Doctoral Dissertation
Doctoral Program in Telecommunications Enginering (37.th cycle)

Advanced Joint Detection and Synchronization Techniques

Barbara Ripani

* * * * *

Supervisors
Prof. Guido Montorsi, Supervisor
PhD. Andrea Modenini, Co-supervisor

Doctoral Examination Committee:
Prof. G. Colavolpe, Università di Parma
Prof. D. Raphaeli, Tel Aviv University

Politecnico di Torino
December 2024

Summary

This thesis investigates the challenge of synchronization in coherent communication systems operating over time-selective channels. Coherent systems, known for their superior performance compared to non-coherent systems, rely on precise synchronization of phase, frequency, and timing between the transmitter and receiver. For many years, conventional receivers have relied upon the phase-locked loop (PLL) to achieve synchronization. PLLs are closed-loop systems that adjust the local oscillator's frequency and phase to align with the input signal. They have been widely adopted in various applications, including wireless communication and satellite navigation.

However, challenges arise due to variations in the channel caused by Doppler shifts, especially in scenarios with low signal-to-noise ratios. These conditions are common in deep-space communications, where residual carrier modulations are often employed to aid synchronization at the cost of power efficiency. Despite the simple design and ease of implementation of the PLL, it can struggle to or even fail in harsh propagation environments.

This thesis uses the factor graph and sum-product algorithm framework to focus on joint phase and frequency estimation. After establishing the foundation concepts and introducing the problem, a review of existing algorithms within this framework follows, examining their strengths and limitations in the context of time-selective fading channels. The purpose is to lay a foundation for deriving new detection and synchronization algorithms specifically designed for challenging propagation environments. These algorithms have been categorized based on whether they address phase noise alone or both phase noise and frequency uncertainties. These contributions aim to enhance system performance and reliability in adverse conditions, providing more robust solutions compared to traditional techniques.

Acknowledgements

I am grateful to my supervisor, Prof. Guido Montorsi, and my co-supervisor, Dr. Andrea Modenini, whose expertise and guidance were vital to the development and completion of this PhD. Their insights and constructive feedback shaped the direction of this thesis and pushed me to reach new levels in my academic journey.

Contents

List of Tables

List of Figures

List of Acronyms

AIR Achievable Information Rate

APP A Posteriori Probability

AWGN Additive White Gaussian Noise

APSK Amplitude Phase-shift Keying

BCJR Bahl-Cock-Jelinek-Raviv

BER Bit Error Rate

BPSK Binary Phase-shift Keying

CCSDS Consultative Committee for Space Data Systems

CER Code Error Rate

CMVM Circular Mean and Variance Matching

dp-BCJR Discrete Phase BCJR

EXIT Extrinsic Information Transfer

ISI Inter-symbol Interference

KF Kalman Filter

LDPC Low Density Parity Check

LEO Low Earth Orbit

LLR Log-likelihood Ratio

LPF Low-pass Filter

MAP Maximum A Posteriori

MF Matched Filter

MSE Mean Square Error

NRZ Non-return-to-zero

PLL Phase Locked Loop

pdf Probability density function

pmf Probability Mass Function

QAM Quadrature Amplitude Modulations

QPSK Quaternary Phase-shift Keying

RSC Recursive Systematic Convolutional

SISO Soft-Input Soft-Output

SNR Signal-to-Noise Ratio

SPA Sum-Product Algorithm

TC Telecommand

Mathematical Notations

Throughout the thesis, the following notation will be adopted.

- Matrices are denoted using bold, underlined uppercase letters (e.g. $\underline{\mathbf{A}}$).
- The transpose of a matrix or vector is indicated by the superscript $^\top$ (e.g. $\underline{\mathbf{A}}^\top$ or $\mathbf{v}^\top$).
- The determinant of a matrix $\underline{\mathbf{A}}$ is indicated as $|\underline{\mathbf{A}}|$.
- The Moore-Penrose pseudo-inverse of a matrix is denoted by the superscript $\dagger$ (e.g. $\underline{\mathbf{A}}^\dagger$).
- Vectors are represented by bold lowercase ($\mathbf{v}$) or uppercase ($\mathbf{V}$) letters.
- The standard basis vectors $\mathbf{e}_i \in \mathbb{R}^n$ are defined as:

$$\mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad (1 \text{ in the } i\text{-th position, } 0 \text{ elsewhere}), \quad \text{for } i = 1, 2, \ldots, n.$$

- The complex conjugate of a complex number is denoted by the superscript * (e.g. a^*).
- The univariate Gaussian distribution is denoted as

$$\mathcal{N}(x; \mu, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

  where x is the random variable, μ is the mean, and σ^2 is the variance of the distribution.

