{\pi^N} \det(\boldsymbol{\Sigma})^{-1} \exp\big(-(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{H}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\big) \tag{C.1}$$

where $\boldsymbol{\mu} \in \mathbb{C}^N$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{C}^{N \times N}$ is the covariance matrix. We denote the pdf as $\mathrm{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

### C.2 Gamma Distribution

The pdf of a gamma-distributed random variable $x \in \mathbb{R}^+$ is given by

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha - 1} \exp(-\beta x) \tag{C.2}$$

where $\alpha \in \mathbb{R}^+$ is the shape parameter, $\beta \in \mathbb{R}^+$ is the rate parameter and $\Gamma(\cdot)$ is the gamma function. The pdf is denoted as $\mathrm{Ga}(x; \alpha, \beta)$.

### C.3 Conditional Gaussian Distribution

Given the vector $\mathbf{x} \in \mathbb{C}^N$ with distribution $\mathrm{CN}\left(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$. Partitioning the variable as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \tag{C.3}$$

and doing the corresponding partitioning of the mean and covariance

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \; \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}. \tag{C.4}$$

The conditional distribution $p(\mathbf{x}_a | \mathbf{x}_b)$ will also be complex normal with mean

$$\boldsymbol{\mu}_{a|b} = \mu_a + \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{C.5}$$

and covariance matrix

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} \boldsymbol{\Sigma}_{ba}. \tag{C.6}$$

See [28, sec 2.3.1] for the derivation for the real case.

### C.4 Bayes Rule for Gaussian Random Variables

Given

$$p(\mathbf{x}) = \mathrm{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{C.7}$$

$$p(\mathbf{y}|\mathbf{x}) = \mathrm{CN}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{B}) \tag{C.8}$$

Then

$$p(\mathbf{y}) = \mathrm{CN}(\mathbf{y}; \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{B} + \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\mathrm{H}}). \tag{C.9}$$

See [28, sec 2.3.3] for the derivation.

### C.5 Product of Gaussian pdfs

The product of two Gaussian pdfs is proportional to a Gaussian pdf

$$\mathrm{CN}(\mathbf{x}; \mathbf{a}, \mathbf{A})\, \mathrm{CN}(\mathbf{x}; \mathbf{b}, \mathbf{B}) = k\, \mathrm{CN}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{C.10}$$

where

$$\boldsymbol{\Sigma} = \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} \tag{C.11}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\left(\mathbf{A}^{-1}\mathbf{a} + \mathbf{B}^{-1}\mathbf{b}\right) \tag{C.12}$$

$$k = \mathrm{CN}\left(\mathbf{a}; \mathbf{b}, \mathbf{A} + \mathbf{B}\right) \tag{C.13}$$

### C.6 Woodbury Matrix Inverse Formula

Given invertible matrices $\mathbf{A}$ of dimension $N \times N$ and $\mathbf{C}$ of dimension $K \times K$ and the rectangular matrices $\mathbf{U}$ of dimension $N \times K$ and $\mathbf{V}$ of dimension $K \times N$:

$$(\mathbf{A} + \mathbf{U}\mathbf{C}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} \tag{C.14}$$

APPENDIX D

# Frequency Domain Covariance Matrix for the Static Multipath Channel

From the equations (2.6), (6.6) and (6.7) we obtain

$$
\begin{aligned}
[\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,k} &= \langle h_i h_k^* \rangle_{p(\boldsymbol{\beta},\boldsymbol{\tau},P)} \\
&= \left\langle \sum_{p=1}^{P} \sum_{q=1}^{P} \left\langle \left\langle \beta_p \beta_q^* \right\rangle_{p(\boldsymbol{\beta}|\boldsymbol{\tau})} \exp(-j2\pi\Delta f(\tau_p i - \tau_q k)) \right\rangle_{p(\boldsymbol{\tau}|P)} \right\rangle_{p(P)} \\
&= \left\langle \sum_{p=1}^{P} \left\langle \left\langle |\beta_p|^2 \right\rangle_{p(\boldsymbol{\beta}|\boldsymbol{\tau})} \exp(-j2\pi\Delta f \tau_p (i-k)) \right\rangle_{p(\boldsymbol{\tau}|P)} \right\rangle_{p(P)} \\
&= \left\langle \sum_{p=1}^{P} \left\langle u \exp\!\left(-\frac{\tau_p}{v}\right) \exp(-j2\pi\Delta f \tau_p (i-k)) \right\rangle_{p(\boldsymbol{\tau}|P)} \right\rangle_{p(P)} \\
&= \left\langle \sum_{p=1}^{P} \int_0^{\tau_{\max}} \frac{u}{\tau_{\max}} \exp\!\left(-\tau_p \left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)\right) d\tau_p \right\rangle_{p(P)} \\
&= \left\langle \sum_{p=1}^{P} \frac{u}{\tau_{\max}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)} \left(1 - \exp\!\left(-\tau_{\max}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)\right)\right) \right\rangle_{p(P)} \\
&= \frac{\mu_P \cdot u}{\tau_{\max}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)} \left(1 - \exp\!\left(-\tau_{\max}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)\right)\right)
\end{aligned}
\tag{D.1}
$$

