UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXXVII Ciclo

Advanced Graph-Based Detection for Wireless Channels

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.ma Prof.ssa Amina Piemontese

Dottoranda: *Elisa Conti*

Anni 2021 - 2024

To my family

Abstract

Low-complexity graph-based algorithms provide a useful and powerful tool to limit the huge computational load typically encountered in the design of wireless communications receivers. This thesis focuses on the development of efficient detection techniques aimed at achieving an optimal trade-off between computational complexity and performance.

Specifically, in the context of single-carrier time-varying channels affected by fading or strong phase noise, this research examines approximate message-passing schemes based on the expectation propagation framework. The primary objective is to develop a detection algorithm that, besides solving the complexity problem, operates effectively even with a reduced number of pilot symbols for the channel estimation process, thereby increasing the payload efficiency and improving adaptability to various communication standards. Moreover, current state-of-the-art algorithms rely on turbo iterations between the detector and decoder to refine estimates of both data and channel parameters. However, this approach introduces significant computational complexity, which is generally impractical, and the communication between the two units can sometimes be unfeasible. These limitations further motivate the research presented in this thesis.

Additionally, this work addresses the detection problem for Orthogonal Time Frequency Space (OTFS)-based transmissions in challenging high-mobility conditions. It proposes a message-passing algorithm that leverages the particular structure of the equivalent channel matrix in the Doppler-delay domain, aiming to reach the performance benchmark with affordable complexity under realistic channel conditions.

The performance of the proposed solutions is evaluated by assessing bit error rate and pragmatic capacity through MATLAB-based numerical simulations. The results demonstrate satisfactory performance, suggesting that the proposed approaches provide a viable low-complexity alternative to the existing methods. These findings offer practically implementable solutions in the receiver design for future research and applications.

Contents

List of Figures

List of Tables

List of Acronyms

APP A Posteriori Probability

APSK Amplitude-Phase Shift Keying

AWGN Additive White Gaussian Noise

BCJR Bahl-Cocke-Jelinek-Raviv

BER Bit Error Rate

BP Belief Propagation

BR Bessel Ratio

CP Cyclic Prefix

CSI Channel State Information

CS Channel Shortening

dp-BCJR discretized-phase BCJR

DVB-S2 Digital Video Broadcasting Satellite Second Generation

EP Expectation Propagation

EPA Extended Pedestrian A

EVA Extended Vehicular A

ETU Extended Typical Urban

FG Factor Graph

FN Factor Node

ISFFT Inverse Symplectic Finite Fourier Transform

i.i.d. independent and identically distributed

ISI Intersymbol Interference

KL Kullback-Leibler

LDPC Low-Density Parity Check

LEO Low Earth Orbit

LLR Log-Likelihood Ratio

MMSE Minimum Mean Square Error

LUT Lookup Table

MAP Maximum A Posteriori

MIMO Multiple-Input Multiple-Output

MP Message-Passing

NTN Non-Terrestrial Networks

OFDM Orthogonal Frequency Division Multiplexing

OTFS Orthogonal Time Frequency Space

PC Pragmatic Capacity

pdf probability density function

PLL Phase-Locked Loop

pmf probability mass function

PN Phase Noise

PS Parallel Schedule

PSD Power Spectral Density

PSK Phase Shift Keying

QAM Quadrature Amplitude Modulation

QPSK Quadrature Phase Shift Keying

RV Random Variable

SFFT Symplectic Finite Fourier Transform

SISO Soft-Input Soft-Output

SNR Signal-to-Noise Ratio

SPA Sum-Product Algorithm

SS Serial Schedule

TP Transparent Propagation

UT User Terminal

VN Variable Node

WI Woodbury Identity

YW Yule-Walker

Introduction

Everything should be made as simple as possible,
but not simpler.

– Albert Einstein

Message-Passing (MP) algorithms are methods designed to simplify computations and inference within graphical models by decomposing complex operations into efficient and localized ones. These techniques play a crucial role in addressing various tasks in digital communications, including decoding, detection, equalization, channel estimation, and many others.

Graphical models, which form the foundation on which MP techniques are built, are powerful tools for representing complex probabilistic relationships between variables. They offer a clear and modular approach for the representation of probabilistic models, making them particularly useful in various fields that involve the study of large numbers of interacting variables [4]. In addition to the widely used Bayesian networks and Markov random fields, another important representation is the undirected bipartite graph called Factor Graph (FG).

A very effective framework operating on the FG is the Sum-Product Algorithm (SPA) which computes – either exactly or approximately – the marginal distributions derived from a global function depending on a set of variables. Several algorithms in artificial intelligence, signal processing and digital communications can be derived as specific instances of the SPA [5]. However, sometimes, the exact application of the SPA is not feasible since it would lead to an unmanageable complexity. Therefore, the adoption of approximated techniques is demanded. In particular, variational methods provide mathematical tools for the approximation of intractable distributions with simpler ones, optimized in order to minimize the information loss [6], and subject to specific constraints. Variational inference methods can be iteratively implemented through MP algorithms operating on the FG representing the probabilistic model of the analyzed system.

In the context of digital communications, one of the main goals of a receiver is to optimally recover the transmitted information, minimizing the error probability. To this purpose, the marginalization of complex functions is involved. Since communication systems typically include multiple Random Variables (RVs), which can be both discrete, such as the information bits, and continuous, such as channel parameters, exact inference is, in most cases, unfeasible. Approximate MP algorithms therefore represent a valuable solution for dealing with complexity-manageable schemes.

This thesis addresses the design of advanced detection techniques aimed at achieving an optimal trade-off between computational complexity and receiver performance within wireless channels.

Overview of the Thesis

This thesis is organized as follows.

After a brief overview in Chapter 1 of some tools and frameworks that will be applied to the analyzed scenarios, the following chapters provide the mathematical details and numerical results of the contributions emerged from the research activity that I carried out during these three-years as a PhD student.

In particular, Chapter 2 addresses the problem of signal detection in flat-fading channels. In this context, receivers based on the *Expectation Propagation (EP)* framework, which is reviewed in Chapter 1, appear to be a very promising solution for performing the approximation of multimodal distributions. However, the integration of this framework to the analyzed context presents some critical issues, which will be tackled throughout the chapter. A new algorithm based on this framework is presented and, through simulation results, we demonstrate its applicability to sparse pilot configurations. The work presented in this chapter led to the publication of the journal paper [7].

In Chapter 3, we provide an analysis of EP-based algorithms for the classical problem of coded transmission on a strong Wiener Phase Noise (PN) channel. Also in this case, the analysis includes possible improvements over the native application of EP. We identify its limitations and propose new strategies which reach the performance benchmark while maintaining low complexity, with a primary focus on challenging scenarios where the state-of-the-art algorithms fail. We presented the outcomes of this research in [8, 9].

Finally, Chapter 4 focuses on Orthogonal Time Frequency Space (OTFS) modulated transmissions over doubly-selective channels. Specifically, we analyze multi-satellite transmissions and standard 3GPP multipath time-variant scenarios. We address the design of an MP detection algorithm which leverages the particular structure of the channel matrix in the Doppler-delay domain. The proposed approach significantly reduces the complexity by acting on the choice of interferers to be considered in the detection process, and on the schedule, by prioritizing the strongest elements. The results discussed in this chapter have been partially presented in [10, 11].

Notation

The following notation will be used throughout this thesis. Bold lower-case letters ($\boldsymbol{a}$) denote vectors, bold upper-case letters ($\boldsymbol{A}$) denote matrices. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and the Hermitian of a matrix or vector, respectively. The superscript $(\cdot)^*$ applied to a scalar term denotes its complex conjugate. Given a square matrix $\boldsymbol{B}$ with dimension $N_B \times N_B$, its generic entry in the i-th row and j-th column, with $i, j = 1, \ldots, N_B$, is denoted by $\boldsymbol{B}(i, j)$. Moreover, we denote with $|\,\boldsymbol{B}\,|$ its determinant. If the argument of $|\cdot|$ is a scalar quantity, we denote its absolute value. We indicate with $\Re[\cdot]$

and $\Im[\cdot]$ the real and imaginary parts of a complex quantity, respectively. A complex circularly symmetric (respectively, real) Gaussian RV x with mean η_x and variance σ_x^2 is denoted by $x \sim \mathcal{N}_{\mathbb{C}}(\eta_x, \sigma_x^2)$ [respectively, by $x \sim \mathcal{N}(\eta_x, \sigma_x^2)$]. We denote the complex (respectively, real) Gaussian probability density function (pdf) with argument x, mean η_x and variance σ_x^2 by $g_{\mathbb{C}}(x, \eta_x, \sigma_x^2)$ [respectively, by $g(x, \eta_x, \sigma_x^2)$]. We adopt a similar notation for a Gaussian vector, whose pdf is written as $g_{\mathbb{C}}\left(\boldsymbol{x}, \boldsymbol{\eta_x}, \boldsymbol{C_x}\right)$, where the inverse of the covariance matrix, $\boldsymbol{\Sigma_x} = \boldsymbol{C_x}^{-1}$, is called *precision*. We denote by $p(\cdot)$ the pdf of continuous variables (vectors) and with a capital $P(\cdot)$ the probability mass function (pmf) of discrete variables as well as the pdf of mixed random vectors. We use $\mathbb{E}_p[\cdot]$ to indicate the expectation under the distribution $p(\cdot)$. When $p(\cdot)$ is not specified, it is implicit from the context. When an approximated distribution $q(\cdot)$ minimizes a generic D divergence measure to the exact distribution $p(\cdot)$ over an approximating family $\mathcal{F}$, we say that $q(\cdot)$ is the D-projection of p onto $\mathcal{F}$. Analytically, we define the projection operation $\mathrm{proj}_D[\cdot]$ as

$$\mathrm{proj}_D[p(\cdot)] = \underset{q(\cdot)\in\mathcal{F}}{\mathrm{argmin}} D\left[p(\cdot) \parallel q(\cdot)\right] .$$

We denote with $\mathbb{I}_N$ the identity matrix with dimension N. When the dimension is not specified, it is implicit from the context.

Chapter 1

Background

1.1 Overview of a General Communication System

Throughout this thesis, we refer to the general communication block diagram shown in Figure 1.1. We consider the transmission of a randomly generated sequence of bits $\boldsymbol{b}$, which is first encoded by a channel encoder. This represents a crucial step before transmitting data over a noisy channel. The channel encoder adds redundancy to the data, thus reducing the effective data rate but enhancing the communication robustness. Specifically, we use Low-Density Parity Check (LDPC) codes, a powerful encoding strategy employed in many communication standards, including 4G, 5G, and Wi-Fi. The resulting sequence of encoded bits is then processed by the Mapper block, which associates groups of bits with constellation symbols (e.g., from Phase Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM) modulation formats). Denoting with M the alphabet cardinality, i.e., the number of symbols composing the constellation, each symbol corresponds to a group of $\log_2 M$ bits.

Then, pilot symbols are inserted in the obtained sequence $\boldsymbol{c}$. These symbols are known data which are transmitted for synchronization purposes, so that the receiver can assess the state of the channel. Depending on the employed

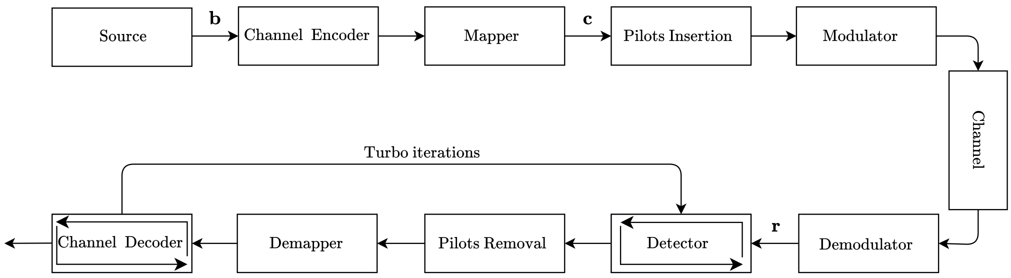


Figure 1.1: General communication block diagram.

communication standard, pilot symbols are placed in different positions, e.g., only in the preamble and postamble of the data packet or periodically inserted within data. Afterwards, the modulator converts the symbols into an analog waveform for the transmission over the wireless communication channel. The transmitted signal can be impaired by various effects which induce a distortion. Beside Additive White Gaussian Noise (AWGN) which is unavoidable, as it represents thermal noise caused by the random motion of electrons within electronic components, we consider additional limiting impairments. Specifically, Chapter 2 focuses on flat-fading channels; in Chapter 3, we analyze the effects of PN, which is a consequence of the instabilities of local oscillators employed both at the transmitter and the receiver; finally, in Chapter 4, we address the challenges associated to doubly-selective channels.

The signal affected by the channel effects undergoes demodulation and sampling. The resulting discrete-time signal $\boldsymbol{r}$ is then passed to the detector unit, which performs both channel estimation and symbol detection based on the optimal decision strategy, i.e., the Maximum A Posteriori (MAP), aiming to minimize the average bit-error probability. In fact, the MAP detector enhances the estimation accuracy by maximizing the posterior pmf of the transmitted data. To this purpose, MP algorithms, like the SPA, represent powerful solutions to perform efficiently the marginalization of the joint posterior pdf, required to perform symbol detection.

It is worth underlying that both the detection and decoding employed strategies fall in the category of soft-output algorithms. As a result, they can potentially iterate both internally and among each other through what are referred to as *turbo* iterations as illustrated in Figure 1.1 (see, e.g., [12, 13] and references therein). When these two units communicate, thus exchanging extrinsic soft information, we refer to joint detection and decoding, as detailed in Section 1.4. In general, the iterations allow for the refinement of both symbols and channel parameters estimates, leading to an improved performance. However, turbo iterations do not represent a practical solution, as they significantly increase complexity.

1.2 Factor Graphs and Sum Product Algorithm

A framework widely used in this work, the FG/SPA tool, is here described using as a reference the tutorial paper [5].

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ be a collection of variables, where x_i takes on values on some (usually finite) domain A_i, and let $F(x_1, x_2, \ldots, x_n)$ be a real-valued function of these variables. Suppose that we want to compute the marginal functions associated to $F(x_1, x_2, \ldots, x_n)$, namely the n functions $F_i(x_i)$ defined as

$$F_i(x_i) \triangleq \sum_{\sim\{x_i\}} F(x_1, x_2, \ldots, x_n)\,, \tag{1.1}$$

where we indicate by $\sum_{\sim\{x_i\}}$ the not-sum or *summary* operator [5], i.e., a sum over all variables except x_i.

The marginalization of a function is a problem common to many applications. Typically, detection problems need the marginalization of a possibly complicated function, depending on many variables. If the function to marginalize can be expressed as the product of simpler functions, each of them depending on a subset of variables, the marginalization turns out to be less complex. The FGs provide a natural graphical description of the factorization of a *global* function into a product of *local* functions. The SPA works on the FG and computes - either exactly or approximately - the marginal functions derived from the global function.

Suppose that $F(x_1, x_2, \ldots, x_n)$ factors into a product of several local functions f_j, each one having a subset X_j of $\{x_1, x_2, \ldots, x_n\}$ as argument, i.e.,

$$F(x_1, x_2, \ldots, x_n) = \prod_{j \in J} f_j(X_j)\,, \tag{1.2}$$

where J is a discrete index set. The FG of the function g is a bipartite graph which has a Variable Node (VN) for each variable x_i, a Factor Node (FN) for each function f_j, and an edge connecting the VN x_i to the FN f_j if and only if x_i is an argument of f_j. The SPA is an MP algorithm working over an FG. It consists of the evaluation of messages at the nodes of the bipartite graph and the exchange of such messages along the edges. It can be proved that if the FG does not have cycles (and hence it admits a tree representation for each variable node) the SPA carries out the exact marginalization of the function represented by the FG.

As shown in Figure 1.2, denoting by $\mu_{x \to f}(x)$ a message sent from the VN x to the FN f, and by $\mu_{f \to x}(x)$ a message in the opposite direction, the message computations performed at VNs and FNs are, respectively,

$$\mu_{x \to f}(x) = \prod_{h \in\, n(x) \setminus \{f\}} \mu_{h \to x}(x) \tag{1.3}$$

$$\mu_{f \to x}(x) = \sum_{\sim \{x\}} \left[f(X) \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right]\,, \tag{1.4}$$

where $n(x)$ is the set of functions f having x as argument, and $X = n(f)$ is the set of arguments of the function f. If the graph is cycle-free, it has a tree representation and the SPA begins by evaluating messages at the leaf nodes, which send to their neighbor nodes an identity message given by $\mu_{x \to f}(x) = 1$ if the leaf is a VN or $\mu_{f \to x}(x) = f(x)$ if it is an FN. Each other node is activated when messages have arrived on all but one of the edges incident on it, and it sends a message on the remaining edge.

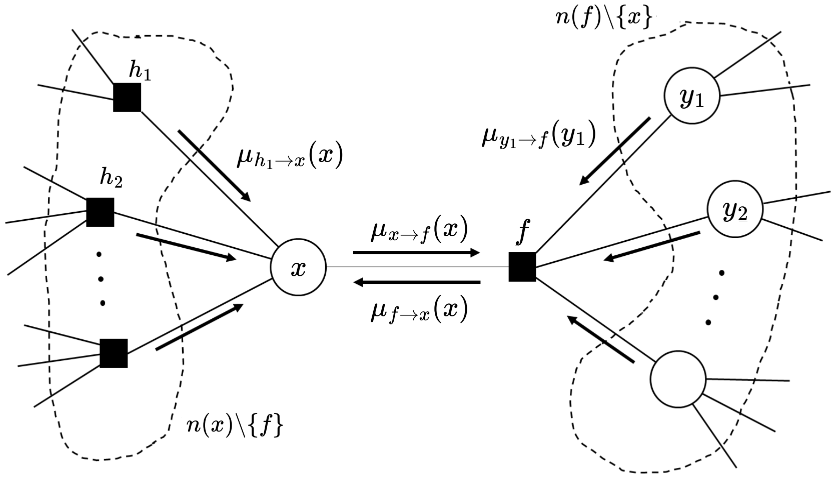


Figure 1.2: An FG fragment where the messages exchanged between FNs and VNs are highlighted.

When two messages have been passed over every edge, one in each direction, the SPA terminates and the marginal functions can be computed as

$$F_i(x_i) = \prod_{f \in n(x_i)} \mu_{f \to x_i}(x_i) \,. \tag{1.5}$$

The SPA can be also used when the underlying FG has cycles, leading to an iterative algorithm. In this case, the convergence to the exact marginalization is not guaranteed, and depends on the adopted schedule, i.e., the order of message computation, and on the structure of the graph. Using graphical techniques described in [5], it is always possible to eliminate the cycles in a graph, at the price of increasing the complexity of the SPA. Nevertheless, for many relevant problems characterized by FGs with cycles, the SPA was found to have good convergence proprieties and to provide a valuable alternative to the exact solution, with an excellent complexity/performance trade-off.

1.3 The Expectation Propagation Framework

The SPA operates without imposing any constraints on messages and sometimes can get exponentially complex. For instance, this situation arises when discrete and continuous variables are mixed within the probabilistic model, as occurs in the scenarios analyzed in Chapters 2 and 3. In these cases, alternative approximate methods are demanded.

The EP framework [14] provides an approximation of the SPA rules. Specifically, it allows to bound the complexity of the algorithm by introducing an exponential family constraint. In particular, the key concept on which EP is based is that the constraint applies on *beliefs*, i.e., marginal distributions which, according to the SPA rules, can be computed via (1.5). EP belongs to a set of MP algorithms, which can be viewed as instances of the same algorithmic structure aiming to the minimization of the *information divergence* when performing the approximation of complex probability distributions [15], i.e., the exact beliefs in (1.5). The various methods distinguish for the measure of divergence that minimize. In particular, EP relies on the minimization of the inclusive Kullback-Leibler (KL) divergence, defined as

$$\mathrm{KL}(p \,\|\, q) = \int_x p(x) \log \frac{p(x)}{q(x)} \, dx \,, \tag{1.6}$$

where we denoted with $p(x)$ the exact distribution and with $q(x)$ the approximating distribution, constrained to belong to a predefined exponential family $\mathcal{F}$. Therefore, $q(x)$ can be written as

$$q(x) \propto \exp \sum_{\ell} g_\ell(x) \lambda_\ell \,, \tag{1.7}$$

where λ_ℓ are the *natural parameters* of the distribution while $g_\ell(x)$ are the fixed *features* of the family. For instance, considering a Gaussian approximating family and working with normalized distributions, the features are $g_1(x) = x$ and $g_2(x) = x^2$.

We say that q is the KL-projection of p onto $\mathcal{F}$ when q belongs to the approximating family $\mathcal{F}$ and minimizes the KL divergence to p, i.e.,

$$q(x) = \operatorname*{argmin}_{q(x)\in\mathcal{F}} \mathrm{KL}(p(x) \,\|\, q(x)) \,. \tag{1.8}$$

A generalization of the KL-divergence is the so-called α-divergence [16] which, depending on the value of $\alpha \in (-\infty, +\infty)$, gives rise to different metrics and, thus, to distinct MP algorithms. Specifically, zero-forcing approaches, which correspond to $\alpha \leq 0$, tend to underestimate the variance of p [15]. For example, in the case of a Gaussian mixture, the approximating distribution will, in the limit case, align with the widest component. On the contrary, inclusive divergence measures ($\alpha \geq 1$) tend to stretch across the mixture modes trying to cover p as much as possible, accordingly to the specific value of α. The KL-divergence corresponds to an α-divergence with $\alpha = 1$ and, therefore, belongs to the family of inclusive divergence measures. However, none approximation is better than the others. The choice is dictated by the properties of p the we want to preserve. If the focus is on the marginal distributions of a complex function, as in the case of interest in Chapters 2 and 3, the most suitable divergence is the KL one.

KL-projection for exponential families leads to solve the minimization problem in (1.8) by selecting the unique distribution belonging to $\mathcal{F}$ whose expectation of the features g_ℓ in (1.7) matches that of p, i.e.,

$$q(x) = \mathrm{proj}_{\mathrm{KL}}[p(x)] \quad \Leftrightarrow \quad \int_x g_\ell(x)\, q(x)\, dx = \int_x g_\ell(x)\, p(x)\, dx \,,\ \forall \ell. \tag{1.9}$$

We therefore call the operation in (1.9) *moment matching*. For instance, if $\mathcal{F}$ is the set of real Gaussian functions, q will be the unique Gaussian whose mean η and variance σ^2 match those of p, i.e.,

$$\eta = \int_x x\, p(x)\, dx = \mathbb{E}_p[x] \,, \tag{1.10}$$

$$\sigma^2 + \eta^2 = \int_x x^2\, p(x)\, dx = \mathbb{E}_p[x^2] \,. \tag{1.11}$$

An example of the KL projection of a Gaussian mixture is reported in Figure 1.3.

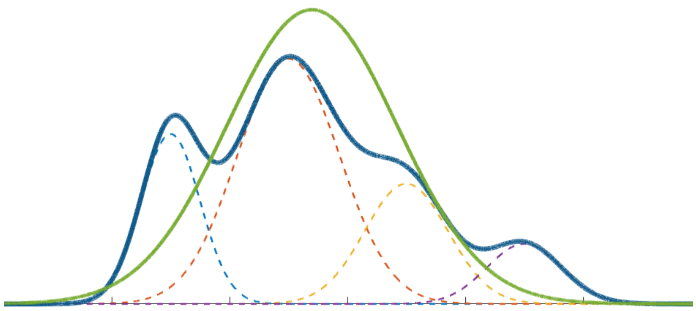


Figure 1.3: Example of a Gaussian mixture (in blue) and its KL-projection (in green).

In the following chapters, detectors based on the EP framework will be discussed. EP will be integrated and combined with the SPA rules obtaining a unified MP framework with the aim of reducing complex distributions to simple *canonical* functions. Specifically, the inherent power of EP is its ability to *merge* information deriving from different portions of the graph and perform an approximation of the overall variable *belief*, instead of the single messages. The resulting algorithm progressively enhances the accuracy of the the local marginal approximations through an *iterative refinement*.

1.4 Iterative Joint Detection and Decoding Schemes

Consider the detection problem when a sequence $\boldsymbol{c} = (c_0, c_1, \ldots, c_{K-1})$ of K symbols is transmitted over an unknown channel. To this purpose, the MAP detection strategy, i.e.,

$$\hat{c}_k = \underset{c_k}{\operatorname{argmax}}\, P(c_k \,|\, \boldsymbol{r}) \,, \tag{1.12}$$

where we denoted with $\boldsymbol{r}$ the discrete-time sufficient statistic of the received signal, is commonly adopted in the receiver design since it minimizes the average symbol error probability. Moreover, the MAP symbol detection strategy provides, as a byproduct, an estimate of the reliability of the decisions through the *a posteriori* probabilities (APPs) which is fundamental in iterative decoding. In fact, Soft-Input Soft-Output (SISO) detection or decoding algorithms work on soft estimates allowing the implementation of joint detection and decoding schemes. An efficient method for the computation of the APPs in (1.12) is provided by the SPA, as discussed in Section 1.2, through a marginalization of the joint pdf of the information symbols and channel parameters ($\boldsymbol{h}$). However, when the sequence of symbols $\boldsymbol{c}$ results from the encoding (and mapping) of a sequence of information bits $\boldsymbol{b}$, the optimal MAP symbol detection strategy requires the inclusion of the code constraints in the marginalization of $P(\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{h} \,|\, \boldsymbol{r})$ with respect to b_k. If the overall system memory becomes too large, the optimal MAP strategy could become unfeasible. Therefore, we typically resort to suboptimal solutions, such as iterative joint detection and decoding schemes, which significantly reduce the receiver complexity while offering performance comparable to the optimal strategy [17]. Furthermore, when the FG has cycles (as is the case, for instance, when employing LDPC codes), the SPA becomes inherently iterative, thereby offering a suboptimal solution. In such iterative concatenated joint detection and decoding schemes, each component block (i.e., the detector and decoder) operates separately, applying the optimal MAP symbol strategy for the individual block, under the assumption that no other memory sources are present in the system. These two blocks can be concatenated due to the reliability estimates which are provided by each block's soft-output. Generally, an iterative concatenated scheme relies on the following core concept: each component block leverages the information provided by the other component block to produce output estimates that become increasingly reliable with the iterations. The detector typically operates first by processing the channel output $\boldsymbol{r}$. Then, it outputs soft-decisions on the information coded symbols $\{c_k\}$ which are forwarded to the decoder. It employs these detector APPs, $P(c_k \,|\, \boldsymbol{r})$, as *a priori* information on $\{c_k\}$ while performing decoding. In turn, the decoder produces soft-decisions on symbols which are then passed to the detector as a priori probabilities, and so on. This iterative process continues until a maximum number of iterations have been performed or convergence is achieved. At the end, hard final decisions on bits $\{b_k\}$ are derived.

In order to speed up the convergence of the iterative process, each component must receive input information that is not self-generated. To this purpose, in [18, 19] the concept of *extrinsic information* has been introduced in order to identify the part of the reliability information produced by each component at the output which does not depend on the information, or "suggestion", from the other component that it processed as input. The FG/SPA tool intrinsically propagates extrinsic information, as highlighted by (1.3)-(1.4).

1.5 Pragmatic Capacity

In some applications, the iterative joint detection and decoding schemes presented in Section 1.4 can be prohibitive due to the high involved computational load. In fact, iterative *turbo* detection, i.e., the reprocessing of the channel observation once the decoder output is available as anticipated in Section 1.1, may lead hardly implementable solutions. Conversely, separate detection and decoding schemes allow to implement detector and decoder in different and independent ways and, sometimes, even in distinct locations. For example, the detector is often implemented in hardware (e.g., in an integrated circuit), while the decoder is typically implemented in software.

In this context, the system performance can be effectively evaluated in terms of *pragmatic capacity*, i.e., a measure of the rate achievable by the detector when no information from the decoder is available. In other words, the pragmatic capacity is a useful performance metric representing the information theoretical rates achievable through separate detection and decoding [20, 21]. As a result, it offers greater practical relevance compared to the typically examined uncoded bit error rate.

Consider the block diagram reported in Figure 1.4. A sequence of $K \log_2 M$ bits $\{b_k\}$, where M denotes the alphabet cardinality, is mapped onto a sequence of K symbols $\{c_k\} \in \mathcal{A}$ which are transmitted over a channel with memory and

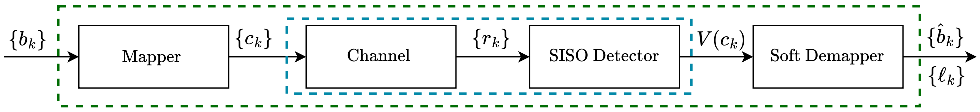


Figure 1.4: Discrete-time equivalent of the communication block diagram considered for the pragmatic capacity computation.

a sequence $\boldsymbol{r}$ of received samples $\{r_k\}$ is obtained. At the receiver, the samples $\{r_k\}$ are processed by a SISO detector followed by a soft-demapper, which provides at the output a decision $\hat{b}_k$ for each transmitted bit b_k and the reliability corresponding to that decision, expressed through the Log-Likelihood Ratio (LLR) ℓ_k, which is given by

$$\ell_k = \ln \frac{P(b_k = 0 \,|\, \boldsymbol{r})}{P(b_k = 1 \,|\, \boldsymbol{r})} \,. \tag{1.13}$$

Notice that the LLR in (1.13) is exact if the detector implements the optimal MAP; otherwise, an approximated LLR will be obtained.

With reference to the communication system in Figure 1.4, the *bit pragmatic capacity* is defined as $I(b_k; \ell_k)$, i.e., the mutual information of the *virtual* channel (highlighted with the outer green box in Figure 1.4) having b_k as input and ℓ_k as output and can be computed as

$$I(b_k; \ell_k) \triangleq H(b_k) - H(b_k \,|\, \ell_k) \,, \tag{1.14}$$

where with $H(\cdot)$ we denote the *entropy*, i.e., a measure of the amount of information required on average to describe a RV [22]. In particular, exploiting the definition of entropy and assuming that the transmitted bits are uniformly distributed, $H(b_k)$ in (1.14) can be expressed as

$$
\begin{aligned}
H(b_k) &= \mathbb{E}\left\{\log_2 \frac{1}{P(b_k)}\right\} &(1.15)\\
&= -P(b_k=0)\log_2 P(b_k=0) - P(b_k=1)\log_2 P(b_k=1) &(1.16)\\
&= -0.5\log_2 \frac{1}{2} - 0.5\log_2 \frac{1}{2} &(1.17)\\
&= 1\,. &(1.18)
\end{aligned}
$$

Analogously, the conditional entropy $H(b_k \,|\, \ell_k)$ of the information bits b_k given the detector's soft-output ℓ_k is

$$
H(b_k \,|\, \ell_k) = \mathbb{E}\left\{\log_2 \frac{1}{P(b_k \,|\, \ell_k)}\right\}\,. \tag{1.19}
$$

In order to compute the posterior pmfs involved in (1.19), we can exploit the definition of LLR in (1.13), obtaining

$$
\begin{aligned}
P(b_k = 0 \,|\, \boldsymbol{r}) &= \frac{\mathrm{e}^{\ell_k}}{1+\mathrm{e}^{\ell_k}} = \frac{1}{1+\mathrm{e}^{-\ell_k}}\,, &(1.20)\\
P(b_k = 1 \,|\, \boldsymbol{r}) &= \frac{1}{1+\mathrm{e}^{\ell_k}}\,. &(1.21)
\end{aligned}
$$

By defining

$$
f(x) = \log_2(1+\mathrm{e}^{-x})\,, \tag{1.22}
$$

it follows that

$$
\begin{aligned}
f(\ell_k) &= -\log_2 P(b_k = 0 \,|\, \boldsymbol{r}) &(1.23)\\
f(-\ell_k) &= -\log_2 P(b_k = 1 \,|\, \boldsymbol{r}) &(1.24)
\end{aligned}
$$

and thus

$$
f(\ell_k(1-2b_k)) = -\log_2 P(b_k \,|\, \boldsymbol{r}) = \log_2 \frac{1}{P(b_k \,|\, \boldsymbol{r})}\,. \tag{1.25}
$$

Hence, Equation (1.14) can be rewritten as

$$
\begin{aligned}
I(b_k;\ell_k) &= 1 - \mathbb{E}\left\{\log_2 \frac{1}{P(b_k \,|\, \ell_k)}\right\} &(1.26)\\
&= 1 - \mathbb{E}\left\{f(\ell_k(1-2b_k))\right\}\,, &(1.27)
\end{aligned}
$$

where in (1.26) we used the results in (1.18)-(1.19), and (1.27) derives from (1.25). The expectation operation $\mathbb{E}[\cdot]$ in (1.27) can be accomplished through Monte Carlo simulations. The resulting (bit) pragmatic capacity can be thus computed as

$$\overline{I}_{\text{PC}} \simeq 1 - \frac{1}{K \log_2 M} \sum_{k=0}^{K \log_2 M - 1} f(\ell_k(1 - 2\bar{b}_k)) \,, \tag{1.28}$$

where $\bar{b}_k$ assumes the value of the actual k-th transmitted bit.

Focusing on *symbols* instead of bits, the pragmatic capacity can alternatively be computed as the achievable rate of the channel induced by the signal constellation at its input and the detector soft-output $V(c_k)$, as highlighted by the blue box in Figure 1.4. Within this formulation, the *symbol pragmatic capacity* directly depends on the employed signal constellation and the soft-output symbol detector. It can be computed as

$$\begin{aligned}
I(c_k; V(c_k)) &\triangleq H(c_k) - H(c_k \,|\, V(c_k)) && (1.29)\\
&= -\sum_{c_k \in \mathcal{A}} P(c_k) \log_2 P(c_k) - \mathbb{E}\left\{\log_2 \frac{1}{P(c_k \,|\, V(c_k))}\right\} && (1.30)\\
&= \frac{1}{M} \sum_{c_k \in A} \log_2 M - \mathbb{E}\left\{\log_2 \frac{1}{V(c_k)}\right\} && (1.31)\\
&= \log_2 M - \mathbb{E}\left\{\log_2 \frac{1}{V(c_k)}\right\} && (1.32)\\
& && (1.33)
\end{aligned}$$

where in (1.31) we assumed that the transmitted symbols are uniformly distributed and we used $V(c_k)$ in place of $P(c_k \,|\, V(c_k))$ since it directly takes on the form of a posterior probability distribution on $c_k \in \mathcal{A}$ for graph-based detectors. Differently, in the case of linear equalizers, $V(c_k)$ is given as the noisy estimate $\hat{c}_k$ which is treated as the output of a (virtual) AWGN channel. Finally, by approximating the mean in (1.32) through a Monte Carlo simulation over K different realizations, we obtain the expression of the (symbol) pragmatic capacity

$$I_{\mathrm{PC}} \simeq \log_2 M - \frac{1}{K} \sum_{k=0}^{K-1} \left\{ \log_2 \frac{1}{V(\bar{c}_k)} \right\} , \tag{1.34}$$

where $V(\bar{c}_k)$ refers to the posterior probability in correspondence to the actual k-th transmitted symbol.

In Chapter 4, we will adopt the symbol pragmatic capacity to test the performance of the analyzed detectors.

Chapter 2

Expectation Propagation for Flat-Fading Channels

2.1 Introduction

Detectors for flat Rayleigh fading channels have been widely studied in the literature (e.g., see [23–25] and references therein). This kind of channels is commonly adopted because it properly represents real scenarios, such as narrowband communications over multipath wireless channels. The framework based on FGs and the SPA provides a powerful tool for the accomplishment of both detection and decoding [5]. However, when this framework is applied jointly to detection and channel estimation, discrete and continuous RVs appear in the FG and the SPA becomes impractical.

A common approach foresees the use of canonical distributions [5]. Focusing on the FG methodology and on the choice of a predefined family which allows some kind of parametrization, different approaches have been analyzed [23, 25]. A canonical distribution that is commonly selected for the messages in this scenario is the Gaussian one. Due to the presence of both discrete and continuous RVs, the *observation* message is typically represented by a multimodal distribution. One of the main differences among the various methods lies in the technique followed for its approximation. In [23], the authors proposed to set the mean to the empirical average of the Gaussian mixture and fix the variance to that of the AWGN. In [25], the approximating pdf is the Gaussian that minimizes the KL divergence from the true message [26] and whose variance depends on the fading sample (i.e., on a VN), which complicates the resulting algorithm. Such a dependence is removed by substituting the fading sample with its estimate performed at the previous iteration. In the separate detection and decoding scenario, this approach coincides with the one of [23]. By reducing the multimodal distributions to single Gaussian ones, the forward and backward recursions of the algorithm correspond to those of a Kalman smoother [27]. Although this approach, based on message projection, is effective in pilot-aided transmission, it eventually fails when the spacing between pilot blocks is longer than a maximum accepted distance, as we will demonstrate in the numerical results.

In this chapter, we focus on a different kind of receiver, which is built upon the EP framework [15] with the goal of addressing this primary limitation. As discussed in Section 1.3, this MP algorithm is based on approximating the distributions of the variables of interest, rather than those of the exchanged messages. This way, a *prior belief* coming from the rest of the FG drives the approximation of intractable messages. The combination of the SPA rules with the approximation deriving from EP leads to a hybrid framework which, in the literature, is commonly referred to as EP-Belief Propagation (EP-BP). This framework has been deeply studied and analyzed in different scenarios, including, e.g., the Multiple-Input Multiple-Output (MIMO) sparse codes multiple access detection [28], the joint channel estimation and decoding for faster-than-Nyquist signaling [29] and also for the transmission over PN channels [8], which will be the core of Chapter 3. As it is inherent to the EP update strategy, improper distributions, e.g., Gaussians with a negative variance, could appear. While in [24] the presence of improper distributions is accepted, and no additional operations are introduced, different approaches are followed by other authors, consisting in either rejecting the message updates that produce improper distributions [30] or, in the case of a multivariate normal distribution, operating on the covariance matrix through a replacement of negative eigenvalues [31]. In [24], the EP framework has been applied to a flat-fading channel, modeled as a complex autoregressive moving-average process. The authors proposed an EP smoothing approach where inner detector iterations are required to achieve convergence and it is tested using differential encoding. However, when using a more powerful encoder, hence working in a lower Signal-to-Noise Ratio (SNR) regime, the approach in [24] is not equally effective and this is the focus of our investigation.

The main contributions that are addressed in this chapter can be summarized as follows.

- We propose an implementation of EP-BP that overcomes its limitations and reduces the algorithmic complexity.
- An improved scheduling which allows to achieve convergence without the aid of multiple inner detector iterations (*iterative refinement* [14]) is adopted.
- A new technique for balancing the variance of the involved Gaussian distributions is introduced.
- The problem of improper distributions is tackled, the scenarios where they can compromise the algorithm operations are identified and a novel strategy to handle these events is presented.

Finally, we demonstrate through simulations that the proposed techniques present a superior performance, in terms of Bit Error Rate (BER) and computational complexity, with respect to state-of-the-art EP implementations [24]. Moreover, the proposed solution ensures a higher robustness against concentrated pilot distributions, laying the foundation for overcoming the limitations related to the classical and well-established Kalman filter.

The rest of the chapter is organized as follows. Section 2.2 introduces the system model. Section 2.3 is devoted to the description of the EP algorithm.

The proposed techniques are introduced in Section 2.4. A complexity comparison among the analyzed algorithms is carried out in Section 2.5. Numerical results are presented in Section 2.6 while Section 2.7 collects some concluding remarks.

2.2 System Model

We consider the transmission of a sequence $\boldsymbol{c} = (c_0, c_1, \ldots, c_{K-1})$ of K complex coded symbols over a flat fading channel. The sequence $\boldsymbol{c}$ is the result of the binary encoding and modulation over an M-ary constellation ($\mathcal{A} = \{a_1, \ldots, a_M\} \subset \mathbb{C}$) of the information bit sequence $\boldsymbol{b}$. Assuming linear modulation with Nyquist shaping pulses, matched filtering, and channel variations slow enough so that no intersymbol interference arises, i.e., channel coherence time much greater than the symbol period T, the discrete-time received signal can be expressed as

$$r_k = g_k c_k + n_k, \quad k = 0, 1, \ldots, K-1 \tag{2.1}$$

where n_k is the AWGN and g_k is the fading process. The vector of noise samples, $\boldsymbol{n} = (n_0, n_1, \ldots, n_{K-1})$, has independent and identically distributed (i.i.d.) complex circularly symmetric components with $n_k \sim \mathcal{N}_{\mathbb{C}}\left(0, 2\sigma^2\right)$.