- The multivariate Gaussian distribution is denoted as

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \underline{\boldsymbol{\Sigma}}) \triangleq \frac{1}{(2\pi|\underline{\boldsymbol{\Sigma}}|)^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \underline{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

where $\mathbf{x}$ is the random vector, $\boldsymbol{\mu}$ is the mean vector, and $\underline{\boldsymbol{\Sigma}}$ is the covariance matrix.

- The univariate von Mises (Tikhonov) distribution is denoted as

$$t(x; z) \triangleq \frac{1}{2\pi I_0(|z|)} \exp\left\{\Re\left[ze^{-jx}\right]\right\}$$

where x is the value of the random variable, and z the complex coefficient.

- The cosine model of the bivariate von Mises is defined as

$$t(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\kappa}) \triangleq \exp\left\{k_1 \cos(x_1 - \mu_1) + k_2 \cos(x_2 - \mu_2) - k_3 \cos(x_1 - \mu_1 + x_2 - \mu_2)\right\} ,$$

where $\mathbf{x}$ is the value of the random vector, $\boldsymbol{\mu} = \begin{bmatrix}\mu_1 & \mu_2\end{bmatrix}^\top$ is the mean and $\boldsymbol{\kappa} = \begin{bmatrix}k_1 & k_2 & k_3\end{bmatrix}^\top$ is the vector of the concentration parameters.

Chapter 1

Introduction

This thesis focuses on the synchronization in coherent communication systems over time-selective channels. Coherent communication systems are known to outperform non-coherent systems. However, they require precise synchronization between the transmitter and the receiver. On the other hand, non-coherent systems, encode information in phase differences between successive transmissions, eliminating the need for precise synchronization and allowing signal recovery without phase ambiguity. Since this approach is simpler, it is preferred when sub-optimal error performance can be accepted [1]. Accurately estimating the transmission parameters, such as phase, frequency, and timing, is necessary in coherent systems. Proper detection can only occur after these reference parameters have been estimated at the receiver side. Phase and frequency estimation is crucial for recovering the carrier frequency, enabling the local oscillator to align the receiver's reference signal with the transmitted signal. Timing recovery is necessary to identify signal peaks, ensuring proper signal sampling. Depending on the communication architecture adopted, it may be required to estimate additional parameters, in addition to phase, frequency, and timing, before signal detection.

The term "time-selective channels" is commonly linked to fading processes; however, in this context, it is used in its standard definition. Specifically, it refers to channels that are influenced by temporal variations due to the relative motion between the transmitter and receiver, which results in Doppler shifts. These channels exhibit changes over time, with the degree of fluctuation being directly related to the relative velocity between the transmitter and the receiver. When the duration of a transmission symbol exceeds the coherence time—the time interval during which the channel can be considered approximately constant—the transmitted signal experiences distortion. A practical example of a time-selective channel is the communication between a ground station on Earth and a satellite in orbit, where the high relative speeds lead to significant time-varying effects on the signal. Communication becomes more challenging when the distance between the transmitter

and the receiver increases. For instance, synchronization in deep space is complicated due to the low signal-to-noise ratios (SNRs) resulting from the large distances involved. Residual carrier modulations can be used on the transmitter side to address this issue. These modulations involve a permanent pilot (the residual carrier) that aids synchronization. This comes at the cost of a small portion of the total power allocated to the carrier transmission. The consultative committee for space data systems (CCSDS) has proposed adopting residual carrier modulations for the up-link (Earth-to-space) in deep space missions [2]. In a low SNR environment affected by highly varying Doppler dynamics, using residual carrier modulations is necessary to ensure a reliable communication link [3].

One of the main factors affecting precise synchronization between the transmitter and the receiver is phase noise. This noise arises due to instabilities in the local oscillators of the transmitter and receiver, and its severity depends on the quality of these components and the modulation format and constellation cardinality. In some applications, there may be discrepancies in the quality of local oscillators at the transmitter and receiver, often due to cost constraints. For example, in the global navigation satellite system, the local oscillator on the satellites is very precise, but the receivers, such as mobile phones, usually have low-cost ones. A misalignment in the local oscillators of the transmitter and receiver can lead to severe performance degradation [4]. Being phase noise an impairment that affects any communication system and can prevent a system from operating correctly, it has been a focus of several studies over the past years [4]–[7].