for all $i, k \in [1 : N + M]$. Now choose $u$ such that the diagonal elements of $\boldsymbol{\Sigma}_{\mathbf{h}}$ are normalized to 1

$$
u = \frac{\tau_{\max}}{\mu_P \cdot v\left(1 - \exp\!\left(-\frac{\tau_{\max}}{v}\right)\right)}. \tag{D.2}
$$

Inserting (D.2) in (D.1) yields

$$
[\boldsymbol{\Sigma}_{\mathbf{h}}]_{i,k} = \frac{1 - \exp\!\left(-\tau_{\max}\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)\right)}{v\left(1 - \exp\!\left(-\frac{\tau_{\max}}{v}\right)\right)\left(\frac{1}{v} + j2\pi\Delta f(i-k)\right)}. \tag{D.3}
$$

Now considering the case of flat power delay profile ($v \to \infty$), the two occurrences of $\frac{1}{v} \to 0$ and

$$
\lim_{v\to\infty} v\left(1 - \exp\!\left(-\frac{\tau_{\max}}{v}\right)\right) = \tau_{\max}, \tag{D.4}
$$

thus

$$\lim_{v\to\infty} [\mathbf{\Sigma_h}]_{i,k} = \frac{1 - \exp(-\tau_{\max} j 2\pi \Delta f (i-k))}{\tau_{\max} j 2\pi \Delta f (i-k)}. \tag{D.5}$$

A similar derivation for the scenario where the number of multipath components $P$ is known is shown in [59]. The results for the normalized covariance matrices are equivalent to what is shown here in (D.3) and (D.5).

APPENDIX E

# Correlation Between Columns of Dictionary

With the dictionary $\mathbf{\Phi}$ defined in Section 5.1 we can derive a closed-form expression for the correlation between two columns, i.e. the entries of the matrix $\mathbf{\Phi}^{\mathrm{H}}\mathbf{\Phi}$.

$$\begin{aligned}
[\mathbf{\Phi}^{\mathrm{H}}\mathbf{\Phi}]_{k,\ell} &= \boldsymbol{\psi}(t_k)^{\mathrm{H}}\boldsymbol{\psi}(t_\ell) && \text{(E.1)}\\
&= \sum_{n=1}^{N+M} \exp(j2\pi\Delta f n t_k)\exp(-j2\pi\Delta f n t_\ell) && \text{(E.2)}\\
&= \sum_{n=1}^{N+M} \exp(j2\pi\Delta f n(t_k - t_\ell)) && \text{(E.3)}\\
&= \sum_{n=0}^{N+M-1} \exp(j2\pi\Delta f n(t_k - t_\ell))\cdot\exp(j2\pi\Delta f(t_k - t_\ell)) && \text{(E.4)}\\
&= D_{N+M}(2\pi\Delta f(t_k - t_\ell))\cdot\exp(j2\pi\Delta f(t_k - t_\ell)) && \text{(E.5)}\\
&= \frac{\sin(\pi\Delta f(N+M)(t_k - t_\ell))}{\sin(\pi\Delta f(t_k - t_\ell))}\exp(j\pi\Delta f(t_k - t_\ell)(N+M+1)) && \text{(E.6)}
\end{aligned}$$

for all $k, \ell \in [1:L]$ and where

$$D_K(x) = \sum_{k=0}^{K-1} \exp(jkx) = \frac{\sin(\frac{1}{2}Kx)}{\sin(\frac{1}{2}x)}\exp\big(jx\tfrac{K-1}{2}\big) \qquad \text{(E.7)}$$

is the Dirichlet kernel [57]. When both the numerator and denominator of the amplitude term in E.7 approaches zero the limiting value of the fraction is used instead.