The fading channel follows Clarke's isotropic scattering model and the fading coefficients $\boldsymbol{g} = (g_0, g_1, \ldots, g_{K-1})$ form a sequence of zero-mean complex Gaussian RVs with unit variance, statistically independent of both $\boldsymbol{c}$ and $\boldsymbol{n}$, with autocorrelation sequence

$$R_g(m) = E\left\{g_{k+m} g_k^*\right\} = J_0(2\pi f_D T m), \tag{2.2}$$

where $J_0(\cdot)$ is the zero-th order Bessel function and f_D is the Doppler spread. At the receiver, the adopted model for the fading is an autoregressive process of order N, AR(N), so that $\boldsymbol{g}$ is approximated by the process $\boldsymbol{h} = (h_0, h_1, \ldots, h_{K-1})$, which obeys[1]

$$h_k = \sum_{n=1}^{N} \rho_n h_{k-n} + \nu_k = \boldsymbol{\rho} \boldsymbol{h}_{k-1}^T + \nu_k \,, \tag{2.3}$$

where $\nu_k \sim \mathcal{N}_{\mathbb{C}}\left(0, 2\sigma_\nu^2\right)$, $\boldsymbol{h}_{k-1} = (h_{k-1}, \ldots, h_{k-N})$ and $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_N)$ are the real AR(N) coefficients. Clearly, there is a channel mismatch between the AR(N) process $\{h_k\}$ and the one modelled by (2.2), hence it is critical to properly set the values of the coefficients $\boldsymbol{\rho}$ in (2.3) and of the random increment's variance $2\sigma_\nu^2$. A classical approach in the literature [23,32] consists in obtaining the coefficients $\boldsymbol{\rho}$ by solving the Yule-Walker (YW) equations $R_h(m) = R_g(m)$, for $m = 1, 2, \ldots, N$, and computing σ_ν^2 as a function of $\boldsymbol{\rho}$ through the constraint $R_h(0) = R_g(0)$. However, in the present context, it turns out that this solution is not the best choice to approximate the actual fading statistics [25]. Therefore, we will select the value of σ_ν^2 that optimizes the performance, as detailed in Section 2.6.

2.3 Detection based on Expectation Propagation

Solving the fading estimation problem through MP algorithms requires the factorization of the joint posterior pdf of the information bits and the fading coefficients, given the received samples. From the system model described above, it can be expressed as

$$\begin{aligned} P(\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{h} | \boldsymbol{r}) &\propto p(\boldsymbol{r} | \boldsymbol{c}, \boldsymbol{h}) p(\boldsymbol{h}) P(\boldsymbol{c} | \boldsymbol{b}) P(\boldsymbol{b}) \\ &\propto \chi(\boldsymbol{c}) p(h_0) \prod_{k=1}^{K-1} p_\nu(h_k - \boldsymbol{\rho} \boldsymbol{h}_{k-1}^T) \prod_{k=0}^{K-1} f_k(c_k, h_k), \end{aligned} \tag{2.4}$$

where the information bits are uniformly and identically distributed, $\chi(\boldsymbol{c})$ is an indicator function representing the code constraints. The factorizations of the pdfs $p(\boldsymbol{h})$ and $p(\boldsymbol{r}|\boldsymbol{c}, \boldsymbol{h})$ in (2.4) come, respectively, from the assumption that the fading process follows the AR(N) model (2.3) and from the fact that the AWGN channel is memoryless, given $\boldsymbol{h}$.

[1] We assume that $h_l = 0$ for $l < 0$.

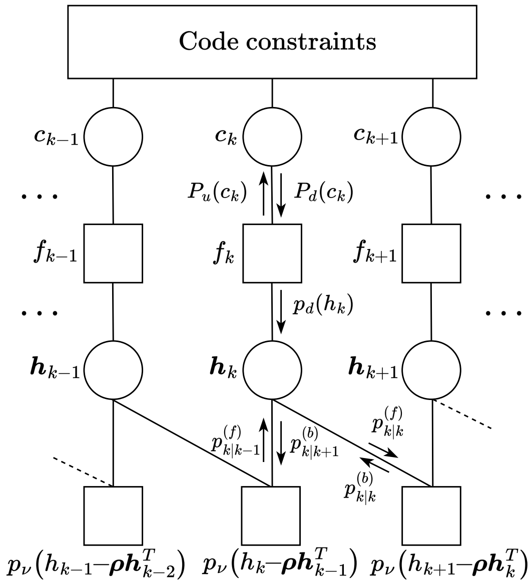


Figure 2.1: FG corresponding to (2.4).

In (2.4), we defined

$$p_\nu(h_k - \boldsymbol{\rho h}^T_{k-1}) \triangleq p(h_k|\boldsymbol{h}_{k-1}) = g_{\mathbb{C}}(h_k, \boldsymbol{\rho h}^T_{k-1}, 2\sigma^2_\nu) \tag{2.5}$$

$$f_k(c_k, h_k) \triangleq p(r_k|c_k, h_k) = g_{\mathbb{C}}(r_k, h_k c_k, 2\sigma^2) \tag{2.6}$$

as the functions associated with the FNs in the FG representation of the joint pdf (2.4), which is reported in Figure 2.1.

Due to the presence of continuous RVs in the FG, we follow the commonly adopted technique based on canonical distributions [5]. In this way, only the parameters characterizing a specific pdf within the family are exchanged in the FG. From (2.3), the distributions that better approximate those of the fading samples are the Gaussian ones. According to the SPA rules [5], outlined in Section 1.2, the *observation* message, i.e., the message from FN f_k to VN $\boldsymbol{h}_k$ is

$$p_d(h_k) \propto \sum_{m=1}^{M} P_d(c_k = a_m) f_k(c_k = a_m, h_k)\,, \tag{2.7}$$

hence it is the mixture of M Gaussian distributions and its exact propagation would be infeasible. Depending on how the message in (2.7) is treated, different algorithms can be derived.

A straightforward projection of the mixture message (2.7) onto a single Gaussian yields the Kalman filtering approach of [23, 25]. Once the projection is performed, all the subsequent messages that are sent along the Markov chain at the bottom of Figure 2.1 belong to the same approximating family because of the closure of the Gaussian family, under both the multiplication and the convolution operations [5, 23].

A different possibility consists in approximating the distributions of VNs, rather than the messages, and this is the key idea on which the EP framework is based. The approximation of $p_d(h_k)$ in (2.7) is thus performed in EP, under the influence of a *temporary prior* distribution which is the product of the incoming messages to the VN $\boldsymbol{h}_k$ from the lower part of the FG. When, on the contrary, a message projection is carried out as in the Kalman filtering approach, so that no prior information is exploited, this is sometimes called *transparent* propagation algorithm [33, 34].

As discussed in Section 1.3, EP is an MP algorithm whose approximations are generated by projecting the variable distribution on a fully-factorized exponential family $\mathcal{F}$, where the projection is performed by minimizing the KL divergence. These two features allow to compute the projection by solving a set of *moment matching* equations [15]. In the present case of a Gaussian approximating family, the distribution resulting from the projection is thus the unique Gaussian whose mean, variance and scale factor equal those of the distribution to be approximated.

The specific algorithmic steps of EP are described in the following. Starting from (2.7), the message $p_d(h_k)$ can be rewritten as a function of the channel parameter h_k as follows

$$p_d(h_k) = \sum_{m=1}^{M} P_m \, g_{\mathbb{C}}(h_k, \eta_m, 2\sigma_m^2) \, , \tag{2.8}$$

where

$$P_m = \frac{P_d(c_k = a_m)}{|a_m|^2} \quad , \quad \eta_m = \frac{r_k}{a_m} \quad , \quad 2\sigma_m^2 = \frac{2\sigma^2}{|a_m|^2} \quad . \tag{2.9}$$

The dependence of P_m and η_m on the time epoch k is omitted for simplicity of notation. The temporary prior distribution $p^{(u)}(\boldsymbol{h}_k)$ is defined as

$$\begin{aligned} p^{(u)}(\boldsymbol{h}_k) &\triangleq p^{(f)}_{k|k-1}(\boldsymbol{h}_k) p^{(b)}_{k|k+1}(\boldsymbol{h}_k) \\ &= g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k-1}, \boldsymbol{C}_{k|k-1}\right) g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k+1}\right) \\ &\propto g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) \ , \end{aligned} \tag{2.10}$$

where the $\propto$ sign derives from the messages normalization, and, by defining $\boldsymbol{\Sigma}_{f,k} \triangleq \boldsymbol{C}^{-1}_{k|k-1}$ and $\boldsymbol{\Sigma}_{b,k} \triangleq \boldsymbol{C}^{-1}_{k|k+1}$ as the precisions of the forward and backward Gaussian messages, respectively, and $\boldsymbol{\Sigma}_{u,k} \triangleq \boldsymbol{C}^{-1}_{u,k}$, the following Gaussian product rules can be applied

$$\boldsymbol{\Sigma}_{u,k} = \boldsymbol{\Sigma}_{b,k} + \boldsymbol{\Sigma}_{f,k} \ , \tag{2.11}$$

$$\boldsymbol{\Sigma}_{u,k}\boldsymbol{\eta}_{u,k} = \boldsymbol{\Sigma}_{f,k}\boldsymbol{\eta}_{k|k-1} + \boldsymbol{\Sigma}_{b,k}\boldsymbol{\eta}_{k|k+1} \ . \tag{2.12}$$

A detailed derivation of the results in (2.10)-(2.12) can be found in Appendix A.

The marginal distribution of $\boldsymbol{h}_k$ is given by the product of all the incoming messages to the corresponding VN, according to the SPA rules [5]

$$\begin{aligned} \hat{p}(\boldsymbol{h}_k) &= p_d(\boldsymbol{h}_k) p^{(f)}_{k|k-1}(\boldsymbol{h}_k) p^{(b)}_{k|k+1}(\boldsymbol{h}_k) \\ &= \sum_{m=1}^{M} P_m \, g_{\mathbb{C}}\left(h_k, \eta_m, 2\sigma_m^2\right) g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{p,k}, \boldsymbol{C}_{p,k}\right) \\ &= \sum_{m=1}^{M} P_m w_m \, g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{MIX,m}, \boldsymbol{C}_{MIX,m}\right) \ , \end{aligned} \tag{2.13}$$

where w_m in (2.13) is the symbol-dependent weight that multiplies the Gaussian distribution corresponding to the m-th symbol in (2.7) and is defined as

$$w_m = g_{\mathbb{C}}\left(\eta_m, \boldsymbol{a}^T\boldsymbol{\eta}_{u,k}, 2\sigma_m^2 + \boldsymbol{a}^T\boldsymbol{C}_{u,k}\boldsymbol{a}\right) \ , \tag{2.14}$$

where $\boldsymbol{a} = (1, 0, \ldots, 0)^T$ is a column vector of dimension N while $\boldsymbol{C}_{MIX,m}$ and $\boldsymbol{\eta}_{MIX,m}$ result:

$$\boldsymbol{\eta}_{MIX,m} = \boldsymbol{\eta}_{u,k} + \boldsymbol{J}_{m,k} \left(\eta_m - \boldsymbol{a}^T \boldsymbol{\eta}_{u,k}\right) , \tag{2.15}$$

$$\boldsymbol{C}_{MIX,m} = \boldsymbol{C}_{u,k} - \boldsymbol{J}_{m,k}\, \boldsymbol{a}^T \boldsymbol{C}_{u,k} , \tag{2.16}$$

$$\boldsymbol{J}_{m,k} = \boldsymbol{C}_{u,k}\, \boldsymbol{a} \left(2\sigma_m^2 + \boldsymbol{a}^T \boldsymbol{C}_{u,k}\, \boldsymbol{a}\right)^{-1} . \tag{2.17}$$

The detailed derivation of (2.14)-(2.17) can be found in Appendix B.

Notice that the variable distribution in (2.13) is still a combination of M Gaussians but the weight of each one, w_m, depends on the distance between the mean of that mode (η_m) and that of the temporary prior distribution. The closer they are, the more the corresponding mode is reinforced by the prior and vice versa, as per (2.14). The approximation of the mixture (2.13) with a simple Gaussian $p^{(EP)}(\boldsymbol{h}_k) = \mathrm{proj}_{\mathrm{KL}}\left[\hat{p}(\boldsymbol{h}_k)\right]$ is performed, as usual in EP, by *matching the moments* of the distributions. The zero-th order moment is the mass of the distribution, i.e, $Z = \sum_{m=1}^{M} P_m w_m$. However, the mass of the messages is not significant in MP algorithms like SPA or EP, therefore we consider $p^{(EP)}(\boldsymbol{h}_k)$ with unit mass (hence a proper pdf) and replace $\hat{p}(\boldsymbol{h}_k)$ in (2.13) with its normalized version

$$p_h(\boldsymbol{h}_k) = \frac{\hat{p}(\boldsymbol{h}_k)}{Z} = \sum_{m=1}^{M} W_m g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{MIX,m}, \boldsymbol{C}_{MIX,m}\right) . \tag{2.18}$$

where $W_m = Z^{-1} P_m w_m$ are defined as normalized weights. The matching of the first order moment is thus performed by equating the mean of the two distributions $\mathbb{E}_{p^{(EP)}}\left[\boldsymbol{h}_k\right] = \mathbb{E}_{p_h}\left[\boldsymbol{h}_k\right]$. It follows that the mean of $p^{(EP)}(\boldsymbol{h}_k)$ is

$$\boldsymbol{\eta}_{EP} = \sum_{m=1}^{M} W_m\, \boldsymbol{\eta}_{MIX,m} \, . \tag{2.19}$$

Then, by matching the second order moments $\mathbb{E}_{p^{(EP)}}[|\boldsymbol{h}_k|^2] = \mathbb{E}_{p_h}[|\boldsymbol{h}_k|^2]$, the covariance matrix of the approximating distribution results to be

$$\boldsymbol{C}_{EP} = \sum_{m=1}^{M} W_m \left(\boldsymbol{C}_{MIX,m} + \boldsymbol{\eta}_{MIX,m} \boldsymbol{\eta}_{MIX,m}^H\right) - \boldsymbol{\eta}_{EP} \boldsymbol{\eta}_{EP}^H \, . \tag{2.20}$$

Now, it is possible to compute the updated message

$$\tilde{p}_d(\boldsymbol{h}_k) \propto g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_d, \boldsymbol{C}_d\right) , \tag{2.21}$$

which is transmitted from the FN f_k to the VN $\boldsymbol{h}_k$ in Figure 2.1, by dividing the approximating Gaussian $p^{(EP)}(\boldsymbol{h}_k)$ by the temporary prior distribution $p^{(u)}(\boldsymbol{h}_k)$ in (2.10). We obtain

$$\boldsymbol{\Sigma}_d = \boldsymbol{\Sigma}_{EP} - \boldsymbol{\Sigma}_{u,k} , \tag{2.22}$$

$$\boldsymbol{\Sigma}_d \boldsymbol{\eta}_d = \boldsymbol{\Sigma}_{EP} \boldsymbol{\eta}_{EP} - \boldsymbol{\Sigma}_{u,k} \boldsymbol{\eta}_{u,k} . \tag{2.23}$$

The approximated message is then propagated following the SPA rules [5]. Further details concerning the computation of the forward message $p^{(f)}_{k|k-1}(\boldsymbol{h}_k)$, the backward message $p^{(b)}_{k|k+1}(\boldsymbol{h}_k)$ and the upward message $P_u(c_k)$ can be found in Appendix C.

2.4 The Proposed Algorithm

The classical EP implementation [24] presents some critical issues which do not lead to a satisfactory performance in the considered scenario. In the following, we present novel techniques which allow to overcome the limitations related to the standard EP structure.

2.4.1 Damping and Boosting Factors

In order to face the EP instabilities, as discussed in the literature, a number of solutions exists, including *damping* [30, 31]. A damped update of a pdf is a member of the approximating family characterized by a convex combination of new parameters (i.e., updated according to the current EP iteration) and old ones (i.e., from previous iteration). However, unlike its common use, in the analyzed scenario the introduction of this correction factor is studied specifically to provide a performance improvement through the balancing of overly confident estimates without the involvement of old updates. In fact, in our implementation, the damping factor, ξ_d, which belongs to the interval $[0, 1]$, is applied to the precision of the approximating message $\tilde{p}_d(h_k)$ only. In this way, the algorithm underestimates the updates reliability as commonly done in the literature when dealing with sub-optimal algorithms. We propose to set this factor to the maximum value of $P_d(c_k)$, i.e., the probability mass function of the symbols. The main advantage of this approach is that a lower number of turbo iterations is required for achieving convergence. At the first turbo iteration[2] (which is the only one in the separate detection and decoding case) this quantity is equal to $1/M$. On the contrary, from the 2nd iteration ahead this value becomes a clearer and clearer representation of the decoder confidence about the transmitted symbols. It follows that the closer this value is to 1, the higher the weight given to the approximating message after the KL projection.

Moreover, we propose an additional technique which allows to obtain a further performance improvement: the use of a *boosting* factor for pilot symbols. Similarly to damping, the precision (2.22) is multiplied by a factor that, this time, is greater than 1, so that its effect is the opposite: it increases the confidence of the observation message in correspondence with a pilot symbol, i.e., when $p_d(h_k)$ is a unimodal Gaussian distribution and, therefore, no approximation is performed. The selected value for the boosting factor is $\xi_b = 2$.

2.4.2 Negative Variances

Due to the subtraction in (2.22), it may occur that the algorithm generates negative variances, so that the approximation of the observation message, $\tilde{p}_d(\boldsymbol{h}_k)$, is not necessarily a valid pdf [24]. This happens when the precision of the prior distribution is larger than the one of the marginal after the KL projection. Since the focus of EP is on the marginal distribution of the channel parameters, the algorithm could accept improper distributions as messages. However, when the scheduling described in Section 2.4.3 is adopted for the separate detection and decoding case, the only message which can present such variances

[2] We will refer to the iterations between detector and decoder as *turbo* iterations.

is the observation one. When turbo iterations are considered, improper distributions could instead propagate giving rise to numerical problems. Hence, in this case, we propose an alternative approach to deal with negative variances, where the algorithm performs a message projection instead of a variable distribution projection. This leads to a significant performance improvement with respect to the propagation of improper distributions as will be shown in Section 2.6. The proposed method consists in approximating the message $p_d(h_k)$ locally with a Gaussian distribution whose mean and variance are obtained through moment matching without the influence of the rest of the network.

The approximating message can be written as

$$\hat{p}_d(h_k) = \mathrm{proj}_{\mathrm{KL}}\left[\sum_{m=1}^{M} \frac{P_d(c_k = a_m)}{|a_m|^2}\, g_{\mathbb{C}}\left(h_k, \frac{r_k}{a_m}, \frac{2\sigma^2}{|a_m|^2}\right)\right] . \tag{2.24}$$

With the constraint on the Gaussian approximating family, the new mean and variance will be

$$\eta_d = \sum_{m=1}^{M} \frac{P_d(c_k = a_m)}{|a_m|^2} \cdot \frac{r_k}{a_m} , \tag{2.25}$$

$$2\sigma_d^2 = \sum_{m=1}^{M} \frac{P_d(c_k = a_m)}{|a_m|^2} \cdot \left(\frac{2\sigma^2}{|a_m|^2} + \left|\frac{r_k}{a_m}\right|^2\right) - |\eta_d|^2 . \tag{2.26}$$

Therefore, if, following the main procedure where the KL divergence minimization is performed on the marginal distribution, negative variances occur, the resulting observation message update is discarded and the new technique is applied.

2.4.3 Scheduling

We assume that the detector subgraph operates first by exchanging *horizontal* messages along the Markov chain in the lower part of the FG shown in Figure 2.1. This message passing, implemented inside the detector, could be iterated with multiple forward-backward passes (N_D) before sending *vertical* messages to the upper part of the FG where the decoder operates. An inner message passing procedure could take place also there until convergence or until a maximum number of decoder iterations (N_C) is reached. The exchange of information, through vertical messages, between detector and decoder can be iterated for N_T times.

With reference to the generic $n_{iter,T}$ turbo iteration, the proposed algorithm structure is outlined in Algorithm 1.

Notice that the forward and backward recursions lead to two different approximations for $p_d(h_k)$. The main advantage of the adopted scheduling is that a satisfying performance is already achieved after a single forward-backward sweep inside the detector.

2.5 Complexity Analysis

The complexity of the considered algorithms is evaluated with reference to one inner detector iteration and for a single time epoch k. We performed a simplified analysis by counting the number of required operations distinguishing among two-(scalar) terms multiplications, additions and accesses to the Lookup Table (LUT) for the computation of nonlinear terms. The operations needed for the matrices inversions can be divided in additions and multiplications depending on the matrix dimension as follows

	Two-terms additions	**Two-terms multiplications**
1×1	0	1
2×2	1	8
3×3	11	30

We considered the operations needed for the observation message approximation, the forward and backward messages computation and the completion step for the three analyzed algorithms: the proposed EP, the classical EP [24] and the Kalman filter. In the following we will indicate the total number of operations for each specific step as a function of the alphabet cardinality M and the order N of the adopted AR model. The literal computation is followed

Algorithm 1 Proposed EP algorithm

1: **Initialize** the messages $p^{(f)}_{0|-1}(\boldsymbol{h}_0)$ and $p^{(b)}_{K-1|K}(\boldsymbol{h}_{K-1})$ by setting $\boldsymbol{\Sigma}_{f,0}$, $\boldsymbol{\Sigma}_{b,K-1}$ to $\mathbb{I}_N$ and $\boldsymbol{\eta}_{0|-1}$, $\boldsymbol{\eta}_{K-1|K}$ to $\mathbf{0}$

2: **for** $k = 0 : K-2$ **do**

3: Compute the *temporary prior* distribution, i.e., update its parameters $\boldsymbol{\eta}_{u,k}$ and $\boldsymbol{\Sigma}_{u,k}$ via (2.11)-(2.12). The backward message $p^{(b)}_{k|k+1}(\boldsymbol{h}_k)$ in (2.10) is set to the one computed at the $(n_{iter,T}-1)$-th turbo iteration. If $n_{iter,T} = 1$, it is set to a uniform message for any value of k

4: Compute the parameters of the *exact marginal* distribution (i.e., for each m, compute the terms in (2.9), (2.14)-(2.17))

5: Perform the *moment matching*, using (2.19)-(2.20)

6: Compute $\tilde{p}_d(\boldsymbol{h}_k)$ via (2.22)-(2.23) and perform either *damping* or *boosting* on $\boldsymbol{\Sigma}_d$ depending on k

7: **if** $(\boldsymbol{\Sigma}_d(1,1) < 0)$ & $(n_{iter,T} > 1)$ **then**

8: Perform the *observation* message projection via (2.25)-(2.26)

9: **end if**

10: Compute the forward messages $p^{(f)}_{k|k}(\boldsymbol{h}_k)$ and $p^{(f)}_{k+1|k}(\boldsymbol{h}_{k+1})$ through the SPA rules [5] (refer to Figure 2.1 and to Appendix C for details)

11: **end for**

12: **for** $k = K-1 : -1 : 1$ **do**

13: Compute $p^{(u)}(\boldsymbol{h}_k)$ via (2.10)-(2.12) where the forward message $p^{(f)}_{k|k-1}(\boldsymbol{h}_k)$ is the one computed at the $(n_{iter,T}-1)$-th turbo iteration. If $n_{iter,T} = 1$, it is set to a uniform message for all ks

14: Repeat the steps 4-9

15: Compute the backward messages $p^{(b)}_{k|k}(\boldsymbol{h}_k)$ and $p^{(b)}_{k-1|k}(\boldsymbol{h}_{k-1})$ through the SPA rules [5]

16: **end for**

17: Multiply the forward message obtained through 2-11 with the backward one computed in 12-16 for each k

18: Compute for each k and each $c_k \in \mathcal{A}$ the probabilities $P_u(c_k)$ to be sent to the decoder as $P_u(c_k) \propto g_{\mathbb{C}}\left(r_k, c_k \boldsymbol{a}^H \boldsymbol{\eta}_{u,k}, 2\sigma^2 + |c_k|^2 \boldsymbol{a}^H \boldsymbol{C}_{u,k} \boldsymbol{a}\right)$, where $\boldsymbol{\eta}_{u,k}$ and $\boldsymbol{C}_{u,k}$ derive from 17

	Two-terms additions	$N = 1$	$N = 2$	**LUT accesses**
Kalman	$2M$	8	8	2
EP proposed	$3N^2(M+1) + N(2M-4) + 4M$	35	86	M
EP classical	$N^2(M+8) + N(5M+11) + 7M + 2$	77	184	M

	Two-terms multiplications	$N = 1$	$N = 2$	**Matrix inversions**
Kalman	$2M + 3$	11	11	0
EP proposed	$N^3M + N^2(2M+4) + 3NM + 6M + 3$	57	147	2 (Dim: N)
EP classical	$N^2(4M+6) + N(12M+16) + 14M + 11$	185	403	4 (Dim: $N+1$)

Table 2.1: Number of operations required for the *observation* message approximation.

by a blue number which refers to the result for $M = 4$ and $N = 1$, and by a red one which refers to the result for $M = 4$ and $N = 2$, according to the parameters used in the simulation results that will be shown in Section 2.6. Notice that these final values also include the operations needed for the matrix inversion. The number of required operations for the observation message approximation, the forward and backward messages computation and, finally, the completion step are reported in Tables 2.1, 2.2, 2.3 and 2.4, respectively.

Notice that the higher computational load involved by the classical EP strategy is mainly due to the scheduling choice which foresees a *backward smoothing*. Now, considering one inner detector iteration, we can count the total number of operations required by the algorithms for each time epoch k considering that, for the EP-based algorithms, the observation message approximation is performed twice, once for the backward recursion and once for the forward recursion; in fact, the two recursions give rise to two different approximations of the message $p_d(h_k)$.

The complexity comparison is shown in Figure 2.2 versus the constellation cardinality M whereas, in Table 2.5, the total number of required operations is reported for $M = 4$. It can be noticed that the additional operations involved by the EP framework lead inevitably to a complexity increase which, relying

	Two-terms additions	$N = 1$	$N = 2$	**LUT accesses**
Kalman	$2N^3 + 4N^2 - 3N$	3	29	0
EP proposed	$2N^3 + 4N^2 - 3N$	3	29	0
EP classical	$2N^3 + 9N^2 + 10N + 3$	26	97	0

	Two-terms multiplications	$N = 1$	$N = 2$	**Matrix inversions**
Kalman	$2N^3 + 4N^2$	9	56	3 (Dim: N)
EP proposed	$2N^3 + 4N^2$	9	56	3 (Dim: N)
EP classical	$2N^3 + 9N^2 + 12N + 5$	44	141	2 (Dim: $N + 1$)

Table 2.2: Number of operations required for the *forward* message computation.

	Two-terms additions	$N = 1$	$N = 2$	**LUT accesses**
Kalman	$2N^3 + 4N^2 - 3N$	3	29	0
EP proposed	$2N^3 + 4N^2 - 3N$	3	29	0
EP classical	$5N^3 + 14N^2 + 13N + 4$	37	137	0

	Two-terms multiplications	$N = 1$	$N = 2$	**Matrix inversions**
Kalman	$2N^3 + 4N^2$	9	56	3 (Dim: N)
EP proposed	$2N^3 + 4N^2$	9	56	3 (Dim: N)
EP classical	$5N^3 + 17N^2 + 19N + 7$	56	183	1 (Dim: $N + 1$)

Table 2.3: Number of operations required for the *backward* message computation.

	Two-terms additions	$N = 1$	$N = 2$	**LUT accesses**
Kalman	$4N^2 - 2N + 2M$	10	23	M
EP proposed	$4N^2 - 2N + 2M$	10	23	M
EP classical	0	0	0	0

	Two-terms multiplications	$N = 1$	$N = 2$	**Matrix inversions**
Kalman	$3N^2 + 4M + 2$	24	54	3 (Dim: N)
EP proposed	$3N^2 + 4M + 2$	24	54	3 (Dim: N)
EP classical	$2M$	8	8	0

Table 2.4: Number of operations required for the *completion* message computation.

	Two-terms additions		**Two-terms multiplications**		**LUT accesses**
	$N = 1$	$N = 2$	$N = 1$	$N = 2$	–
Kalman	**24**	**97**	**53**	**177**	**6**
EP proposed	**86**	**261**	**148**	**452**	**12**
EP classical	**217**	**602**	**478**	**1138**	**8**

Table 2.5: Complexity comparison for $M = 4$ and $N_T = 1$ at a fixed time epoch k.

on our implementation, is still limited and reasonably comparable to the one of the Kalman filter. On the other hand, a considerable complexity increase is registered for the classical strategy [24].

2.6 Simulation Results

The performance of the proposed algorithm was analyzed in terms of BER versus E_b/N_0, E_b being the received signal energy per information bit and $N_0 = 2\sigma^2$ the one-sided noise Power Spectral Density (PSD). We considered a $(3, 6)$-regular rate-$1/2$ LDPC code with length 4000 and Quadrature Phase Shift Keying (QPSK) modulation ($M = 4$). In all simulations, in order to

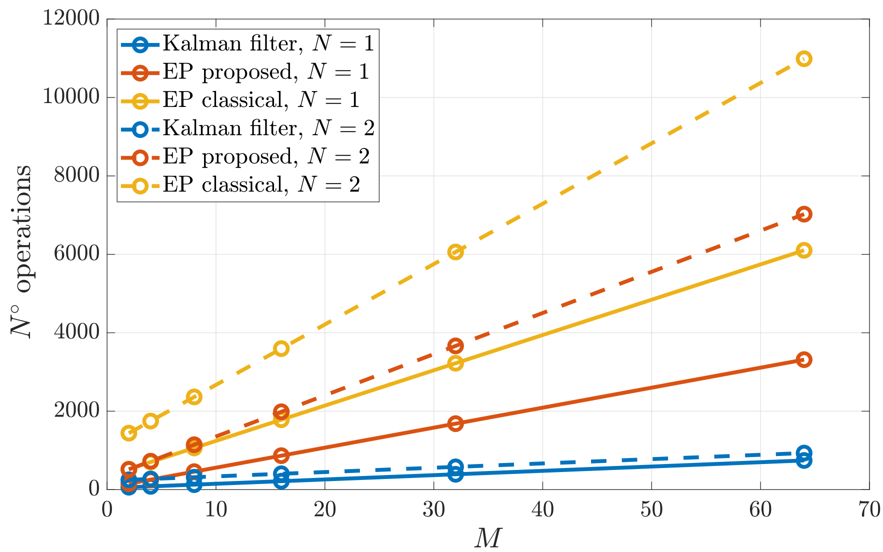


Figure 2.2: Complexity comparison.

allow the convergence of the algorithm, pilot symbols were inserted in the transmitted codeword.[3] The presence of pilots involves a slight decrease of the effective information rate, resulting in an increase in the required SNR. This increase has been artificially introduced in the curve labeled "All Pilots", inserted as ideal bound, for the sake of comparison. Hence, the gap between the "All Pilots" and the other curves is uniquely due to the need for fading estimation/compensation, and not to the decrease due to pilot symbols. The curves labeled "Kalman filter" in Figures 2.3 and 2.5 refer to the algorithm proposed in [25]. The algorithms were tested for both the AR(1) and AR(2) fading models. The values of the parameter $\boldsymbol{\rho}$ were set according to the YW equations, whereas the increment variance σ_ν^2 was set to the one minimizing the BER, found through computer simulations. In fact, the YW equations lead to a value of the increment variance σ_ν^2 which does not allow the detector to work

[3] We use the notation {number of pilots in a block}/{blocks distance} to describe a pilot distribution.

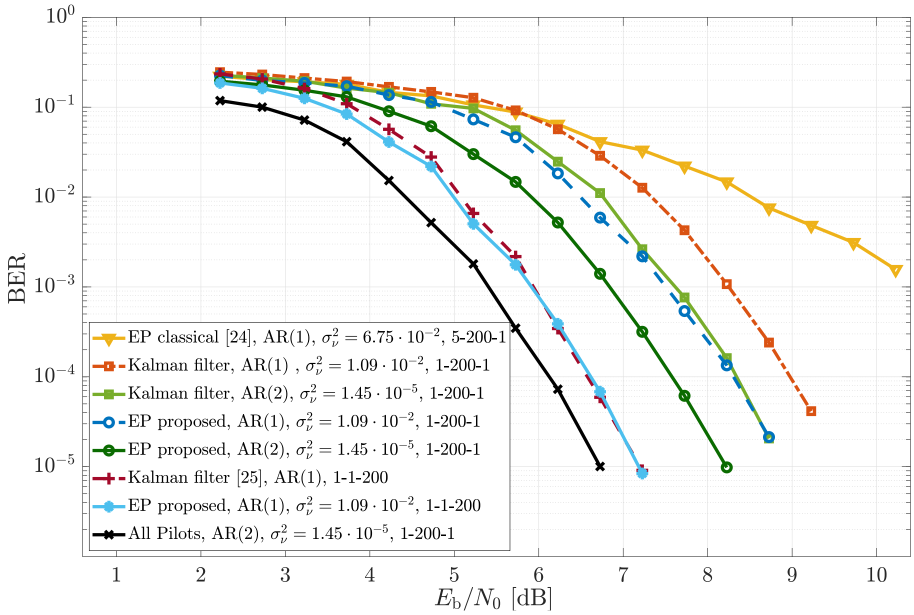


Figure 2.3: Performance for the 1/20 pilots distribution with $f_D T = 10^{-2}$.

properly. This can be explained by the fact that the YW equations are studied for the approximation of the autocorrelation function in (2.2) with $R_h(m)$ by matching the first N samples and computing σ_ν^2 from the constraint

$$R_h(0) = R_g(0)\,. \tag{2.27}$$

Satisfying all the above conditions the resulting autocorrelation function does not follow accurately the true one $R_g(m)$. Therefore, we can either optimize the $\boldsymbol{\rho}$ values or the σ_ν^2 one. Consider, for instance, the AR(1) model. In Figure 2.4 we compare the exact autocorrelation function $R_g(m)$ with the approximation performed matching the first sample and computing σ_ν^2 from (2.27) (as suggested by the YW equations) with the one obtained optimizing σ_ν^2 or, equivalently ρ through simulations. As can be noticed, the application of the YW equations would lead to a much larger duration (i.e., the memory of the process) than the true one.

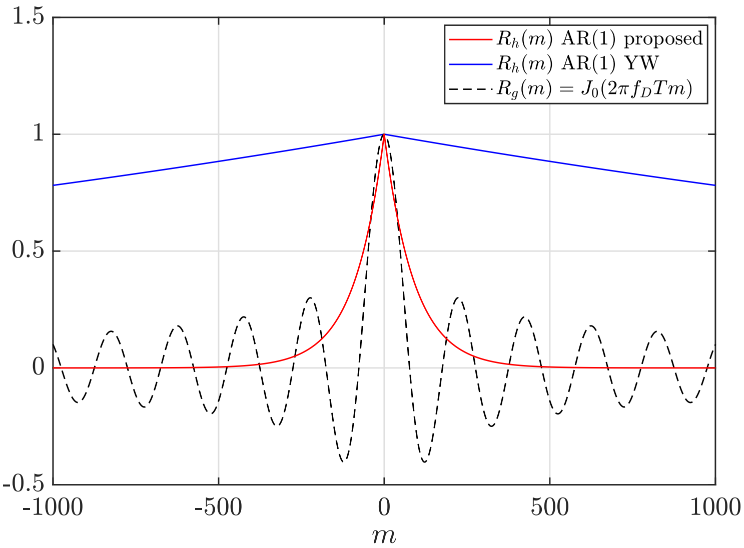


Figure 2.4: Comparison among the AR(1) model autocorrelation function deriving from the solution of YW equations, the proposed approximation and the exact distribution.

In Figure 2.3, we compared the performance of the proposed algorithm with both the Kalman filter and the "batch-EP" of [24], labeled with "classical EP", when the normalized Doppler bandwidth of the fading process $f_D T$ was set to 10^{-2} and one pilot symbol was inserted in every block of 20 transmitted symbols. Firstly, we analyzed the practical case of separate detection and decoding.[4] Figure 2.3 shows that the proposed EP algorithm can achieve a performance gain of approximately 0.9 dB with respect to the Kalman filter for both the AR(1) and the AR(2) models. Moreover, a significant gain is observed comparing the proposed technique with the classical EP of [24]. Regarding the latter algorithm, the authors proposed an EP smoothing approach where an iterative refinement of the observation approximations is foreseen. Consistently with that used in [24], a maximum number (N_D) of 5 inner detector iterations

[4]In order to indicate the number of inner detector, inner decoder and turbo iterations, the notation N_D-N_C-N_T is used.

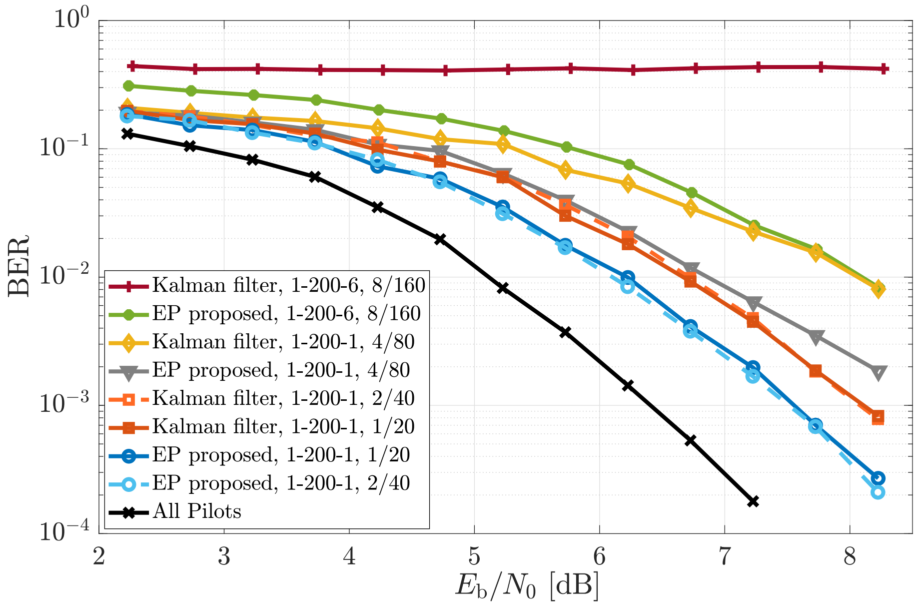


Figure 2.5: Performance for different pilots distribution with $f_D T = 5 \cdot 10^{-3}$ and $\sigma_\nu^2 = 8 \cdot 10^{-3}$.

was allowed, and it resulted to be sufficient for achieving convergence. For this algorithm, the curve shown in Figure 2.3 was obtained using the AR(1) model since the AR(2) model did not yield further improvement.

In Figure 2.3, the BER curves related to the joint iterative detection and decoding, where a maximum of 200 turbo iterations is allowed, are also reported with reference to the AR(1) fading model. It is possible to notice that the curves related to the proposed EP algorithm and the Kalman filter (optimized according to [25]) are approximately overlapped. In fact, when a sufficient number of turbo iterations is performed, the decoder provides symbol probabilities approaching the ones passed in correspondence with a known symbol. Therefore, the observation message is composed by a single Gaussian and the operations performed by the two detectors coincide.

In Figure 2.5, keeping fixed the effective information rate, different pilot distributions were considered in order to show the higher robustness of EP with respect to the Kalman filter against pilot symbols arrangements in sequences separated by longer blocks of code symbols. The normalized Doppler bandwidth was set to $5 \cdot 10^{-3}$. The AR(1) fading model was adopted for the sake of simplicity, however, similar conclusions can be drawn for AR(2). It is possible to notice that both the proposed algorithm and the Kalman filter are practically insensitive to a change in the pilots blocks distance from 20 to 40 symbols. Differently from the EP-based algorithm, increasing the gap to 80 symbols leads to a significant loss for the Kalman filtering approach. Finally, considering a pilots distance of 160 code symbols, the EP detector is still able to provide reasonable fading estimates and, with the aid of 6 turbo iterations, it loses approximately 2 dB from the case where the 1/20 pilots distribution is adopted. On the contrary, in the same conditions, the Kalman filter does not work because of its lower robustness against longer pilots spacing.

The proposed solution in this last situation adopts the observation message projection strategy presented in Section 2.4.2. In order to show the advantages of this approach, we compared the algorithm performance with and without the handling of improper distributions in Figure 2.6. It can be noted that a considerable gain can be achieved by blocking the improper distributions propagation. This is due to the fact that the adopted pilots configuration makes the algorithm operating conditions harder. Therefore, the estimate obtained by the *prior* is not often in accordance with any of the observation message modes, leading to a rather wide approximating marginal distribution. Negative variances are thus typically generated after the observation message update, and allowing their propagation and combination with those from previous iterations leads to poor performance. On the other hand, by considering more favorable conditions, the performance gap becomes less relevant.