For several years, conventional receivers have relied on PLLs to achieve synchronization across various applications, ranging from wireless communications to satellite navigation. Examples include NASA's deep space network [8], the deep space transponder used in several European projects for carrier acquisition and tracking [3], [9], and global positioning systems [10], [11]. However, as channel conditions become more complex and synchronization becomes increasingly challenging, more advanced synchronization techniques have been explored. To achieve performance approaching capacity when the receiver does not know the transmission reference parameters, iterative detection and decoding schemes have been studied for different channel models. Examples are [5], [12]–[15] which adopt the factor graph and sum-product algorithm (SPA) framework [16], [17] to derive joint detection and decoding algorithms. The authors in [12], [13] assumed perfect synchronization and developed joint channel estimation and data detection algorithms. While [13] focused on a channel model with Rayleigh fading and co-channels inference, [12] considered a correlated Rayleigh fading channel. Focusing on phase tracking in the presence of strong phase noise, the authors in [5] and [14] used the same framework to derive iterative detection and decoding algorithms. In [15], the authors considered coupling the problems of phase noise and channel estimation in time-selective fading channels with data detection and developed a joint detection and decoding algorithm.

This thesis addresses the problem of phase and frequency estimation in time-selective channels, utilizing the factor graphs and SPA framework. Phase noise estimation has been widely studied, with numerous approaches developed for signal detection and decoding over phase noise channels. These algorithms can generally be divided into two groups: those that jointly perform detection and decoding, and those that treat these tasks separately. The first group of algorithms modifies the encoder to estimate the phase jointly with the soft decoder output (see [6] and references therein). For instance, rotationally invariant codes have been used to derive low-complexity solutions that implement the exact *maximum a posteriori* (MAP) symbol detection strategy with a forward-backward phase probability density function estimator [18]. The resulting joint detection and decoding algorithm do not require pilot symbols and has been proven to be robust [6]. The second class considers the detector and decoder blocks to work separately in a soft-decision-directed mode [5], [14]. This can sometimes involve iteratively exchanging soft information to refine the reliability of their estimation. While effective, these iterative methods are often computationally complex and unsuitable for real-time applications due to memory and latency constraints.

Concerning the problem of joint phase and frequency estimation, the authors in [3] highlighted the superiority of Kalman-based synchronization techniques to replace the PLLs for carrier synchronization in challenging environments. The Kalman filter [19] is an optimal Bayesian estimator widely adopted to predict the state of a system using its noisy observations. Building on the work in [20], the authors in [21] adopt a bank of extended Kalman filters to develop a nonlinear Bayesian estimator for phase and frequency tracking. However, this work is limited to the case of analog demodulation of phase and frequency-modulated signals. Joint phase and frequency estimation in the digital domain, particularly in time-selective channels, remains an under-explored area. To the best of the authors' knowledge, only the authors in [22] addressed the problem and developed an iterative detection and decoding algorithm for channels affected by time-varying phase noise and a frequency offset. This thesis specifically focuses on the design of joint detection and synchronization algorithms optimized for challenging propagation environments with poor SNRs and low data rates. Such environments are characterized by rapidly varying phase and frequency noise processes, which exhibit higher temporal variability than the dynamics of the transmitted signal. Addressing these challenges is crucial for improving the performance and reliability of communication systems under adverse conditions.

The remainder of this thesis is organized as follows. Chapter 2, provides an overview of the channel models found in the literature and discusses potential models for describing the phase process. Additionally, it introduces fundamental concepts at the basis of the factor graph and SPA-based techniques which will be explored in the thesis. Chapter 3, describes the reference state-of-the-art detection algorithms based on the said framework. These algorithms will be categorized

into two groups based on the tracking problem they aim to solve: those addressing phase only, and those addressing both phase and frequency variations. The chapters that follow represent the primary contributions of this thesis. Chapter 4 establishes a connection between the algorithms derived from the factor graphs and SPA framework and standard techniques like the PLL. This link is intended to assist readers who are less familiar with these advanced algorithms in placing them into a more recognizable context. First, the chapter derives the Kalman filter as a specific instance of the SPA while recalling the duality with a PLL. Additionally, it introduces an alternative family of PLLs based on the Tikhonov distribution, and compares these methods in terms of their inherent differences, tracking accuracy, and potential advantages in phase noise channels. Chapter 5 explores the use of oversampling techniques to develop detection algorithms capable of phase tracking in strong phase noise channels. It begins by introducing an algorithm in the Bahl-Cock-Jelinek-Raviv (BCJR, [23]) form, followed by a discussion of a reduced-complexity algorithm based on the Tikhonov distribution. Chapter 6 addresses the challenges of phase and frequency tracking through a case study on telecommand (TC) reception in deep space missions. Advanced synchronization algorithms to track both the carrier and sub-carrier in low SNR environments accounting for phase and frequency uncertainties are proposed. Chapter 7 takes a different approach by utilizing Gaussian mixtures to develop an algorithm for the simultaneous detection and synchronization of phase and frequency. Finally, conclusions are presented in Chapter 8.