Denote the reduced dictionary $\mathbf{\Phi}_A$ that consists of every $A$th row of $\mathbf{\Phi}$, i.e. $\mathbf{\Phi}_A$ contains row $[1, 1+A, 1+2A, \cdots, \lceil\frac{N+M}{A}\rceil]$ of $\mathbf{\Phi}$. The correlation between the $k$th and the $\ell$th columns can then be written as

$$[\mathbf{\Phi}_A^{\mathrm{H}}\mathbf{\Phi}_A]_{k,\ell} = \sum_{n=1}^{\lceil \frac{N+M}{A} \rceil} \exp(j2\pi\Delta f(A(n-1)+1)t_k)\exp(-j2\pi\Delta f(A(n-1)+1)t_\ell) \tag{E.8}$$

$$= \sum_{n=0}^{\lceil \frac{N+M}{A} \rceil - 1} \exp(j2\pi\Delta f nA(t_k - t_\ell))\exp(j2\pi\Delta f n(t_k - t_\ell)) \tag{E.9}$$

$$= D_{\lceil \frac{N+M}{A} \rceil}(2\pi\Delta f A(t_k - t_\ell)) \cdot \exp(j2\pi\Delta f n(t_k - t_\ell)) \tag{E.10}$$

$$= \frac{\sin(\pi\Delta f A\lceil \frac{N+M}{A} \rceil (t_k - t_\ell))}{\sin(\pi\Delta f A(t_k - t_\ell))} \cdot \exp\big(j\pi\Delta f(t_k - t_\ell)(A(\lceil \tfrac{N+M}{A} \rceil - 1) + 2)\big) \tag{E.11}$$

If $\frac{N+M}{A}$ is an integer the expression can be simplified as

$$[\mathbf{\Phi}_A^{\mathrm{H}}\mathbf{\Phi}_A]_{k,\ell} = \frac{\sin(\pi\Delta f(N+M)(t_k - t_\ell))}{\sin(\pi\Delta f A(t_k - t_\ell))} \cdot \exp(j\pi\Delta f(t_k - t_\ell)(N+M-A+2)) \tag{E.12}$$

APPENDIX F

# Upper Bound for the Best $P$-term Approximation Error

Let $\mathbf{h}_P$ denote the best approximation of $\mathbf{h}$ one can obtain by using a linear combination of $P$ columns from the fixed dictionary $\boldsymbol{\Phi}$ (as defined in Section 5.1) with delay resolution $T_\mathrm{d}$. The approximation $\mathbf{h}_P$ is thus given by (5.13).

The approach to derive an upper bound for the best $P$-term approximation error is to derive an upper bound for a naïve $P$-term approximation. The approach is originally devised in [52], however there is a minor mistake in the derivation and different scaling and notation is used for the dictionary. Now define the naïve $P$-term approximation as

$$\mathbf{h}' = \sum_{p=1}^{P} \alpha'_p \boldsymbol{\psi}(t'_p) \tag{F.1}$$

where $\alpha'_p = \beta_p b_p$ with $b_p$ to be found and $t'_p = T_\mathrm{d}\,\mathrm{round}(\frac{\tau_p}{T_\mathrm{d}})$ is the delay in the search grid that is closest to the actual delay $\tau_p$. Then

$$\begin{aligned} ||\mathbf{h} - \mathbf{h}_P||_2 &\leq ||\mathbf{h} - \mathbf{h}'||_2 && \text{(F.2)}\\ &= \left|\left| \sum_{p=1}^{P} \beta_p \boldsymbol{\psi}(\tau_p) - \sum_{p=1}^{P} \beta_p b_p \boldsymbol{\psi}(t'_p) \right|\right|_2 && \text{(F.3)}\\ &\leq \sum_{p=1}^{P} |\beta_p| \, ||\boldsymbol{\psi}(\tau_p) - b_p \boldsymbol{\psi}(t'_p)||_2 && \text{(F.4)} \end{aligned}$$

Minimizing (F.4) using standard least-squares approach yields the solution

$$b_p = \frac{\boldsymbol{\psi}(t'_p)^\mathrm{H} \boldsymbol{\psi}(\tau_p)}{||\boldsymbol{\psi}(t'_p)||_2^2} = \frac{\boldsymbol{\psi}(t'_p)^\mathrm{H} \boldsymbol{\psi}(\tau_p)}{N+M} \tag{F.5}$$

Now insert the solution into (F.4) to obtain

$$\begin{aligned} \text{(F.4)} &= \sum_{p=1}^{P} |\beta_p| \sqrt{||\boldsymbol{\psi}(\tau_p)||_2^2 - \frac{|\boldsymbol{\psi}(t'_p)^\mathrm{H} \boldsymbol{\psi}(\tau_p)|^2}{N+M}} && \text{(F.6)}\\ &= \sum_{p=1}^{P} |\beta_p| \sqrt{(N+M)\left(1 - \left|\frac{D_{N+M}(2\pi\Delta f(t'_p - \tau_p))}{N+M}\right|^2\right)} && \text{(F.7)}\\ &\leq \sqrt{(N+M)\left(1 - \left|\frac{D_{N+M}(\pi\Delta f T_\mathrm{d})}{N+M}\right|^2\right)} \sum_{p=1}^{P} |\beta_p| && \text{(F.8)} \end{aligned}$$