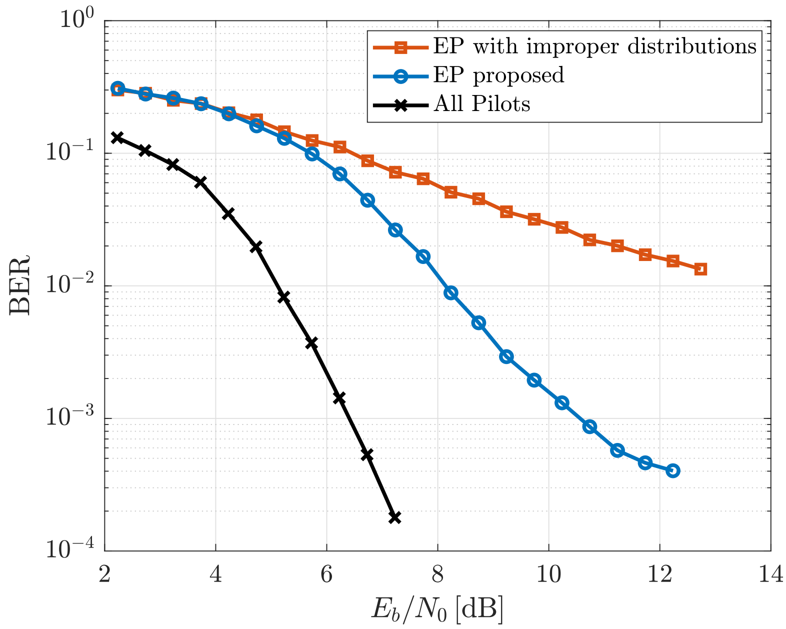


Figure 2.6: Performance comparison of the EP algorithm with and without the propagation of improper distributions. The pilots distribution is the 8/160 one, and the normalized Doppler bandwidth $f_d T = 5 \cdot 10^{-3}$. The selected σ_ν^2 optimizing the performance is set to $8 \cdot 10^{-3}$. The simulated iterations are 1-200-7.

2.7 Conclusions

In this chapter, we proposed a new solution for signal detection over channels with flat-fading through EP. The presented method allows to solve the numerical instabilities related to the traditional procedure provided in the literature. We showed through numerical results the advantages in terms of BER that can be achieved in the separate detection and decoding case with respect to the classical Kalman filter approach and we demonstrated the higher robustness of this algorithm against more concentrated pilots distributions.

The advantages of this framework offer a variety of possibilities for new research directions, such as its application to the multiple-input multiple-output channel model and the study of more severe fading conditions, for instance frequency selectivity. Moreover, we could extend our analysis to other kinds of divergence measures for the involved approximations.

Chapter 3

Phase Noise Detection via Expectation Propagation

3.1 Introduction

Detection algorithms for channels affected by a time-varying PN have received a lot of attention in the literature considering either linear or continuous-phase modulations and different scenarios (see, e.g., [35–41] and references therein). This is because in many communication links PN must be considered one of the major impairments. Examples are represented by scenarios employing a high carrier frequency, such as (coherent) optical communications, communications from geostationary satellites, etc.

One of the most effective algorithms is that proposed in [35]. It is derived using the framework based on FGs and the SPA. In a scenario like the one at hand, where continuous RVs (the channel phase) appear in the FG, a common approach to implement the SPA is to resort to the use of canonical distributions [42] as tackled in Chapter 2. In particular, in [35], the messages representing the a-posteriori pdfs of the channel phase are represented through Tikhonov distributions, which can be described by a single complex parameter. Although suboptimal, the resulting algorithm exhibits an excellent trade-off between performance and complexity. The suboptimality is related to the joint presence of discrete (the code symbols) and continuous RVs in the graph, which brings up mixture pdfs with exponential proliferation, approximated in [35] with unimodal distributions. The presence of distributed pilot symbols is thus required to make the algorithm bootstrap. In fact, the algorithm in [35] based on the Tikhonov parametrization is very sensitive to the positioning of pilot symbols, so that increasing the pilot spacing causes a performance degradation. In particular, when pilot symbols are concentrated at both ends of the codeword, as a preamble/postamble, or, more in general, when a maximum distance between consecutive pilot blocks is exceeded, it is known to fail [35, 43]. This occurs since this kind of algorithm cannot achieve any effective phase detection without the aid of extrinsic information, provided by the decoding part of the FG during turbo iterations, or without distributed pilots (with sufficiently close spacing). Therefore, other alternative solutions have to be analyzed to this purpose.

As an extension of [35], the results of Raphaeli and coworkers [40, 44] are obtained by letting mixture messages propagate one step further into the FG; their exponential proliferation being limited by an appropriate pruning of the mixture components, performed at the level of the Markov chain governing PN. This way, the phase uncertainty inherent in the observation of channel output is free to interact with the provisional estimation of previous phase samples (or with the following ones, in backward block processing). Such a mixture message reduction approach, however, is characterized by a considerable complexity.

The same objective of letting channel observations interact with provisional estimates of adjacent phase samples can be achieved by the EP algorithm [14, 15, 45, 46], where similar mixture reduction techniques are applied to the marginal distributions of the variables of interest, rather than to individual messages as discussed in Section 1.3. Despite the projection of messages or marginals do have some features in common, and can even reduce to the same algorithm in some cases [34, 47, 48], there is a profound conceptual difference between them. In fact, only in the latter case, a message coming from a channel observation is merged, i.e., multiplied, with the (provisional) *prior belief* on the destination variable that comes from the rest of the FG. This is the key to the potential success of EP in many similar applications, including, e.g., transmission over fading channels [7, 24] as discussed in Chapter 2, multiple-input multiple-output detection schemes [49] and the design of low complexity receivers within the orthogonal time-frequency space modulation [50, 51].

The primary technical contribution of the work presented in this chapter is the effective application of the EP framework to the detection problem in PN channels, for the first time. Other attempts made in the past, published online or in conference venues [43, 52], had a limited range of application and were only partially successful. Their most common limits, which are here overcome, are: (i) the need to corroborate the phase estimation process with "turbo" iterations, between detector and decoder and/or (ii) the puncturing of data with closely spaced pilot symbols. Our first approach to solve these two limits [8] proved to be effective only for binary modulations and needing an ad-hoc post-processing of messages. In particular, turbo iterations do not represent a practical solution, since a separate (and sequential) detection of channel parameters and decoding is desirable in the design of digital receivers. On the contrary, the challenge of making EP work with separate phase detection and decoding has not been solved. In fact, previous analyses have shown that, despite its ability to bootstrap the phase estimation process from a proper block of preamble/postamble pilots, the native EP algorithm is not able to effectively refine the phase estimates beyond the first phase-detecting iteration [8]; and this is true even when pilot blocks are close enough, a situation where the algorithm in [35] shows a better performance. Up to now, the EP framework has been successfully proposed for many applications, though not specifically for PN, despite it is an historical impairment in radio transmissions. Hence, the work presented in this chapter discusses the solution of an old problem through the first successful application of an existing approach (EP). Nonetheless, EP is not a standard technique, readily available for a "blind" application. Rather, it is merely a probabilistic inference framework where changing the analyzed scenario requires the full development of a new algorithm, just leveraging the core concept behind EP, which is the marginal distribution approximation through moment matching. In particular, relying on this framework involves:

- the proper selection of a suitable approximating family;
- the derivation of the associated moment matching equation(s) (KL projection);
- the approximation of non-invertible functions or non-closed form solutions;
- the analytical calculation of all the subsequent messages in parametric message passing.

The additional contribution of this work concerns identifying EP weaknesses and the underlying reasons. To enhance and overcome the limited performance of the resulting algorithm, we propose novel solutions that yield a flexible algorithm capable of adapting to various pilot positioning and iterations scheme (whether separate or joint detection and decoding) always outperforming the state-of-the-art solutions.

The rest of this chapter is organized as follows. In Section 3.2, we introduce the system model and the FG representation of the joint *posterior* pdf of the information symbols and channel parameters. Section 3.3 briefly describes the detection algorithms that are used for comparison purposes. In Section 3.4, we derive an EP algorithmic solution suited for the system model in Section. 3.2. Amendments and modifications of the derived algorithm are discussed in Section 3.5. The message scheduling is outlined in Section 3.6. A complexity comparison among the analyzed algorithms is carried out in Section 3.7. Finally, in Section 3.8 we show and discuss the numerical results and, in Section 3.9, we draw some concluding remarks.

3.2 System Model and Related Factor Graph

In the system we wish to investigate, a sequence of K complex coded symbols $\boldsymbol{c} = [c_0, c_1, \cdots, c_{K-1}]$, modulated over an M-ary constellation (with alphabet $\mathcal{A} = \{a_1, \ldots, a_M\} \subset \mathbb{C}$), is transmitted over an AWGN channel also affected by Wiener PN, so that the received samples are

$$r_k = c_k \mathrm{e}^{j\theta_k} + n_k \,, \tag{3.1}$$

where $\{n_k\}$ is a sequence of independent (complex and circularly symmetric) Gaussian noise samples, i.e., $n_k \sim \mathcal{N}_{\mathbb{C}}(0, 2\sigma^2)$, with zero mean and given variance per component σ^2, while the PN sequence $\boldsymbol{\theta} = [\theta_0, \theta_1, \cdots, \theta_{K-1}]$ follows the Wiener model, where each sample

$$\theta_k = \theta_{k-1} + \Delta_k \tag{3.2}$$

results from the previous one plus a zero-mean (real) Gaussian increment $\Delta_k \sim \mathcal{N}(0, \sigma_\Delta^2)$ whose variance σ_Δ^2 dictates the severity of PN. Even though the receiver design is based on the Wiener model (3.2), which is commonly used to describe PN of a free-running oscillator, the proposed algorithm proves to be effective also when PN samples are generated according to other models, such as the one representing the PN of a Phase-Locked Loop (PLL) oscillator [53].

The FG representing the joint distribution of the latent variables $\boldsymbol{\theta}$ and $\boldsymbol{c}$, conditioned on the value of the observed variables $\boldsymbol{r} = [r_0, r_1, \cdots, r_{K-1}]$ is reported in Figure 3.1. The subscripts of messages identify their direction in the FG (*u*p, *d*own, *f*orward, *b*ackward), whereas the subscript of VNs and FNs coincides with a time index.

The joint posterior distribution of all latent system variables is

$$\begin{aligned} P(\boldsymbol{c}, \boldsymbol{\theta} \,|\, \boldsymbol{r}) &\propto p(\boldsymbol{r} \,|\, \boldsymbol{c}, \boldsymbol{\theta})\, P(\boldsymbol{c}, \boldsymbol{\theta}) && (3.3)\\ &\propto p(\boldsymbol{r} \,|\, \boldsymbol{c}, \boldsymbol{\theta})\, P(\boldsymbol{c})\, p(\boldsymbol{\theta}) && (3.4)\\ &\propto \chi(\boldsymbol{c})\, p(\theta_0) \prod_{k=1}^{K-1} p(\theta_k \,|\, \theta_{k-1}) \prod_{k=0}^{K-1} f_k(c_k, \theta_k) \,, && (3.5) \end{aligned}$$

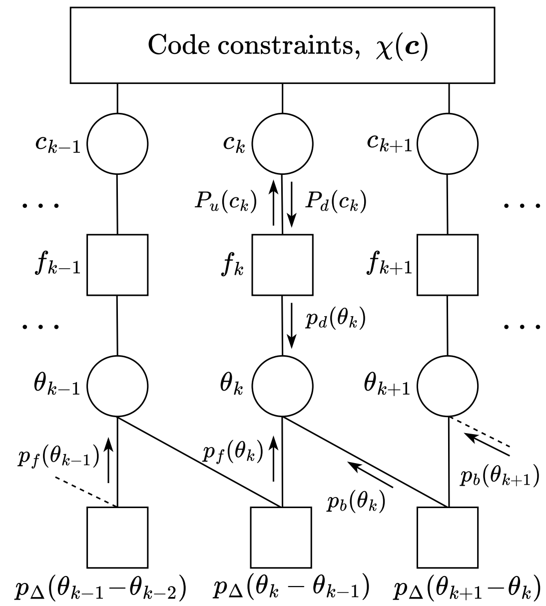


Figure 3.1: FG representing the joint probability distribution of PN samples and coded symbols.

where $\propto$ is the proportionality symbol and $\chi(\boldsymbol{c})$ is an indicator function representing the code constraints. We assume that all the allowed codewords in the employed codebook are equally likely and, in (3.4), we exploited the inherent *a priori* independence of coded symbols and PN sequence.

The FN f_k implements the observation of the received samples in (3.1), hence

$$\begin{aligned} f_k(c_k, \theta_k) &= p(r_k \mid c_k, \theta_k) = g_{\mathbb{C}}(r_k, c_k \mathrm{e}^{j\theta_k}, 2\sigma^2) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}\left|r_k - c_k \mathrm{e}^{j\theta_k}\right|^2\right) \end{aligned} \tag{3.6}$$

is a complex Gaussian distribution when seen as a function of the received sample r_k.

Besides f_k, the only other FNs in the FG of Figure 3.1 (if we disregard the check nodes included in the code sub-graph) are the ones related to the conditional pdfs in (3.5) that implement the Markov chain governing the Wiener PN,

$$p(\theta_k \mid \theta_{k-1}) \triangleq p_\Delta(\theta_k - \theta_{k-1}) = g(\theta_k, \theta_{k-1}, \sigma_\Delta^2)\,, \tag{3.7}$$

which are Gaussian too, as per (3.2). The distribution in (3.7) provides an estimates of θ_k exploiting the correlation among consecutive PN samples.

Owing to the fact that the additive sources of randomness in (3.1) and (3.2) are Gaussian, the FG in Figure 3.1 seems to represent a Gaussian belief network, where all messages are Gaussian [54] and the message passing procedure resembles the simple operations of a Kalman smoother. This is clearly not true, for two reasons. First, the problem at hand entails a "mixed model", where continuous ($\boldsymbol{\theta}$) and discrete ($\boldsymbol{c}$) variables coexist, which produces probabilistic mixtures in the FG, specifically arising from the messages $p_d(\theta_k)$ which are linear combinations of simpler component distributions. In addition, (3.6) is a Gaussian pdf only if seen as a function of r_k whereas the FNs f_k send downward messages $p_d(\theta_k)$ to the VNs θ_k, hence (3.6) must be seen as a function of θ_k. By manipulating (3.6) as follows, we obtain

$$\begin{aligned}
f_k(c_k, \theta_k) &\propto \exp\left(-\frac{1}{2\sigma^2}\left(|r_k|^2 - 2\Re\left\{r_k c_k^* \mathrm{e}^{-j\theta_k}\right\} + |c_k|^2\right)\right) && (3.8)\\
&\propto \exp\left(-\frac{1}{2\sigma^2}\left(-2\Re\left\{r_k c_k^* \mathrm{e}^{-j\theta_k}\right\} + |c_k|^2\right)\right) && (3.9)\\
&= \exp\left(-\frac{|c_k|^2}{2\sigma^2} + \frac{1}{\sigma^2}\Re\left\{r_k c_k^* \mathrm{e}^{-j\theta_k}\right\}\right) && (3.10)\\
&= \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right)\exp\left(\Re\left[\frac{r_k c_k^*}{\sigma^2}\mathrm{e}^{-j\theta_k}\right]\right) && (3.11)\\
&= \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) 2\pi I_0(|z_k|)\frac{1}{2\pi I_0(|z_k|)}\exp\left(\Re\left[z_k \mathrm{e}^{-j\theta_k}\right]\right) && (3.12)\\
&= \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) 2\pi I_0(|z_k|) t(\theta_k, z_k)\,, && (3.13)
\end{aligned}$$

where

- in (3.9) we neglected the term depending on r_k only since it is a constant with respect to both θ_k and c_k. Being the FN f_k connected to both the variables c_k and θ_k, each term depending on, at least, one of them cannot be discarded;

- in (3.12) we defined $z_k = r_k c_k^* / \sigma^2$;
- in (3.13) we denoted with $t(\theta_k, z_k)$ the general Tikhonov distribution with argument θ_k and complex parameter z_k, i.e.,

$$t(\theta_k; z_k) = \frac{1}{2\pi I_0(|z_k|)} \exp\left(\Re\left[z_k \mathrm{e}^{-j\theta_k}\right]\right) . \tag{3.14}$$

Therefore, in (3.13), it is possible to notice that the obtained expression for the FN f_k is proportional to a Tikhonov distribution [35] when seen as a function of θ_k.

In directional statistics [55], the phase $\angle z_k$ corresponds to the *circular mean* of the circular RV θ_k in (3.14), while the magnitude $|z_k|$ is a measure of its precision, i.e., of the inverse of the variance $\sigma^2_{\theta_k} = 1 - \frac{I_1(|z_k|)}{I_0(|z_k|)} \in [0; 1]$, where $I_p(x)$ is the modified Bessel function of the first kind of order p.[1]

The Bayesian inference problem for the estimation of symbols and channel parameters cannot be solved by the plain SPA since the message

$$p_d(\theta_k) = \sum_{m=0}^{M-1} P_d(c_k = a_m) f_k(c_k = a_m, \theta_k) \propto \sum_{m=0}^{M-1} \alpha_k^m t(\theta_k; z_k^m) , \tag{3.15}$$

where $z_k^m = r_k a_m^* / \sigma^2$, is a Tikhonov mixture. Its coefficients α_k^m, defined as

$$\alpha_k^m = P_d(c_k = a_m) \exp\left(-\frac{|a_m|^2}{2\sigma^2}\right) 2\pi I_0(|z_k^m|) , \tag{3.16}$$

depend on the extrinsic information on each symbol (i.e., on the values of its pmf for each of the possible M symbols a_m), as provided by the soft-input soft-output decoding part of the FG (top of Figure 3.1), as well as on the magnitude of the symbols. In the SPA, the mixture messages (3.15) would propagate through the bottom half of the FG and eventually proliferate to produce

[1] More in general, the p-th trigonometric moment of the Tikhonov variable θ_k is $\mathbb{E}_t[\exp(jp\theta_k)] = \exp(jp\angle z_k) \frac{I_p(|z_k|)}{I_0(|z_k|)}$, so that the phase of the first trigonometric moment is the circular mean.

untractable mixtures with an exponentially increasing number of components. Approximate inference is thus demanded, which can be performed in a variety of ways.

3.3 Message passing algorithms taken from the literature

In the following, we will briefly outline the algorithms that we tested for comparison purposes in Section 3.8.

3.3.1 Discretization of Channel Parameters: discretized-phase BCJR

The most straightforward approach to approximate message passing is to discretize the distribution of θ_k, with a given number of samples N_θ, so that all the latent system variables appear to be discrete and modelled by their pmf. This produces an algorithm called *discretized-phase BCJR (dp-BCJR)* in [38], since the forward-backward message passing procedure, along the Markov chain in the bottom half of the FG, is the same as that of the celebrated Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm. This approach has a large accuracy being able to follow any kind of distribution, thus we shall implement this algorithm, denoted as *dp-BCJR* in the simulation results, as a performance benchmark. In fact, when N_θ is sufficiently large the performance loss is negligible: it gets close to the performance of the ideal SPA. However, it leads to an impractical solution because, as it shown in Section 3.7, it is characterized by extremely high complexity. Unlike the other message passing algorithms we will focus on, it does not constrain the approximating distribution to belong to a certain family of functions.

3.3.2 Transparent Propagation algorithms

A totally different approach relies on projecting messages and/or marginals onto a selected family of approximating distributions, usually in the exponential form (so that the family is closed under the multiplication operation),

$$q(x) = \exp\left(\sum_i \eta_i g_i(x)\right), \tag{3.17}$$

where η_i are called natural parameters and the functions $g_i(x)$ are the features of the family. This way, only the natural parameters η_i are updated, for each distribution, resulting in a parametric message passing procedure. In [35], the Tikhonov approximating family is selected quite naturally, since it is known to be the marginal distribution of θ_k, conditioned on the values taken by the corresponding symbol and by the corresponding channel output, i.e., $p(\theta_k \mid c_k, r_k) \propto f_k(c_k, \theta_k)$, as per (3.6). The Gaussian parameterization has also been analyzed in [35]. However, simulation results show that using the Tikhonov parameterization improves the performance. In fact, being a close approximation to the *wrapped* Gaussian defined in the interval $[-\pi, \pi]$, the Tikhonov distribution better represents PN samples following their statistics more accurately. Curiously, the way in which message $p_d(\theta_k)$ has been projected onto a Tikhonov pdf in [35] relies on the Gaussian expression (3.6) of the FN, so that (3.15) is approximated by the Gaussian pdf with minimum KL divergence from the mixture, which is further interpreted as a Tikhonov message towards θ_k.

A more natural solution would have been to project the Tikhonov mixture (3.15) onto a Tikhonov pdf with minimum KL divergence, which is achieved (for this as well as for any other exponential approximating family) by matching the expectations of the features [15]. We refer to this approach, along with the Tikhonov parameterization of [35], as Transparent Propagation (TP) algorithms (the reason will be clear later). The features of a Tikhonov pdf like (3.14) are $\cos(\theta_k)$ and $\sin(\theta_k)$, associated with the natural parameters $\Re[z_k]$ and $\Im[z_k]$ respectively, so that their expectations can be jointly computed as the $\Re/\Im$ parts of the first trigonometric moment $\mathbb{E}[\exp(j\theta_k)]$. Exploiting known results [55] and the linearity of expectation, the mixture in (3.15) is approximated by the Tikhonov pdf $p_d^{TP}(\theta_k) = t(\theta_k; z_k^{TP})$, which achieves the minimum KL divergence when z_k^{TP} obeys the following *moment matching* equation

$$\frac{I_1(|z_k^{TP}|)}{I_0(|z_k^{TP}|)} \mathrm{e}^{j\angle z_k^{TP}} = \sum_{m=0}^{M-1} \overline{\alpha}_k^m \frac{I_1(|z_k^m|)}{I_0(|z_k^m|)} \mathrm{e}^{j\angle z_k^m}\,, \tag{3.18}$$

where $\overline{\alpha}_k^m$ stands for the normalized version of the coefficients in (3.16), i.e.,

$$\sum_{m=0}^{M-1} \overline{\alpha}_k^m = 1\,. \tag{3.19}$$

A detailed derivation of (3.18) can be found in Appendix D. The complex equation (3.18) yields the circular mean $\angle z_k^{TP}$ as well as the variance of PN, related to the Bessel Ratio (BR) of the magnitude $|z_k^{TP}|$ [44]. Despite the different analytical approach, the two ways of projecting the mixture message $p_d(\theta_k)$ described above, i.e., that of [35] and the one in (3.18), yield extremely similar results, as verified by numerical simulation in all PN scenarios that we analyzed. For this reason, their equivalent performance is collectively shown under the *TP* label in the simulation results in Section 3.8. We consider this approach in the performance analysis because it is the best low-complexity technique available in the literature.

3.4 The Expectation Propagation Framework

As the TP algorithm, EP [14] propagates through the FG just the natural parameters characterizing the selected approximating family. However, unlike TP, in the EP algorithmic framework [14], it is the entire marginal of each latent variable that is approximated/projected instead of the single message; in our case,

$$p(\theta_k) = p_d(\theta_k)p_f(\theta_k)p_b(\theta_k) = p_d(\theta_k)p_u(\theta_k)\,, \tag{3.20}$$

of which the message $p_d(\theta_k)$ only represents one factor. In (3.20), message $p_u(\theta_k) \triangleq p_f(\theta_k)p_b(\theta_k)$ is defined as the product of forward and backward messages, and is sent upwards, opposite to message $p_d(\theta_k)$, although not reported in the FG in Figure 3.1. Indeed, the conceptual meaning of the two factors of the marginal in (3.20) is the following: $p_d(\theta_k)$ is an *observation message*, which carries information on θ_k after the channel observation entailed in the FN (3.13), while $p_u(\theta_k)$ plays the role of a temporary *prior belief* for the PN variable θ_k, as provided by the rest of the FG. Denoting the projection operation as[2]

$$\mathrm{proj}[p(\cdot)] = \arg \min_{q(\cdot)\in\mathcal{F}} \mathrm{KL}\left[p(\cdot) \parallel q(\cdot)\right] , \tag{3.21}$$

where $\mathcal{F}$ is the approximating family, the approximating message

$$p_d^{EP}(\theta_k) = \frac{\mathrm{proj}[p_d(\theta_k)p_u(\theta_k)]}{p_u(\theta_k)} \tag{3.22}$$

is computed and sent in place of the mixture $p_d(\theta_k)$. In this way, the argument of the minimization in (3.21) is the divergence among variable distributions, rather than messages. Consequently, the *observation* message approximation is also influenced by the rest of the network through $p_u(c_k)$. After the observation $f_k(c_k, \theta_k)$, the estimated marginal for the PN variable θ_k is thus updated in a different way, compared to the plain SPA rule (3.20), and expressed as

$$p^{EP}(\theta_k) = p_d^{EP}(\theta_k)p_u(\theta_k)\,, \tag{3.23}$$

which is equal to the numerator of (3.22) and thus belongs, by construction, to the approximating family (Tikhonov, in our case).

The projection operation in (3.22) still amounts to a moment matching operation like (3.18), despite a marginal is projected, in EP, rather than a message, as in TP. Of course, in the case of EP, the temporary prior $p_f(\theta_k)p_b(\theta_k)$ must be accounted for in the right hand side of (3.18), by simply adding the (complex) parameters $z_{k,f}$ and $z_{k,b}$ of the forward and backward messages to

[2] We denote with $\mathrm{KL}[p(\cdot) \,||\, q(\cdot)]$ the KL divergence of an approximated distribution $q(\cdot)$ from the exact distribution $p(\cdot)$.

those of each mixture component, i.e., to z_k^m (hence the coefficients α_k^m must be calculated accordingly). Therefore, from a conceptual standpoint, the mixture to be projected is now the result of the interaction between the prior belief, deriving from the Markov chain at the bottom half of the FG in Figure 3.1, and the current channel observation, represented by message $p_d(\theta_k)$. The expression for the marginal mixture parameters thus becomes

$$z_{k,MIX}^m = z_{k,u} + z_k^m \,, \tag{3.24}$$

where we denoted with $z_{k,u} = z_{k,f} + z_{k,b}$ the temporary prior parameter.

The general approach to approximate message passing can be either a projection of the marginal pdf of each VN (onto the selected approximating family), as prescribed by EP, or a projection of individual mixture messages onto the same family, as described above for TP. The TP algorithm can also be seen as a special case of EP. In fact, when the EP algorithm generates an informationless prior (i.e., the product of messages $p_f(\theta_k)p_b(\theta_k)$ in (3.22) represents a uniform distribution), it is unable to modify the shape of the marginal distribution of the VN. In such cases, we can *transparently* shift it out of the projection operator. This way, it simplifies with the denominator and (3.22) reduces to the simple projection of message $p_d(\theta_k)$. This is the reason for which we denote the message projection approach as TP [47]. It can be demonstrated that in some specific problems, the TP and EP approaches lead to the same solution [48], although in general they differ remarkably.

No matter if one of the TP algorithms or if EP is adopted, the forward and backward messages $p_f(\theta_k)$ and $p_b(\theta_k)$ are assumed to belong to the Tikhonov family too, despite the FNs in the Markov chain of the FG, e.g., $p(\theta_{k+1} \mid \theta_k)$, introduce a convolution between the Tikhonov message obtained multiplying $p_d^{TP/EP}(\theta_k)$ with $p_f(\theta_k)$ and a Gaussian like (3.7), for the computation of $p_f(\theta_{k+1})$. As it is shown in the Appendix of [35], if the PN variance σ_Δ^2 is not exaggeratedly large then $p_f(\theta_{k+1})$ is very well approximated by a Tikhonov pdf whose parameter $|z_{f,k+1}|$ is less than that of the incoming message, i.e., $\left|z_{f,k} + z_k^{TP/EP}\right|$, while the circular mean is the same (see (36)-(38) in [35], for details). This corresponds conceptually to the fact that the Wiener PN update in (3.2) does not bias the temporary estimate for θ_{k+1} but decreases its precision, according to its Gaussian variance. A similar conclusion holds for the backward messages too.

3.5 The EP Modification

The EP algorithm has been already applied to transmission over Wiener PN channels in [43] (and later in [52]), in the absence of distributed pilot symbols, achieving a good performance when turbo iterations However, existing literature does not demonstrate successful operation of EP with separate phase detection and decoding.

As it is known, the performance of TP algorithms, such as [35], significantly deteriorates by increasing the pilot spacing, leading to complete failure when they are concentrated at both ends of the codeword. In contrast, in these scenarios, EP represents a viable alternative, due to its ability to self-sustain the process of phase estimation across a long block of payload coded symbols.

However, the native EP algorithm is not able to effectively refine the phase estimates beyond the first phase-detecting iteration [8], resulting in unsatisfying performance both with distributed pilots (where it is outperformed by TP algorithms) and with more concentrated pilots (where, unable to benefit from inner detector iterations, its performance is limited as shown in Section 3.8).

Through a detailed numerical analysis of EP, we found that its failure is confined to some critical data packets that, due to a long sequence of noisy observations, bring the sequence of precision values $|z_{k,f/b}^{EP}|$ to a collapse, while propagating forward or backward messages, $p_f(\theta_k)$ and $p_b(\theta_k)$. Once a critical lower threshold is exceeded, the precision approaches zero and is rarely able to recover, even if the channel in (3.1) outputs reliable observations, i.e., samples r_k with little noise. In order to make EP overcome these limiting situations, we introduce the modifications described hereafter, which improve its robustness even in very challenging conditions, as further verified in Section 3.8.

3.5.1 Bessel Ratio Approximation

A first countermeasure that we introduce in the classical EP algorithm concerns the way in which the BR $I_1(x)/I_0(x)$ is inverted. Since $|z_{k,f/b}^{EP}|$ are calculated after the EP message (3.22), which is in turn computed by applying the moment matching (3.18), we verified that the occasional collapse of their value is largely due to a numerical instability entailed in the approximation of the BR in (3.18) whose inverse, not available in closed form, is necessary to obtain $|z_k^{EP}|$. In fact, despite the approximation, referred to as (b_1),

$$I_1(|z|)/I_0(|z|) \simeq \exp(-0.5/|z|)\,, \tag{3.25}$$

which is commonly adopted in the literature [43, 52], is very close to the exact function when $|z| \gg 0$, it is not equally accurate for lower values of $|z|$, e.g, in the range $[0, 5]$ as shown in Figure 3.2. When this traditional approximation (3.25) is adopted, the moment matching equation (3.18) becomes

$$\mathrm{e}^{-0.5/|z_k^{EP}|+j\angle z_k^{EP}} = \sum_{m=0}^{M-1} \overline{\alpha}_{k,MIX}^{m} \mathrm{e}^{-0.5/|z_{k,MIX}^{m}|+j\angle z_{k,MIX}^{m}}\,, \tag{3.26}$$

where, differently from [43], we adopted BR approximation also in the right-hand side. In fact, since the exponential approximation is not perfectly overlapped with the true BR, as seen in Figure 3.2, using the same *mapping* function in both sides of the equation yields better results. In [8], we followed a different approach, by replacing the BR straightforwardly with its argument $|z|$, so that the resulting moment matching equation,

$$z_k^{EP} = \sum_{m=0}^{M-1} \overline{\alpha}_{k,MIX}^{m} z_{k,MIX}^{m}\,, \tag{3.27}$$

gives the Tikhonov parameter for the marginal of θ_k as a simple linear combination of the mixture components' parameters. In [8], our aim was not to accurately approximate the BR but rather to avoid the compression of the $|z_k^{EP}|$ values that, as stated, seems to be inherent to the native EP message passing, due to the shape of the BR (and of its inverse), and tends to depress the level of confidence in the transmitted messages. Although this strategy showed satisfactory performance, in [8], the resulting algorithm requires an additional monitoring of the precision values $|z_{k,f/b}|$, along the forward and backward propagation of $p_{f/b}(\theta_k)$, in order to identify sudden drops of the precision values, which would lead inevitably to errors in phase detection. Such a monitoring is implemented in [8] as a *post-processing* of the computed precision values, followed by the rejection of messages whose parameters are critical. This allows the algorithm to withstand challenging scenarios, e.g, a gap of 4000 symbols among preamble and postamble without the aid of turbo iterations.

In this work, we follow a different approach focusing on the accuracy of the inverse BR approximation. In fact, because of the very steep shape of the BR in this range, even approximations that are apparently quite similar may produce very different results. Besides the simple and apparently satisfactory approximation in (3.25), we analyzed a number of piecewise approximations in which the range of precision values $|z|$ (hence for the BR values) is split in two intervals that are conceptually related to the initial, steeply increasing, part of the BR and to the saturating part that follows. A Taylor series expansion (b_2), i.e., polynomials of increasing degree, were considered, as well as exponential profiles like (3.25), that are particularly suited for the saturating part of the BR. The simplest instance, among these, is a piecewise linear function where the saturating (rightmost) part of the BR is approximated by a constant, equal to its asymptotic value 1, so that the overall approximation has the shape of a *soft limiter* (b_3). The free coefficients and parameters of all the approximating polynomials that we considered were found by best matching the average values of the derivatives of the actual BR function on the considered range for $|z|$.

As will be seen, a good compromise between simplicity and performance, in terms of achievable results, is a piecewise function composed by a parabola, for low values of $|z|$, followed by an exponential profile in the saturating range. We will refer to this approximation as (b_4). Therefore, the sought magnitude of the Tikhonov parameter is $|z| = f^{-1}(BR)$, where $f^{-1}(\cdot)$ denotes the following inverse BR approximation,

$$f^{-1}(BR) = \begin{cases} 2.55 - 3.02 \cdot \sqrt{0.71 - BR}, & BR \leq 0.59 \\ -0.5/\log(BR) + 0.55, & \text{elsewhere} \end{cases}, \tag{3.28}$$

where the numerical parameters are found numerically through the curve-fitting method as stated above. In particular, concerning the parabolic piece, the approximating polynomial is

$$f(|z|) = -0.11\,|z|^2 + 0.56\,|z|\,. \tag{3.29}$$

We set $f(|z|) = BR$ and manipulate (3.29) as

$$|z|^2 - \frac{0.56}{0.11}\,|z| = -\frac{BR}{0.11} \tag{3.30}$$

$$\left(|z| - \frac{0.56}{2 \cdot 0.11}\right)^2 = \left(\frac{0.56}{2 \cdot 0.11}\right)^2 - \frac{BR}{0.11}\,. \tag{3.31}$$

Taking the square root of both members, we obtain

$$|z| = 2.55 - \sqrt{6.48 - 9.09 \cdot BR} \tag{3.32}$$

$$= 2.55 - \sqrt{9.09 \cdot (0.71 - BR)} \tag{3.33}$$

and, therefore, the inverse function

$$f^{-1}(BR) = |z| = 2.55 - 3.02 \cdot \sqrt{0.71 - BR}\,. \tag{3.34}$$

The exponential profile that we choose for the approximation of the saturating region is

$$BR = f(|z|) = \exp\left\{-\frac{0.5}{|z| - 0.55}\right\}\,. \tag{3.35}$$

Directly inverting this function we obtain the 2^{nd} range approximation of the inverse BR:

$$\log(BR) = -\frac{0.5}{|z| - 0.55} \tag{3.36}$$

$$f^{-1}(BR) = |z| = -\frac{0.5}{\log(BR)} + 0.55\,. \tag{3.37}$$

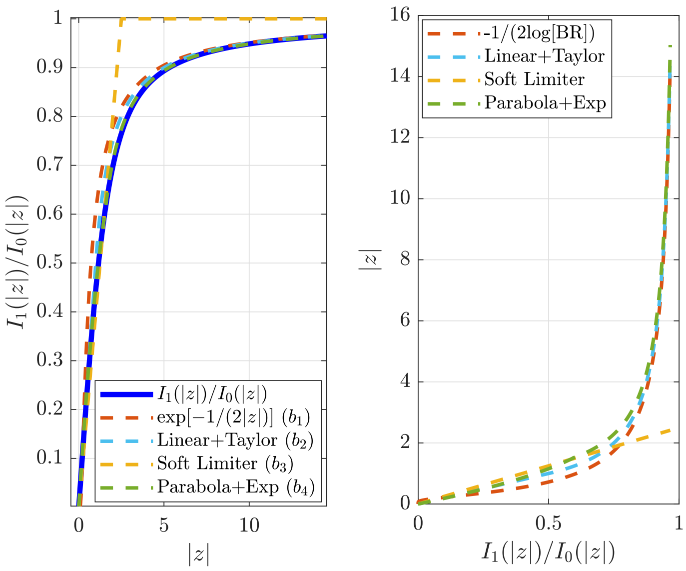


Figure 3.2: On the left, the comparison among the BR, its commonly adopted approximation (b_1) and the proposed (and tested) ones, among which the employed approximation (b_4) in Section 3.8. On the right, the comparison between the approximations for the inverse BR.

Figure 3.2 shows the discussed approximating curves for BR, along with their inverse, and the system performance associated to each of them will be analyzed in Section 3.8. In particular, simulation results demonstrate the significant advantages related to the approximation (b_4) which leads to a higher accuracy in the estimation of the Tikhonov precision $|z|$ obtained by solving the moment matching equation.

3.5.2 Rejection Strategy

The rejection of critical messages is, indeed, a well assessed technique within the EP framework, whenever the projection and division operations (see (3.22)) make *improper distributions* arise [7, 54]. These distributions are easily recognizable when dealing with Gaussian mixtures, since the resulting message is characterized by a negative variance. When, on the contrary, the approximating family is that of Tikhonov distributions, as in the present case, any value for the parameter z_k in (3.14) characterizes a valid distribution. Although improper distributions can sometimes be allowed to propagate along the graph, when the EP algorithms faces critical conditions it is usually advantageous to discard the corresponding updates [7] and specific rejection criteria [54] become useful, and often necessary, for ensuring the success of EP.

Typically, improper messages arise whenever there is an inconsistency between the two factors of the distribution to be projected in (3.22), i.e, the *temporary prior*, represented by the parameter $z_{k,u}$ in (3.24), and the observation message $p_d(\theta_k)$, which, being a mixture, is represented by the set of values z_k^m in (3.24), one for each possible transmitted symbol.

The *modified moment matching equation*, including the approximation discussed in Section 3.5.1, is

$$f(|z_k^{EP}|)\mathrm{e}^{j\angle z_k^{EP}} = \sum_{m=0}^{M-1} \overline{\alpha}_{k,MIX}^m \frac{I_1(|z_{k,MIX}^m|)}{I_0(|z_{k,MIX}^m|)} \mathrm{e}^{j\angle z_{k,MIX}^m} , \tag{3.38}$$

where the definition of $\alpha_{k,MIX}^m$ is analogous to that in (3.16) and $\overline{\alpha}_{k,MIX}^m$ are normalized coefficients, as in (3.19) . Before the calculation of (3.38), a real-time monitoring is introduced on the mixture parameters $z_{k,MIX}^m$ that result from (3.24). We define the angle

$$\gamma_k^m = \arg\left[z_{k,MIX}^m z_{k,u}^*\right] , \tag{3.39}$$

where the superscript m refers to the m-th mixture mode, which quantifies the inconsistency between the phase estimate brought about by each mode and the one provided by the prior distribution. Considering the PN evolution until a given time epoch k, the argument of $z_{k,u}$ typically yields a stable estimate of PN, which should be refined by the current channel observation. When this is very noisy, or there is little agreement between the phase angles, i.e., the absolute value of (3.39) is large, the inclusion of the observation message can introduce spurious phase slips in the PN estimate. We verified that this problem occurs when the estimated marginal distribution of θ_k before the EP projection, i.e., $p_d(\theta_k)p_u(\theta_k)$, is a mixture of modes (associated to the complex parameter $z^m_{k,MIX}$) which is spread around the (temporary) prior distribution of PN, $p_u(\theta_k)$ (associated to the complex parameter $z_{k,u}$). In this case, the projection produces a rather flat Tikhonov distribution, i.e., characterized by a small precision $\left|z_k^{EP}\right|$ and a phase estimate $\arg\left[z_k^{EP}\right]$ which gets far from both the prior and the mixture modes, thus creating an uncertainty condition in the estimation process from which it is sometimes difficult to recover, in the absence of pilot symbols.