Chapter 2

Technical Background

This chapter provides essential background information to lay the foundation for the rest of the thesis. It introduces key concepts and frameworks to help the reader understand the following chapters. The remainder of this chapter is organized as follows. Section 2.1 introduces the basic channel models that will be analyzed throughout the thesis, setting the stage for further discussion and application. Next, Section 2.2 presents existing models used to describe the phase process, while Section 2.3 briefly discusses different pilot distribution strategies. Section 2.4 covers the development of iterative techniques, beginning with turbo codes and the underlying turbo principle, which laid the groundwork for modern coding strategies. Section 2.5 examines iterative decoding methods based on factor graphs and the message-passing sum-product algorithm. Finally, Section 2.6 introduces the MAP detection strategy and demonstrates how it can be derived within the frameworks of factor graphs and the sum-product algorithm.

2.1 Time Selective Channels

Accurate channel modeling is crucial for the development of effective detection and synchronization algorithms. This section provides an overview of the channel models that will be examined throughout this thesis. The goal is to establish a solid mathematical framework essential for the subsequent development and analysis of the joint detection and synchronization algorithms. A general discrete-time complex base-band channel model that takes into account the time selectivity of the channel is

$$y_k = a_k x_k + w_k \,, \tag{2.1}$$

where x_k is the symbol transmitted at the k-th time interval, w_k are independent identically distributed (i.i.d.) complex Gaussian random variables with a variance σ^2 per component. In the literature, it is common to find two models to describe the channel gain process a_k:

- **Complex circular model** In this model, the channel coefficient a_k is a complex-valued fading process described as

$$a_k = a_k^R + j a_k^I \,, \tag{2.2}$$

where the real and imaginary components a_k^R and a_k^I evolve as independent Gaussian processes. This model captures the essential characteristics of a time-correlated fading channel. However, it does not account for phase fluctuations, such as those introduced by the instabilities of the oscillator, which are significant in practical communication systems.

- **Log-normal complex model** The log-normal model is described by

$$a_k = e^{h_k + j\theta_k} \,, \tag{2.3}$$

where h_k and θ_k are independent single or multi-pole filtered Gaussian processes (e.g. [24]). This model is particularly useful in scenarios where the channel experiences multiplicative log-normal fading, which is common in environments with shadowing effects [25]. Here, h_k represents the log-normal fading component while θ_k models the phase noise. When h_k is constantly zero over time, the channel model in (2.3) simplifies to

$$y_k = x_k e^{j\theta_k} + w_k \,, \tag{2.4}$$

a widely adopted phase noise channel model [6]. This simplification is often used to isolate and study the impact of phase noise on system performance, and it will be the reference model in the following chapter.

2.2 Phase Process Models

There are several models used in the literature to simulate the discrete phase process $\{\theta_k\}_{k=0}^{K-1}$. This section will describe the most common models found in the literature that will be referenced throughout this thesis.

2.2.1 Wiener Model

The discrete-time Wiener model is commonly adopted in the literature to study the performance of communications systems affected by phase noise. According to the Wiener model, the difference between two adjacent phase noise samples is distributed according to a Gaussian distribution

$$\theta_k - \theta_{k-1} \sim \mathcal{N}(\theta_k - \theta_{k-1}; 0, \sigma_\Delta^2) \,. \tag{2.5}$$

Therefore, the sequence of phase samples evolves as

$$\theta_k = \theta_{k-1} + \delta_k \,, \tag{2.6}$$

where δ_k is a i.i.d. real Gaussian random variable having zero mean, variance σ_Δ^2. The term δ_k can be interpreted as the instantaneous value of a white frequency noise process, given by the difference between two successive phase noise samples. From (2.6), it is possible to recognize a type of Markov process, where the probability of the system of being in the state θ_k at time k only depends on the previous state θ_{k-1}. Mathematically,

$$p(\theta_k|\theta_{k-1}, \theta_{k-2}, ..., \theta_0) = p(\theta_k|\theta_{k-1}) = p_\Delta(\theta_k - \theta_{k-1}) \,, \tag{2.7}$$

with

$$p_\Delta(\theta) = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \sum_{i=-\infty}^{\infty} \exp\left\{-\frac{(\theta - 2\pi i)^2}{2\sigma_\Delta^2}\right\} \; \theta \in [0,2\pi] \,, \tag{2.8}$$

being a wrapped Gaussian pdf in $[0, 2\pi]$.