Finally squaring both sides yields

$$||\mathbf{h} - \mathbf{h}_P||_2^2 \leq \left(1 - \left|\frac{D_{N+M}(\pi \Delta f T_\mathrm{d})}{N+M}\right|^2\right) \left(\sum_{p=1}^{P} |\beta_p|\right)^2 (N+M) \tag{F.9}$$

Note (F.4) comes from the triangle inequality and it holds with equality if and only if all the error terms $\mathbf{e}_p = \boldsymbol{\psi}(\tau_p) - b_p \boldsymbol{\psi}(t'_p)$ are aligned, i.e. if and only if $\mathbf{e}_p$ for all $p \in [2 : P]$ can be obtained from $\mathbf{e}_1$ by multiplication of non-negative, real scalars. The inequality (F.8) holds with equality if and only if the true delays $\tau_p$ for all $p \in [1 : P]$ are located exactly in between two delays of the search grid. We can thus not expect the upper bound to be tight.

APPENDIX G

# Oracle Estimator

The MSE of the 'oracle' estimator is often used in estimation of sparse signals as a lower bound for the MSE of the sparse estimate [62]. It can be shown that for a particular definition of unbiasedness for a sparse signal estimator the MSE of the oracle estimator is equal to the unbiased Cramer-Rao bound [62]. Note that this does not imply that there is no biased estimator with better performance. With the observation model $\mathbf{y} = \mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\beta} + \mathbf{w}$ the oracle estimator for $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}}_o = (\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau}))^{\dagger}\mathbf{y} \tag{G.1}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. Notice that the oracle estimator needs the knowledge of $\mathbf{X}$ and $\boldsymbol{\tau}$ and it is therefore not a practical estimator (if we knew $\mathbf{X}$ there is no reason to estimate $\boldsymbol{\beta}$), but it is still useful for comparison. In this project we are more concerned about the estimate of $\mathbf{h}$ rather than $\boldsymbol{\beta}$ and we thus compute $\hat{\mathbf{h}}_o = \boldsymbol{\Psi}(\boldsymbol{\tau})\hat{\boldsymbol{\beta}}_o$. Treating $\boldsymbol{\tau}$ and $\boldsymbol{\beta}$ as fixed unknown parameters and averaging the estimation error over realizations of the data $\mathbf{x}_{\mathcal{D}}$ and noise vector $\mathbf{w}$, the MSE of this estimator reads

$$\left\langle \left|\left| \boldsymbol{\Psi}(\boldsymbol{\tau})\boldsymbol{\beta} - \boldsymbol{\Psi}(\boldsymbol{\tau})\hat{\boldsymbol{\beta}}_o \right|\right|_2^2 \right\rangle_{p(\mathbf{w})p(\mathbf{x}_{\mathcal{D}})} = \left\langle \left|\left| \boldsymbol{\Psi}(\boldsymbol{\tau})(\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau}))^{\dagger}\mathbf{w} \right|\right|_2^2 \right\rangle_{p(\mathbf{w})p(\mathbf{x}_{\mathcal{D}})} \tag{G.2}$$

Assuming full column rank of $\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau})$ such that

$$(\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau}))^{\dagger} = (\boldsymbol{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau}))^{-1}\boldsymbol{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}\mathbf{X}^{\mathrm{H}} \tag{G.3}$$

the MSE is given by

$$\left\langle \left|\left| \mathbf{h} - \hat{\mathbf{h}}_o \right|\right|_2^2 \right\rangle_{p(\mathbf{w})p(\mathbf{x}_{\mathcal{D}})} = \lambda^{-1} \left\langle \mathrm{trace}\left( (\boldsymbol{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}\mathbf{X}^{\mathrm{H}}\mathbf{X}\boldsymbol{\Psi}(\boldsymbol{\tau}))^{-1}\boldsymbol{\Psi}(\boldsymbol{\tau})^{\mathrm{H}}\boldsymbol{\Psi}(\boldsymbol{\tau}) \right) \right\rangle_{p(\mathbf{x}_{\mathcal{D}})} \tag{G.4}$$

If phase-shift keying is used for all data and pilot symbols and we normalize the symbols to unit power we have $\mathbf{X}^{\mathrm{H}}\mathbf{X} = \mathbf{I}$ and the MSE thus becomes $\frac{P}{\lambda}$ where $P$ is the number of multipath components and $\lambda$ is the noise precision. The MSE of the oracle estimator is thus proportional to the number of multipath components and inversely proportional to the noise precision.