In order to avoid such situations, the specific *rejection strategy* that we adopt is thus based on the inconsistency angles (3.39). We assume that the m-th mixture mode is inconsistent if $|\gamma_k^m| > \Gamma_{th}$, i.e, if its absolute angular deviation from the prior phase angle $\arg\left[z_{k,u}\right]$ exceeds a given threshold Γ_{th}. Before performing the projection (3.38), at each time epoch k, the algorithm checks the rejection condition,

$$\sum_{m=0}^{M-1} \mathbb{1}(|\gamma_k^m| > \Gamma_{th}) > \overline{M}\,, \tag{3.40}$$

where $\mathbb{1}(\cdot)$ is an indicator function equal to 1 if the condition is true and 0 otherwise, to verify if the number of inconsistent modes is above a given limit $\overline{M} < M$. When (3.40) is satisfied, the observation message $p_d(\theta_k)$ is considered unreliable and the information contained in it is rejected by setting z_k^m to zero in (3.24). Hence, both $p_d(\theta_k)$ and $p_d^{EP}(\theta_k)$ are treated as uniform distributions, which do not carry any information on the phase, whose marginal equals that of the prior, i.e., a Tikhonov distribution with parameter to $z_{k,u}$. In Figure 3.3, with reference to a generic time epoch k, an example of the angles γ_k^m , $m = 1, \ldots, 4$ defined in (3.39) for a QPSK modulation is provided. Specifically, the phase estimates brought by each mixture mode is compared with the estimate of the prior without the influence of the channel observation. Since the angles in (3.39) describe the phase difference, the normalized parameters

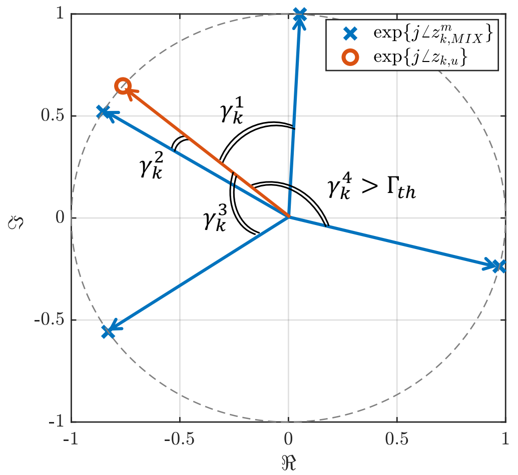


Figure 3.3: An example of the angles describing the consistency between the estimate provided by the prior ($z_{k,u}$) and the mixture distribution ($z_{k,MIX}^m$) before performing the projection, with reference to a generic time epoch k and assuming QPSK modulation.

associated to the two involved distributions are plotted. In the case shown in Figure 3.3, one inconsistent mode is identified.

The values of Γ_{th} and $\overline{M}$ (3.40) depend on the modulation format (M) and on the severity of PN (σ_Δ). Especially at high SNRs, there can be more than one suitable pair and, possibly, multiple rejection conditions can be adopted by logical disjunction ($\vee$) as, e.g.,

$$\left(\sum_{m=0}^{M-1} \mathbb{1}(|\gamma_k^m| > \Gamma_{th,1}) > \overline{M}_1\right) \vee \left(\sum_{m=0}^{M-1} \mathbb{1}(|\gamma_k^m| > \Gamma_{th,2}) > \overline{M}_2\right) \tag{3.41}$$

in the case of two message rejection conditions, whose parameters should be set such that $\Gamma_{th,1} < \Gamma_{th,2}$ and $\overline{M}_1 > \overline{M}_2$. In any case, as stated, the rejection condition is evaluated before the EP projection operation, thus reducing the number of unnecessary operations.

3.5.3 Message Damping

Another solution to overcome the convergence problems of EP, which is known in the literature, is the use of "*damped*"updates [46, 54]. In iterative parametric message passing, the damping technique consists in propagating a convex combination of the new parameter value and the old one. In our case, instead of computing from (3.22)

$$z_{k,d}^{EP} = z_k^{EP} - z_{k,u} \,, \tag{3.42}$$

which is the parameter of $p_d^{EP}(\theta_k)$ at the current n-th iteration, the linear combination

$$z_{k,d}^{(n)} = \xi \cdot (z_k^{EP} - z_{k,u}) + (1-\xi) \cdot z_{k,d}^{(n-1)} \tag{3.43}$$

is used, which partly recovers the value computed at the previous, $(n-1)$-th, inner detector iteration. The value for the damping parameter ξ must be selected, as discussed in Section 3.8, where we apply damping and demonstrate its advantages through numerical results.

3.6 Message Scheduling and Algorithm Summary

Regarding message scheduling, for any algorithm, we take advantage of the structure of the FG, which evidences the logical separation of the decoding part from the phase detection part, corresponding to the upper and lower halves of Figure 3.1. Indeed, $P_d(c_k)$ is the extrinsic information on the k-th code symbol, sent from the decoder to the phase detector, as well as $p_u(\theta_k)$ which, as stated, represents the prior message in (3.20) and is thus seen as a temporary estimate of PN, sent towards the decoder.

We shall assume that the phase-detector subgraph operates first, by exchanging *horizontal* messages along the Markov chain in the lower part of Figure 3.1, so as to update $p_f(\theta_k)$ and $p_b(\theta_k)$ by message passing (see [35], to recall the rules of computation). This horizontal message passing could be iterated, with multiple passes, before sending vertical messages to the decoder subgraph so that the phase estimate could be refined. A wide variety of message scheduling can be adopted, such as a *flooding* schedule [5], a forward filtering followed by either a backward *filtering* or a backward *smoothing* and so on. After a detailed comparison, we concluded that the best performing scheduling is a forward/backward filtering, composed by two independent passes, one for each direction. This schedule is the most natural one as it results from the direct application of the SPA on the detector subgraph. In fact, the Markov chain it comprises resembles that of the BCJR algorithm, where independent forward and backward recursions have a probabilistic interpretation. Therefore, during the first inner-detector iteration the temporary prior parameter $z_{k,u}$ is equal to either $z_{k,f}$ or $z_{k,b}$, depending on the considered direction. In other words, the opposite message has no impact on the observation message approximation. On the other hand, from the 2nd iteration onwards, the temporary prior is determined by the current message in the analyzed direction entering the variable node θ_k and the previous iteration's opposite message, i.e.,

$$z_{k,u}^{(n)} = z_{k,f}^{(n)} + z_{k,b}^{(n-1)} \tag{3.44}$$

during the forward pass and vice versa for the backward pass. The number of "inner" detector iterations is referred to as N_D.

After N_D phase-detection iterations, *vertical messages* $p_u(\theta_k)$ are sent from the phase detector to the upper part of the FG, implementing the decoder, where an inner message passing procedure takes place, until convergence or until a maximum number of decoding iterations is reached.

The exchange of information, through vertical messages, between detector and decoder can be iterated for N_T times,[3] so as to yield a refinement of both the symbols' and the phase samples' estimates, from which the other receiver half can benefit. On the other hand, detector-decoder iterations are considered impractical and they are usually avoided in the implementation of digital receivers, especially at high baud rates. Since phase detection is performed be-

[3]We recall that the notation N_D-N_C-N_T is used to indicate the number of inner detector, inner decoder and turbo iterations.

	Two-terms additions	**Two-terms multiplications**	**LUT accesses**
TP [35]	$7M + 12$	$11M + 22$	$2M + 2$
EP	$16M + 14$	$36M + 17$	$12M + 8$
dp-BCJR	$5N_\theta^2 + (18M - 6)N_\theta - (M + 2)$	$N_\theta(14M + 1) + 1$	$2N_\theta^2 + N_\theta(3M - 1) - M$

Table 3.1: Complexity comparison per code symbol per iteration

fore symbol decoding, the forward/backward messages $p_{f/b}(\theta_k)$ propagated at the first turbo iteration cannot benefit from any symbol estimate, hence phase estimation is accomplished (locally) in a non data-aided fashion, while a data-aided phase estimation occurs, instead, at time epochs where pilot symbols are inserted. In the case of turbo iterations, on the contrary, a (provisional) soft-decision-directed strategy is implemented, thanks to the extrinsic information $P_d(c_k)$, even for the payload symbols.

With reference to the generic $n_{iter,T}$ turbo iteration, the proposed algorithm structure is outlined in Algorithm 2.

3.7 Complexity Analysis

We performed a simplified analysis of the computational complexity of the proposed algorithm based on the EP framework, the TP algorithm proposed in [35] and the dp-BCJR. Referring to the k-th time epoch, we considered the operations needed for the computation of the observation message $p_d(\theta_k)$ approximation, the forward $p_f(\theta_k)$ and backward $p_b(\theta_k)$ messages and, finally, the message $P_u(c_k)$. In order to compute the algorithmic complexity, we counted the number of required operations distinguishing among two real terms multiplications, additions and accesses to the LUT for the computation of nonlinear functions. The total number of required operations is reported in Table 3.1. As it can be seen, the complexity of all the algorithms increases linearly with the constellation cardinality M. However, for the dp-BCJR algorithm, it is much higher. In fact, a critical parameter which has a strong impact on the

Algorithm 2 Proposed EP algorithm

1: **Initialization**:
2: $z_{0,f} \leftarrow 0$, $z_{K-1,b} \leftarrow 0$
3: **Inputs**: P_d from the decoder if $n_{iter,T} > 1$, received samples $\boldsymbol{r}$
4: **Output**: P_u
5: **for** $n = 1$ to N_D **do**
6:     **for** $k = 0$ to $K - 2$ **do**
7:         Compute the *temporary prior* $z_{k,u}^{(n)} \leftarrow z_{k,f}^{(n)} + z_{k,b}^{(n-1)}$
8:         $\forall a_m \in \mathcal{A}$, $m = 1, \ldots, M$, compute:
9:         $\circ$ $z_k^m \leftarrow r_k a_m^* / \sigma^2$
10:         $\circ$ $z_{k,MIX}^m \leftarrow z_{k,u} + z_k^m$
11:         $\circ$ γ_k^m via (3.39)
12:         **if** rejection condition (3.40) is met **then**
13:             $z_k^{EP} \leftarrow z_{k,u}$
14:         **else**
15:             Compute $\overline{\alpha}_{k,MIX}^m$ through (3.16) and (3.19) with $z_k^m \leftarrow z_{k,MIX}^m$
16:             Perform *Tikhonov mixture projection* via (3.38) and (3.28).
            In particular:
17:             $\circ$ Compute the *first circular moment* (CM) of (3.20)
              through the right-hand side of (3.38)
18:             $\circ$ $z_k^{EP} \leftarrow f^{-1}(|\mathrm{CM}|) \exp\{j \arg[\mathrm{CM}]\}$, using (3.28) for $f^{-1}(\cdot)$
19:         **end if**
20:         Compute $z_{k,d}^{EP} \leftarrow z_k^{EP} - z_{k,u}$
21:         For non-pilots only: apply *damping* as in (3.43)
22:         Apply the SPA rules [5] to compute:
23:         $\circ$ $z_{k+1,f} \leftarrow z_{k,f} + z_{k,d}^{EP}$
24:         $\circ$ $z_{k+1,f} \leftarrow z_{k+1,f} / \left(1 + \sigma_\Delta^2 |z_{k+1,f}|\right)$
25:     **end for**
26:     **for** $k = K - 1$ to 2 **do**
27:         Compute the *temporary prior* $z_{k,u}^{(n)} \leftarrow z_{k,b}^{(n)} + z_{k,f}^{(n-1)}$
28:         Repeat steps 8-21
29:         Apply the SPA rules [5] to compute:
30:         $\circ$ $z_{k-1,b} \leftarrow z_{k,b} + z_{k,d}^{EP}$
31:         $\circ$ $z_{k-1,b} \leftarrow z_{k-1,b} / \left(1 + \sigma_\Delta^2 |z_{k-1,b}|\right)$
32:     **end for**
33:     Compute $\forall a_m \in \mathcal{A}$ and $\forall k$ the upward message P_u through SPA rules as
34:     $P_u \leftarrow \exp\{-\frac{|a_m|^2}{2\sigma^2}\} I_0(|z_{k,f} + z_{k,b} + \frac{r_k a_m^*}{\sigma^2}|)$
35: **end for**

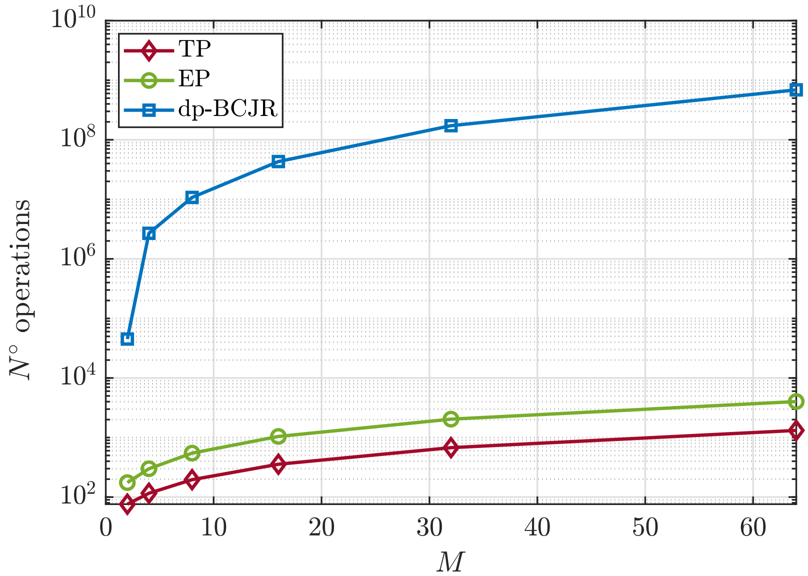


Figure 3.4: Comparison among the complexity of the analyzed algorithms.

computational load is the number of discretization levels N_θ. An empirical rule for the choice of this parameter is provided in [35], which suggests that it is linearly related to M according to $N_\theta = 8M$. We found that the value obtained through this rule of thumb is not sufficient for an accurate quantization of the PN variables, which requires at least 512 levels in the QPSK case.

In order to visualize the complexity gap among the analyzed algorithms, the number of required operations is plotted versus M in Figure 3.4. Note that the curve labelled *EP* refers to the complexity of its native variant. However, applying damping involves only 6 additional operations per symbol within the observation message approximation (performed twice, once for each recursion) and the computation of (3.39)-(3.40) for the adoption of rejection strategies requires approximately $9M$ operations more. It is worth noting that, when the rejection condition is satisfied, no mixture projection is performed, thus saving several operations. It is possible to observe that the EP and TP algorithms have a comparable complexity whereas the dp-BCJR is characterized by a much higher computational load and hence represents a totally impractical solution. Because of this reason and since, with a sufficiently large N_θ, achieves the performance of the plain SPA, it is considered here just as a performance benchmark.

3.8 Simulation Results

We analyzed the performance of different receivers for a coded transmission, employing an LDPC encoder, through the system under investigation. BER is plotted versus the SNR E_b/N_0, where $N_0 = 2\sigma^2$ is the variance of noise samples in (3.1) and E_b is the average energy per information (payload) bit.

In all simulation results, in order to allow the algorithms convergence, pilot symbols, which are known at the receiver, were inserted in the transmitted codeword. The presence of pilots involves a slight decrease of the effective information rate, resulting in an increase I in the required average energy per bit E_b, equal to

$$I = 10 \log_{10} \left(1 + \frac{N_p}{K} \right) \text{ [dB]} \, , \tag{3.45}$$

where N_p is the total number of inserted pilots in a sequence of K information symbols. In order to make a fair comparison, this increase has been artificially introduced in the curve referring to the *Known Phase* scenario, even though pilot symbols were useless in this case since there was no need for phase estimation. Thus, the Known Phase curve shows the performance of the adopted LDPC code. Furthermore, we included the *All Pilots* curve, which represents an ideal bound since a "genie-aided" receiver is assumed to perform the estimation of PN samples resorting to a perfect knowledge of the entire transmitted codeword. Also in this case, the E_b increase, I, to which the *non-ideal* algorithms are subject due to the insertion of pilot symbols, has been accounted for by right-shifting the ideal All Pilots curve for comparison purposes. As discussed in Section 3.7, the *dp-BCJR* algorithm is regarded as a performance benchmark. The number of phase discretization levels is set to $N_\theta = 512$ for all the analyzed modulation formats.

The performance of the algorithms analyzed in Figures 3.5, 3.6 and 3.8 refers to the practical case of separate detection and decoding, i.e., the absence of turbo iterations ($N_T = 1$). The maximum number of LDPC decoding iterations N_C is set to 200.

In Figures 3.5 and 3.6, we considered the transmission of a block of 2000 coded symbols punctured by distributed pilot symbols. The transmitted sequence started with a pilot, then one pilot symbol was regularly alternated with 19 payload symbols, amounting to a 5.3% overhead The payload symbols were the output of a $(3, 6)$-regular LDPC code with rate-1/2 and length 4000 [1]. The QPSK modulation format ($M = 4$) was adopted, since it is robust to the strong PN assumed with $\sigma_\Delta = 6°$.[4]

The focus of Figure 3.5 is on the performance comparison among the various BR approximations involved in the EP framework. A relevant and surprising aspect which emerges from Figure 3.5 is how apparently similar approximations, in terms of the BR curve shape, as seen in Figure 3.2, can lead to very different performance results, in terms of phase estimation accuracy and of the resulting error rates. Figure 3.5 also shows the performance related to the adoption of the modified moment matching equation (3.27) that we proposed in [8]. The reasoning behind the weak performance of this approach is that, for carrying out a fair comparison, all the reported strategies in Figure 3.5 differ solely in the choice of the BR approximation. Therefore, the solution of [8] is reported in the absence of post-processing, which is, instead, a relevant algorithmic part complementing the simpler moment matching equation (3.27). As previously anticipated, to avoid the constraint on ad-hoc post-processing, we shifted our attention to more accurate approximations. In particular, the most

[4]This standard deviation of PN increments $\sigma_\Delta = 6°$ corresponds to a relative bandwidth parameter [53], i.e., the ratio between the phasor 3-dB bandwidth ($f_{3\text{-dB}}$) and the signal bandwidth, equal to $8.7 \cdot 10^{-4}$. For instance, possible parameters generating such value are $f_{3\text{-dB}} = 8.7 \cdot 10^2$ Hz and baudrate equal to 1 MHz.

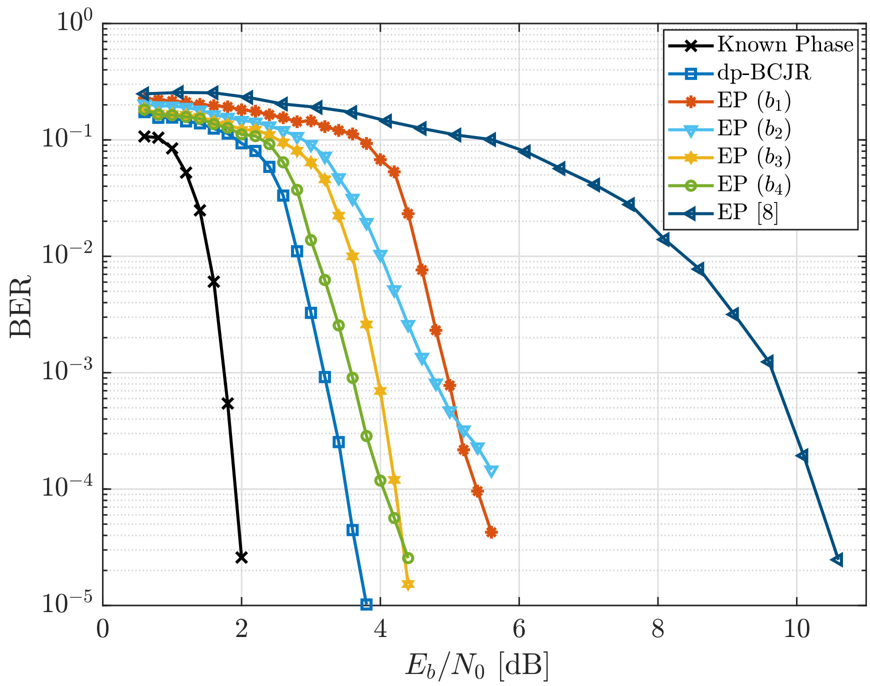


Figure 3.5: Performance for an LDPC-coded [1] QPSK modulation in the separate detection and decoding scenario ($N_T = 1$) employing different BR approximations. The standard deviation of PN increments is $\sigma_\Delta = 6°$ and (1/19) pilot symbols are distributed among the payload.

satisfying one is (b_4), whose inverse is reported in (3.28), resulting to be the closest to the dp-BCJR benchmark. Therefore, in the following, we will focus on this BR approximation as the best alternative to the standard (b_1), which will be still included in the forthcoming simulation results for comparison.

In Figure 3.6, we tested the performance of the EP-based algorithms comparing them with the algorithms presented in Section 3.3 in the same scenario of Figure 3.5. Although representing an ideal situation, the performance of the All Pilots algorithm shows a loss of about 0.2 dB, compared to the Known Phase curve, since the strong PN still has to be estimated in the presence of

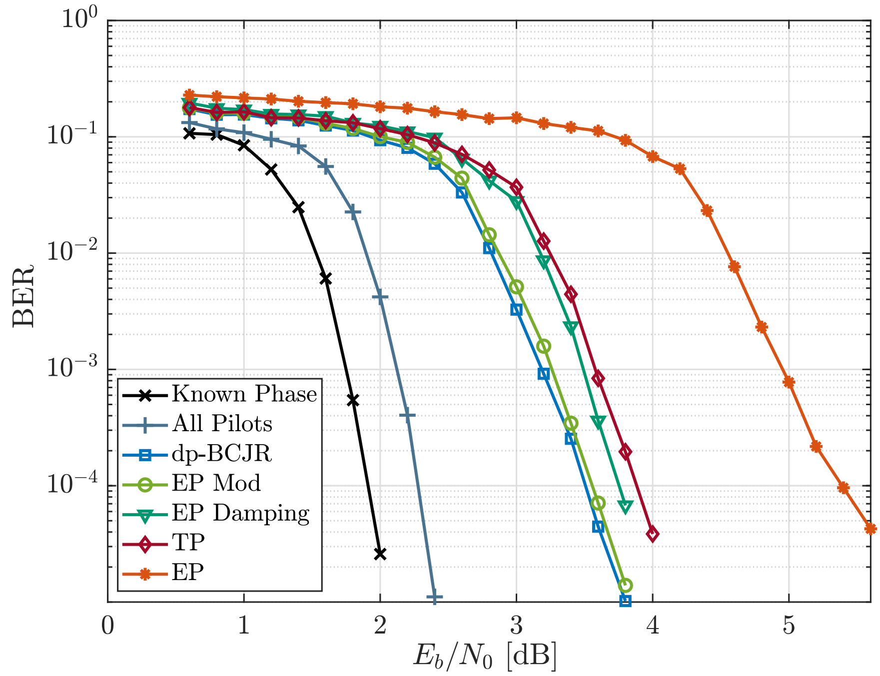


Figure 3.6: Performance for an LDPC-coded [1] QPSK modulation in the separate detection and decoding scenario ($N_T = 1$). The standard deviation of PN increments is $\sigma_\Delta = 6°$ and (1/19) pilot symbols are distributed among the payload (for other parameters, see text).

AWGN.

The performance of the dp-BCJR algorithm is more than 1 dB away from the theoretical All Pilots reference. The TP algorithm of [35] shows 0.4 dB of penalty, compared to the dp-BCJR limit. Therefore, thanks to the help of distributed pilots among the payload, a satisfactory performance is achieved by this low-complexity algorithm. Moving to the primary object of our investigation, we compared the *native* implementation of the EP algorithm (labelled *EP* in figures), which employs the exponential BR approximation (b_1), with two improved variants. The first one only exploits the *damping* technique de-

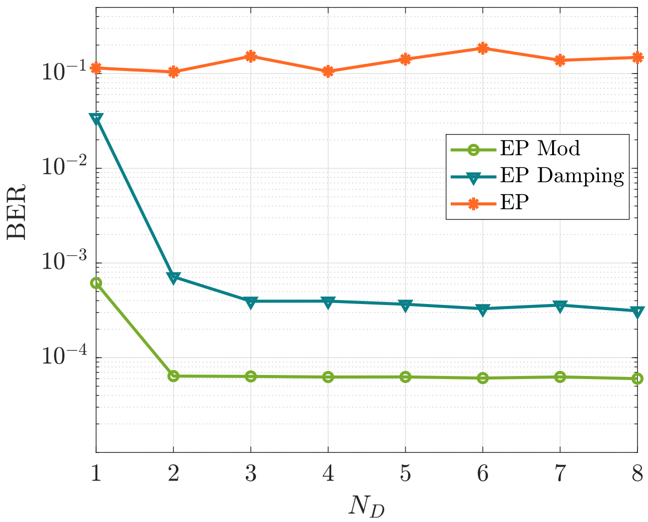


Figure 3.7: Convergence analysis of the EP-based algorithms in the scenario considered in Figure 3.6 at $E_b/N_0 = 3.6\,\text{dB}$.

scribed in Section 3.5.3, it is labelled *EP Damping* in figures and it still employs the BR approximation (b_1), if not otherwise specified. The second one relies, in addition to damping, on the inverse BR approximation (b_4) and on the message rejection strategies discussed in Section 3.5.2 (labelled *EP Mod* in figures). From Figure 3.6, we notice that the native EP algorithm loses about 2 dB, compared to the dp-BCJR benchmark, a penalty that increases towards higher SNRs, where a slight change of slope can be noticed. In this scenario, where the distance among consecutive pilot symbols is short, the introduction of damping allows to solve the convergence problems of EP, outperforming the TP algorithm and reducing the loss to 0.3 dB. In Figure 3.7, we analyzed the convergence of the EP-based algorithms with reference to the scenario of Figure 3.6. In particular, we performed a study of the BER at $E_b/N_0 = 3.6\,\text{dB}$ versus the number of inner detector iterations (N_D). As can be observed from the convergence analysis reported in Figure 3.7, we found out that the native EP does not benefit from inner detector iterations and we, therefore, set

$N_D = 1$. On the contrary, the convergence of its damped version is reached after 3 inner iterations and an optimized damping coefficient ($\xi = 0.05$). The performance of the Modified EP algorithm (*EP Mod*) shows that the adoption of a more accurate moment matching equation (3.38) combined with a suitable message rejection strategy allows to reach the performance benchmark, with a remarkably lower complexity, compared to dp-BCJR. Therefore, the Modified EP algorithm outperforms all the above analyzed algorithms. Additionally, as emerges from Figure 3.7, this algorithm converges faster than EP Damping, requiring only 2 iterations for phase estimation refinements. In this case, the optimal ξ results to be 0.4. We investigated by simulation the parameters adopted for the rejection condition (3.40) and found that $\Gamma_{th} = \pi/2$ and $\overline{M} = 0$ achieve the optimal performance reported in Figure 3.6.

In Figure 3.6, however, the successful operation of the proposed solution occurs in a context where the placement of pilots is quite favorable. Nevertheless, in many communications standards the position of pilots is predefined. Therefore, with the aim of designing a receiver able to cope with sparser pilots' distributions, in Figure 3.8 we tested the same algorithms analyzed above in the case of the *Digital Video Broadcasting Satellite Second Generation (DVB-S2)* system, where pilot symbols are organized into bursts of 36 symbols every 1440 information symbols with the exception of a longer sequence of 90 symbols in the preamble and postamble [2]. We considered the standardized LDPC code with codeword length 64800 and rate 1/2 whereas the analyzed modulation formats are QPSK and 16QAM. For this system, we assumed $\sigma_\Delta = 1°$ for the PN and we still focused on receivers employing separate detection and decoding, i.e., $N_T = 1$.

As seen in Figure 3.8, the algorithm that is most vulnerable to the sparser distribution of pilots is TP, as expected. In fact, in order to reach a near-optimal performance with this pilots placement, it requires multiple turbo iterations as shown in [35]. On the contrary, in the analyzed scenario, where $N_T = 1$, the TP shows a loss of 1 dB with respect to the dp-BCJR benchmark for the QPSK modulation. Moving to a higher constellation cardinality, the

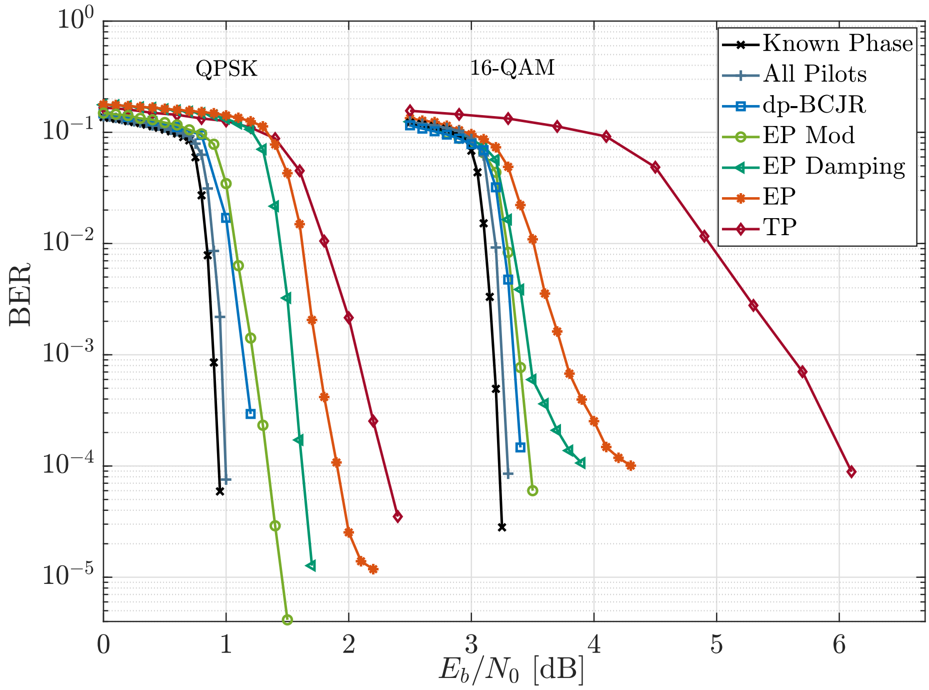


Figure 3.8: Performance for QPSK and 16QAM modulations in the separate detection and decoding scenario ($N_T = 1$) employing the DVB-S2 LDPC code (rate 1/2) and pilots positioning (36/1440) [2]. The standard deviation of PN increments is $\sigma_\Delta = 1°$ (for other parameters, see text).

loss considerably increases, reaching almost 3 dB for 16QAM. The penalty is reduced to about 0.7 dB for both modulation formats, using the native EP algorithm, which however shows the onset of a floor around 10^{-4} for the 16QAM and around 10^{-5} for the QPSK. The adoption of damping ($\xi = 0.4$) improves the performance but an error floor is still present for the 16QAM while for the QPSK there is still a residual loss with respect to the benchmark. It is only with the proposed Modified EP algorithm that the BER curves in Figure 3.8 practically coincide with those of the dp-BCJR benchmark. In this scenario, where the interval between consecutive pilot blocks (1440 symbols) is much longer compared to Figure 3.6 (19 symbols), the convergence of *EP* and *EP Damping* is not guaranteed. As a result, the reported BER curves correspond to a single pass in the detector, which yields the best possible performance. Conversely, the modifications introduced in *EP Mod* enable the algorithm to benefit from multiple passes: the refinement of phase estimates converges after $N_D = 2$ inner detector iterations.

The employed rejection condition is in the form (3.41) with

$$(\Gamma_{th,1}, \Gamma_{th,2}) = \left(\frac{\pi}{12}, \frac{\pi}{6}\right) \tag{3.46}$$

for both modulations, and

$$\left(\overline{M}_1, \overline{M}_2\right) = \begin{cases} (1,0)\,, & \text{for QPSK} \\ (12,7), & \text{for 16QAM} \end{cases} . \tag{3.47}$$

Even though the considered algorithms have been derived under the assumption of Wiener PN, they result to be effective also for other PN models. For instance, in Figure 3.9, we tested the EP-based algorithms in a scenario where the generated PN is characterized by the PSD defined in Table H.1 of [3]. We employed the 32-Amplitude-Phase Shift Keying (APSK) modulation format and the DVB-S2 standard with code rate 9/10 and pilots positioning (36/1440). The detection algorithms still assumed a Wiener PN where the value of σ_Δ to be used for the channel parameter estimation is evaluated through simulations by generating a long sequence of PN samples. In this scenario, the TP algorithm loses more than 4 dB with respect to the All Pilots benchmark. In contrast, the native *EP* algorithm, based on the BR approximation (b_1), performs significantly better. However, its behavior is still not satisfying: the BER curve goes down close to the benchmark (losing less than 0.2 dB) till around 10^{-4}, then a change of slope is registered. Using the proposed BR inversion (b_4), included in *EP Mod*, allows to avoid this effect and to practically overlap to the performance benchmark losing less than 0.05 dB, without the aid of rejection conditions. Both curves are obtained in the absence of damping ($\xi = 1$) and with a single pass in the detector, differing only in the inverse BR approximation.

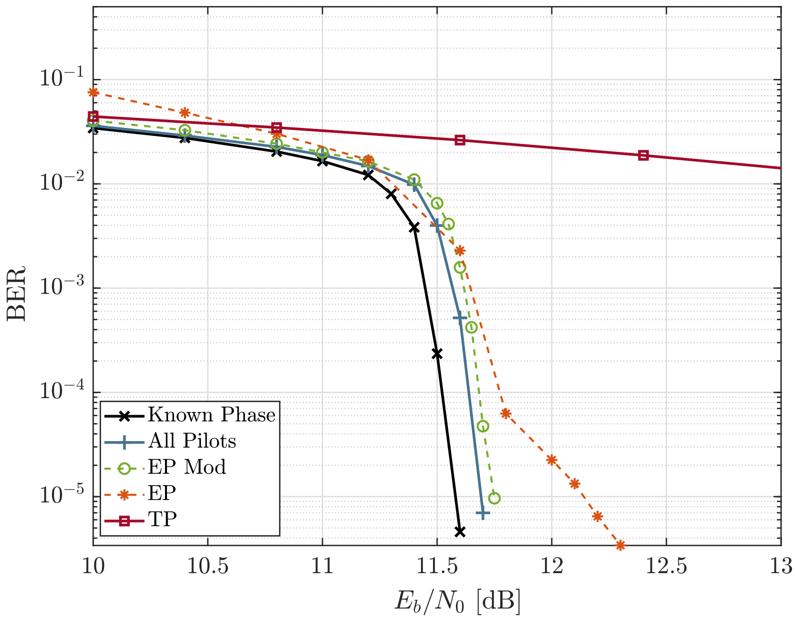


Figure 3.9: Performance for 32-APSK modulation in separate detection and decoding scenario employing the DVB-S2 LDPC code (rate 9/10) and pilots positioning (36/1440). The PN PSD is defined in Table H.1 (*"critical"*) of [3].

In Figure 3.10, we analyzed the most challenging scenario where pilot symbols are concentrated in the preamble and postamble of the data packet, both consisting of 90 symbols. We still employed the DVB-S2 LDPC-code with rate 1/2 and the QPSK modulation, so that a block of 32400 information symbols is transmitted without any pilot in-between. As a result, this situation is particularly critical for channel estimation algorithms, which have to perform the estimates without any synchronization reference over a long information sequence. We assumed Wiener PN with $\sigma_\Delta = 1°$ and, since none of the tested receivers could achieve a satisfying performance when $N_T = 1$, we considered the joint detection and decoding scenario, setting the maximum number of allowed iterations to 1-1-200. Besides the algorithms discussed in the previous

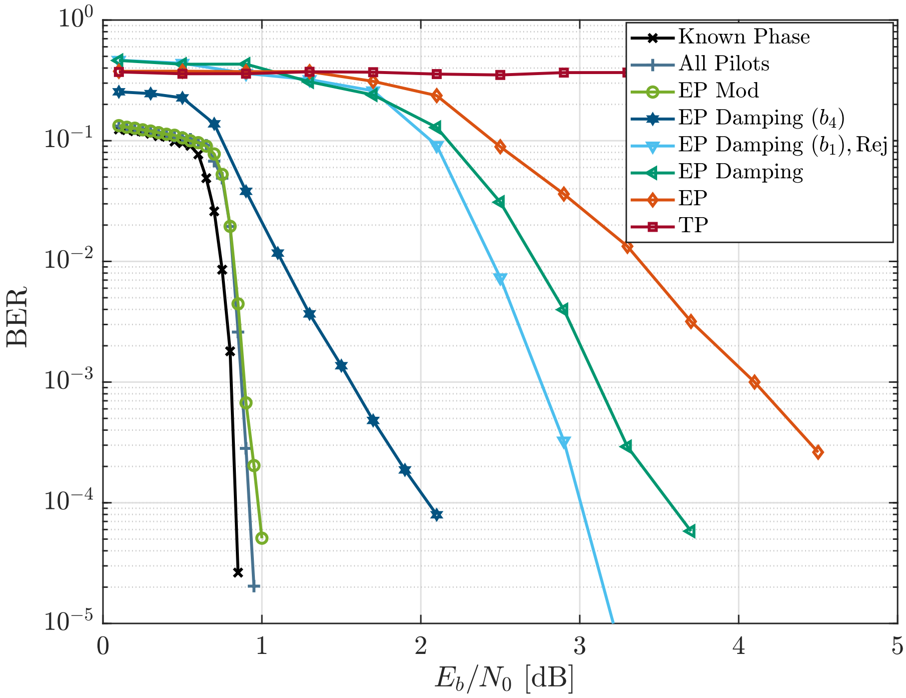


Figure 3.10: Performance for QPSK modulation in the joint detection and decoding scenario (1-1-200 iterations) employing the DVB-S2 LDPC code (rate 1/2) and concentrated pilots, positioned in a preamble and postamble each composed by 90 symbols and separated by 32400 information symbols. The standard deviation of PN increments is $\sigma_\Delta = 1°$.

analyses, in Figure 3.10 we included two more cases, labelled *EP Damping* (b_4) and *EP Damping* (b_1), *Rej*, to demonstrate how each modification made to the native EP contributes to the performance gain.

Due to the absence of distributed pilots, the TP algorithm fails completely, as expected. In fact, in the presence of a (non pilot) payload symbol c_k, the observation message $p_d(\theta_k)$ is a uniform pdf, that does not carry any information, since the decoder has no knowledge of the transmitted bits. Hence, the TP algorithm cannot bootstrap, even with multiple turbo iterations. On the contrary, the algorithms based on the EP framework benefit from the exchange of messages between detector and decoder because the PN estimates approximation is carried out after combining the observation message $p_d(\theta_k)$ with the prior distribution $p_u(\theta_k)$. Nevertheless, as evidenced by Figure 3.10, the native implementation of EP yields unsatisfactory performance and the use of damping allows to gain about 1 dB but the gap from the ideal All Pilots curve is still significant.

The rejection conditions applied to EP Damping affect the slope of the BER curve, resulting in a steeper behavior as highlighted in Figure 3.10 (see the curve labelled *EP Damping* (b_1)*, Rej*). However, the performance loss with respect to the All Pilots algorithm is about 2 dB. In contrast, a more accurate BR approximation (b_4) improves the code threshold as shown by the curve labelled *EP Damping* (b_4). Nevertheless, the slope of the curve is similar to the one of EP Damping with the standard BR approximation (b_1). Only by jointly applying these two modifications is it possible to reach the benchmark, almost perfectly overlapping with the All Pilots curve. For this scenario we set $\xi = 0.5$ and we exploited, also inside the rejection strategy, the knowledge of the current estimate of the symbols pmf $P_d(c_k)$, provided by the decoder. In fact, we set[5]

$$(\Gamma_{th,1}, \Gamma_{th,2}) = \left(\frac{\pi}{6}, \frac{\pi}{4}\right) , \tag{3.48}$$

$$(\overline{M}_1, \overline{M}_2) = ([2 \max P_d(c_k)], [\max P_d(c_k)]) \tag{3.49}$$

so that the clearer the decoder's belief regarding the transmitted symbol, the more stringent the condition to be met for rejection. Moreover, with reference to the n-th turbo iteration, we apply message rejection only if (3.41) and

$$\max_{c_k \in \mathcal{A}} P_d^{(n)}(c_k) \leq \max_{c_k \in \mathcal{A}} P_d^{(n-1)}(c_k) , \tag{3.50}$$

are both satisfied. In fact, condition (3.50) identifies the cases where the iterations are not properly refining the symbols and PN estimates. Finally, at high

[5]We denote with $[\cdot]$ the rounding function to the closest integer.

SNRs, the value of σ^2 that is passed to the detector is slightly larger than the actual one to avoid the overestimation of the message reliability as commonly done in the literature when dealing with suboptimal iterative algorithms.

3.9 Conclusions

We analyzed the EP algorithm operating in a digital receiver for transmissions over channels affected by Wiener PN. We identified and discussed the weaknesses of the native implementation of EP and examined various supplementary techniques that could be applied to enhance its performance. Our investigation highlighted the challenges that arise in applying the EP framework to the considered context and, while solutions taken from the existing literature proved beneficial, they still presented some relevant limitations. By carefully analyzing the message passing procedure, we found that the EP rules of computation for the messages can sometimes induce a collapse of the level of confidence (precision) in the resulting estimates, hence lead to phase slips and subsequent errors.

To address this, we introduced a more accurate approximation of the message projection rule of EP and proposed a novel message rejection strategy aimed at identifying and discarding misleading messages. These countermeasures prevent the appearance of spurious phase detection errors so as to reach the performance of the reference benchmark even in the most challenging scenarios where either the separation of phase detection and decoding is required, or the absence of distributed pilots is imposed by the adopted standard.

Chapter 4

Detection for OTFS-Based Transmissions

4.1 Introduction

Orthogonal Time Frequency Space (OTFS) modulation has emerged as a promising candidate waveform for 6G networks, particularly due to its robustness in challenging wireless environments, e.g., doubly-selective channels characterized by high Doppler shifts (see [56, 57] and references therein). The OTFS modulation format relies on the principle that wireless channels can be effectively modeled in the Doppler-delay domain, which is the one where doubly-selective channels found their most *sparse* representation. This establishes the basis of the working principle of OTFS, which encodes data in such domain to cope with Doppler shifts in multipath propagation channels, typical of dynamically changing environments. As the use cases for wireless communication networks continuously evolve, high mobility scenarios are becoming increasingly widespread [58]. Therefore, since these challenging conditions can create severe performance degradation, the adoption of a proper waveform plays a crucial role.