The Wiener phase noise model is popular because of its simplicity. It can be characterized by a single parameter: the variance of the phase increments. Additionally, it benefits from Markovian properties, making it suitable for simply deriving detection algorithms. It is considered a valid model as it has been shown to accurately describe the phase noise process in terms of real parameters of local oscillators [26]. Even though more accurate phase noise models have been proposed (see [26] and references therein), algorithms designed using the Wiener model have been proven to perform well under those more accurate phase noise models.

2.2.2 Stochastic Phase and Time-Varying Frequency Offset

A commonly adopted model that describes not only the phase noise but also variations of the signal due to relative movement between the transmitter and the receiver, is the following

$$\theta_k = \theta_{k-1} + 2\pi f_{k-1} T + \pi \dot{f}_{k-1} T^2 + \delta_k \tag{2.9a}$$

$$f_k = f_{k-1} + \dot{f}_{k-1} T + \varepsilon_k \tag{2.9b}$$

$$\dot{f}_k = \dot{f}_{k-1} + \rho_k \,, \tag{2.9c}$$

where δ_k, ε_k, and ρ_k, are three i.i.d Gaussian random variables having zero mean and variance σ_Δ^2, σ_f^2, and $\sigma_{\dot{f}}^2$, respectively. The model has been widely adopted to derive PLL and PLL-like algorithms [27]–[29] able to track the phase of a signal affected by phase noise and time-varying frequency offset. This model is beneficial in scenarios where the transmitter and receiver are subject to dynamic environments, such as relative motion (e.g., Doppler effects in vehicular or satellite communications), non-ideal synchronization, or fluctuating propagation conditions. The

stochastic components of the model, represented by the random variables δ_k, ε_k, and ρ_k, account for imperfections in the system, such as oscillator drift, jitter, and rapid changes in frequency, making it highly adaptable for analyzing real-world conditions.

2.2.3 Deterministic Phase and Time-Varying Frequency Offset

Another discrete-time phase model follows a deterministic evolution

$$\theta_k = \theta_{k-1} + 2\pi \left(f_{k-1} kT + \frac{1}{2} \dot{f}_{k-1} k^2 T^2 \right) . \tag{2.10}$$

In this model, the phase at the k-th discrete time interval, θ_k, depends on the previous time step's phase, frequency, and frequency drift. The first term inside the parentheses, accounts for the linear accumulation of phase due to the constant frequency offset, leading to a uniform phase shift over time. The second term instead represents the additional phase shift arising from the frequency drift, which introduces a nonlinear, quadratic dependence on time. This quadratic term indicates that, as time progresses, the effect of the frequency drift becomes more pronounced, resulting in faster phase accumulation. This model has been widely adopted and it is at the basis of the PLL theory [30]. Additionally, by incorporating a Gaussian random variable into the model in (2.10), it is possible to account for a phase noise according to the Wiener model, adopted in [22].

2.2.4 Sinusoidal Doppler Model

The sinusoidal Doppler model simulates the Doppler effect caused by relative motion between the transmitter and receiver, particularly in environments with periodic motion, such as satellite or vehicular communications. The phase evolution due to Doppler can be expressed as

$$\theta(t) = 2\pi \int_{-\infty}^{t} f_D \sin \left(\frac{f_R}{f_D} \tau \right) d\tau . \tag{2.11}$$

The integral in (2.11) evaluates to

$$\theta(t) = -2\pi \frac{f_D^2}{f_R} \cos \left(\frac{f_R}{f_D} t \right) , \tag{2.12}$$

where f_D represents the maximum Doppler shift, and f_R is the frequency drift mathematically represented by the slope of the sinusoidal function. The sinusoidal model effectively simulates the Doppler shift caused by oscillatory relative motion, such as a satellite orbiting around the Earth. In this model, the frequency shift

varies periodically as the satellite moves toward and away from the receiver on the Earth, creating a sinusoidal pattern. This is particularly useful when the satellite repeatedly passes over the same region, causing a predictable Doppler shift that varies with time. The model allows for accurate simulation of phase changes due to Doppler effects in systems where the relative velocity is not constant but instead follows a cyclic or sinusoidal trajectory. The sinusoidal Doppler model has been standardized for use in computer simulations, such as in satellite-to-ground links [31], providing a practical framework to evaluate system performance under realistic motion and propagation conditions. Also in this case, it is possible to account for phase noise by including a Gaussian random variable into (2.11).