APPENDIX H

# Description of Simulation Framework

This appendix contains the documentation for the simulation software developed for the project and some of the design considerations that have been made in the development.

The purpose of the simulation framework is to evaluate the performance of the devised OFDM receivers in different simulation scenarios. As we are interested in estimating the BER performance of the receivers the results must be averaged over a large number of realizations. In the following some of the design considerations to meet this requirement is discussed.

**Programming Environment** Due to the short time span of the project, fast code development is of high priority when it comes to choice of programming environment. Matlab is chosen over other candidates such as Python and R, because a larger number of existing tools for modeling communication systems is available for Matlab.

**Parallelization and Resumability** When running a large number of simulations it is desirable to be able to run several simulations in parallel to decrease the time needed for obtaining results. Therefore we use the parallel computing toolbox of Matlab to achieve this. The simulations are designed to be resumable such that an additional number of Monte Carlo iterations can be performed after an initial completion of the simulation.

**Testing and Validation** To validate the functionality of the implemented software we use the idea of unit testing, in which small parts of the software are tested individually. If small parts of the framework fail the tests there is no reason to believe that the framework is valid as a whole. To aid the setup and execution of tests the Matlab xUnit [70] framework is used.

## H.1 Structural Overview

The main components of the simulation framework is located in the `src/` folder. We employ the object-oriented programming paradigm and files starting with lowercase letters are functions and files starting with uppercase letters are classes. There are three main types of classes, the signal class, the decoder class and the receiver class.

**Signal Class** Defines the parameters of the simulation scenario and generates signals and channel realizations. The base class is `Signal.m`, from which the two subclasses `SignalExp.m` and `Signal3GPP.m` inherits a lot of functionality but redefines the channel generation method to generate realization of the exponential and the 3GPP channel models respectively. After setting the properties of a signal object, a new realization of the channel and signals is obtained by calling the `randomize()` method.

**Decoder Class** Performs demapping, deinterleaving and decoding as described in Subsection 4.4.4. Can be configured to use the Coded Modulation Library [51] or our own implementation based on BP.

**Receiver Class** The files prefixed with `Rx` each defines one receiver class. This includes the sparse estimation algorithm `RxHierarch.m`, the non-sparse algorithm `RxFreq.m`, the oracle estimator `RxOracle.m` and a receiver with perfect knowledge of the channel coefficients $\mathbf{h}$, `RxPerfectCSI.m`. Each receiver has an `rx()` method that takes a signal object and a decoder object as argument, processes the observed signal and returns a performance matrix. The rows of the performance matrix are BER, MSE of the channel estimate and the value of $\langle\lambda\rangle_{q(\lambda)}$ and each column corresponds to one outer iteration of the algorithm.

The unit tests are located in the `test/` folder. The tests in `test/TestDecoder.m` verifies that the different versions of the decoder decodes without error in a noise-free setting and `test/TestSignal.m` verifies that the SNR of the generated signals matches the targeted SNR. All the unittests can be executed by running the `test/tests_run.m` script.

To run a simulation scenario the `simulate.m` function is used. It is possible to sweep over any parameter of the signal class and the `simulate` function parallelize over the parameter sweep as well as the different algorithms to be evaluated.

The following example shows how to run a simulation scenario using the 3GPP ETU channel model and sweeping the SNR in dB as $[5, 10, 15, 20, 25, 30]$. The algorithms to be evaluated are the sparse channel estimation algorithm in Chapter described in 5 using a group size of 10, the receiver in Chapter 4 using a group size of 100 and the oracle estimator. The simulation uses 8 workers (processes) and averages over 200 Monte Carlo simulations for the 5, 10, 15 and 20 dB scenarios and 1000 Monte Carlo iterations for the 25 and 30 dB scenarios. The result is saved to the file `snr_etu_test.mat`.

```
addpath '../src' '../cml'
dbstop if error
rxh = RxHierarch();
rxh.known_lambda = false;
rxf.group_size = 10;
rxf = RxFreq();
rxf.group_size = 100;
rxf.knwon_lambda = false;
```

```
algs = {rxh, rxf, RxOracle()};
trellis = poly2trellis(4,[13 15],13);
dec = Decoder(trellis, [13 15]);
dec.use_cml = true;
sig = Signal3GPP(trellis, 'etu');

monte_carlo_iters = [200 200 200 200 1000 1000];
parm.SNR_DB = [5 10 15 20 25 30];
parm.pilot_spacing = 10;
parm.num_subcarriers = 600;
parm.num_columns = 150;
simulate('../results/snr_etu_test.mat', ...
 dec, sig, parm, algs, monte_carlo_iters, 8);
```

The result can be plotted afterwards using the `plot_result.m` function. To add more Monte Carlo iterations to the same simulation scenario, simply change the numbers in the vector `monte_carlo_iters` and rerun the script and more iterations will be added to the existing result.