Furthermore, the advancement of 6G wireless systems will be enhanced by non-terrestrial devices, such as high-altitude platforms and space-born satellites, which will integrate and complement terrestrial networks enabling seamless connectivity and supporting the rapidly growing data rate demands [59,60]. Additionally, mega-constellations of Low Earth Orbit (LEO) satellites can be exploited to enable macro-diversity schemes allowing each user to be jointly served by multiple satellites.

This scenario is equivalent to the one where the information-bearing signal is transmitted through a multipath time-varying channel. However, in the multi-satellite scenario, the multiple signal copies are generated by different transmitters rather than resulting from reflections in the propagation environment. Under this context, high Doppler shifts directly derive from the relative motion of LEO satellites with respect to User Terminal (UT) on ground. For this reason, to fully exploit the advantages of diversity, the most promising waveform is the OTFS.

Several studies have already shown the advantages of OTFS with respect to other modulation formats. For instance, [61] provides a comparison between Orthogonal Frequency Division Multiplexing (OFDM) and OTFS in sparse channel conditions (e.g., outdoor scenarios with few reflectors). The analysis conducted in [62, 63] focuses instead on the application of OTFS in Non-Terrestrial Networks (NTN) and demonstrates the increased robustness of OTFS compared to OFDM in handling the significant Doppler effects typical of satellite communications. Other relevant contributions address the integration of OTFS with (massive) MIMO satellite systems [64, 65].

The focus of this chapter is on detection strategies for OTFS-based transmissions over general doubly-selective channels, assuming perfect Channel State Information (CSI) at the receiver. Effective detection is fundamental to leverage the potential of OTFS in real channels applications, as it directly impacts the reliability and performance in both terrestrial and satellite communications. Firstly, we show the advantages of OTFS with respect to OFDM, which is the primary competing conventional waveform. Then, focusing on OTFS, we analyze the state-of-the-art detection techniques and propose a novel soft-output algorithm based on MP. Differently from the low-complexity approaches available in the literature [51, 57], the presented algorithm leverages the particular structure assumed by the channel matrix in the Doppler-delay domain. Moreover, it introduces an innovative scheduling approach, where the strongest interferers are prioritized in the detection process. We consider separate detection and decoding, i.e., without "turbo" reprocessing of the decoder output, and test the system performance by evaluating the *pragmatic capacity*, which offers a great practical relevance as discussed in Section 1.5. We examine diverse channel conditions. Specifically, the considered terrestrial scenario is a standard 3GPP multipath fading channel profile [66], whereas the simulated satellite scenarios are based on the *Starlink* constellation, considering various numbers of satellites in visibility with the UT.

For comparison purposes, we tested (i) the linear Minimum Mean Square Error (MMSE) block equalizer, which is characterized by an extremely high complexity due to a large-dimensional matrix inversion, (ii) an MP-based detection scheme firstly proposed in [67] and applied to OTFS in [68], and (iii) low-complexity algorithms taken from the existing literature based on message approximation through Gaussian distributions [51, 57, 69]. Simulation results demonstrate that the proposed solution achieves an optimal trade-off between complexity and performance compared to the alternative state-of-the-art solutions.

The rest of this chapter is organized as follows. Section 4.2 provides an overview of the analyzed scenarios. Then, Section 4.3 presents the system model suitable for the description of both contexts. Section 4.4 is dedicated to the review of the state-of-the-art detectors. The proposed algorithm is summarized and discussed in Section 4.5. In Section 4.6, we report and comment the numerical results and, finally, Section 4.7 provides some concluding remarks.

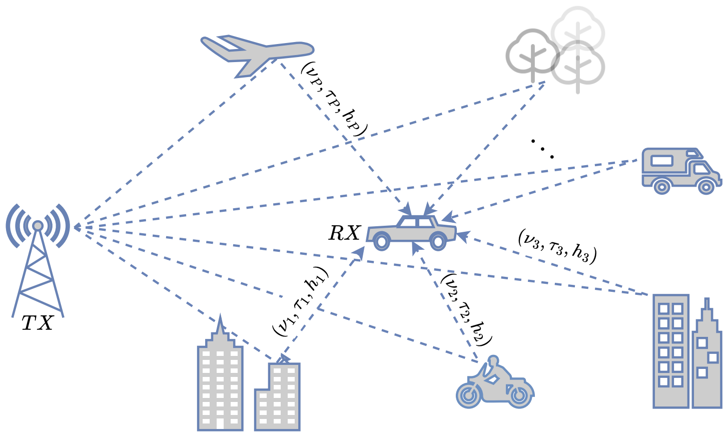


Figure 4.1: An example of the first analyzed scenario. The transmitted signal encounters multiple (possibly) moving reflectors during its propagation towards the receiver.

4.2 Analyzed Scenarios

We consider an OTFS-based transmission with reference to the scenarios depicted in Figures 4.1 and 4.2.

In particular, the scenario of Figure 4.1 is an example of high-mobility multipath channel characterized by scatterers moving at different speeds and directions. The multiple reflected, diffracted and scattered copies of the transmitted signal reach the receiver attenuated in power, delayed in time and shifted in phase and/or frequency. All the received components are summed together at the receiver, generating a distorted signal with respect to the transmitted one. In fact, the multiple replicas can overlap constructively but also destructively.

In the scenario of Figure 4.2, we instead consider a multi-satellite communication system consisting of LEO satellites transmitting the same OTFS-modulated signal to a single-antenna UT on ground. Among the various kinds of non-terrestrial devices, LEO satellites, which are located at an altitude of 500-2000 km, have attracted particular interest. Unlike traditional geostationary and medium Earth orbit satellites, which operate at altitudes of 35786 km and 5000-20000 km, respectively, LEO satellites offer lower propagation latency and reduced path loss. Macro-diversity schemes, which are generated by employing mega-constellations of LEO satellites, allow to enhance the system reliability, especially in scenarios where the line-of-sight link between the satellite and the UT is obstructed by physical impairments on ground. Given that satellites occupy different positions in the sky and follow independent trajectories, shadowing effects impact satellite-to-UT links in a mutually independent way. As a results, diversity is essential not only for ensuring a uniform throughput but also for drastically reducing the link outage probability. However, signals from diverse satellites typically reach the UT at distinct time epochs and are also affected by different Doppler shifts and phases. Therefore, while higher diversity can significantly improve performance, the UT must effectively combine the received information to benefit from multi-satellite configurations. Because of the relative movement of each satellite with respect to the UT, only P satellites are supposed to be in visibility, where P can be much smaller than the total number of satellites composing the constellation. Each satellite orbits at a different distance from the UT and is characterized by a distinct relative speed.

4.3 System Model

We consider a general system model that can be adopted for both scenarios described in Section 4.2, as in each case the received signal results from the superposition of P replicas of the same OTFS-based transmission. These received components are affected by distinct propagation delay $\{\tau_p\}_{p=1}^P$, Doppler shifts $\{\nu_p\}_{p=1}^P$ and complex gains $\{h_p\}_{p=1}^P$. Since P is typically small, the wireless channel presents a sparse representation in the Doppler-delay domain, as emerged by the Doppler-delay spread function

$$\gamma(\nu, \tau) = \sum_{p=1}^{P} h_p \delta(\nu - \nu_p) \delta(\tau - \tau_p) \,. \tag{4.1}$$

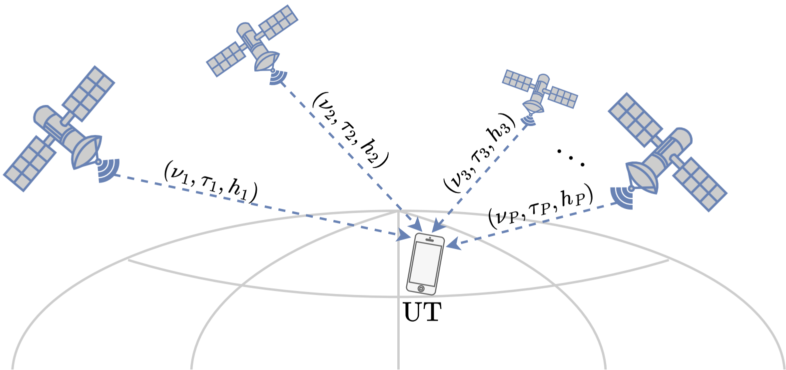


Figure 4.2: A representation of the second analyzed scenario. P satellites serve the same UT to realize macro-diversity.

We will assume that the transmission system is designed such that the propagation delays and the Doppler shifts can be considered as constant for the duration of a transmitted frame.

Within the OTFS modulation, data-symbols $\{x[k, l]\}$ (belonging to a finite alphabet $\mathcal{A}$) for $k = 0, 1, \ldots, N-1$ and $l = 0, 1, \ldots L-1$ are arranged into an $N \times L$ grid in the Doppler-delay domain. These symbols are assumed to be spaced by $1/NT$ in the Doppler domain and $1/L\Delta f$ in the delay domain, where T is the symbol duration and Δf is the subcarrier spacing, i.e., $\Delta f = B/L$ being B the total available bandwidth. The values of T and Δf are usually selected in such a way that $\max_p\{\tau_p\} < T$ and $\max_p\{\nu_p\} < \Delta f$. It is worth to underline that we do not make the simplifying unrealistic assumption that Doppler and delay shifts are integer multiples of the symbol grid. As described by Figure 4.3, symbols are then converted to the time-frequency domain through the so-called *Inverse Symplectic Finite Fourier Transform (ISFFT)*, i.e., a Fourier transform with respect to the delay and an inverse transform with respect to the Doppler. Therefore, the resulting expression of samples in

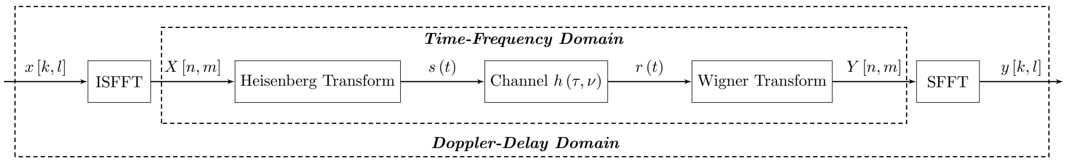


Figure 4.3: Block diagram of an OTFS-based system.

the time-frequency domain is

$$X[n,m] = \sum_{k=0}^{N-1} \sum_{l=0}^{L-1} x[k,l] e^{\jmath 2\pi \left(\frac{nk}{N} - \frac{ml}{L}\right)} \tag{4.2}$$

for $n = 0, \ldots, N-1$ and $m = 0, \ldots, L-1$.

Consequently, the continuous-time transmitted signal $s(t)$ can be expressed as

$$s(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{L-1} X[n,m] p_{\text{tx}}(t - nT) e^{\jmath 2\pi m \Delta f (t-nT)}, \tag{4.3}$$

i.e., symbol $X[n,m]$ is transmitted at time n and over subcarrier m, and $p_{\text{tx}}(t)$ is a transmit shaping pulse. The transform applied in (4.3), which generates the signal $s(t)$ from symbols $\{X[n,m]\}$, is usually called *Heisemberg transform* in the OTFS literature. The general expression in (4.3) can be specialized to represent any of the multicarrier modulation formats available in the literature by selecting a particular shaping pulse and a proper value of T and Δf. For instance, setting $\Delta f = 1/T$ and adopting a rectangular shaping pulse with duration T, (4.3) represents a classical OFDM modulation with properly precoded information symbols.

Modeling the time-varying impulse response of the doubly selective channel as

$$h(t,\tau) = \sum_{p=1}^{P} h_p e^{\jmath 2\pi \nu_p t} \delta(\tau - \tau_p) \tag{4.4}$$

where $\{h_p\}_{p=1}^{P}$ are the complex channel gains accounting for the different path losses, the complex envelope of the received signal $r(t)$ can be expressed as

$$r(t) = \sum_{p=1}^{P} h_p s_p(t - \tau_p) e^{\jmath 2\pi \nu_p t} + w(t) \,, \tag{4.5}$$

where, for the multi-satellite scenario of Figure 4.2, $s_p(t)$ is the complex envelope of the signal transmitted by the p-th satellite whereas, in the scenario of Figure 4.1 where a single transmitter is employed, $s_p(t) = s(t)$. Moreover, $w(t)$ models the complex AWGN whose real and imaginary components have power spectral density σ^2. In the case of multiple transmitters, the signal $s_p(t)$ includes the factors $\widetilde{\nu}_p$, $\widetilde{\theta}_p$ and $\widetilde{\tau}_p$ required for the compensation of the frequency offset, phase and delay, respectively, performed by the P satellites with reference to one point on the Earth, referred to as *ideal UT position*. The signal $s_p(t)$ can thus be expressed as

$$s_p(t) = e^{\jmath(2\pi\widetilde{\nu}_p t + \widetilde{\theta}_p)} s(t - \widetilde{\tau}_p) \,, \tag{4.6}$$

where $s(t)$ is the ordinary information-bearing waveform sent by all the P satellites.

At the receiver side, the resulting signal at the output of a filter matched to $p_{\mathrm{rx}}(t) e^{\jmath 2\pi f t}$, where $p_{\mathrm{rx}}(t)$ is a receive shaping pulse, is

$$Y(t, f) = \int r(t') p^*_{\mathrm{rx}}(t' - t) e^{-\jmath 2\pi f (t' - t)} \, dt' \,. \tag{4.7}$$

By sampling $Y(t, f)$ at $t = nT$ and $f = m\Delta f$, we obtain the samples $\{Y[n, m]\}$. In the OTFS literature, the filtering of the received signal with the bank of matched filters plus sampling is usually called *Wigner transform*. Finally, through the *Symplectic Finite Fourier Transform (SFFT)*, it is possible to obtain the received samples $\{y[k, l]\}$, for $k = 0, \ldots, N-1$ and $l = 0, \ldots, L-1$ in the Doppler-delay domain:

$$y[k, l] = \frac{1}{NL} \sum_{n=0}^{N-1} \sum_{m=0}^{L-1} Y[n, m] e^{\jmath 2\pi\left(-\frac{nk}{N} + \frac{ml}{L}\right)} \,. \tag{4.8}$$

In the following, we will assume $p_{\text{rx}}(t) = p_{\text{tx}}(t)$, $\Delta f = 1/T$, and that $p_{\text{tx}}(t)$ is a rectangular pulse with duration T. Under these assumptions, the noise samples affecting the useful signal in $\{Y[n,m]\}$ are white. Moreover, since the SFFT does not color the noise, the same consideration holds also for the noise samples affecting $\{y[k,l]\}$. In the following, they will be omitted for the sake of notational simplicity.

Under the further assumption of absence of interblock interference, the received samples $y[k,l]$ can be expressed as [61, 68]

$$y[k,l] = \sum_{k',l'} x[k',l'] g_{k,k'}[l,l'] \,, \tag{4.9}$$

where the Intersymbol Interference (ISI) coefficient of the Doppler-delay pair $[k',l']$ seen by sample $[k,l]$ is given by [61, 68]

$$g_{k,k'}[l,l'] = \sum_{p=1}^{P} h_p e^{\jmath 2\pi \nu_p \tau_p} \mathbf{\Psi}^p_{k,k'}[l,l'] \,. \tag{4.10}$$

The entries of the channel matrix $\mathbf{\Psi}^p_{k,k'}$ in (4.10) are defined as

$$\begin{aligned}\mathbf{\Psi}^p_{k,k'}[l,l'] \simeq & \frac{1}{NL} \frac{1 - e^{\jmath 2\pi (k'-k+\nu_p NT)}}{1 - e^{\jmath 2\pi \frac{(k'-k+\nu_p NT)}{N}}} \frac{1 - e^{\jmath 2\pi (l-l'-\tau_p L\Delta f)}}{1 - e^{\jmath 2\pi \frac{(l-l'-\tau_p L\Delta f)}{L}}} e^{\jmath 2\pi \nu_p \frac{l'}{L\Delta f}} \\ & \cdot \begin{cases} 1 & \text{if } l' \in [0, L-1-\lceil \tau_p / (T/L) \rceil] \\ e^{-\jmath 2\pi \left(\frac{k'}{N} + \nu_p T\right)} & \text{if } l' \in [L - \lceil \tau_p / (T/L) \rceil, L-1] \end{cases} ,\end{aligned} \tag{4.11}$$

where the approximation derives from the use of a discrete sum in place of the integral in the cross-ambiguity function between the transmit and receive shaping pulses, which is included in the expression of $\{Y[n,m]\}$. For more details refer to [68]. It follows from (4.9) that each symbol $x[k,l]$ in the Doppler-delay grid can, in principle, interfere with any other symbol within the same OTFS block. We therefore obtain a linear system characterized by a two-dimensional ISI. This is the reason for which the receiver has to process the entire OTFS block and could not take into account a subset of samples only. The magnitude of $\boldsymbol{\Psi}^p_{k,k'}[l,l']$ depends on $\{k,l,k',l'\}$ through the differences $k-k'$ and $l-l'$. Defining the Dirichlet kernel function

$$D_n(x) = \frac{1-e^{\jmath 2\pi x}}{1-e^{\jmath 2\pi x/n}}\,,$$

we can express

$$\begin{aligned}\left|\boldsymbol{\Psi}^p_{k,k'}[l,l']\right| \simeq & \frac{1}{NL}\left|D_N(k'-k+\nu_p NT)\right| \\ & \cdot \left|D_L(l-l'-\tau_p L\Delta f)\right|\,.\end{aligned}$$

The input-output equation (4.9) can be organized in matrix form. Writing the $N\times L$ matrices of transmitted symbols and received samples as NL-dimensional column vectors (stacking the columns of the corresponding matrices on top of each other), we obtain the block-wise input-output relation in the form [61]

$$\boldsymbol{y} = \boldsymbol{\Psi}\boldsymbol{x} + \boldsymbol{w}\,, \tag{4.12}$$

where, defining $h'_p = h_p e^{\jmath 2\pi\nu_p\tau_p}$,

$$\boldsymbol{\Psi} = \sum_{p=1}^{P} h'_p \boldsymbol{\Psi}_p \tag{4.13}$$

and matrices $\{\boldsymbol{\Psi}_p\}$ have dimension $NL\times NL$ and are obtained by properly stacking the entries of the $L\times L$ matrices $\boldsymbol{\Psi}^p_{k,k'}$ for $k,k'=0,\ldots,N-1$, while $\boldsymbol{w}$ denotes the AWGN with zero mean and covariance matrix $2\sigma^2\mathbb{I}_{NL}$, where $\mathbb{I}_{NL}$ is the $NL\times NL$ identity matrix.

The channel effect on each transmitted symbol in the Doppler-delay grid can be described as follows. As an example, consider the transmission over the time-frequency selective channel in (4.4) of an OTFS block composed by all zero symbols except one (placed anywhere in the grid) with enough energy to be distinguishable. The p-th received component shows the transmitted symbol (circularly) shifted in the grid according to the pair (ν_p,τ_p). Moreover, if the involved channel parameters are integer multiples of the grid resolution, the

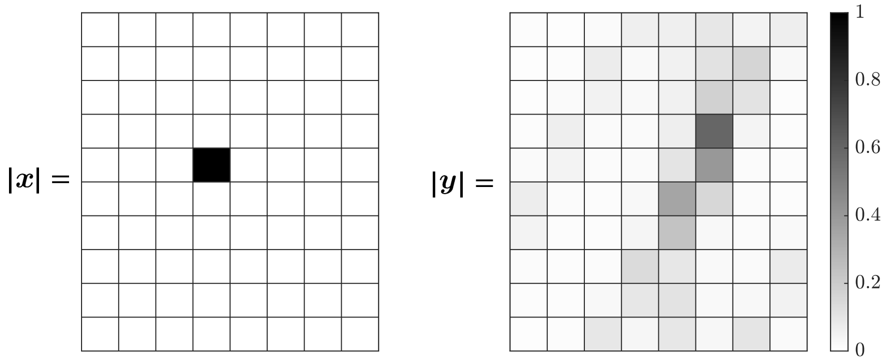


Figure 4.4: Example of the channel effect on a generic transmitted symbol. In the left subfigure, the transmitted symbol (magnitude) in the Doppler-delay grid. In the right subfigure, the received energy distribution after the propagation through the multipath channel.

whole received energy is confined in a single grid position. Otherwise, in the general and realistic scenario of Doppler and delay with a fractional component, an energy leakage, which is dictated by the Dirichlet kernel functions included in the channel matrix $\boldsymbol{\Psi}$, appears in the positions surrounding the peak value. Due to the system's linearity, the overall effect of the multiple paths results in the superposition of the shifts associated with each component. In Figure 4.4, an example of the symbol shift in a multipath scenario is shown.

4.4 State-of-the-art Detection Algorithms

In this section, detection algorithms for OTFS-based transmissions are described. Firstly, in Section 4.4.1, the MMSE detector is presented. Then, Section 4.4.2 describes the Channel Shortening (CS) technique, proposed in [70] for MIMO channels and here applied to OTFS detection. Since both these two approaches have a very good performance but an impractical complexity, different low-complexity soft-output algorithms exploiting the framework based on FGs and the SPA have been developed [51, 57, 68].

It is possible to distinguish these MP algorithms for OTFS detection in two categories:

- the first one, presented in Section 4.4.3, is based on *discrete* MP (i.e., the incoming and outgoing messages from a VN have size M)[1];
- the second category, discussed in Section 4.4.4, focuses on *continuous* MP algorithms, which are derived under the assumption that the information symbols are continuous RVs, whose pdf is approximated by a complex Gaussian distribution.

Moreover, inside the first family we can further differentiate between the algorithms working on the Ungerboeck observation model [68] and those derived from the Forney model.

In general, for FG-based algorithms, depending on how the posterior pmf of the modulation symbols $\boldsymbol{x}$, i.e.,

$$P(\boldsymbol{x} \,|\, \boldsymbol{y}) \propto p(\boldsymbol{y} \,|\, \boldsymbol{x}) P(\boldsymbol{x}) \,, \tag{4.14}$$

is factorized, different algorithms can be derived.

4.4.1 MMSE block equalizer

Denoting with x_k the k-th entry of the column vector $\boldsymbol{x}$ in (4.12), the detector provides at its output the noisy estimate $\hat{x}_k$, $k = 1, \ldots, NL$. This detector is derived under the assumption that $\boldsymbol{x}$ is a Gaussian vector. The optimal estimator $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ according to the MMSE criterion results to be linear with the observation $\boldsymbol{y}$ and has thus expression

$$\hat{\boldsymbol{x}} = \boldsymbol{C}\boldsymbol{y} \,, \tag{4.15}$$

where $\boldsymbol{C}$ is a proper $NL \times NL$ matrix that can be found by using the orthogonality principle [71], stating that the error and the observations must be orthogonal, i.e,

$$\mathbb{E}\{\boldsymbol{y}(\hat{\boldsymbol{x}} - \boldsymbol{x})^H\} = \boldsymbol{0} \,, \tag{4.16}$$

[1] We recall that M denotes the constellation cardinality.

or, equivalently

$$\mathbb{E}\{\boldsymbol{y}\hat{\boldsymbol{x}}^H\} = \mathbb{E}\{\boldsymbol{y}\boldsymbol{x}^H\}\,. \tag{4.17}$$

Note that $\boldsymbol{y}(\hat{\boldsymbol{x}} - \boldsymbol{x})^H$ is a $NL \times NL$ matrix. Thus, (4.16) is indeed a linear system with N^2L^2 equations and N^2L^2 unknowns. By substituting (4.15) in (4.17), we obtain

$$\mathbb{E}\{\boldsymbol{y}\boldsymbol{y}^H\}\boldsymbol{C}^H = \mathbb{E}\{\boldsymbol{y}\boldsymbol{x}^H\}\,. \tag{4.18}$$

By assuming uncorrelated symbols with mean zero and mean square value A^2, i.e., $E\{\boldsymbol{x}\boldsymbol{x}^H\} = A^2\mathbb{I}_{NL}$, we have

$$\begin{aligned}
\mathbb{E}\{\boldsymbol{y}\boldsymbol{y}^H\} &= \mathbb{E}\{(\boldsymbol{\Psi}\boldsymbol{x} + \boldsymbol{w})(\boldsymbol{\Psi}\boldsymbol{x} + \boldsymbol{w})^H\} = \boldsymbol{\Psi}\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^H\}\boldsymbol{\Psi}^H + \mathbb{E}\{\boldsymbol{w}\boldsymbol{w}^H\} \\
&= A^2\boldsymbol{\Psi}\boldsymbol{\Psi}^H + 2\sigma^2\mathbb{I}_{NL}\,, \\
\mathbb{E}\{\boldsymbol{y}\boldsymbol{x}^H\} &= \mathbb{E}\{(\boldsymbol{\Psi}\boldsymbol{x} + \boldsymbol{w})\boldsymbol{x}^H\} = \boldsymbol{\Psi}\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^H\} = A^2\boldsymbol{\Psi}
\end{aligned}$$

and thus (4.18) becomes

$$\left[A^2\boldsymbol{\Psi}\boldsymbol{\Psi}^H + 2\sigma^2\mathbb{I}_{NL}\right]\boldsymbol{C}^H = A^2\boldsymbol{\Psi}\,.$$

By defining $\overline{\sigma}^2 = 2\sigma^2/A^2$, we finally obtain

$$\boldsymbol{C} = \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1} \tag{4.19}$$

having exploited the fact that matrix $\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]$ is Hermitian. Hence, substituting (4.19) in (4.15), we obtain

$$\hat{\boldsymbol{x}} = \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{y}\,, \tag{4.20}$$

or, equivalently,

$$\hat{\boldsymbol{x}} = \left[\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y}\,. \tag{4.21}$$

In fact, right multiplying (4.20) by $\left[\boldsymbol{\Psi}^H\right]^{-1}\boldsymbol{\Psi}^H = \mathbb{I}_{NL}$, we obtain

$$\begin{aligned}
\hat{\boldsymbol{x}} &= \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\left[\boldsymbol{\Psi}^H\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y} && (4.22)\\
&= \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}^H\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right)\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y} && (4.23)\\
&= \boldsymbol{\Psi}^H\left[\left(\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \overline{\sigma}^2\mathbb{I}_{NL}\right)\boldsymbol{\Psi}^H\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y} && (4.24)\\
&= \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}^H\right]^{-1}\left[\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y} && (4.25)\\
&= \left[\boldsymbol{\Psi}^H\boldsymbol{\Psi} + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi}^H\boldsymbol{y}\,, && (4.26)
\end{aligned}$$

where in (4.23) and (4.25) the following known property on the inverse of matrices product has been applied. Given two generic matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, it holds

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}\,. \tag{4.27}$$

Finally, for the computation of the pragmatic capacity we need the *a-posteriori* pmf of the transmitted symbols x_k given the detector soft-output $\hat{x}_k$, i.e., $P(x_k\,|\,\hat{x}_k)$. To this purpose, the estimate $\hat{x}_k$ is treated as the output of a (virtual) AWGN channel as mentioned in Section 1.5. We can thus express the k-th symbol as a function of the relative estimate $\hat{x}_k$ as

$$x_k = \hat{x}_k + \epsilon_k\,, \tag{4.28}$$

where ϵ_k is the zero-mean estimation error whose variance $\sigma^2_{\epsilon,k}$ corresponds to the element in position (k,k) of the following error covariance matrix:

$$\begin{aligned}
\boldsymbol{C}_\epsilon &= \mathbb{E}\left[(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{x}-\hat{\boldsymbol{x}})^H\right] && (4.29)\\
&= \mathbb{E}\left[\boldsymbol{\epsilon}(\boldsymbol{x}-\hat{\boldsymbol{x}})^H\right] && (4.30)\\
&= \mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{x}^H\right] - \mathbb{E}\left[\boldsymbol{\epsilon}\hat{\boldsymbol{x}}^H\right] && (4.31)
\end{aligned}$$

where, in (4.30), $\boldsymbol{\epsilon}$ is a error column vector with entries ϵ_k, $k = 1,\ldots,NL$. Then, exploiting the orthogonality principle in (4.16), we obtain

$$\begin{aligned}
\boldsymbol{C}_\epsilon &= \mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{x}^H\right] - \underbrace{\mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{y}^H\right]}_{=\boldsymbol{0}}\left\{\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\right\}^H\boldsymbol{\Psi} && (4.32)\\
&= \mathbb{E}\left[(\boldsymbol{x}-\hat{\boldsymbol{x}})\boldsymbol{x}^H\right] && (4.33)\\
&= \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^H\right] - \mathbb{E}\left[\hat{\boldsymbol{x}}\boldsymbol{x}^H\right] && (4.34)\\
&= A^2\mathbb{I}_{NL} - \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\mathbb{E}\left[(\boldsymbol{\Psi}\boldsymbol{x}+\boldsymbol{w})\boldsymbol{x}^H\right] && (4.35)\\
&= A^2\left(\mathbb{I}_{NL} - \boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + \overline{\sigma}^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi}\right) && (4.36)\\
&= A^2\left(\mathbb{I}_{NL} + \frac{\boldsymbol{\Psi}^H\boldsymbol{\Psi}}{\overline{\sigma}^2}\right)^{-1} && (4.37)\\
&= \left(\frac{\mathbb{I}_{NL}}{A^2} + \frac{\boldsymbol{\Psi}^H\boldsymbol{\Psi}}{2\sigma^2}\right)^{-1} && (4.38)
\end{aligned}$$

where (4.37) is obtained applying the Woodbury matrix identity.

It follows that the required pmf is

$$P(x_k \,|\hat{x}_k) = g_{\mathbb{C}}(x_k, \hat{x}_k, \sigma^2_{\epsilon,k})\,. \tag{4.39}$$

The complexity of this detector is $\mathcal{O}\big((NL)^3\big)$, since its implementation requires the inversion of the $NL \times NL$ matrix in (4.20). In a time-varying scenario, when the channel matrix $\boldsymbol{\Psi}$ changes at every block, and for typical values of N and L (e.g., $N = 50$ and $L = 64$), this complexity cannot be afforded. A simplified MMSE detector has been proposed in [72] which is, however, based on the simplified assumption that ν_p and τ_p are multiple of the grid spacings in the Doppler-delay domain. In a practical scenario, when this assumption is not fulfilled, a significant performance degradation is observed.

4.4.2 Channel shortening

Another possible solution for the OTFS detection is based on the use of the CS technique. The CS approach was first proposed in 1973 by Falconer and Magee [73] with the aim of reducing the computational complexity of the Viterbi algorithm. In fact, this last algorithm, as well as other detection methods working on a trellis diagram such as BCJR [74], can quickly become unmanageable when the size of the trellis increases, that is, when the channel memory Λ is high and/or the dimension M of the adopted constellation grows, being the number of states defined by M^{Λ}. In order to solve this complexity problem two main directions can be followed: (1) to work on the complete trellis but exploring only a fraction of it, and (2) to build a reduced trellis which is, then, processed with full complexity. The CS belongs to the second category of algorithms since this detector approximates the true channel with a *shorter* channel. This approach simplifies the optimal MAP detector, however, its complexity exceeds the one of the MMSE, which can be seen as a special case of CS with minimum complexity.

After the first investigation on CS [73], many other works were published on this topic. A general derivation of CS for ISI and MIMO channels is proposed by Rusek and Prlja in [70]. This work proposes a general technique for the design of an optimal receiver for a given reduced complexity. In particular, the optimal channel shortening filter and the optimal branch labels of the reduced trellis are derived in closed form. These closed form expressions are obtained under the assumption of Gaussian inputs however, in [70], the authors verify that the performance is not affected by the adoption of commonly used discrete constellations.

In this section, we will briefly review the main results of [70] for a general system model described by (4.12).

Assuming the channel matrix $\boldsymbol{\Psi}$ to be perfectly known, the conditional pdf of the received samples $\boldsymbol{y}$ given the symbols $\boldsymbol{x}$ can be computed as

$$p\left(\boldsymbol{y}|\boldsymbol{x}\right) = \frac{1}{\left(2\pi\sigma^2\right)^{NL}} \exp\left(-\frac{\left\|\boldsymbol{y}-\boldsymbol{\Psi}\boldsymbol{x}\right\|^2}{2\sigma^2}\right) \tag{4.40}$$

$$\propto \exp\left(-\frac{\left\|\boldsymbol{y}-\boldsymbol{\Psi}\boldsymbol{x}\right\|^2}{2\sigma^2}\right) \tag{4.41}$$

$$\propto \exp\left(-\frac{\boldsymbol{y}^H\boldsymbol{y}-2\Re\{\boldsymbol{y}^H\boldsymbol{\Psi}\boldsymbol{x}\}+\boldsymbol{x}^H\boldsymbol{G}\boldsymbol{x}}{2\sigma^2}\right), \tag{4.42}$$

where in (4.41) the proportionality operator removes a constant factor independent of $\boldsymbol{x}$ and in (4.42) we defined

$$\boldsymbol{G} \triangleq \boldsymbol{\Psi}^H\boldsymbol{\Psi}\,. \tag{4.43}$$

According to the CS detector, (4.42) can be rewritten as

$$\tilde{p}\left(\boldsymbol{y}|\boldsymbol{x}\right) \propto \exp\left(-\frac{\boldsymbol{y}^H\boldsymbol{y}-2\Re\{\boldsymbol{y}^H\boldsymbol{\Psi}^r\boldsymbol{x}\}+\boldsymbol{x}^H\boldsymbol{G}^r\boldsymbol{x}}{N_r}\right), \tag{4.44}$$

where the mismatched noise density N_r and the matrices $\boldsymbol{\Psi}^r$ and $\boldsymbol{G}^r$ are subject to optimization. Neglecting the irrelevant terms for the detection process and including N_r into $\boldsymbol{\Psi}^r$ and $\boldsymbol{G}^r$, we can redefine $\tilde{p}\left(\boldsymbol{y}|\boldsymbol{x}\right)$ as

$$\tilde{p}\left(\boldsymbol{y}|\boldsymbol{x}\right) \triangleq \exp\left(2\Re\{\boldsymbol{x}^H(\boldsymbol{\Psi}^r)^H\boldsymbol{y}\}-\boldsymbol{x}^H\boldsymbol{G}^r\boldsymbol{x}\right)\,. \tag{4.45}$$

We can notice from (4.45) that the need of trellis processing is related uniquely to the matrix $\boldsymbol{G}^r$. In order to reduced the complexity, it is necessary to *shorten* this matrix, by constraining $\boldsymbol{G}^r$ to satisfy the following condition

$$\boldsymbol{G}^r(m,n) = 0 \quad \text{if } |m-n| > \lambda\,, \tag{4.46}$$

where λ is the selected length of the reduced channel response. Therefore, the number of required states for the algorithm working on the reduced trellis decreases to $|\mathcal{A}|^\lambda$, thus showing a complexity reduction factor of $|\mathcal{A}|^{(\Lambda-\lambda)}$, where typically $\Lambda \gg \lambda$, with respect to the MAP detector based on (4.42).

The optimal front-end-filter is derived in [70] and results to be

$$\boldsymbol{\Psi}^r = \left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + 2\sigma^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi}[\boldsymbol{G}^r + \mathbb{I}_{NL}]\,. \tag{4.47}$$

The resulting filter in (4.47) equals the MMSE filter (independent of the choice on λ) compensated by the trellis processing, which is represented through $\boldsymbol{G}^r + \mathbb{I}_{NL}$.

The steps for the computation of $\boldsymbol{G}^r$ can be summarized as follows.

1. Compute
$$\boldsymbol{B} \triangleq -\boldsymbol{\Psi}^H\left[\boldsymbol{\Psi}\boldsymbol{\Psi}^H + 2\sigma^2\mathbb{I}_{NL}\right]^{-1}\boldsymbol{\Psi} + \mathbb{I}_{NL}\,. \tag{4.48}$$

2. Define the submatrices $\tilde{\boldsymbol{B}}_k^\lambda$ of $\boldsymbol{B}$ as
$$\tilde{\boldsymbol{B}}_k^\lambda = \begin{bmatrix} B_{k+1,k+1} & \cdots & B_{k+1,\min(NL,k+\lambda)} \\ \vdots & \ddots & \vdots \\ B_{\min(NL,k+\lambda),k+1} & \cdots & B_{\min(NL,k+\lambda),\min(NL,k+\lambda)} \end{bmatrix}, \tag{4.49}$$
and the row vectors
$$\boldsymbol{b}_k^\lambda = [B_{k,k+1},\ldots,B_{k,\min(NL,k+\lambda)}]\,, \tag{4.50}$$
$$\boldsymbol{u}_k^\lambda = [u_{k,k+1},\ldots,u_{k,\min(NL,k+\lambda)}] \tag{4.51}$$
where $B_{m,n} = \boldsymbol{B}(m,n)$ and $u_{m,n} = \boldsymbol{U}(m,n)$ are the elements in the m-th row and n-th column of the matrices $\boldsymbol{B}$ and $\boldsymbol{U}$, respectively.

3. Compute the scalar terms

$$c_n = B_{n,n} - \boldsymbol{b}_n^{\lambda}(\tilde{\boldsymbol{B}}_n^{\lambda})^{-1}(\boldsymbol{b}_n^{\lambda})^H \,. \tag{4.52}$$

4. Construct the upper triangular matrix $\boldsymbol{U}$ from the row vectors

$$\boldsymbol{u}_n^{\lambda} = -u_{n,n}\boldsymbol{b}_n^{\lambda}(\tilde{\boldsymbol{B}}_n^{\lambda})^{-1} \tag{4.53}$$

   and the entries in the main diagonal, defined as

$$u_{n,n} = \frac{1}{\sqrt{c_n}} \,. \tag{4.54}$$

5. Finally, compute the optimal $\boldsymbol{G}^r$ as

$$\boldsymbol{G}^r = \boldsymbol{U}^H\boldsymbol{U} - \mathbb{I}_{NL} \,. \tag{4.55}$$

For a detailed derivation refer to [70].

As previously mentioned, the detector based on CS, as well as the MMSE equalizer, has a very good performance but typically a still too high complexity for some applications. In fact, even its implementation requires the inversion of an $NL \times NL$ matrix, thus a complexity proportional to $(NL)^3$. In a time-variant scenario, when the matrix $\boldsymbol{\Psi}$ changes at every block, for large values of the product NL, this complexity cannot be afforded.

4.4.3 Discrete MP

A. Matrix $\boldsymbol{G}$-based algorithm (MP$_{\boldsymbol{G}}$)

Among the MP algorithms working on FGs, an effective one, here referred to as MP$_{\boldsymbol{G}}$, is proposed in [67] and recently applied to OTFS in [68]. Differently from the algorithms that will be discussed in Section 4.4.4, it works on a processed version of $\boldsymbol{\Psi}$. In the following, the main algorithmic steps will be summarized. The factorization of the conditional pdf in (4.42) can be rewritten as

$$p\left(\boldsymbol{y}|\boldsymbol{x}\right) \propto \exp\left(\frac{2\Re\left\{\boldsymbol{x}^H\boldsymbol{z}\right\} - \boldsymbol{x}^H\boldsymbol{G}\boldsymbol{x}}{2\sigma^2}\right) \,, \tag{4.56}$$

where $\boldsymbol{z}$ is defined as

$$\boldsymbol{z} = \boldsymbol{\Psi}^H \boldsymbol{y}\,, \tag{4.57}$$

and the expression of $\boldsymbol{G}$ is the one of (4.43). From (4.56), it is possible to notice that $\boldsymbol{z}$ is a sufficient statistic for symbols detection. In fact, the operation in (4.57) corresponds to a matched filtering. The algorithm is thus developed on the Ungerboeck observation model since it performs the detection process operating on $\boldsymbol{G}$. Expressing in (4.56) the matrices products explicitly in terms of their elements, it is possible to factorize the conditional pmf in (4.14) as

$$P(\boldsymbol{x} \,|\, \boldsymbol{y}) \propto \prod_{k=1}^{NL} \left[P_k\left(x_k\right) F_k\left(x_k\right) \prod_{l<k} I_{k,l}\left(x_k, x_l\right) \right]\,, \tag{4.58}$$

which is based on the assumption that the modulation symbols x_k are treated by the detector as i.i.d. with given (typically uniform) *a-priori* pmfs $P_k(x_k)$, $k = 1, \ldots, NL$, and we defined

$$F_k\left(x_k\right) \triangleq \exp\left[\frac{1}{\sigma^2}\Re\left\{ z_k x_k^* - \frac{1}{2} G_{k,k} \left|x_k\right|^2 \right\}\right]\,, \tag{4.59}$$

$$I_{k,l}\left(x_k, x_l\right) \triangleq \exp\left[-\frac{1}{\sigma^2}\Re\left\{G_{k,l} x_l x_k^*\right\}\right]\,, \tag{4.60}$$

where $G_{k,l}$ denotes the element in the k-th row and l-th column of $\boldsymbol{G}$.

The corresponding FG is shown in Figure 4.5. The resulting MP$_{\boldsymbol{G}}$ algorithm consists in a direct application of the SPA to the obtained graph. The details of the resulting MP algorithm can be found in [67]. It can be summarized as follows. Define $V_k\left(x_k\right)$ as the product of all messages incoming to the variable node x_k, namely

$$V_k\left(x_k\right) = P_k\left(x_k\right) F_k\left(x_k\right) \prod_{j \neq k} \nu_{k,j}\left(x_k\right)\,, \tag{4.61}$$

which is proportional to the estimated *posterior* marginal distribution of x_k. Following the SPA rules

- the message $\mu_{k,j}$ transmitted from VN x_k to the FN $I_{k,j}$ is computed as

$$\mu_{k,j}\left(x_k\right) = V_k\left(x_k\right) / \nu_{k,j}\left(x_k\right)\,; \tag{4.62}$$

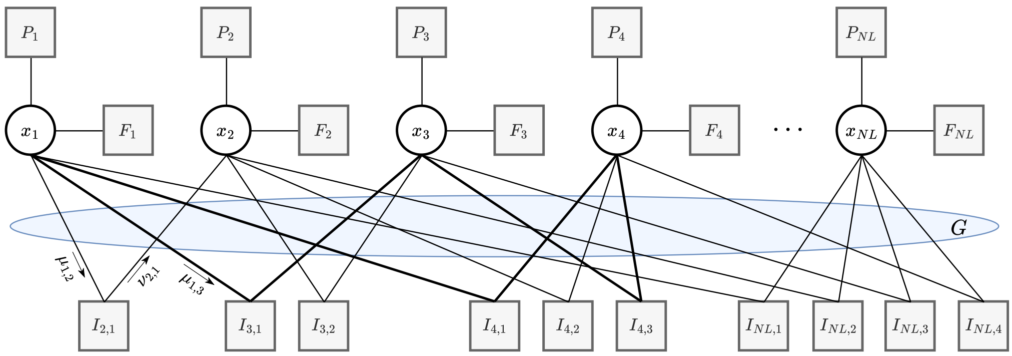


Figure 4.5: Structure of the FG for the Matrix $\boldsymbol{G}$-based algorithm.

- the message $\nu_{k,j}$ transmitted from FN $I_{k,j}$ to VN x_k can, instead, be expressed as

$$\nu_{k,j}\left(x_k\right) = \sum_{x_j \in \mathcal{A}} I_{k,j}\left(x_k, x_j\right) \mu_{k,j}\left(x_j\right) . \tag{4.63}$$

It is worth noting that changing the order in which messages are computed and propagated, i.e., the schedule, different performance can be obtained. The adopted schedules in the simulation results reported in Section 4.6 are the *Parallel Schedule (PS)* and the *Serial Schedule (SS)*. For details, refer to [67].

An important property of this approach is that, as highlighted in Figure 4.5, the FG has girth (i.e., minimum length cycles) 6. Even if it is known that the SPA generates exact marginal distributions when the graph is cycle-free, in loopy graphs with large girth the local neighborhood of each node is "tree-like" [5].

Moreover, the complexity of the MP$_{\boldsymbol{G}}$ algorithm is linear in the number of interferers per symbol Z [67] and quadratic in the constellation size M. In the simulation results, we selected the $S < Z$ terms bringing the highest amount of interference, where the value of S is set such that the complexity constraints are satisfied. The overall computational load is thus $\mathcal{O}(n_{iter}NLSM^2)$, where n_{iter} denotes the number of allowed inner detector iterations.

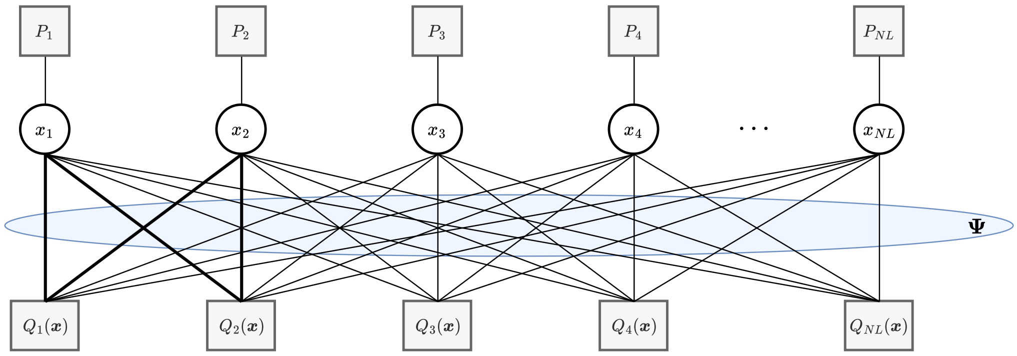


Figure 4.6: Structure of the FG for the Matrix $\boldsymbol{\Psi}$-based algorithm.

B. Matrix $\boldsymbol{\Psi}$-Based Algorithm (MP$_{\boldsymbol{\Psi}}$)

We will now focus on a different factorization of the posterior pdf which leads to a different FG structure. In fact, (4.14) can be written as

$$P(\boldsymbol{x} \mid \boldsymbol{y}) \propto \prod_{k=1}^{NL} p(y_k \mid \boldsymbol{x}) P_k(x_k) \,. \tag{4.64}$$

We define the FNs

$$Q_k(\boldsymbol{x}) \triangleq p(y_k \mid \boldsymbol{x}) = \exp\left(-\frac{\|y_k - \boldsymbol{\Psi}_k \boldsymbol{x}\|^2)}{2\sigma^2}\right), \tag{4.65}$$

where $\boldsymbol{\Psi}_k$ denotes the k-th row of the channel matrix $\boldsymbol{\Psi}$. The resulting FG is shown in Figure 4.6. This FG has girth equal to 4, as highlighted in Figure 4.6, therefore lower than that of [57] working on the FG of Figure 4.5. As directly follows from the FNs definition (4.65), the j-th VN is connected to a number of FNs equal to the number of nonzero elements in the j-th column of $\boldsymbol{\Psi}$ and the k-th FN is connected to all the symbols corresponding to the positions of the nonzero elements of the k-th row of $\boldsymbol{\Psi}$. Since in the general case of delay and Doppler shifts with a fractional component the channel matrix is (almost) full, the FG presents an extremely high number of edges. Denoting with l_k

the number of nonzero elements in the k-th row of $\boldsymbol{\Psi}$ which is, therefore, the degree of the FN $Q_k(\boldsymbol{x})$, the exact application of the SPA would require to sum over $l_k - 1$ discrete variable taking M possible values in $\mathcal{A}$. Thus, the computational load is M^{l_k-1} which, especially for large constellations and/or real channel conditions, is highly impractical. In fact, the message sent from FN $Q_k(\boldsymbol{x})$ to VN x_j is

$$\mu_{Q_k \to x_j}(x_j) = \sum_{\sim x_j} Q_k(\boldsymbol{x}) \prod_{l \in n(k) \setminus \{j\}} \mu_{x_l \to Q_k}(x_l)\,, \tag{4.66}$$

where $n(k)$ is the set of VNs connected to FN $Q_k(\boldsymbol{x})$. Instead, the message transmitted from VN x_j to FN Q_k, applying the SPA rules, is

$$\mu_{x_j \to Q_k}(x_j) = \prod_{l \in m(j) \setminus \{k\}} \mu_{Q_l \to x_j}(x_j)\,, \tag{4.67}$$

where $m(j)$ is the set of FNs connected to the VN x_j. Due to the impractical complexity, several solutions have been proposed to simplify the exchanged messages. Among the approximating algorithms working on the FG shown in Figure 4.6, one of the most successful ones is that proposed in [57]. We will denote this algorithm as $\mathrm{MP}_{\boldsymbol{\Psi}}$. In [57], the authors propose to adopt a Gaussian approximation for the interfering symbols in the computation at the FN $Q_k(\boldsymbol{x})$, which decreases the computational load. In fact, the approach proposed in [57] relies on the following approximation. When computing the message from FN $Q_k(\boldsymbol{x})$ to VN x_j, the k-th received sample, which can be expressed as

$$y_k = \psi_{k,j} x_j + \sum_{l \in n(k) \setminus \{j\}} \psi_{k,l} x_l + w_k \tag{4.68}$$

where w_k is the k-th component of the AWGN vector $\boldsymbol{w}$, $\psi_{k,j}$ represents the element in the k-th row and j-th column of $\boldsymbol{\Psi}$ and $n(k)$ is the set of VNs connected to FN $Q_k(\boldsymbol{x})$, is assumed to be

$$\tilde{y}_k = \psi_{k,j} x_j + z_{k \to j} \tag{4.69}$$

where $z_{k\to j}$ is the approximation of the interference term, modeled as a Gaussian random variable. It follows that

$$z_{k\to j} \sim \mathcal{N}_{\mathbb{C}}(\eta_{k,j}^{(n)}, \nu_{k,j}^{(n)})\,, \tag{4.70}$$

where n denotes the iteration number and

$$\begin{aligned}
\eta_{k,j}^{(n)} &= \mathbb{E}\left[z_{k\to j}\right] && (4.71)\\
&= \mathbb{E}\left[\sum_{l\in n(k)\setminus\{j\}} \psi_{k,l}x_l + w_k\right] && (4.72)\\
&= \sum_{l\in n(k)\setminus\{j\}} \psi_{k,l}\mathbb{E}\left[x_l\right] && (4.73)\\
&= \sum_{l\in n(k)\setminus\{j\}} \psi_{k,l}\sum_{s=1}^{M}\mu_{x_l\to Q_k}^{(n-1)}(x_l = a_s)a_s && (4.74)
\end{aligned}$$

where we recall that $a_s \in \mathcal{A}$, $s = 1,\ldots,M$, are the constellation symbols, and

$$\begin{aligned}
\nu_{k,j}^{(n)} &= \mathrm{Var}\left[\sum_{l\in n(k)\setminus\{j\}} \psi_{k,l}x_l + w_k\right] && (4.75)\\
&= \sum_{l\in n(k)\setminus\{j\}} \left|\psi_{k,l}\right|^2 \mathrm{Var}\left\{x_l\right\} + 2\sigma^2 && (4.76)
\end{aligned}$$

where

$$\mathrm{Var}\left\{x_l\right\} = \sum_{s=1}^{M}\mu_{x_l\to Q_k}^{(n-1)}(x_l = a_s)|a_s|^2 - \left|\sum_{s=1}^{M}\mu_{x_l\to Q_k}^{(n-1)}(x_l = a_s)a_s\right|^2. \tag{4.77}$$

Therefore, the message in (4.66) is replaced by

$$\mu_{Q_k\to x_j}^{(n)}(x_j = a_s) \propto \exp\left\{-\frac{\left|y_k - \psi_{k,j}a_s - \eta_{k,j}^{(n)}\right|^2}{\nu_{k,j}^{(n)}}\right\}. \tag{4.78}$$

Then, at the VN x_j the corresponding *partial* pmf to be sent to FN Q_k is computed by substituting in (4.67) the just derived approximated messages from FNs Q_l with $l \in m(j)\setminus\{k\}$ to VN x_j.

In this way, a significant performance reduction can be obtained, achieving an overall complexity of $\mathcal{O}(n_{iter}NLSM)$, where S denotes the number of elements of each row and column of $\mathbf{\Psi}$ which are considered for the detection process. It is worth specifying that, in order to increase the sparsity of $\mathbf{\Psi}$ and, therefore, decrease the number of edges in the graph of Figure 4.6, the detector proposed in [57] considers the delay shifts as integer multiples of the grid resolution. For this reason, the number S of nonzero elements per row/column is limited and much smaller than the channel matrix dimension. However, the above assumption is generally not satisfied by real channels and the matrix $\mathbf{\Psi}$ is typically full. Moreover, making the detector work on an approximated channel matrix constructed by rounding the delay shifts to integer multiples of the grid leads to a significant performance degradation. Therefore, in the simulation results presented in Section 4.6, we make the $\text{MP}_{\mathbf{\Psi}}$ detector work on the exact matrix $\mathbf{\Psi}$ and we set the value of S according to the complexity constraints and the choice of the interfering terms per symbol is done by selecting the most significant entries in terms of energy.

4.4.4 Continuous MP

In [51], other possible approximated MP approaches are described, aiming to a complexity reduction of the exact SPA applied to the FG in Figure 4.6. In particular, the algorithms discussed in this section are derived based on the assumption that the transmitted symbols x_k, for $k = 1, \ldots, NL$, are continuous RVs, characterized by a complex Gaussian distribution. For this reason, we refer to them as *continuous* MP algorithms.

A. Approximate MP using Gaussian Approximation (AMP-GA)

Differently from the approach of [57], outlined in Section 4.4.3.B, the AMP-GA algorithm solves the complexity problem of (4.66) and (4.67) introducing an approximation on the VNs: the discrete pmf of symbols is substituted with a Gaussian distribution so that only its mean and variance have to be computed and propagated through the graph.

According to this approach, each discrete M-dimensional message, sent from a VN to a FN is approximated by the Gaussian pdf minimizing the KL divergence. The parameters characterizing the Gaussian distribution are, therefore, obtained through *moment matching*. Denoting with $\mu^{(n)}_{x_l \to Q_k}$ the discrete message sent from VN x_j to FN Q_k at the n-th inner detector iteration and with $\eta^{(n)}_{x_j \to Q_k}$ and $\nu^{(n)}_{x_j \to Q_k}$ the mean and variance of the approximated message, respectively, it follows that

$$\eta^{(n)}_{x_j \to Q_k} = \sum_{x_j \in \mathcal{A}} x_j \mu^{(n)}_{x_j \to Q_k}(x_j) \tag{4.79}$$

$$\nu^{(n)}_{x_j \to Q_k} = \sum_{x_j \in \mathcal{A}} |x_j|^2 \mu^{(n)}_{x_j \to Q_k}(x_j) - |\eta^{(n)}_{x_j \to Q_k}|^2 \,. \tag{4.80}$$

Substituting the discrete message $\mu_{x_l \to Q_k}(x_l)$ in (4.66) with a Gaussian distribution and considering the VNs x_l, $l \in n(k) \backslash \{j\}$ as continuous RVs, we obtain

$$\mu^{(n)}_{Q_k \to x_j}(x_j) = \int_{\sim x_j} Q_k(\boldsymbol{x}) \prod_{l \in n(k) \backslash \{j\}} g_{\mathbb{C}}(x_l, \eta^{(n)}_{x_l \to Q_k}, \nu^{(n)}_{x_l \to Q_k}) \tag{4.81}$$

$$\propto g_{\mathbb{C}}(\psi_{k,j} x_j, \eta^{(n)}_{Q_k \to x_j}, \nu^{(n)}_{Q_k \to x_j}) \tag{4.82}$$

where

$$\eta^{(n)}_{Q_k \to x_j} = y_k - \sum_{l \in n(k) \backslash \{j\}} \psi_{k,l} \eta^{(n)}_{x_l \to Q_k} \,, \tag{4.83}$$

$$\nu^{(n)}_{Q_k \to x_j} = 2\sigma^2 + \sum_{l \in n(k) \backslash \{j\}} |\psi_{k,l}|^2 \nu^{(n)}_{x_l \to Q_k} \,. \tag{4.84}$$

Finally, (4.67) becomes

$$\mu^{(n)}_{x_j \to Q_k}(x_j = a_s) = \frac{g_{\mathbb{C}}\left(x_j = a_s, \zeta^{(n)}_{x_j \to Q_k}, \gamma^{(n)}_{x_j \to Q_k}\right)}{\sum_{m=1}^{M} g_{\mathbb{C}}\left(a_m, \zeta^{(n)}_{x_j \to Q_k}, \gamma^{(n)}_{x_j \to Q_k}\right)} \,, \quad a_s \in \mathcal{A} \tag{4.85}$$

where

$$\gamma_{x_j \to Q_k}^{(n)} = \left(\sum_{l \in m(j) \setminus \{k\}} \frac{|\psi_{l,j}|^2}{\nu_{Q_l \to x_j}^{(n-1)}} \right)^{-1}, \tag{4.86}$$

$$\zeta_{x_j \to Q_k}^{(n)} = \gamma_{x_j \to Q_k}^{(n)} \sum_{l \in m(j) \setminus \{k\}} \frac{\psi_{l,j}^* \eta_{Q_l \to x_j}^{(n-1)}}{\nu_{Q_l \to x_j}^{(n-1)}}. \tag{4.87}$$

The discrete distribution computed in (4.85) is then used in (4.79)-(4.80) to find the mean and variance of the approximated Gaussian message transmitted from VN x_j to FN Q_k. The symbols marginal distribution is then computed through the SPA rules by multiplying all the incoming messages in each VN.

B. Approximate MP using EP (AMP-EP)

The EP framework can be used to further reduce the complexity of the AMP-GA approach, outlined in Section 4.4.4.A. In particular, the simplification concerns the message in (4.85). In fact, the symbol belief

$$\beta^{(n)}(x_j = a_s) \triangleq \frac{\prod_{k \in m(j)} \mu_{Q_k \to x_j}^{(n-1)}(x_j = a_s)}{\sum_{m=1}^{M} \prod_{k \in m(j)} \mu_{Q_k \to x_j}^{(n-1)}(x_j = a_m)}, \quad a_s \in \mathcal{A}, \tag{4.88}$$

where the M discrete values of $\mu_{Q_k \to x_j}^{(n-1)}(x_j)$ are computed from (4.82) as

$$\mu_{Q_k \to x_j}^{(n-1)}(x_j = a_s) \propto \exp \left\{ -\frac{\left| \psi_{k,j} a_s - \eta_{Q_k \to x_j}^{(n-1)} \right|^2}{\nu_{Q_k \to x_j}^{(n-1)}} \right\}, \tag{4.89}$$

is replaced by a Gaussian pdf $\hat{\beta}^{(n)}(x_j) = g_{\mathbb{C}}(x_j, \hat{\eta}_{x_j}^{(n)}, \hat{\tau}_{x_j}^{(n)})$, instead of approximating the message $\mu_{x_j \to Q_k}(x_j)$. From the approximate symbol belief, it is, then, possible to compute the approximate message $\hat{\mu}_{x_j \to Q_k}(x_j)$. The parameters $\hat{\eta}_{x_j}^{(n)}$ and $\hat{\tau}_{x_j}^{(n)}$ of $\hat{\beta}^{(n)}(x_j)$ can be computed by minimizing the KL divergence among the exact and the approximated Gaussian distribution, therefore, exploiting moment matching technique, we obtain

$$\hat{\eta}_{x_j}^{(n)} = \mathbb{E}_{\beta^{(n)}(x_j)}[x_j]\,, \tag{4.90}$$

$$\hat{\tau}_{x_j}^{(n)} = \mathbb{E}_{\beta^{(n)}(x_j)}[|x_j|^2] - |\hat{\eta}_{x_j}^{(n)}|^2\,. \tag{4.91}$$

Finally, the parameters of the message from VN to FN directly derive from the ratio among two Gaussian pdfs, i.e.,

$$\hat{\tau}_{x_j\to Q_k}^{(n)} = \left(\frac{1}{\hat{\tau}_{x_j}^{(n)}} - \frac{|\psi_{k,j}|^2}{\nu_{Q_k\to x_j}^{(n-1)}}\right)^{-1}\,, \tag{4.92}$$

$$\hat{\eta}_{x_j\to Q_k}^{(n)} = \hat{\tau}_{x_j\to Q_k}^{(n)}\left(\frac{\hat{\eta}_{x_j}^{(n)}}{\hat{\tau}_{x_j}^{(n)}} - \frac{\psi_{k,j}^{*}\eta_{Q_k\to x_j}^{(n-1)}}{\nu_{Q_k\to x_j}^{(n-1)}}\right)\,. \tag{4.93}$$

4.5 The Proposed Algorithm

In this section, we propose a new low-complexity MP algorithm which belongs to the family of *discrete* MP and, similarly to the MP$_{\boldsymbol{G}}$ approach discussed in Section 4.4.3.A, works on the processed channel matrix $\boldsymbol{G}$ defined in (4.43). Differently from the existing solutions in the OTFS literature (see [51, 57, 68] and references therein), this algorithm takes advantage of the specific structure assumed by the matrix $\boldsymbol{G}$ which is analyzed in Section 4.5.1. The main algorithmic steps are then presented in Section 4.5.2.

4.5.1 Considerations on the channel matrix

The matrix $\boldsymbol{G}$ is characterized by a particular structure. In fact, it is

- *sparse*, as it represents a processed version of the channel matrix $\boldsymbol{\Psi}$ in the Doppler-delay domain;
- arranged in *diagonals*: its entries (i.e., the coefficients responsible for determining the amount of interference) are organized in a relatively ordered way. Specifically, the most relevant ones, i.e., the terms with the

largest magnitudes, are located along diagonals. Each diagonal consists of elements with comparable energy, with only a few exceptions. This structure is also partially present in $\boldsymbol{\Psi}$, though to a lesser extent.

Specifically, from (4.43) and (4.13), we can write $\boldsymbol{G}$ as

$$\boldsymbol{G} = \left(\sum_{p=1}^{P} h'_p \boldsymbol{\Psi}_p\right)^H \left(\sum_{t=1}^{P} h'_t \boldsymbol{\Psi}_t\right) \tag{4.94}$$

$$= \underbrace{\sum_{p=1}^{P} \left|h'_p\right|^2 \boldsymbol{\Psi}_p^H \boldsymbol{\Psi}_p}_{\boxed{1}\,:\,\text{Main Diagonal}} + \underbrace{\sum_{p=1}^{P} \sum_{t \neq p} (h'_p)^* h'_t \boldsymbol{\Psi}_p^H \boldsymbol{\Psi}_t}_{\boxed{2}\,:\,\text{Subdiagonals}} \,. \tag{4.95}$$

The first term in (4.95) identifies the main diagonal. In fact, it can be demonstrated that

$$\boldsymbol{\Psi}_p^H \boldsymbol{\Psi}_p = \mathbb{I}_{NL}\,, \quad \forall p = 1, \ldots, P\,. \tag{4.96}$$

Moreover, equation (4.96) is verified for any pair (ν_p, τ_p), no matter if they are or not integer multiples of the grid spacings. The second term in (4.95) is the sum of $P(P-1)$ matrices which give rise to the subdiagonals.

In the special case where τ_p and ν_p are integer multiples of the symbol grid, the matrix $\boldsymbol{\Psi}_p$ associated with the path p has one nonzero element in each row and column, all placed along (typically incomplete) diagonals. Since the product $\boldsymbol{\Psi}_p^H \boldsymbol{\Psi}_t$ in (4.95) shifts the diagonal positions without altering the number of elements per row/column, $\boldsymbol{G}$ results to be composed by $P(P-1)+1$ nonzero elements in each row and column. From (4.13), it is clear that, under these unrealistic assumptions, $\boldsymbol{\Psi}$ is sparser than $\boldsymbol{G}$, as it contains only P nonzero entries per row and column.

However, in the general case of real channels where (ν_p, τ_p) are not integer multiples of the grid granularity, the above considerations about the precise matrix structure are not valid anymore. In fact, the energy is not strictly *confined* along diagonals but rather spreads to positions near the peak value.

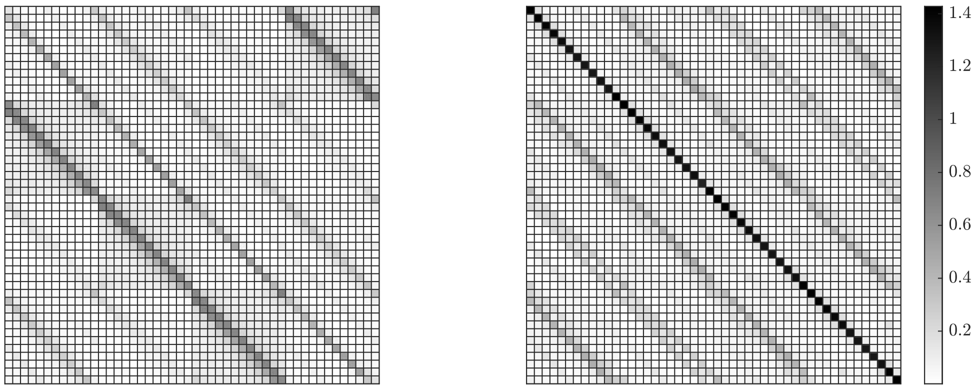


Figure 4.7: Comparison of the energy distribution for the matrix $\boldsymbol{\Psi}$ (on the left) and $\boldsymbol{G}$ (on the right).

Therefore, working on $\boldsymbol{G}$ can be beneficial because the product of $\boldsymbol{\Psi}$ with its Hermitian allows to obtain a more defined and compact matrix structure. An example of this behavior can be seen in Figure 4.7 where, on the left we reported the structure of the channel matrix $\boldsymbol{\Psi}$ by showing a map of the energy distribution and on the right that of the related matrix $\boldsymbol{G}$.

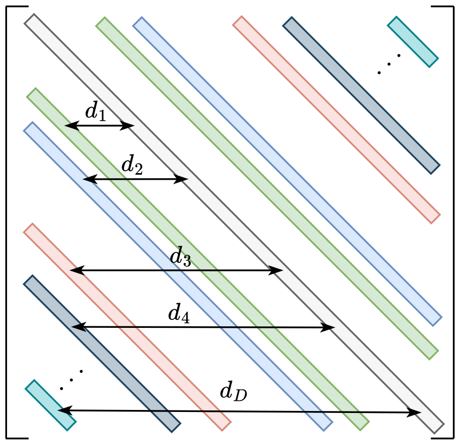
4.5.2 Algorithm structure

The proposed algorithm thus takes advantage of the structure assumed by the matrix $\boldsymbol{G}$ and can be summarized as follows. Firstly, the energy of each diagonal is computed as

$$E_j = \sum_{k=1}^{NL-j} |G_{j+k,k}|\,, \quad j = 1, \ldots, NL-1\,. \tag{4.97}$$

Notice that, since $\boldsymbol{G}$ is Hermitian, considering the upper triangular submatrix would yield the same result. Then, select the first D diagonals with the highest energy to be used in the detection process, where D is chosen in accordance to the complexity constraints or/and optimized in order to achieve the best trade-off between complexity and performance. The greatest D energy values are sorted in decreasing order and collected in the vector $\overline{\boldsymbol{E}}$. The detection algorithm operates on the reduced Hermitian matrix $\boldsymbol{G}^r$, which is composed by D nonzero diagonals in the lower triangular submatrix and the corresponding D in the upper one, and has the following structure:

$$\boldsymbol{G}^r = \tag{4.98}$$



where $d_i\,, i = 1, \ldots, D$ is the distance in terms of number of elements between the main diagonal and the i-th subdiagonal. Analytically,

$$G^r_{m,n} = \begin{cases} G_{m,n}, & \text{if } m = n \,\vee\, |m-n| \in \{d_i\}_{i=1}^{D} \\ 0, & \text{elsewhere} \end{cases} \tag{4.99}$$

where $G^r_{m,n}$ denotes the element in the m-th row and n-th column of $\boldsymbol{G}^r$. It is worth to underline that a longer distance from the main diagonal does not imply a reduction in the magnitude of its entries. With reference to the conditional pdf in (4.56), by substituting $\boldsymbol{G}^r$ in place of $\boldsymbol{G}$ and explicitly expressing the matrix operations in terms of their components, we obtain

$$\boldsymbol{x}^H \boldsymbol{z} = \sum_{k=1}^{NL} z_k x_k^* \,, \tag{4.100}$$

$$\begin{aligned}
\boldsymbol{x}^H \boldsymbol{G}^r \boldsymbol{x} &= \sum_{k=1}^{NL} G^r_{k,k} |x_k|^2 + \sum_{k=1}^{NL} x_k^* \sum_{l \neq k} G^r_{k,l} x_l \qquad (4.101)\\
&= \sum_{k=1}^{NL} G_{k,k} |x_k|^2 + \sum_{k=1}^{NL} \sum_{j=1}^{D} 2\Re\{G_{k,k-d_j} x_{k-d_j} x_k^*\} \,. \qquad (4.102)
\end{aligned}$$

Therefore, we can rewrite the approximate *a-posteriori* pmf as

$$\tilde{P}(\boldsymbol{x}|\boldsymbol{y}) \propto \prod_{k=1}^{NL} \left[P_k(x_k) F_k(x_k) \prod_{j=1}^{D} I_{k,k-d_j}(x_k, x_{k-d_j}) \right] \,, \tag{4.103}$$

where the factor $F_k(x_k)$ is defined in (4.59) and

$$I_{k,k-d_j}(x_k, x_{k-d_j}) \triangleq \exp\left(-\frac{1}{\sigma^2} \Re\left\{ G_{k,k-d_j} x_{k-d_j} x_k^* \right\} \right) \,. \tag{4.104}$$

Considering the generic VN x_k, a portion of the FG corresponding to (4.103) is shown in Figure 4.8.

Each couple of diagonals in the matrix $\boldsymbol{G}^r$ gives rise to a set of d_j parallel branches implementing independently the BCJR algorithm with unitary memory. In fact, each BCJR involves NL/d_j symbols and takes as input information a partial marginal distribution $T_k^{(d_j)}(x_k)$ defined as

$$T_k^{(d_j)}(x_k) = P_k(x_k) F_k(x_k) \prod_{i \neq j} \underbrace{\alpha_{in}^{(d_i)}(x_k) \beta_{in}^{(d_i)}(x_k)}_{O_k^{(d_i)}(x_k)} \tag{4.105}$$

where $\alpha_{in}^{(d_i)}(x_k)$ and $\beta_{in}^{(d_i)}(x_k)$ are the incoming messages to the VN x_k in Figure 4.8, and their product, $O_k^{(d_i)}(x_k)$, represents the output information about the k-th symbol generated by the BCJR corresponding to the diagonal at distance d_j. Notice that in (4.105), the messages $\alpha_{in}^{(d_i)}(x_k)$ and $\beta_{in}^{(d_i)}(x_k)$

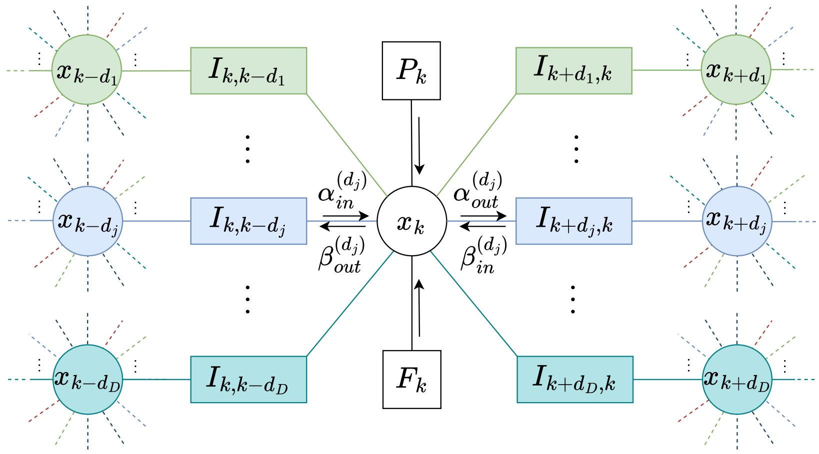


Figure 4.8: Partial structure of the FG for the proposed algorithm.

correspond to their most recent updates and, in order to account for the extrinsic information on the VN x_k, we exclude from the product those messages that refer to the considered j-th BCJR. For easier visualization, we report in Figure 4.9 a section of the FG of Figure 4.8, partially reordered for the sake of readability, where the incoming and outgoing messages related to VN x_k are highlighted. With reference to the messages in Figure 4.9, by applying the SPA updating rules, $\alpha_{in}^{(d_j)}$, $\alpha_{out}^{(d_j)}$, $\beta_{in}^{(d_j)}$ and $\beta_{out}^{(d_j)}$ can be recursively computed by means of the following forward and backward recursions:

$$\alpha_{in}^{(d_j)}(x_k) = \sum_{\sim\{x_k\}} \alpha_{out}^{(d_j)}(x_{k-d_j}) I_{k,k-d_j}(x_k, x_{k-d_j}) \tag{4.106}$$

$$\alpha_{out}^{(d_j)}(x_k) = \alpha_{in}^{(d_j)}(x_k) T_k^{(d_j)}(x_k) \tag{4.107}$$

$$\beta_{in}^{(d_j)}(x_k) = \sum_{\sim\{x_k\}} \beta_{out}^{(d_j)}(x_{k+d_j}) I_{k,k+d_j}(x_{k+d_j}, x_k) \tag{4.108}$$

$$\beta_{out}^{(d_j)}(x_k) = \beta_{in}^{(d_j)}(x_k) T_k^{(d_j)}(x_k)\,. \tag{4.109}$$

Concerning the scheduling, the set of BCJR algorithms that is *activated* first is the one involving the largest interference, i.e., $\overline{E}_M = \max(\overline{\boldsymbol{E}})$ (placed

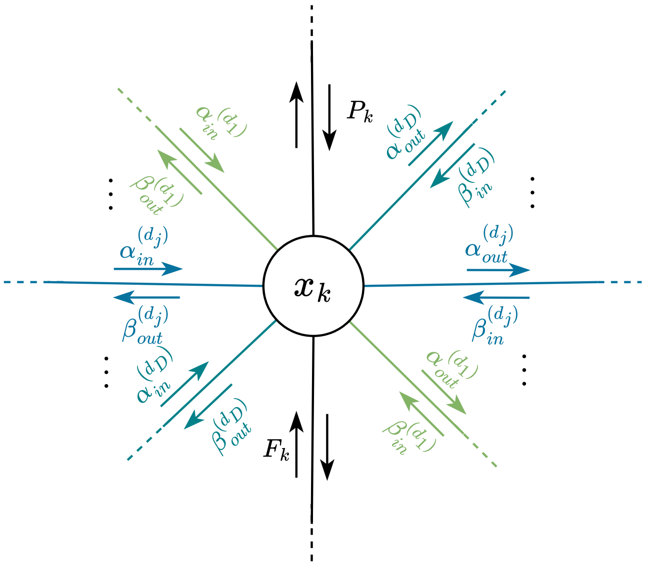


Figure 4.9: Overview of the outgoing and incoming messages to the VN x_k.

at distance $\overline{d}_M$ from the main diagonal), with input information

$$T_k^{(\overline{d}_M)}(x_k) = P_k(x_k)F_k(x_k)\,.$$

Then, we proceed following the decreasing energy order, i.e., the one dictated by the entries of $\overline{\boldsymbol{E}}$. This way, we are prioritizing the strongest diagonals of $\boldsymbol{G}^r$.

The proposed algorithm benefits from inner iterations. However, it achieves convergence very quickly: typically, just 3 iterations result to be sufficient. After the allowed iterations have been performed, the final estimate for the modulation symbols is computed as

$$P(x_k \,|\, \boldsymbol{y}) \propto P_k(x_k)F_k(x_k)\prod_{i=1}^{D}\alpha_{in}^{(d_i)}(x_k)\beta^{(d_i)}(x_k)\,. \tag{4.110}$$

Similarly to the graph of the MP$_{\boldsymbol{G}}$ [67, 68], the FG on which the proposed algorithm operates has girth 6. However, although they are quite long, a known issue of the SPA, as applied to FGs with cycles, is the overestimation of the reliability of the propagated messages. A possible solution is the adoption in (4.59) and (4.104) of a value of σ^2 greater than the exact one. The effectiveness of this algorithm comes from its ability to leverage the matrix structure and operate along the diagonals, allowing for different optimizations of σ^2 based on the specific diagonal being considered. In particular, we found out that it is sufficient to optimize only two variances: one for the first BCJR set and another for the remaining $D-1$. The complexity of the proposed approach is $\mathcal{O}(n_{iter}NLSM^2)$, where S is the number of interferers per symbol and depends linearly on the selected value D of diagonals.

It is possible to demonstrate that, being derived from the same channel matrix $\boldsymbol{G}$, the FG of the proposed algorithm can be traced back to that of the MP$_{\boldsymbol{G}}$. However, the primary distinction between the two algorithms lies in their scheduling strategies. In fact, as will be shown in Section 4.6, adopting the proposed operating criterion allows to achieve a superior performance and faster convergence in all the analyzed scenarios.

4.6 Performance Analysis

We tested the performance of the proposed algorithm and, for comparison purposes, we examined (i) the MMSE detector, presented in Section 4.4.1 and used as a benchmark and, among the graph-based strategies, we selected (ii) the MP$_{\boldsymbol{G}}$ (see Section 4.4.3.A), since it operates on the same channel matrix of the proposed algorithm, and (iii) the MP$_{\boldsymbol{\Psi}}$, discussed in Section 4.4.3.B, as a low-complexity alternative. Moreover, for the sake of completeness, in Section 4.6.1, the performance of the AMP-GA and EP algorithms, outlined in Section 4.4.4, is also shown. We tested the different receivers considering a fixed constraint on the computational load. The MP detectors perform a maximum of 3 inner iterations and the selection of the interfering terms for the detection process is based on their energy as discussed in Section 4.4. A simplified complexity analysis of the graph-based detectors is reported in Table 4.1, where

	Two-terms additions	Two-terms multiplications	LUT accesses
$\boldsymbol{G}$-based	$4M^2S + 2MS + 5M$	$\frac{7}{2}M^2S + 6M$	M^2S
$\text{MP}_{\boldsymbol{\Psi}}$	$8MS + 8S - M - 1$	$8MS + 6S$	MS
AMP-GA	$7MS + 19S + 3M - 6$	$6MS + 19S + 3M$	MS
AMP-EP	$4MS + 19S + 6M - 5$	$4MS + 25S + 7M + 2$	M

Table 4.1: Complexity comparison per code symbol per iteration for the tested graph-based detectors.

we differentiated between two (real and scalar) terms multiplications, additions and accesses to the LUT. We recall that the number of considered interferers per symbol is denoted with S. The MMSE detector is allowed to operate at full complexity as it is used only for reference but it cannot be practically implemented in most cases. Moreover, in all the simulated scenarios, we assumed QPSK modulation for the information symbols if not otherwise stated, and perfect CSI at the receiver. In Section 4.6.1, we focus on the performance analysis within standard 3GPP multipath channel profiles, while Section 4.6.2 is dedicated to the multi-satellite scenario.

4.6.1 Multipath terrestrial channel

In this section, we evaluate the pragmatic capacity of the analyzed detectors, measured in bit/s/Hz, versus the received SNR, with reference to the Extended Pedestrian A (EPA), Extended Vehicular A (EVA), and Extended Typical Urban (ETU) standard 3GPP channel profiles [66]. The channel characteristics are specified through the *power delay profile* and maximum Doppler spread of the considered propagation environment. Specifically, the power delay profiles are reported in Tables 4.2, 4.3, 4.4, and they represent low, medium and high delay spread environments, respectively. The path delays $\{\tau_p\}_{p=1}^{P}$ are specified in the first row of the tables, while the channel gains $\{h_p\}_{p=1}^{P}$ are derived from the power profile in the second row and each of them is assumed to be char-

Excess tap delays (ns)	0	30	70	90	110	190	410
Relative power (dB)	0.0	−1.0	−2.0	−3.0	−8.0	−17.2	−20.8

Table 4.2: EPA model.

Excess tap delays (ns)	0	30	150	310	370	710	1090	1730	2510
Relative power (dB)	0.0	−1.5	−1.4	−3.6	−0.6	−9.1	−7.0	−12.0	−16.9

Table 4.3: EVA model.

acterized by a random, uniformly distributed, channel phase. The maximum Doppler shift ν_{max} of the overall multipath channel directly depends on the the maximum speed u_{max} of the receiver, according to

$$\nu_{max} = \frac{u_{max} f_c}{c}, \tag{4.111}$$

where f_c is the carrier frequency and c is the speed of light. In particular, the value of u_{max} is selected depending on the simulated channel model, e.g., for the EVA model, the maximum UT speed is set to 100 Kmph. Moreover, we assumed that a single Doppler shift is associated with the p-th delay path and is generated using the Jakes' formula

$$\nu_p = \nu_{max} \cos(\theta_p), \quad p = 1, \dots, P, \tag{4.112}$$

where θ_p is uniformly distributed in $[-\pi, \pi]$. The other relevant simulation parameters are given in Table 4.5.

Excess tap delays (ns)	0	50	120	200	230	500	1600	2300	5000
Relative power (dB)	−1.0	−1.0	−1.0	0.0	0.0	0.0	−3.0	−5.0	−7.0

Table 4.4: ETU model.

Parameter	Value
Carrier frequency (f_c)	4 GHz
Nº of subcarriers (L)	64
Nº of OTFS symbols (N)	16
Subcarrier spacing (Δ_f)	15 kHz
Symbol duration (T)	66.7 µs
Modulation alphabet	QPSK, 16QAM
UT speed (u_{max})	5, 100, 300 Kmph
Channel estimation	Ideal

Table 4.5: Simulation parameters employed for the terrestrial channel.