### H.2 Running Time and Speedup

To get an idea of the running time of the implemented algorithms we measure the time it takes to process 5 OFDM blocks sequentially. The signal parameters are as in Chapter 6 and the channel model is as in Table 6.2 and Table 6.3. We use the two-layer sparse estimation algorithm and the robust algorithm as described in Section 6.2. The algorithms are run on a desktop PC with a 2.67 GHz Intel Core i7 920 Quad-core CPU, 6 GB RAM, Matlab R2013a on Arch Linux (kernel version 3.9.2). Each algorithm uses 20 outer iterations. The results are generated by running the script `running_time.m` and the produced output is presented in Table H.1.

In test 1 we use our own Matlab implementation of the BCJR algorithm for decoding as described in Subsection 4.4.4. In the other test setups we use the CML library to do the decoding. In test setup 3 and 4 px_mex refers to the use of a MEX versions of the mapping and demapping functions. The demapping function `src/pde2bi.m` is used to compute the messages $m^{\mathrm{BP}}_{f_{\mathrm{M}_n}\to c_q^{(n)}}(c_q^{(n)})$ for all $n \in [1:N]$ and all $q \in [1:Q]$ as depicted for $n = 1, q = 1$ and $Q = 4$ on Figure H.1a. For each of the $QN$ messages we need to sum over the $2^Q$ valid configurations of $f_{\mathrm{M}_n}(x_{i_n}, \mathbf{c}^{(n)})$. These computations are not well suited for fast computation in the standard Matlab programming language. Similarly the mapping function `src/pbi2de.m` computes the messages $m^{\mathrm{BP}}_{f_{\mathrm{M}_n}\to x_{i_n}}(x_{i_n})$ for all $n \in [1:N]$ as shown in Figure H.1b. To speed up the computation of these messages we have implemented `src/pde2bi_cg.m` and `src/pbi2de_cg.m`. These functions are equivalent to the aforementioned functions but they are better suited for generating Matlab MEX files using the Matlab coder, which translates Matlab functions into C code and MEX files. Executing the build script `src/mex_build.m` generates the MEX versions of these functions.

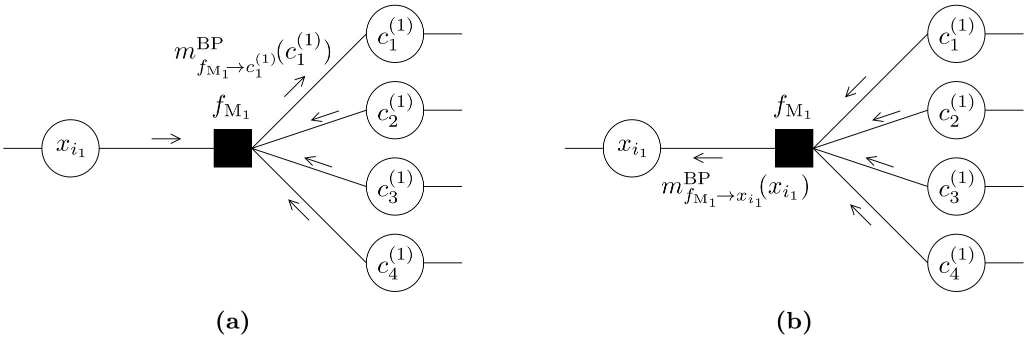


**Figure H.1:** *Example of the messages corresponding to (a) demapping and (b) mapping operations.*

First notice from Table H.1 how employing the CML library greatly reduces the running time of the implemented algorithms. In this example the reduction is approximately 70 seconds for processing 5 OFDM blocks. By using the MEX versions of the mapping and demapping functions the running time is further reduced by 10 seconds. By reducing the group size to $G = 10$ for the sparsity-aware algorithm and $G = 100$ for the receiver in Chapter 4 we obtain a further reduction in running time. Keep in mind that the sparsity-aware algorithm uses several sub-iterations in the channel estimation part of the factor graph and therefore it has higher running times for small group sizes than the other algorithm. We have observed that using even smaller group sizes e.g. $G = 1$ does not reduce the running time of the Matlab implementation because of the larger book-keeping overhead incurred by using smaller group sizes.