In Figures 4.10, 4.11 and 4.13, we tested the analyzed detectors employing the QPSK modulation format, whereas in Figure 4.12 we analyzed the 16QAM. As mentioned in Section 4.5.2, MP algorithms working on graph with cycles typically require an optimization of the value of the AWGN variance σ^2. For the proposed solution, as well as for the other graph-based algorithms, the optimal value depends both on the simulated SNR and on the channel parameters.

Figure 4.10 shows the pragmatic capacity comparison when the channel follows the EPA model. According to this scenario, there are $P = 7$ received components. Moreover, it is characterized by a low delay spread, a reduced maximum Doppler shift corresponding to $u_{max} = 5$ Kmph, and a sharply decreasing power profile as the arrival time increases. It can be observed that, in this situation, all the detection algorithms perform identically and exhibit a poor performance at high SNRs.

This behavior could be explained as follows. Considering a maximum UT speed of 5 Kmph, all the received components will be characterized by Doppler shifts always lower than the Doppler resolution, i.e., $1/(NT) = \Delta_f/N \simeq 937$ Hz, based on the system parameters reported in Table 4.5. The same consideration also holds for the delays, which are on the order of nanoseconds (see

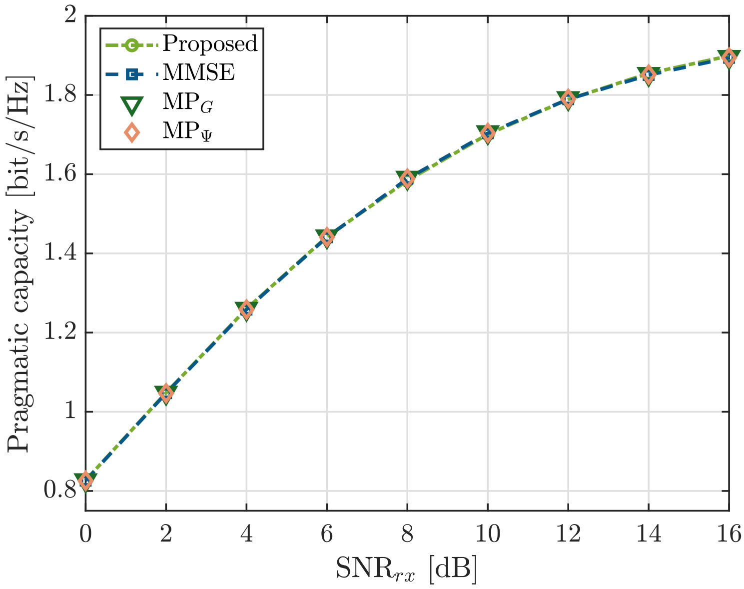


Figure 4.10: Performance comparison in terms of pragmatic capacity for QPSK modulation alphabet using the EPA channel model.

Table 4.2) and, therefore, correspond to a fraction of the delay resolution, i.e., $1/(L\Delta_f) = 1.0417\,\mu\text{s}$. For this reason, depending on the random channel phase, the overlap of multiple received components may generate constructive interference, resulting in a (processed) channel matrix $\boldsymbol{G}$ that is nearly diagonal. In this case, a symbol-by-symbol detector could achieve satisfactory performance. However, destructive interference may occur, causing the useful signals from different replicas to cancel each other out. In such cases, even more advanced detectors are unable to operate efficiently due to the lack of enough information for the detection process. This explains the poor performance of the tested receivers. Additionally, this simulation results suggest that depending on the environment dynamics, different system settings could be more suitable. For instance, still referring the EPA model, it has been observed that increasing the Doppler resolution, i.e., decreasing the subcarrier spacing and/or increasing

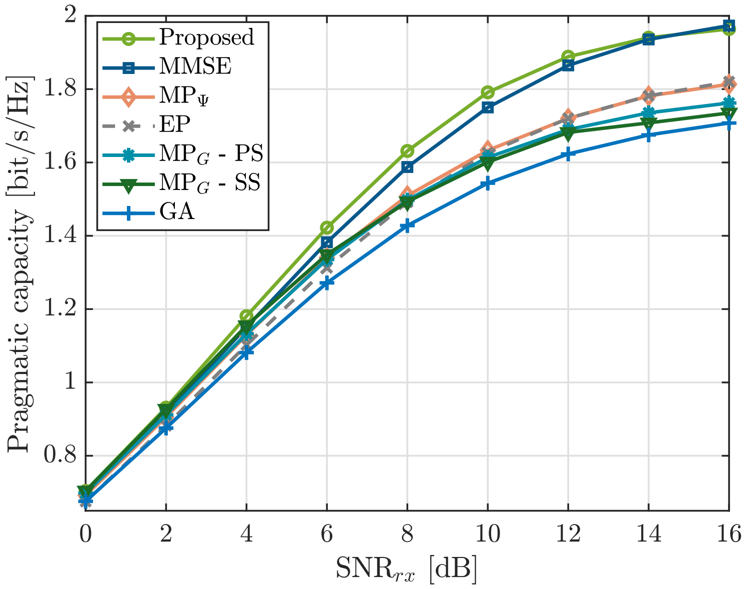


Figure 4.11: Performance comparison in terms of pragmatic capacity for QPSK modulation alphabet using the EVA channel model.

the number of OTFS symbols in the same block, can enhance performance.

In contrast, considering the EVA model, different performance is achieved by the detectors, as shown in Figure 4.11. In this case, $P = 9$ multipath components reach the receiver generating a medium delay spread and the maximum speed is assumed to be $u_{max} = 100\,\text{Kmph}$. As can be noticed, the proposed solution provides the best result. In fact, it outperforms the MMSE benchmark which, however, shows a little penalty. On the contrary, a weaker behavior can be observed for the other graph-based algorithms, which are tested at equal complexity. In fact, they saturate before reaching the highest values of pragmatic capacity because they are unable to handle the significant ISI. Figure 4.11 also shows the performance of the continuous MP algorithms described in Section 4.4.4. As can be observed, the AMP-EP approach overlaps with the $\text{MP}_{\boldsymbol{\Psi}}$ algorithm, whereas the AMP-GA exhibits a non-negligible loss and is outperformed by all the other tested algorithms. Because of the selected constellation size, i.e., $M = 4$, the complexity advantages brought by continuous MP algorithms are limited (see Table 4.1). In fact, they are allowed to consider nearly the same number S of interferers per symbol as the $\text{MP}_{\boldsymbol{\Psi}}$ to satisfy the complexity constraint. Consequently, among the $\boldsymbol{\Psi}$-based algorithms, we focus solely on the $\text{MP}_{\boldsymbol{\Psi}}$ for comparison purposes in the following analyses. It is worth specifying that, as previously anticipated, the analyzed algorithms are constrained to perform a maximum of 3 iterations. However, each algorithm requires a different number of iterations to converge. Therefore, given a fixed overall computational load, an important aspect concerns the optimal trade-off between the number of iterations and the number of interferers considered in the detection process. This will be the focus of future investigations.

In Figure 4.12, we assessed the performance of the proposed OTFS detector using a higher-order constellation, specifically the 16QAM, under the same operating conditions of Figure 4.11, except for the grid dimensions, which are reduced to $L = 20$ and $N = 16$ for simplicity. Unlike the QPSK case, the MMSE benchmark and the proposed solution present different behaviors. In fact, in the low SNR range, the proposed algorithm, along with the other graph-based approaches, outperforms the MMSE detector. However, at high SNRs, none of the analyzed detectors reach the MMSE benchmark. Specifically, the $\text{MP}_{\boldsymbol{\Psi}}$ and $\text{MP}_{\boldsymbol{G}}$ algorithms exhibit very similar behavior, with a saturating pragmatic capacity below 3 bit/s/Hz, whereas the proposed algorithm outperforms both. In particular, in the medium SNR range, it demonstrates a superior performance also compared to the benchmark, which eventually surpasses it as the SNR increases.

Finally, Figure 4.13 reports the results associated to the adoption of the ETU channel model, still characterized by $P = 9$ multipath components but a greater delay spread with respect to the other considered models and more balanced path gains, leading to challenging interference conditions. Moreover, for this scenario, we assumed $u_{max} = 300\,\text{Kmph}$. As emerges from Figure 4.13, the overall behavior is comparable to the one of Figure 4.11, with the main

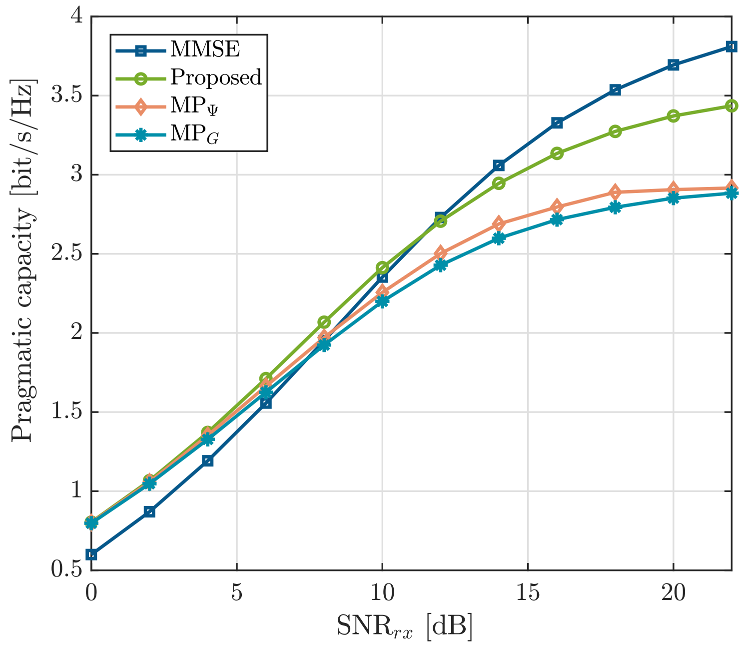


Figure 4.12: Performance comparison in terms of pragmatic capacity for 16QAM modulation alphabet using the EVA channel model.

difference that the MP$_{\boldsymbol{\Psi}}$ and the MP$_{\boldsymbol{G}}$ algorithms exhibit a saturating performance at lower values of pragmatic capacity compared to the case reported in Figure 4.11. This aspect reflects the harder operating conditions. On the other hand, the satisfying performance of the MMSE and the proposed solution demonstrates the robustness of the OTFS modulation to high magnitudes of Doppler shifts.

As a final remark on this consideration, we compared in Figure 4.14 the OTFS with the widely used OFDM modulation considering the EVA channel model. According to this modulation format, which is at the basis of 4G/5G mobile communication systems, the information symbols are multiplexed on closely spaced orthogonal subcarriers, creating parallel streams until the com-

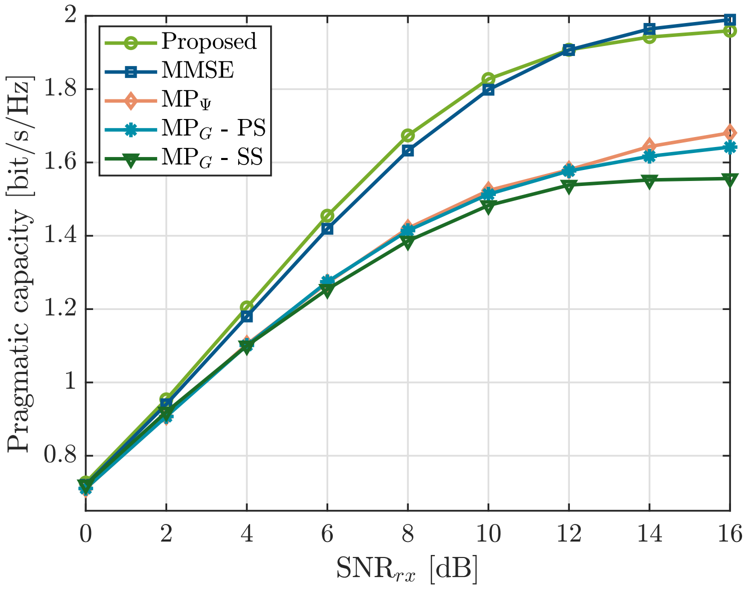


Figure 4.13: Performance comparison in terms of pragmatic capacity for QPSK modulation alphabet using the ETU channel model.

munication channel does not break up the orthogonality. OFDM is, therefore, a reliable waveform in frequency-selective channels, such as the *static* multipah wireless channel [58]. However, it is affected by some limitations in high-mobility scenarios, e.g., the severe loss of orthogonality. As shown in Figure 4.14, where detection is performed through the MMSE equalizer, the OTFS-based communication significantly outperforms the OFDM one. The superior performance of OTFS, especially for high spectral efficiency, reflects its robustness against high Doppler shifts and suitability for time-frequency selective channels. On the contrary, multiple Dopplers are hard to equalize within OFDM, as discussed in [58]. Moreover, the loss of the OFDM modulation is also caused by the Cyclic Prefix (CP) insertion, whose duration was set to 7% of T. While necessary to address the delay spread and mitigate subsequent ISI, it inevitably imposes a penalty on the achievable information rate.

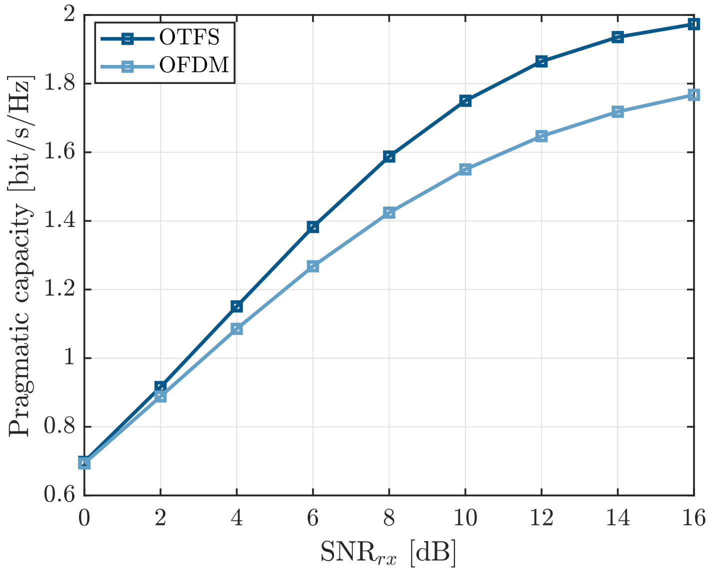


Figure 4.14: Performance comparison between OTFS and OFDM adopting the EVA channel model with $u_{max} = 100\,\text{Kmph}$ and using the MMSE detector.

4.6.2 Multi-satellite communications

Concerning the multi-satellite scenario, we considered an OTFS-based system with carrier frequency 5 GHz, system bandwidth of 0.24 MHz, $L = 64$ and $N = 50$, resulting in a subcarrier spacing of 3750 Hz and a symbol duration $T = 1/\Delta f$ of $267\,\mu$s, if not otherwise stated. We considered P satellites in visibility with the UT, belonging to the SpaceX *Starlink* constellation, whose orbital parameters are derived from [75]. The UT on ground has a fixed position. As mentioned in Section 4.3, we assumed that the P satellites perfectly compensate for delay and Doppler shifts at the *ideal UT position*. At this location, the signal contributions from the transmitting satellites arrive simultaneously. Then, we considered an *offset* distance on the order of kilometers of the UT from the ideal position, such that the residual uncompensated delay and Doppler shifts satisfy $\max_p\{\tau_p\} < T$ and $\max_p\{\nu_p\} < \Delta f$. We assumed that

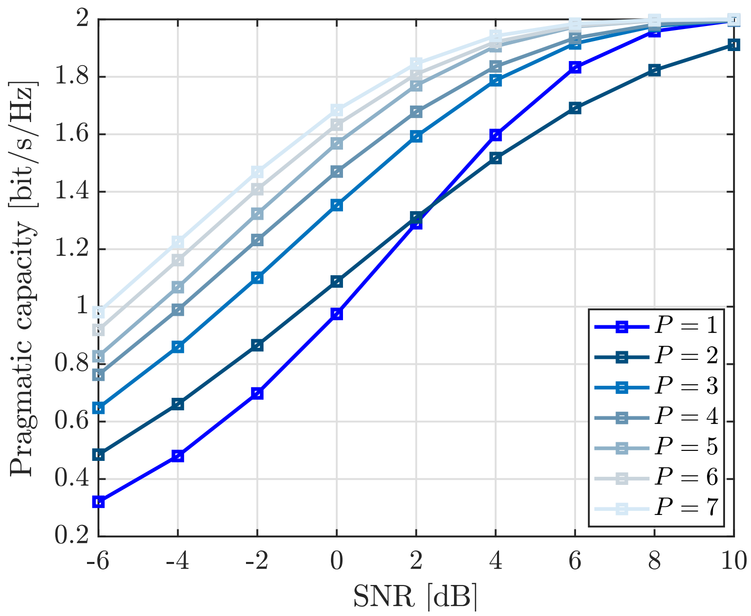


Figure 4.15: Pragmatic capacity exploiting multi-satellite diversity and using the MMSE detector.

the considered UT is served by the P closest satellites, ordered by increasing slant range. We evaluated the performance in terms of pragmatic capacity versus the SNR of the first path (i.e., the one characterized by the largest gain). The energy of all the considered paths is normalized to the shortest slant range attenuation.

Before moving to the analysis of the different detection strategies, we analyzed in Figures 4.15 and 4.16 the performance of the MMSE detector employing various numbers of transmitting satellites. In particular, the aim of Figure 4.15, where we considered $P = 1, \dots, 7$, is to verify the performance gain related to the adoption of macro-diversity configurations, whereas Figure 4.16 focuses on the comparison of the OTFS modulation with the OFDM one. As demonstrated in Figure 4.15, for $P > 2$ a higher number of satellites serving the same UT simultaneously allows to significantly improve the

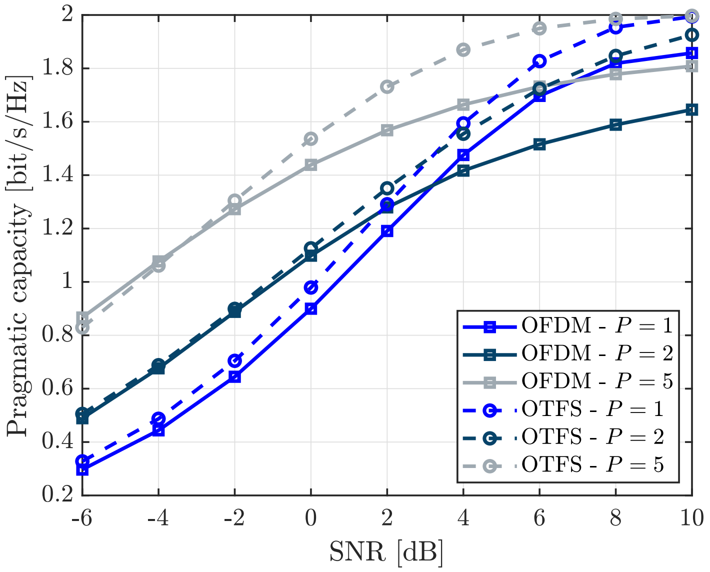


Figure 4.16: Performance comparison between OTFS and OFDM in terms of pragmatic capacity with $P = 1, 2, 5$ satellites serving simultaneously the same UT.

system performance, thus confirming the advantages that diversity schemes applied to OTFS-based communications could bring. Increasing the number of transmitting satellites from 1 to 2 results in a performance gain only in the low-to-medium SNR range when the MMSE detector is employed, as shown in Figure 4.15. This performance degradation at high spectral efficiency is addressed adopting other kinds of detectors, as discussed below. Figure 4.16 shows, instead, the performance comparison between OTFS and OFDM modulation formats considering multiple transmitting satellites. It can be observed that, for all analyzed values of P, the OTFS-based communication achieves superior performance compared to OFDM one. Additionally, from Figure 4.16 emerges that multi-satellite diversity enables a significant performance gain even for the OFDM modulation. The curves of Figure 4.16 are obtained with

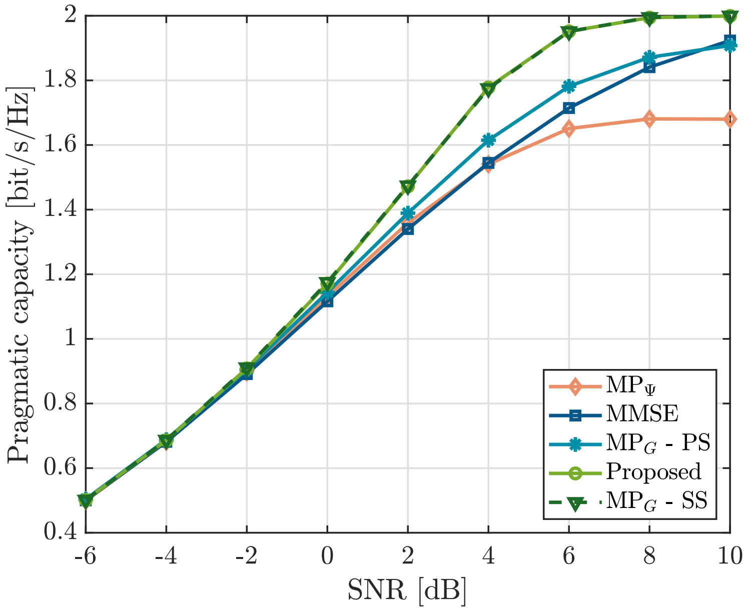


Figure 4.17: Performance comparison in terms of pragmatic capacity with $P = 2$ satellites in visibility with the UT.

subcarrier spacing $\Delta f = 15.625$ kHz, symbol time $T = 64$ μs, length of the CP for OFDM $T_{\mathrm{CP}} = 4.69$ μs, $M = 32$ and $N = 50$.

The performance of the other analyzed detectors is reported in Figures 4.17 and 4.18. Firstly, the scenario considered in Figure 4.17 foresees the presence of two transmitting satellites. In this context, the proposed detector and the MP$_{\boldsymbol{G}}$ based on a SS achieve the same pragmatic capacity, outperforming the MP$_{\boldsymbol{\Psi}}$ at fixed complexity and the MMSE. Moreover, we also tested the MP$_{\boldsymbol{G}}$ with a PS and found out that this implementation requires a higher number of iterations for achieving convergence. In fact, even if it reaches the MMSE, it shows a significant performance loss with respect to the proposed algorithm, as can be observed in Figure 4.17.

In Figure 4.18, we analyzed the detectors performance when the same OTFS signal is transmitted simultaneously by $P = 5$ satellites. The MP$_{\boldsymbol{G}}$ has

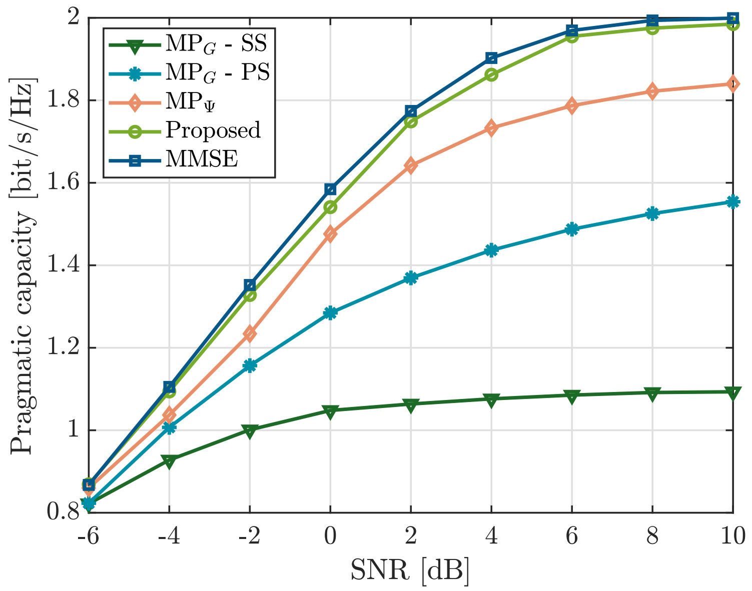


Figure 4.18: Performance comparison in terms of pragmatic capacity with $P = 5$ satellites serving simultaneously the UT.

been tested with both schedules and it can be observed that, in this context, the implementation relying on the PS results to be much more effective than that based on the SS. Although this reversal in performance between SS and PS may at first surprise, the interplay between the FG complexity and the scheduling strategy is a known issue. Such that an ad-hoc scheduling is sometimes needed to implement an MP algorithm on the FG, modeling some specific problems that would otherwise become intractable [33,34,76]. However, in this case, both MP$_{\boldsymbol{G}}$ algorithms are outperformed by all the other algorithms, showing that a higher number of paths, and a resulting reduced channel sparsity, causes a performance degradation. The largest pragmatic capacity is achieved by the MMSE, which is practically reached by the proposed algorithm. On the contrary, the MP$_{\boldsymbol{\Psi}}$ saturates at approximately 1.8 bit/s/Hz and it is thus outperformed by the proposed solution. As mentioned in Section 4.6.1, future studies will include convergence analyses to investigate the optimal trade-off between the number of iterations and the number of interferers per symbol.

4.7 Conclusions

In this chapter, we presented the state-of-the-art detection algorithms for OTFS-based transmissions. In particular, we discussed different families of algorithms, such as linear equalizers and graph-based approaches, each of them characterized by a different computational load, methodology and performance. Moreover, we proposed a detection strategy for OTFS modulation based on MP. We considered two communication scenarios: standard 3GPP multipath fading channels, characterized by different power delay profiles, and a LEO multi-satellite transmission system, where the multipath channel is doubly-selective because of the high Doppler shifts and channel time-variance, typically encountered in such contexts. The proposed algorithm leverages the sparsity and the particular structure of the channel matrix to enhance the detection efficiency. Moreover, it relies on an innovative schedule which, elaborating the strongest interferers first, allows to speed up the convergence process. Performance comparisons with state-of-the-art methods demonstrated that our approach offers an optimal trade-off between complexity and performance in both scenarios.

Conclusions

In this thesis, we addressed the development of graph-based detection and channel estimation techniques for wireless channels. The main goal of this work was to solve the problems related to unfeasible detection strategies through approximate MP techniques without impacting the performance, therefore minimizing the information loss induced by the approximation.

In particular, with reference to single-carrier transmissions, Chapters 2 and 3 focused on the design of joint detection and channel estimation algorithms for channels impaired by two strongly limiting factors in communication systems, i.e., fading and phase noise, respectively. In both cases, the analysis centered on the integration of the EP framework in the receiver design, in order to enable the interaction with the SPA computational rules to block the propagation of intractable, multimodal distributions, which otherwise would create a prohibitive complexity. Differently from other existing approaches for the approximation of unmanageable probability distributions, the power of EP is its inherent ability to combine prior knowledge of system variables with channel observations. However, before "packaging" a self-sustainable and flexible algorithm, thus able to work even with rather far pilots groups, several challenges emerged and achieving satisfactory performance required improved approximations and ad hoc adjustments in the way in which the probability distributions of latent variables - both data and channel parameters - are combined and projected. Simulation results revealed the relevant advantages of the proposed algorithms in the two study cases, demonstrating their applicability to various communication standards.

Future developments on this topic will focus on the analysis of different kinds of divergence measures, besides the KL one, for the reduction of the mixtures generated by the SPA. In fact, in some critical situations, exclusive divergence measures that ignore far away modes from the main mass could be beneficial for the channel estimation process.

In Chapter 4, we moved the focus to OTFS-based multi-carrier transmissions over doubly-selective channels. The relevance of this waveform lies in its robustness in challenging, dynamically changing, wireless environments, which are becoming more and more widespread. We considered two different scenarios, i.e., multipath time-variant 3GPP terrestrial channel profiles and a LEO multi-satellite transmission, specifically designed to enhance the system reliability. We addressed the detection problem under perfect CSI conditions. To this purpose, we reviewed the main state-of-the-art algorithms, highlighting the advantages and weak aspects of the different approaches. After a deep analysis of the equivalent channel matrix in the Doppler-delay domain, we proceeded operating on its processed version, showing a more defined and compact structure. We then discussed the proposed MP solution, which takes advantage of the arrangement of its main entries along diagonals. Furthermore, the introduction of an enhanced scheduling, where the strongest interferers are processed first within the detection operations, allowed to significantly speed up convergence, leading to an optimal trade-off between complexity and performance.

Regarding future developments, the analysis will be extended to examine the impact of non-ideal knowledge of CSI, assessing the sensitivity of the proposed detection strategy to imperfect channel estimates. Additionally, future studies will address the derivation of a closed-form expression for the optimal value of the AWGN variance used by the detector, as a function of the simulated SNR and the channel parameters. Finally, with reference to the considered multi-satellite scenario, radio resource management algorithms can be studied in order to determine the optimal satellite-user association and thereby enhance the system throughput.

Appendix A

Product of Multivariate Gaussian Distributions

In the following, we will derive the result in (2.10)-(2.12) by demonstrating that the product among two Gaussian distributions with the same argument $\boldsymbol{h}_k$ yields another Gaussian distribution with argument $\boldsymbol{h}_k$, whose covariance matrix ($\boldsymbol{C}_{u,k}$) and mean ($\boldsymbol{\eta}_{u,k}$) are given by (2.11) and (2.12), respectively. This result is multiplied by a distribution independent of $\boldsymbol{h}_k$, hence just a coefficient in our derivation. The presence of this last term justifies the use of the $\propto$ sign in (2.10).

The steps for the computation of $p^{(u)}(\boldsymbol{h}_k)$ are reported in the following.

$$\begin{aligned}
p^{(u)}(\boldsymbol{h}_k) &\triangleq p^{(f)}_{k|k-1}(\boldsymbol{h}_k)p^{(b)}_{k|k+1}(\boldsymbol{h}_k) \\
&= g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k-1}, \boldsymbol{C}_{k|k-1}\right) g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k+1}\right) \\
&= \frac{1}{\pi^{2N}|\boldsymbol{C}_{k|k-1}| \cdot |\boldsymbol{C}_{k|k+1}|} \exp\left\{-(\boldsymbol{h}_k - \boldsymbol{\eta}_{k|k-1})^H \boldsymbol{C}^{-1}_{k|k-1}(\boldsymbol{h}_k - \boldsymbol{\eta}_{k|k-1})\right\} \\
&\qquad \cdot \exp\left\{-(\boldsymbol{h}_k - \boldsymbol{\eta}_{k|k+1})^H \boldsymbol{C}^{-1}_{k|k+1}(\boldsymbol{h}_k - \boldsymbol{\eta}_{k|k+1})\right\}
\end{aligned} \tag{A.1}$$

Developing the products in the exponent, we obtain

$$
\begin{aligned}
&-(\boldsymbol{h}_k-\boldsymbol{\eta}_{k|k-1})^H \boldsymbol{C}_{k|k-1}^{-1}(\boldsymbol{h}_k-\boldsymbol{\eta}_{k|k-1})-(\boldsymbol{h}_k-\boldsymbol{\eta}_{k|k+1})^H \boldsymbol{C}_{k|k+1}^{-1}(\boldsymbol{h}_k-\boldsymbol{\eta}_{k|k+1}) \\
&=-[\boldsymbol{h}_k^H(\boldsymbol{C}_{k|k-1}^{-1}+\boldsymbol{C}_{k|k+1}^{-1})\boldsymbol{h}_k-2\Re(\boldsymbol{h}_k^H(\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}))+ \\
&\quad+\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{\eta}_{k|k+1}^H \boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}] \\
&=-[\boldsymbol{h}_k^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{h}_k-2\Re(\boldsymbol{h}_k^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k})+\boldsymbol{\eta}_{u,k}^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}]+ \\
&\quad-[\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{\eta}_{k|k+1}^H \boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}-\boldsymbol{\eta}_{u,k}^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}]\,,
\end{aligned}
\tag{A.2}
$$

where in (A.2) we defined $\boldsymbol{C}_{u,k}^{-1}$ and $\boldsymbol{\eta}_{u,k}$ as

$$
\boldsymbol{C}_{u,k}^{-1}=\boldsymbol{C}_{k|k-1}^{-1}+\boldsymbol{C}_{k|k+1}^{-1}\,, \tag{A.3}
$$

$$
\boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}=\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}\,. \tag{A.4}
$$

Substituting (A.2) in the exponential function of (A.1), it follows that

$$
\begin{aligned}
p^{(u)}(\boldsymbol{h}_k)&=\frac{1}{\pi^N|\boldsymbol{C}_{u,k}|}\cdot\frac{|\boldsymbol{C}_{u,k}|}{\pi^N|\boldsymbol{C}_{k|k-1}|\cdot|\boldsymbol{C}_{k|k+1}|} \\
&\quad\cdot\exp\{-[\boldsymbol{h}_k^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{h}_k-2\Re(\boldsymbol{h}_k^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k})+\boldsymbol{\eta}_{u,k}^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}]\} \\
&\quad\cdot\exp\{-[\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{\eta}_{k|k+1}^H \boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}-\boldsymbol{\eta}_{u,k}^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}]\} \\
&=g_{\mathbb{C}}\left(\boldsymbol{h}_k,\boldsymbol{\eta}_{u,k},\boldsymbol{C}_{u,k}\right)\cdot\frac{|\boldsymbol{C}_{u,k}|}{\pi^N|\boldsymbol{C}_{k|k-1}|\cdot|\boldsymbol{C}_{k|k+1}|} \\
&\quad\cdot\exp\{-[\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{\eta}_{k|k+1}^H \boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1}-\boldsymbol{\eta}_{u,k}^H \boldsymbol{C}_{u,k}^{-1}\boldsymbol{\eta}_{u,k}]\}\,.
\end{aligned}
\tag{A.5}
$$

The exponential term in (A.5) can be written as

$$
\begin{aligned}
&\exp\{-[\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{\eta}_{k|k+1}^H \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+ \\
&\qquad-(\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1})^H \boldsymbol{C}_{u,k}(\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{\eta}_{k|k-1}+\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1})]\} \\
&=\exp\{-[\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{c,k}^{-1}\boldsymbol{\eta}_{k|k-1}-2\Re(\boldsymbol{\eta}_{k|k-1}^H \boldsymbol{C}_{c,k}^{-1}\boldsymbol{\eta}_{c,k})+\boldsymbol{\eta}_{c,k}^H \boldsymbol{C}_{c,k}^{-1}\boldsymbol{\eta}_{c,k}]\} && \text{(A.6)} \\
&=\pi^N\cdot|\boldsymbol{C}_{c,k}|\cdot g_{\mathbb{C}}\left(\boldsymbol{\eta}_{k|k-1},\boldsymbol{\eta}_{k|k+1},\boldsymbol{C}_{k|k-1}+\boldsymbol{C}_{k|k+1}\right) && \text{(A.7)}
\end{aligned}
$$

where in (A.6) we defined $\boldsymbol{C}_{c,k}^{-1}$ and $\boldsymbol{\eta}_{c,k}$ as

$$\begin{aligned}
\boldsymbol{C}_{c,k}^{-1} &= \boldsymbol{C}_{k|k-1}^{-1} - \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{C}_{u,k}\boldsymbol{C}_{k|k-1}^{-1} \\
&= (\boldsymbol{C}_{k|k-1} + \boldsymbol{C}_{k|k+1})^{-1} \qquad \text{(A.8)} \\
\boldsymbol{C}_{c,k}^{-1}\boldsymbol{\eta}_{c,k} &= \boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{C}_{u,k}\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1} \\
&= \boldsymbol{C}_{k|k-1}^{-1}(\boldsymbol{C}_{k|k-1}^{-1} + \boldsymbol{C}_{k|k+1}^{-1})^{-1}\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{\eta}_{k|k+1} \\
&= [\boldsymbol{C}_{k|k+1}(\boldsymbol{C}_{k|k-1}^{-1} + \boldsymbol{C}_{k|k+1}^{-1})\boldsymbol{C}_{k|k-1}]^{-1}\boldsymbol{\eta}_{k|k+1} \\
&= (\boldsymbol{C}_{k|k-1} + \boldsymbol{C}_{k|k+1})^{-1}\boldsymbol{\eta}_{k|k+1} \,,
\end{aligned}$$

and (A.8) derives from the Woodbury Identity (WI).
Finally, substituting (A.7) in (A.5), it follows

$$\begin{aligned}
p^{(u)}(\boldsymbol{h}_k) &= \frac{\pi^N|\boldsymbol{C}_{u,k}|\cdot|\boldsymbol{C}_{c,k}|}{\pi^N|\boldsymbol{C}_{k|k-1}|\cdot|\boldsymbol{C}_{k|k+1}|} g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) \\
&\quad \cdot g_{\mathbb{C}}\Big(\boldsymbol{\eta}_{k|k-1}, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k-1} + \boldsymbol{C}_{k|k+1}\Big) \\
&= |\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{C}_{u,k}\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{C}_{c,k}| \cdot g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) \qquad \text{(A.9)} \\
&\quad \cdot g_{\mathbb{C}}\Big(\boldsymbol{\eta}_{k|k-1}, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k-1} + \boldsymbol{C}_{k|k+1}\Big) \\
&= g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) \cdot g_{\mathbb{C}}\Big(\boldsymbol{\eta}_{k|k-1}, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k-1} + \boldsymbol{C}_{k|k+1}\Big) \,, \qquad \text{(A.10)}
\end{aligned}$$

where we can obtain (A.10) from (A.9) by considering that

$$\begin{aligned}
|\boldsymbol{C}_{k|k-1}^{-1}\boldsymbol{C}_{u,k}\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{C}_{c,k}| &= |\boldsymbol{C}_{k|k-1}^{-1}(\boldsymbol{C}_{k|k-1}^{-1} + \boldsymbol{C}_{k|k+1}^{-1})^{-1}\boldsymbol{C}_{k|k+1}^{-1}\boldsymbol{C}_{c,k}| \\
&= |[\boldsymbol{C}_{k|k+1}(\boldsymbol{C}_{k|k-1}^{-1} + \boldsymbol{C}_{k|k+1}^{-1})\boldsymbol{C}_{k|k-1}]^{-1}\boldsymbol{C}_{c,k}| \\
&= |(\boldsymbol{C}_{k|k+1} + \boldsymbol{C}_{k|k-1})^{-1}\boldsymbol{C}_{c,k}| \\
&= |\,\mathbb{I}\,| \\
&= 1\,.
\end{aligned}$$

In the context under consideration, the 2nd factor in (A.10) can be neglected being a constant term in the variable of interest ($\boldsymbol{h}_k$). Therefore the Gaussian product reduces to $p^{(u)}(\boldsymbol{h}_k) \propto g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right)$.

Appendix B

Product of a Multivariate and a Univariate Gaussian Distribution

In this appendix, we provide derivation details on the product in (2.13) among a general multivariate Gaussian distribution

$$g_{\mathbb{C}}\left(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{C}\right) = \frac{1}{\pi^N |\boldsymbol{C}|} \exp\left\{-(\boldsymbol{x}-\boldsymbol{\eta})^H \boldsymbol{C}^{-1} (\boldsymbol{x}-\boldsymbol{\eta})\right\} \tag{B.1}$$

where N is the dimension of the column vector $\boldsymbol{x}$, and the Gaussian distribution

$$g_{\mathbb{C}}\left(\boldsymbol{a}^H \boldsymbol{x}, y, 2\sigma^2\right) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2\sigma^2} |\boldsymbol{a}^H \boldsymbol{x} - y|^2\right\} \tag{B.2}$$

depending on a single variable included in $\boldsymbol{x}$, which is selected by the scalar product with $\boldsymbol{a}$. In the analyzed case, $\boldsymbol{a} = (1, 0, \ldots, 0)^T$.

Multiplying (B.1) with (B.2), we obtain

$$
\begin{aligned}
& g_{\mathbb{C}}\left(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{C}\right) g_{\mathbb{C}}\left(\boldsymbol{a}^H \boldsymbol{x}, y, 2\sigma^2\right) = \frac{1}{2\pi^{N+1}\sigma^2 |\boldsymbol{C}|} \\
& \cdot \exp\left\{-\left[\boldsymbol{x}^H \left(\boldsymbol{C}^{-1} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right)\boldsymbol{x} - 2\Re\left(\boldsymbol{x}^H\left(\boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{y\boldsymbol{a}}{2\sigma^2}\right)\right) + \boldsymbol{\eta}_1^H \boldsymbol{C}_1^{-1}\boldsymbol{\eta}_1\right]\right\} \\
& \cdot \exp\left\{-\left[\boldsymbol{\eta}^H \boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{|y|^2}{2\sigma^2} - \boldsymbol{\eta}_1^H \boldsymbol{C}_1^{-1}\boldsymbol{\eta}_1\right]\right\} ,
\end{aligned}
$$

where $\boldsymbol{C}_1$ and $\boldsymbol{\eta}_1$ are defined as

$$
\begin{aligned}
\boldsymbol{C}_1 &= \left(\boldsymbol{C}^{-1} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right)^{-1} , \\
\boldsymbol{\eta}_1 &= \boldsymbol{C}_1\left(\boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{y\boldsymbol{a}}{2\sigma^2}\right) .
\end{aligned}
$$

It follows that

$$
\begin{aligned}
& g_{\mathbb{C}}\left(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{C}\right) g_{\mathbb{C}}\left(\boldsymbol{a}^H \boldsymbol{x}, y, 2\sigma^2\right) = \underbrace{\frac{1}{\pi^N |\boldsymbol{C}_1|} \exp\left\{-(\boldsymbol{x} - \boldsymbol{\eta}_1)^H \boldsymbol{C}_1^{-1}(\boldsymbol{x} - \boldsymbol{\eta}_1)\right\}}_{g_{\mathbb{C}}(\boldsymbol{x}, \boldsymbol{\eta}_1, \boldsymbol{C}_1)} \\
& \cdot \frac{|\boldsymbol{C}_1|}{2\pi\sigma^2|\boldsymbol{C}|} \exp\left\{-\left[\boldsymbol{\eta}^H \boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{|y|^2}{2\sigma^2} - \boldsymbol{\eta}_1^H \boldsymbol{C}_1^{-1}\boldsymbol{\eta}_1\right]\right\} .
\end{aligned}
\tag{B.3}
$$

The determinant of $\boldsymbol{C}_1$ can be rewritten as

$$
\begin{aligned}
|\boldsymbol{C}_1| &= \frac{1}{|\boldsymbol{C}_1^{-1}|} \\
&= \frac{1}{\left|\boldsymbol{C}^{-1} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right|} = \frac{1}{\left|\boldsymbol{C}^{-1} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\boldsymbol{C}\boldsymbol{C}^{-1}\right|} \\
&= \frac{1}{\left|\left[\mathbb{I} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\boldsymbol{C}\right]\boldsymbol{C}^{-1}\right|} = \frac{1}{\left|\mathbb{I} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\boldsymbol{C}\right| \cdot \left|\boldsymbol{C}^{-1}\right|} \\
&= \frac{|\boldsymbol{C}|}{\left|\mathbb{I} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\boldsymbol{C}\right|} = \frac{|\boldsymbol{C}|}{1 + \frac{\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}{2\sigma^2}} ,
\end{aligned}
$$

where the equality in the last row can be easily verified $\forall \boldsymbol{a}, \boldsymbol{C}$.

Substituting this result and the definition of $\boldsymbol{\eta}_1$ in (B.3), we obtain

$$g_{\mathbb{C}}(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{C})\, g_{\mathbb{C}}\left(\boldsymbol{a}^H \boldsymbol{x}, y, 2\sigma^2\right) = g_{\mathbb{C}}(\boldsymbol{x}, \boldsymbol{\eta}_1, \boldsymbol{C}_1) \cdot \frac{|\boldsymbol{C}|}{1 + \frac{\boldsymbol{a}^H \boldsymbol{C} \boldsymbol{a}}{2\sigma^2}} \cdot \frac{1}{2\sigma^2 \pi |\boldsymbol{C}|}$$
$$\cdot \exp\left\{ - \left[\boldsymbol{\eta}^H \boldsymbol{C}^{-1} \boldsymbol{\eta} + \frac{|y|^2}{2\sigma^2} - \left(\boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{y\boldsymbol{a}}{2\sigma^2} \right)^H \boldsymbol{C}_1 \left(\boldsymbol{C}^{-1}\boldsymbol{\eta} + \frac{y\boldsymbol{a}}{2\sigma^2} \right) \right] \right\}. \tag{B.4}$$

Performing the involved operations in the exponent, we can rewrite (B.4) in the following form

$$g_{\mathbb{C}}(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{C})\, g_{\mathbb{C}}\left(\boldsymbol{a}^H \boldsymbol{x}, y, 2\sigma^2\right) = g_{\mathbb{C}}(\boldsymbol{x}, \boldsymbol{\eta}_1, \boldsymbol{C}_1) \cdot \frac{1}{\pi(2\sigma^2 + \boldsymbol{a}^H \boldsymbol{C} \boldsymbol{a})}$$
$$\cdot \exp\left\{ - \left[\underbrace{\frac{|y|^2}{2\sigma^2}\left(1 - \frac{\boldsymbol{a}^H \boldsymbol{C}_1 \boldsymbol{a}}{2\sigma^2}\right)}_{\boxed{1}} - \underbrace{\frac{2\Re(y\boldsymbol{\eta}^H \boldsymbol{C}^{-1}\boldsymbol{C}_1 \boldsymbol{a})}{2\sigma^2}}_{\boxed{2}} + \underbrace{\boldsymbol{\eta}^H \boldsymbol{C}^{-1}(I - \boldsymbol{C}_1 \boldsymbol{C}^{-1})\boldsymbol{\eta}}_{\boxed{3}} \right] \right\} \tag{B.5}$$

We will analyze separately the terms $\boxed{1}, \boxed{2}, \boxed{3}$ in (B.5). In the following expressions we will exploit the WI to rewrite the covariance matrix $\boldsymbol{C}_1$ as

$$\begin{aligned} \boldsymbol{C}_1 &= \left(\boldsymbol{C}^{-1} + \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2} \right)^{-1} \\ &\overset{\text{(WI)}}{=} \boldsymbol{C} - \boldsymbol{C}\boldsymbol{a}(2\sigma^2 + \boldsymbol{a}^H \boldsymbol{C} \boldsymbol{a})^{-1} \boldsymbol{a}^H \boldsymbol{C}\,. \end{aligned} \tag{B.6}$$

The term $\boxed{1}$ can be manipulated as follows

$$
\begin{aligned}
\frac{|y|^2}{2\sigma^2}\left(1-\frac{\boldsymbol{a}^H\boldsymbol{C}_1\boldsymbol{a}}{2\sigma^2}\right) &= \frac{|y|^2}{2\sigma^2}\left(\frac{2\sigma^2-\boldsymbol{a}^H\boldsymbol{C}_1\boldsymbol{a}}{2\sigma^2}\right)\\
&= \frac{|y|^2}{(2\sigma^2)^2}\left(2\sigma^2-\boldsymbol{a}^H\left[\boldsymbol{C}-\boldsymbol{C}\boldsymbol{a}(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\right]\boldsymbol{a}\right)\\
&= \frac{|y|^2}{(2\sigma^2)^2}\left(2\sigma^2-\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\left[1-(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right]\right)\\
&= \frac{|y|^2}{(2\sigma^2)^2}\left(2\sigma^2-\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\left[\frac{2\sigma^2}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\right]\right)\\
&= \frac{|y|^2}{(2\sigma^2)^2}\left(\frac{(2\sigma^2)^2}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\right)\\
&= \frac{|y|^2}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\,. \qquad \text{(B.7)}
\end{aligned}
$$

The term $\boxed{2}$ can be rewritten as

$$
\begin{aligned}
\frac{2\Re(y\boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{C}_1\boldsymbol{a})}{2\sigma^2} &= 2\Re\left\{\frac{1}{2\sigma^2}\left(y\boldsymbol{\eta}^H\boldsymbol{C}^{-1}\left[\boldsymbol{C}-\boldsymbol{C}\boldsymbol{a}(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\right]\boldsymbol{a}\right)\right\}\\
&= 2\Re\left\{\frac{1}{2\sigma^2}\left(y\boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{C}\boldsymbol{a}\left[1-(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right]\right)\right\}\\
&= 2\Re\left\{\frac{1}{2\sigma^2}\left[y\boldsymbol{\eta}^H\boldsymbol{a}\left(\frac{2\sigma^2}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\right)\right]\right\}\\
&= \frac{2\Re\left\{y\boldsymbol{\eta}^H\boldsymbol{a}\right\}}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\,. \qquad \text{(B.8)}
\end{aligned}
$$

Finally, the term $\boxed{3}$ becomes

$$
\begin{aligned}
\boldsymbol{\eta}^H\boldsymbol{C}^{-1}(I-\boldsymbol{C}_1\boldsymbol{C}^{-1})\boldsymbol{\eta} &= \boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{\eta}-\boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{C}_1\boldsymbol{C}^{-1}\boldsymbol{\eta}\\
&= \boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{\eta}-\boldsymbol{\eta}^H\boldsymbol{C}^{-1}\left[\boldsymbol{C}-\boldsymbol{C}\boldsymbol{a}(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\right]\boldsymbol{C}^{-1}\boldsymbol{\eta}\\
&= \boldsymbol{\eta}^H\boldsymbol{C}^{-1}\boldsymbol{C}\boldsymbol{a}(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{C}^{-1}\boldsymbol{\eta}\\
&= |\boldsymbol{a}^H\boldsymbol{\eta}|^2(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})^{-1}\,. \qquad \text{(B.9)}
\end{aligned}
$$

Substituting the expression (B.7)-(B.9) in (B.5), we obtain the final result

$$
\begin{aligned}
& g_{\mathbb{C}}\left(\boldsymbol{x},\boldsymbol{\eta},\boldsymbol{C}\right) g_{\mathbb{C}}\left(\boldsymbol{a}^H\boldsymbol{x},y,2\sigma^2\right) = g_{\mathbb{C}}\left(\boldsymbol{x},\boldsymbol{\eta}_1,\boldsymbol{C}_1\right)\cdot\frac{1}{\pi(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})} \\
& \cdot\exp\left\{-\frac{1}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\left[|y|^2-2\Re\left\{y\boldsymbol{\eta}^H\boldsymbol{a}\right\}+|\boldsymbol{a}^H\boldsymbol{\eta}|^2\right]\right\} \\
& = g_{\mathbb{C}}\left(\boldsymbol{x},\boldsymbol{\eta}_1,\boldsymbol{C}_1\right)\cdot\frac{1}{\pi(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a})}\exp\left\{-\frac{1}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}}\left|y-\boldsymbol{a}^H\boldsymbol{\eta}\right|^2\right\} \\
& = g_{\mathbb{C}}\left(\boldsymbol{x},\boldsymbol{\eta}_1,\boldsymbol{C}_1\right)\cdot g_{\mathbb{C}}\left(y,\boldsymbol{a}^H\boldsymbol{\eta},2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right) \\
& = g_{\mathbb{C}}\left(\boldsymbol{x},\underbrace{\left(\boldsymbol{C}^{-1}+\frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right)^{-1}\left(\boldsymbol{C}^{-1}\boldsymbol{\eta}+\frac{y\boldsymbol{a}}{2\sigma^2}\right)}_{\boldsymbol{\eta}_1},\underbrace{\left(\boldsymbol{C}^{-1}+\frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right)^{-1}}_{\boldsymbol{C}_1}\right) \\
& \quad\cdot g_{\mathbb{C}}\left(y,\boldsymbol{a}^H\boldsymbol{\eta},2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right)\,.
\end{aligned}
\tag{B.10}
$$

The mean value $\boldsymbol{\eta}_1$ of the first Gaussian distribution can be rewritten as

$$
\begin{aligned}
\boldsymbol{\eta}_1 &= \left(\boldsymbol{C}^{-1}+\frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2}\right)^{-1}\left(\boldsymbol{C}^{-1}\boldsymbol{\eta}+\frac{y\boldsymbol{a}}{2\sigma^2}\right) \\
&\overset{\text{(WI)}}{=}\left[\boldsymbol{C}-\boldsymbol{C}\boldsymbol{a}\left(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right)^{-1}\boldsymbol{a}^H\boldsymbol{C}\right]\left(\boldsymbol{C}^{-1}\boldsymbol{\eta}+\frac{y\boldsymbol{a}}{2\sigma^2}\right) \\
&= \boldsymbol{\eta}+\boldsymbol{C}\boldsymbol{a}\frac{y}{2\sigma^2}-\boldsymbol{C}\boldsymbol{a}\left(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right)^{-1}\boldsymbol{a}^H\boldsymbol{\eta}+ \\
&\quad -\boldsymbol{C}\boldsymbol{a}\left(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right)^{-1}\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\frac{y}{2\sigma^2} \\
&= \boldsymbol{\eta}+\boldsymbol{C}\boldsymbol{a}\frac{\boldsymbol{\eta}-\boldsymbol{a}^H\boldsymbol{\eta}}{2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}} \\
&= \boldsymbol{\eta}+\underbrace{\boldsymbol{C}\boldsymbol{a}\left(2\sigma^2+\boldsymbol{a}^H\boldsymbol{C}\boldsymbol{a}\right)^{-1}}_{\boldsymbol{J}}\left(\boldsymbol{\eta}-\boldsymbol{a}^H\boldsymbol{\eta}\right),
\end{aligned}
\tag{B.11}
$$

whereas its covariance matrix $\boldsymbol{C}_1$, starting from (B.6) and introducing the above-defined vector $\boldsymbol{J}$, as

$$\boldsymbol{C}_1=\boldsymbol{C}-\boldsymbol{J}\boldsymbol{a}^H\boldsymbol{C}\,.$$

Appendix C

Details on Message Propagation

With reference to the FG in Figure 2.1, in this appendix we derive the expression of the parameters characterizing the distribution of the forward, backward and upward messages.

Once that the approximated *observation* message $\tilde{p}_d(\boldsymbol{h}_{k-1})$ is computed according to (2.22)-(2.23) applied to the time epoch $k-1$, the forward message $p^{(f)}_{k-1|k-1}(\boldsymbol{h}_{k-1})$ is computed as

$$\begin{aligned} p^{(f)}_{k-1|k-1}(\boldsymbol{h}_{k-1}) &= \tilde{p}_d(\boldsymbol{h}_{k-1}) p^{(f)}_{k-1|k-2}(\boldsymbol{h}_{k-1}) \\ &= g_{\mathbb{C}}\left(\boldsymbol{h}_{k-1}, \boldsymbol{\eta}_d, \boldsymbol{C}_d\right) g_{\mathbb{C}}\left(\boldsymbol{h}_{k-1}, \boldsymbol{\eta}_{k-1|k-2}, \boldsymbol{C}_{k-1|k-2}\right) \\ &\propto g_{\mathbb{C}}\left(\boldsymbol{h}_{k-1}, \boldsymbol{\eta}^{(f)}_{k-1|k-1}, \boldsymbol{C}^{(f)}_{k-1|k-1}\right), \end{aligned} \tag{C.1}$$

where in (C.1) we defined

$$\begin{aligned} \boldsymbol{C}^{-1}_{k-1|k-1} &= \boldsymbol{C}^{-1}_d + \boldsymbol{C}^{-1}_{k-1|k-2}, \\ \boldsymbol{C}^{-1}_{k-1|k-1}\boldsymbol{\eta}^{(f)}_{k-1|k-1} &= \boldsymbol{C}^{-1}_d \boldsymbol{\eta}_d + \boldsymbol{C}^{-1}_{k-1|k-2}\boldsymbol{\eta}_{k-1|k-2}. \end{aligned}$$

We can now compute the message $p^{(f)}_{k|k-1}(\boldsymbol{h}_k)$ as follows

$$\begin{aligned}
p^{(f)}_{k|k-1}(\boldsymbol{h}_k) &= \int p^{(f)}_{k-1|k-1}(\boldsymbol{h}_{k-1}) p(h_k|\boldsymbol{h}_{k-1})\, dh_{k-N} \\
&\propto \int g_{\mathbb{C}}\left(\boldsymbol{h}_{k-1}, \boldsymbol{\eta}^{(f)}_{k-1|k-1}, \boldsymbol{C}^{(f)}_{k-1|k-1}\right) g_{\mathbb{C}}\left(h_k, \boldsymbol{\rho}^H \boldsymbol{h}_{k-1}, 2\sigma_\nu^2\right) dh_{k-N} \\
&\propto g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k-1}, \boldsymbol{C}_{k|k-1}\right) ,
\end{aligned}$$

where $\boldsymbol{\eta}_{k|k-1}$ and $\boldsymbol{C}_{k|k-1}$ are defined as

$$\begin{aligned}
\boldsymbol{\eta}_{k|k-1} &= \boldsymbol{F}\boldsymbol{\eta}^{(f)}_{k-1|k-1} \\
\boldsymbol{C}_{k|k-1} &= \boldsymbol{F}\boldsymbol{C}^{(f)}_{k-1|k-1}\boldsymbol{F}^H + \boldsymbol{g}\boldsymbol{g}^H ,
\end{aligned}$$

where

$$\boldsymbol{F} = \begin{bmatrix} \rho_1 & \rho_2 & \rho_3 & \dots & \rho_{N-1} & \rho_N \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & & \\ 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{N\times N} = \begin{bmatrix} \boldsymbol{\rho}^H \\ \left[\mathbb{I}_{N-1}, \boldsymbol{0}_{(N-1)\times 1}\right] \end{bmatrix} ,$$

and

$$\boldsymbol{g} = \begin{bmatrix} \sqrt{2\sigma_\nu^2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{N\times 1} .$$

A similar procedure can be followed for the computation of the backward message $p^{(b)}_{k|k+1}(\boldsymbol{h}_k)$ which, finally, results to be

$$p^{(b)}_{k|k+1}(\boldsymbol{h}_k) \propto g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{k|k+1}, \boldsymbol{C}_{k|k+1}\right)$$

where, defining

$$\hat{\boldsymbol{g}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sqrt{2\sigma^2}_\nu/\rho_N \end{bmatrix}_{N\times 1} ,$$

$$\left(\boldsymbol{C}^{(b)}_{k+1|k+1}\right)^{-1} = \boldsymbol{C}_d^{-1} + \boldsymbol{C}^{-1}_{k+1|k+2} ,$$
$$\left(\boldsymbol{C}^{(b)}_{k+1|k+1}\right)^{-1} \boldsymbol{\eta}^{(b)}_{k+1|k+1} = \boldsymbol{C}_d^{-1}\boldsymbol{\eta}_d + \boldsymbol{C}^{-1}_{k+1|k+2}\boldsymbol{\eta}_{k+1|k+2} ,$$

the parameters $\boldsymbol{\eta}_{k|k+1}$ and $\boldsymbol{C}_{k|k+1}$ of the backward message can be computed as

$$\boldsymbol{\eta}_{k|k+1} = \boldsymbol{F}^{-1}\boldsymbol{\eta}^{(b)}_{k+1|k+1} ,$$
$$\boldsymbol{C}_{k|k+1} = \boldsymbol{F}^{-1}\boldsymbol{C}^{(b)}_{k+1|k+1}(\boldsymbol{F}^{-1})^H + \hat{\boldsymbol{g}}\hat{\boldsymbol{g}}^H .$$

Having computed the forward and backward messages in the two recursions, we can calculate their product, for each time epoch k, obtaining the message $p^{(u)}(\boldsymbol{h}_k)$ whose parameters are defined in (2.11) and (2.12).

Now, it is possible to compute the upward message $P_u(c_k)$ from the FN f_k to the VN c_k of the FG in Figure 2.1. According to the SPA, the symbol probability distribution of $c_k \in \mathcal{A} = \{a_1, \ldots, a_m\}$ can be calculated as

$$\begin{aligned} P_u(c_k = a_m) &= \int p_u(\boldsymbol{h}_k) f_k(a_m, h_k)\, d\boldsymbol{h}_k \\ &\propto \int g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) g_{\mathbb{C}}\left(r_k, a_m h_k, 2\sigma^2\right)\, d\boldsymbol{h}_k \\ &= \frac{1}{|a_m|^2} \int g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) g_{\mathbb{C}}\left(h_k, \frac{r_k}{a_m}, \frac{2\sigma^2}{|a_m|^2}\right) d\boldsymbol{h}_k \\ &= \frac{1}{|a_m|^2} \int g_{\mathbb{C}}\left(\boldsymbol{h}_k, \boldsymbol{\eta}_{u,k}, \boldsymbol{C}_{u,k}\right) g_{\mathbb{C}}\left(\boldsymbol{a}^H\boldsymbol{h}_k, \frac{r_k}{a_m}, \frac{2\sigma^2}{|a_m|^2}\right) d\boldsymbol{h}_k , \end{aligned} \tag{C.2}$$

where in (C.2), $\boldsymbol{a} = (1, 0, \ldots, 0)^T$ is a column vector with length N. By applying the result of Appendix B, we obtain

$$P_u(c_k = a_m) = \frac{1}{|a_m|^2} \int g_{\mathbb{C}} \left(\frac{r_k}{a_m}, \boldsymbol{a}^H \boldsymbol{\eta}_{u,k}, \frac{2\sigma^2}{|a_m|^2} + \boldsymbol{a}^H \boldsymbol{C}_{u,k} \boldsymbol{a} \right)$$
$$\cdot\, g_{\mathbb{C}} \left(\boldsymbol{h}_k, \left(\boldsymbol{C}_{u,k}^{-1} + |a_m|^2 \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2} \right)^{-1} \left(\boldsymbol{C}_{u,k}^{-1} \boldsymbol{\eta}_{u,k} + \frac{r_k a_m^* \boldsymbol{a}}{2\sigma^2} \right), \left(\boldsymbol{C}_{u,k}^{-1} + |a_m|^2 \frac{\boldsymbol{a}\boldsymbol{a}^H}{2\sigma^2} \right)^{-1} \right) d\boldsymbol{h}_k$$
$$= \frac{1}{|a_m|^2} g_{\mathbb{C}} \left(\frac{r_k}{a_m}, \boldsymbol{a}^H \boldsymbol{\eta}_{u,k}, \frac{2\sigma^2}{|a_m|^2} + \boldsymbol{a}^H \boldsymbol{C}_{u,k} \boldsymbol{a} \right)$$
$$= g_{\mathbb{C}} \left(r_k, a_m \boldsymbol{a}^H \boldsymbol{\eta}_{u,k}, 2\sigma^2 + |a_m|^2 \boldsymbol{a}^H \boldsymbol{C}_{u,k} \boldsymbol{a} \right) .$$

This message represents the final output of the detector and is, therefore, the information that is processed by the decoding unit.

Appendix D

Moment Matching Equation for Tikhonov Distributions

In the following, the derivation of the moment matching equation (3.18) is provided.

Let $p(\theta_k)$ be the Tikhonov mixture distribution, representing, e.g, the *observation* message in the case of TP or the exact *variable* distribution in the case of EP, that has to be approximated. The approximating distribution $q(\theta_k)$ is required to belong to a predefined family $\mathcal{F}$, which, in the analyzed case, is the set of Tikhonov distributions. We say that $q(\theta_k)$ is the KL-projection of $p(\theta_k)$ onto $\mathcal{F}$ as reported in (3.21). We can express $p(\theta_k)$ as

$$p(\theta_k) = \sum_{m=0}^{M-1} \overline{\alpha}_k^m t(\theta_k; z_k^m) \tag{D.1}$$

where $\overline{\alpha}_k^m$ with $m = 0, \ldots, M-1$ add up to 1 so that $p(\theta_k)$ is normalized. The approximating distribution $q(\theta_k)$ is a Tikhonov distribution with unknown parameter $\hat{z}_k$

$$q(\theta_k) = t(\theta_k; \hat{z}_k)\,. \tag{D.2}$$

which has to be computed via *moment matching*. Considering normalized distributions, therefore already satisfying the matching of the 0^{th} order moment

$$\mathbb{E}_p[1] = \int_{2\pi} p(\theta_k)\,\mathrm{d}\theta_k = \int_{2\pi} q(\theta_k)\,\mathrm{d}\theta_k = \mathbb{E}_q[1]\,, \tag{D.3}$$

we directly set the matching of the first trigonometric moment $\mathbb{E}[\mathrm{e}^{j\theta_k}]$ as

$$\mathbb{E}_q[\mathrm{e}^{j\theta_k}] = \mathbb{E}_p[\mathrm{e}^{j\theta_k}]\,, \tag{D.4}$$

since its $\Re/\Im$ parts give the expectations of the features of the Tikonov distribution, respectively $\mathbb{E}[\cos(\theta_k)]$ and $\mathbb{E}[\sin(\theta_k)]$. We express the generic Tikhonov distribution as

$$\begin{aligned} t(\theta_k;\hat{z}_k) &= \frac{1}{2\pi I_0(|\hat{z}_k|)}\exp\left(\Re\left[\hat{z}_k\mathrm{e}^{-j\theta_k}\right]\right) && \text{(D.5)}\\ &= \frac{1}{2\pi I_0(|\hat{z}_k|)}\exp\left(\Re\left[|\hat{z}_k|\mathrm{e}^{j(\angle\hat{z}_k-\theta_k)}\right]\right) && \text{(D.6)}\\ &= \frac{1}{2\pi I_0(|\hat{z}_k|)}\exp\left\{|\hat{z}_k|\cos(\angle\hat{z}_k-\theta_k)\right\} && \text{(D.7)} \end{aligned}$$

The first trigonometric moment of a Tikhonov distribution can be computed as

$$\begin{aligned} \mathbb{E}_q[\mathrm{e}^{j\theta_k}] &= \int_{2\pi}\mathrm{e}^{j\theta_k}t(\theta_k;\hat{z}_k)\,\mathrm{d}\theta_k && \text{(D.8)}\\ &= \frac{1}{2\pi I_0(|\hat{z}_k|)}\int_{2\pi}\mathrm{e}^{j\theta_k}\exp\left\{|\hat{z}_k|\cos(\angle\hat{z}_k-\theta_k)\right\}\,\mathrm{d}\theta_k && \text{(D.9)}\\ &= \frac{\mathrm{e}^{j\angle\hat{z}_k}}{2\pi I_0(|\hat{z}_k|)}\int_{2\pi}\mathrm{e}^{j\varphi}\exp\left\{|\hat{z}_k|\cos(\varphi)\right\}\,\mathrm{d}\varphi && \text{(D.10)}\\ &= \frac{\mathrm{e}^{j\angle\hat{z}_k}}{I_0(|\hat{z}_k|)}\underbrace{\frac{1}{2\pi}\int_{2\pi}\mathrm{e}^{j\varphi}\exp\left\{|\hat{z}_k|\cos(\varphi)\right\}\,\mathrm{d}\varphi}_{I_1(|\hat{z}_k|)} && \text{(D.11)}\\ &= \frac{I_1(|\hat{z}_k|)}{I_0(|\hat{z}_k|)}\mathrm{e}^{j\angle\hat{z}_k}\,, && \text{(D.12)} \end{aligned}$$

where in (D.10) we set $\varphi = \theta_k - \angle\hat{z}_k$. The first trigonometric moment of the

Tikhonov mixture is:

$$\mathbb{E}_p[\mathrm{e}^{j\theta_k}] = \int_{2\pi} \mathrm{e}^{j\theta_k} p(\theta_k)\,\mathrm{d}\theta_k \tag{D.13}$$

$$= \sum_{m=0}^{M-1} \overline{\alpha}_k^m \int_{2\pi} \mathrm{e}^{j\theta_k} t(\theta_k; z_k^m)\,\mathrm{d}\theta_k \tag{D.14}$$

$$= \sum_{m=0}^{M-1} \overline{\alpha}_k^m \frac{I_1(|z_k^m|)}{I_0(|z_k^m|)} \mathrm{e}^{j\angle z_k^m}\,. \tag{D.15}$$

Substituting (D.12) and (D.15) in (D.4) we obtain the moment matching equation:

$$\frac{I_1(|\hat{z}_k|)}{I_0(|\hat{z}_k|)} \mathrm{e}^{j\angle \hat{z}_k} = \sum_{m=0}^{M-1} \overline{\alpha}_k^m \frac{I_1(|z_k^m|)}{I_0(|z_k^m|)} \mathrm{e}^{j\angle z_k^m}\,. \tag{D.16}$$

Bibliography

[1] D. J. C. MacKay, "Regular LDPC online database," available: *http://www.inference.phy.cam.ac.uk/mackay/.*

[2] ETSI, "ETSI - DVBS2 74r13, Digital Video Broadcasting (DVB): Second generation framing structure, channel coding and modulation systems for Braoadcasting, Interactive Services, News Gathering and other broadband satellite applications," 2003.

[3] ETSI EN 302 307-2 Digital Video Broadcasting (DVB), Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications, Part II: S2-Extensions (DVB-S2X), Available on ETSI web site (http://www.etsi.org).

[4] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008. [Online]. Available: http://dx.doi.org/10.1561/2200000001

[5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[6] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, pp. 183–233, 1999.

[7] E. Conti, A. Piemontese, G. Colavolpe, and A. Vannucci, "Expectation propagation for flat-fading channels," *IEEE Wireless Commun. Letters*, vol. 12, no. 11, pp. 1846–1850, 2023.

[8] G. Colavolpe, E. Conti, A. Piemontese, and A. Vannucci, "The difficult road of expectation propagation towards phase noise detection," in *ICC 2023 - IEEE Intern. Conf. Commun.*, Rome, Italy, May 2023, pp. 4640–4645.

[9] E. Conti, A. Vannucci, A. Piemontese, and G. Colavolpe, "Phase noise detection via expectation propagation," submitted to *IEEE Trans. Commun.*, undergoing 2nd review round, 2024.

[10] E. Conti, A. Piemontese, T. Foggi, G. Colavolpe, and A. Vannucci, "Efficient message-passing detection for multi-satellite systems using OTFS modulation," to be presented at *IEEE Aerospace Conference*, Big Sky, MT, USA, 2025.

[11] C. Sacchi, A. Ugolini, A. Piemontese, T. Foggi, E. Conti, C. D'Andrea, and F. Babich, "A unified software-defined radio framework for flexible waveform design in non-terrestrial networks," to be presented at *IEEE Aerospace Conference*, Big Sky, MT, USA, 2025.

[12] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 55, pp. 754–767, May 2002.

[13] A. Chindapol and J. Ritcey, "Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 944–957, 2001.

[14] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. UAI*, Aug. 2001.

[15] ——, "Divergence measures and message passing," Microsoft Research Cambridge, Tech. Rep., 2005, mSR-TR-2005-173 (2005).

[16] H. Zhu and R. Rohwer, "Bayesian invariant measurements of generalization," *Neural Processing Letters*, vol. 2, pp. 28–31, 1995.

[17] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 591–600, May 1996.

[18] C. Berrou, A. Glavieux, and P. Thitmajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. IEEE Intern. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[19] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, October 1996.

[20] A. Kavčić, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1636–1652, Jul. 2003.

[21] J. B. Soriaga, H. Pfister, and P. Siegel, "Determining and approaching achievable rates of binary intersymbol interference channels using multistage decoding," *IEEE Trans. Inform. Theory*, vol. 53, pp. 1416–1429, Apr. 2007.

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

[23] H. Niu, M. Shen, J. A. Ritcey, and H. Liu, "A factor graph approach to iterative channel estimation and LDPC decoding over fading channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1345–1350, Jul. 2005.

[24] Y. Qi and T. P. Minka, "Window-based expectation propagation for adaptive signal detection in flat-fading channels," *IEEE Trans. Wireless Commun.*, vol. 6, no. 1, pp. 348–355, 2007.

[25] A. Barbieri, A. Piemontese, and G. Colavolpe, "On the ARMA approximation for fading channels described by the Clarke model with applications to Kalman-based receivers," *IEEE Trans. Wireless Commun.*, vol. 8, no. 2, pp. 535–540, 2009.

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, Inc., 1991.

[27] M. I. Jordan and C. Bishop, "An introduction to graphical models," 2001.

[28] W. Yuan, N. Wu, Q. Guo, Y. Li, C. Xing, and J. Kuang, "Iterative receivers for downlink MIMO-SCMA: Message passing and distributed cooperative detection," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3444–3458, 2018.

[29] N. Wu, W. Yuan, Q. Guo, and J. Kuang, "A hybrid BP-EP-VMP approach to joint channel estimation and decoding for FTN signaling over frequency selective fading channels," *IEEE Access*, vol. 5, pp. 6849–6858, 2017.

[30] M. Seeger, "Bayesian Gaussian process models: PAC-bayesian generalisation error bounds and sparse approximations," *PhD thesis, University of Edinburgh*, 2003.

[31] A. Gelman, A. Vehtari, P. Jylänki, T. Sivula, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. Robert, "Expectation propagation as a way of life: A framework for bayesian inference on partitioned data," *arXiv preprint arXiv:1412.4869*, 2014.

[32] J. Heo and K. M. Chugg, "Adaptive iterative detection for turbo codes on flat fading channels," in *WCNC*, Chicago, IL, Sep. 2000.

[33] A. Vannucci, G. Colavolpe, R. Pecori, and L. Veltri, “Estimation of a Gaussian source with memory in bursty impulsive noise,” in *2019 IEEE ISPLC*, 2019, pp. 1–6.

[34] A. Vannucci, G. Colavolpe, and L. Veltri, “Estimation of correlated Gaussian samples in impulsive noise,” *IEEE Commun. Letters*, vol. 24, no. 1, pp. 103–107, 2020.

[35] G. Colavolpe, A. Barbieri, and G. Caire, “Algorithms for iterative decoding in the presence of strong phase noise,” *IEEE J. Select. Areas Commun.*, vol. 23, no. 9, pp. 1748–1757, Sep. 2005.

[36] A. Barbieri and G. Colavolpe, “Simplified soft-output detection of CPM signals over coherent and phase noise channels,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2486–2496, Jul. 2007.

[37] G. Colavolpe, “Communications over phase noise channels: a tutorial review,” *International Journal of Satellite Communications and Networking*, vol. 32, pp. 167–185, May/June 2014, article first published online: July 2013.

[38] M. Peleg, S. Shamai (Shitz), and S. Galán, “Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel,” *IEE Proc. Commun.*, vol. 147, pp. 87–95, Apr. 2000.

[39] R. Krishnan, M. R. Khanzadi, T. Eriksson, and T. Svensson, “Soft metrics and their performance analysis for optimal data detection in the presence of strong oscillator phase noise,” *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2385–2395, 2013.

[40] A. Kreimer and D. Raphaeli, “Efficient low-complexity phase noise resistant iterative joint phase estimation and decoding algorithm,” *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 4199–4210, 2018.

[41] H. Mehrpouyan, A. A. Nasir, S. D. Blostein, T. Eriksson, G. K. Karagiannidis, and T. Svensson, “Joint estimation of channel and oscillator phase noise in MIMO systems,” *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4790–4807, 2012.

[42] B. D. Woerner and W. E. Stark, “Trellis-coded direct-sequence spread-spectrum communications,” *IEEE Trans. Commun.*, vol. 42, pp. 3163–3170, Dec. 1994.

[43] G. Colavolpe and A. Modenini, “Iterative carrier synchronization in the absence of distributed pilots for low SNR applications,” in *6th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TTC’13).* European Space Agency, 2013.

[44] S. Shayovitz and D. Raphaeli, “Message passing algorithms for phase noise tracking using tikhonov mixtures,” *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 387–401, 2016.

[45] T. Heskes and O. Zoeter, “Expectation propagation for approximate inference in dynamic Bayesian networks,” in *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, Jul. 2002.

[46] A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. Robert, “Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data,” 2014. [Online]. Available: https://arxiv.org/abs/1412.4869

[47] A. Mirbadin, A. Vannucci, and G. Colavolpe, “Expectation propagation and transparent propagation in iterative signal estimation in the presence of impulsive noise,” in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 175–179.

[48] A. Mirbadin, A. Vannucci, G. Colavolpe, R. Pecori, and L. Veltri, “Iterative receiver design for the estimation of gaussian samples in

impulsive noise," *Applied Sciences*, vol. 11, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/2/557

[49] G. Yao, H. Chen, and J. Hu, "An improved expectation propagation based detection scheme for MIMO systems," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2163–2175, 2021.

[50] H. Li, Y. Dong, C. Gong, Z. Zhang, X. Wang, and X. Dai, "Low complexity receiver via expectation propagation for OTFS modulation," *IEEE Commun. Letters*, vol. 25, no. 10, pp. 3180–3184, 2021.

[51] D. G. C. Bandeira, D. Le Ruyet, M. Pischella, and J. C. M. Mota, "Performance evaluation of low-complexity algorithms for orthogonal time-frequency space modulation," *Journal of Communication and Information Systems*, vol. 35, no. 1, p. 138–149, Jun. 2020. [Online]. Available: https://www.jcis.sbrt.org.br/jcis/article/view/691

[52] L. Szczecinski, H. Bouazizi, and A. Aharony, "Parametric phase tracking via expectation propagation," 2020. [Online]. Available: https://arxiv.org/abs/2005.01844

[53] A. Piemontese, G. Colavolpe, and T. Eriksson, "Discrete-time models and performance of phase noise channels," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2211–2225, 2024.

[54] M. Seeger, "Bayesian Gaussian process models: PAC-bayesian generalisation error bounds and sparse approximations," *PhD thesis, University of Edinburgh*, 2003.

[55] K. V. Mardia and P. E. Jupp, *Directional statistics.* Wiley Online Library, 2000, vol. 2.

[56] R. Hadani, S. Rakib, A. F. Molisch, C. Ibars, A. Monk, M. Tsatsanis, J. Delfeld, A. Goldsmith, and R. Calderbank, "Orthogonal time frequency space (OTFS) modulation for millimeter-wave communications systems,"

in *2017 IEEE MTT-S Intern. Microwave Symposium (IMS)*, 2017, pp. 681–683.

[57] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6501–6515, 2018.

[58] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications.* Academic Press, 2022.

[59] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, 2019.

[60] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Network*, vol. 35, no. 2, pp. 244–251, 2021.

[61] L. Gaudio, G. Colavolpe, and G. Caire, "OTFS vs. OFDM in the presence of sparsity: A fair comparison," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4410–4423, 2022.

[62] S. Buzzi, G. Caire, G. Colavolpe, C. D'Andrea, T. Foggi, A. Piemontese, and A. Ugolini, "Multi-satellite diversity through the use of OTFS," in *39th Intern. Commun. Satellite Systems Conf. (ICSSC 2022)*, 2022.

[63] S. Buzzi, G. Caire, G. Colavolpe, C. D'Andrea, T. Foggi, A. Piemontese, and A. Ugolini, "LEO satellite diversity in 6G non-terrestrial networks: OFDM vs. OTFS," *IEEE Commun. Letters*, vol. 27, no. 11, pp. 3013–3017, 2023.

[64] B. Shen, Y. Wu, J. An, C. Xing, L. Zhao, and W. Zhang, "Random access with massive MIMO-OTFS in LEO satellite communications," *IEEE J. Select. Areas Commun.*, vol. 40, no. 10, pp. 2865–2881, 2022.

[65] A. S. Bora, K. T. Phan, and Y. Hong, "Spatially correlated MIMO-OTFS for LEO satellite communication systems," in *2022 IEEE Intern. Conf. Commun. Workshops (ICC Workshops)*, 2022, pp. 723–728.

[66] ETSI, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) Radio Transmission and Reception, Version 8.6.0," document 3GPP TS 36.104, Jul. 2009.

[67] G. Colavolpe, D. Fertonani, and A. Piemontese, "SISO detection over linear channels with linear complexity in the number of interferers," *IEEE J. Sel. Topics in Signal Proc.*, vol. 5, pp. 1475–1485, Dec. 2011.

[68] L. Gaudio, M. Kobayashi, G. Caire, and G. Colavolpe, "On the effectiveness of OTFS for joint radar parameter estimation and communication," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5951–5965, 2020.

[69] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, "Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 902–915, 2014.

[70] F. Rusek and A. Prlja, "Optimal channel shortening for MIMO and ISI channels," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 810–818, Feb. 2012.

[71] H. L. Van Trees, *Detection, Estimation, and Modulation Theory - Part I.* John Wiley & Sons, 1968.

[72] S. Tiwari, S. S. Das, and V. Rangamgari, "Low complexity lmmse receiver for otfs," *IEEE Commun. Letters*, vol. 23, pp. 2205–2209, Dec. 2019.

[73] D. D. Falconer and F. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell System Tech. J.*, vol. 52, no. 9, pp. 1541–1562, Nov. 1973.

[74] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[75] `https://celestrak.org//`.

[76] A. Mirbadin, E. Kiani, A. Vannucci, and G. Colavolpe, “Estimation of gaussian processes in markov-middleton impulsive noise,” in *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, 2019, pp. 68–73.

Acknowledgements

During these three years of Ph.D. course, I had the opportunity to collaborate with people driven by a strong passion for their work, who, day by day, gave me new motivation to move forward.

First and foremost, I wish to express my deep gratitude to my supervisor Amina for having accompanied me along this challenging journey in the research world. Thank you for everything you have taught me, for your inspirational and expert guidance, encouragement, and the continuous scientific and moral support.

I am extremely grateful to Giulio for guiding me over the last eight years and shaping my professional growth from the very beginning. Thank you for all the opportunities, the valuable advices and feedbacks.

I want to extend my sincere gratitude to Armando for the insightful discussions and suggestions, and for his essential help in this research.

To each member of my research group, the SPADiC Lab, thank you for welcoming me on board. It has been a pleasure to work with you.

I want to thank Michelangelo, my colleague and friend, for adding a bit of *yellow* even in the darkest moments.

I am profoundly thankful to my parents and my sister Silvia for their unconditional love and precious support throughout every chapter of my life. Having you by my side is my greatest luck.

A special thanks to Federico for being a constant source of strength and motivating me to look beyond my fears.

I would like to dedicate a final heartfelt thought to my grandparents. Your light will always bright my path.