| Test | MEX | Group size $G$ | Sparsity-aware | Robust |
|---|---|---|---|---|
| 1 | BP | $(150, 600)$ | 98 | 100 |
| 2 | CML | $(150, 600)$ | 23 | 27 |
| 3 | CML, px_mex | $(150, 600)$ | 13 | 17 |
| 4 | CML, px_mex | $(10, 100)$ | 9 | 4 |

**Table H.1:** *Running time in seconds for the implemented algorithms for decoding 5 OFDM blocks. As we are using $N + M = 600$ subcarriers and the number of columns of the dictionary $\mathbf{\Phi}$ is $L = 150$, a group size of $(150, 600)$ corresponds to 1 group for the sparsity-aware receiver and the robust receiver respectively.*

Be aware that the running times are highly dependent on the choice of hardware platform and the specific details of the software implementation. Therefore we can not generalize the results shown here to conclude on the general performance of the algorithms.

# Bibliography

[1] International Telecommunication Union, "The World in 2013: ICT Facts and Figures," White Paper, Feb. 2013.

[2] Nomor Research, "Progress on LTE Advanced - The New 4G Standard," White Paper, Jul. 2008.

[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[4] E. Riegler, G. Kirkelund, C. Manchon, M. Badiu, and B. Fleury, "Merging belief propagation and the mean field approximation: A free energy approach," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013.

[5] A. P. Worthen and W. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 843–849, 2001.

[6] C. Navarro Manchón, G. E. Kirkelund, E. Riegler, L. P. B. Christensen, and B. H. Fleury, "Receiver Architectures for MIMO-OFDM Based on a Combined VMP-SP Algorithm," *ArXiv e-prints*, Nov. 2011.

[7] M.-A. Badiu, G. Kirkelund, C. Manchon, E. Riegler, and B. Fleury, "Message-passing algorithms for channel estimation and decoding using approximate inference," in *IEEE International Symposium on Information Theory (ISIT)*, Jul. 2012, pp. 2376–2380.

[8] Y. Zhu, D. Guo, and M. L. Honig, "A message-passing approach for joint channel estimation, interference mitigation, and decoding," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, Dec. 2009.

[9] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Prentice Hall, Aug. 2001.

[10] *Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception*, 3GPP Technical Specification 36.104 version 10.1.0 Release 10, Jan. 2011.

[11] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[12] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[13] C. R. Berger, S. Zhou, W. Chen, and P. Willett, “Sparse channel estimation for OFDM: Over-complete dictionaries and super-resolution,” in *IEEE 10th Workshop on Signal Processing Advances in Wireless Communications*, Jun. 2009, pp. 196–200.

[14] W. Bajwa, A. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[15] G. Tauböck and F. Hlawatsch, “A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2008, pp. 2885–2888.

[16] M. E. Tipping, “Sparse Bayesian learning and the Relevance Vector Machine,” *Journal of Machine Learning Research*, vol. 1, pp. 211–244, Jun. 2001.

[17] D. P. Wipf and B. D. Rao, “Sparse Bayesian learning for Basis Selection,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[18] N. L. Pedersen, C. N. i Manchon, D. Shutin, and B. H. Fleury, “Application of bayesian hierarchical prior modeling to sparse channel estimation,” in *IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 3487–3492.

[19] R. Prasad and C. R. Murthy, “Bayesian learning for joint sparse OFDM channel estimation and data detection,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010, pp. 1–6.

[20] P. Schniter, “A message-passing receiver for BICM-OFDM over unknown clustered-sparse channels,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 8, pp. 1462–1474, 2011.

[21] S. Sesia, I. Toufik, and M. Baker, Eds., *LTE - The UMTS Long Term Evolution: from Theory to Practice.* John Wiley & Sons Ltd, 2009.

[22] P. Bello, “Characterization of randomly time-variant linear channels,” *IEEE Transactions on Communications Systems*, vol. 11, no. 4, pp. 360–393, Dec. 1963.

[23] R. Duda, P. Hart, and D. Stork, *Pattern classification*, ser. Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems, networks of plausible inference.* Morgan Kaufmann Publishers, 1988.

[25] J. Winn and C. M. Bishop, "Variational message passing," *Journal of Machine Learning Research*, no. 6, pp. 661–694, Apr. 2005.

[26] E. Xing, M. Jordan, and S. Russell, "A generalized mean field algorithm for variational inference in exponential families," in *Proceedings of the Nineteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, 2003, pp. 583–591.

[27] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelhood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, 1977.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer Science+Business Media, LLC, 2006.

[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons Ltd., 2006.

[30] B. Hu, "A variational Bayesian framework divergence minimization as message passing," Ph.D. dissertation, Department of Electronic Systems, Aalborg University, Denmark, Jul. 2010.

[31] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, ser. Compressed Sensing: Theory and Applications. Cambridge University Press, 2012.

[32] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.

[33] N. L. Pedersen, D. Shutin, C. N. Manchón, and B. H. Fleury, "Sparse estimation using Bayesian hierarchical prior modeling for real and complex models," Apr. 2012, unpublished arXiv:1108.4324v2.

[34] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[35] T. T. Cai, G. Xu, and J. Zhang, "On the recovery of sparse signals via l1 minimization," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3388–3397, Jul. 2009.

[36] D. L. Donoho and M. Elad, "Elad m 2003 optimally sparse representation in general (non-orthogonal) dictionaries via l1 minimization," in *Proc. Natl Acad. Sci. USA 100*, 2003, p. 2197–2202.

[37] S. S. Chen, D. L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[38] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[39] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with application to wavelet decomposition," in *Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 1993, pp. 40–44.

[40] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[41] M. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.

[42] T. Park and G. Casella, "The Bayesian Lasso," *Journal of the American Statistical Association*, pp. 681–686, Jun. 2008.

[43] S. Babacan, R. Molina, and A. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 53–63, Jan. 2010.

[44] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables,*. Dover Publications, Incorporated, 1974.

[45] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and dc programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, Dec. 2009.

[46] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 2003, pp. 3–6.

[47] S. Ji, D. Dunson, and L. Carin, "Multitask compressive sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 92–106, Jan. 2009.

[48] T. L. Hansen, P. B. Jørgensen, N. L. Pedersen, C. N. Manchón, and B. H. Fleury, "Bayesian compressed sensing with unknown measurement noise level," Apr. 2013, unpublished.

[49] V. V. Williams, "Breaking the Coppersmith-Winograd barrier. unpublished manuscript," 2011.

[50] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[51] Iterative Solutions. (2013) Coded modulation library. Version 1.12. [Online]. Available: http://www.iterativesolutions.com/Matlab.htm

[52] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Applied and Computational Harmonic Analysis*, 2012, submitted for publication.

[53] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, Aug. 2005.

[54] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, May 2011.

[55] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, "Compressed sensing off the grid," in *50th Annual Allerton Conference*, Oct. 2012, pp. 778–785.

[56] S. Bourguignon and H. Carfantan, "New methods for fitting multiple sinusoids from irregularly sampled data," *Statistical Methodology*, vol. 5, no. 4, pp. 318–327, Jul. 2008.

[57] E. W. Weisstein, "Exponential Sum Formulas." [Online]. Available: http://mathworld.wolfram.com/ExponentialSumFormulas.html

[58] B. Jorgensen, *Statistical Properties of the Generalized Inverse Gaussian Distribution (Lecture Notes in Statistics 9).* Springer-Verlag New York Inc, 1982.

[59] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, and P. O. Börjesson, "Ofdm channel estimation by singular value decomposition," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 931–938, Jul. 1998.

[60] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*, 1st ed. Prentice Hall, Apr. 1993.

[61] Y. Li, L. Cimini, and N. Sollenberger, "Robust channel estimation for ofdm systems with rapid dispersive fading channels," *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 902–915, Jul. 1998.

[62] Z. Ben-Haim and Y. C. Eldar, "The Cramér-Rao bound for estimating a sparse parameter vector," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, Jun. 2010.

[63] P. M. Lee, *Bayesian Statistics - an introduction*, 3rd ed. Arnold, 2004.

[64] N. L. Pedersen, C. N. Manchón, and B. H. Fleury, "Low complexity sparse bayesian learning for channel estimation using generalized mean field," Apr. 2013, unpublished.

[65] A. Greenbaum, *Iterative methods for solving linear systems.* SIAM Philadelphia, 1997.

[66] M.-A. Badiu, C. N. Manchón, and B. H. Fleury, "Message-passing receiver algorithm with reduced-complexity channel estimation," *IEEE Communications Letters*, 2013, accepted for publication.

[67] D. Schafhuber, G. Matz, and F. Hlawatsch, "Adaptive wiener filters for time-varying channel estimation in wireless OFDM systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2003, pp. IV–688–91.

[68] F. A. Dietrich and W. Utschick, "Pilot-assisted channel estimation based on second-order statistics," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, Mar. 2005.

[69] G. Kirkelund, C. Manchon, L. Christensen, E. Riegler, and B. Fleury, "Variational message-passing for joint channel estimation and decoding in mimo-ofdm," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010, pp. 1–6.

[70] S. Eddins. (2010) Matlab xunit test framework. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/22846-matlab-xunit-test-framework