2.3 Pilot Symbols for Detection and Tracking

Pilot symbols play a crucial role in the detection schemes discussed throughout this thesis. They are particularly significant in scenarios where no other information is available enabling reliable signal detection and estimation in varying channel conditions. There are several strategies for inserting pilot symbols within the transmitted signal. One common approach is to distribute pilot symbols at regular intervals. This periodic placement ensures consistent references, facilitating iterative decoding processes. For instance, studies such as [5], [14] have demonstrated that inserting a pilot for every 20 symbols can yield satisfactory performance. An alternative strategy involves sending pilot symbols in bursts, as in the case of the DVB-S2 standard [32]. This burst-based approach concentrates pilot information within specific time intervals, which can be advantageous in rapidly varying channels. These pilot insertion strategies will be referred to as *interspersed*. Another strategy is the transmission of *superimposed* pilot symbols, obtained by substituting

$$x_k = c_k + \gamma p_k \,, \tag{2.13}$$

in (2.1). In this case, x_k represents a linear combination of c_k, the linearly modulated symbol, and p_k the pilot symbol [33]. The power dedicated to the pilots is scaled by a factor γ, which represents the portion of power allocated to the transmission of pilots compared to the total power. Superimposed pilots have been considered in several studies [33]–[35], offering improved power and bandwidth efficiency compared to the classical interspersed technique. The effectiveness of different methods for distributing pilot symbols depends on specific application requirements and the detection algorithm used. Each method has advantages and limitations depending on the application scenarios. In the following sections, we will explore how the interspersed and superimposed insertion strategies impact the capability of the detection algorithm to achieve good performance.

2.4 The Turbo Principle

In 1993, the invention of turbo codes marked a significant milestone in the field of communication theory. These error correction codes were capable of achieving performance close to the Shannon limit [36]. Turbo codes were originally designed by interconnecting two constituent encoders, recursive systematic convolutional codes, in a parallel concatenation scheme, as shown in Figure 2.1. Figure 2.1 depicts how the input information bits are processed by two convolutional encoders, one at the upper side and the other at the lower side. The upper encoder takes the input bits, while the interleaved version of the same bits is given to the lower encoder. The output stream is a combination of the information bits (systematic) and the parity outputs from both encoders. However, the systematic output generated by the lower encoder is the same as that of the upper encoder, but in a different order due to the interleaving process. Therefore, it is redundant and can be discarded. In the block diagram, the overall rate is $\frac{1}{3}$, but higher code rates can be obtained by puncturing, i.e. removing some parity bits from the codeword according to a specific matrix [37]. The interleaver plays a key role in this coding scheme by permuting the input of the upper encoder before it is fed to the lower encoder, which is crucial for improving error correction performance. The *turbo principle*—the decoding principle of turbo

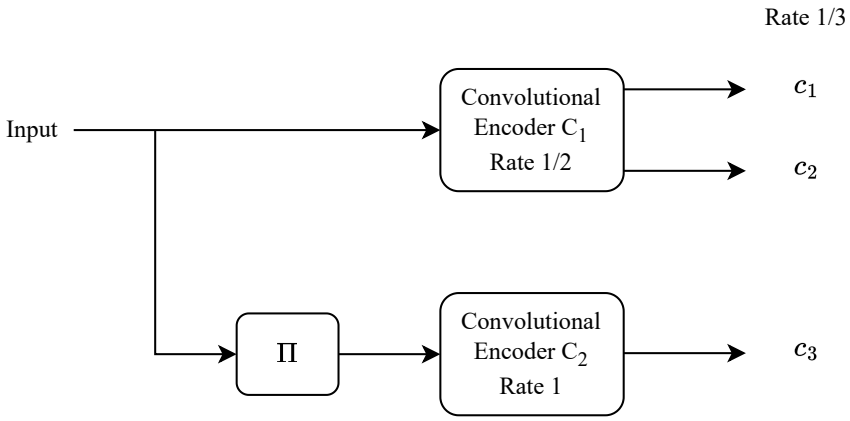


Figure 2.1: Block diagram of a turbo encoder.

codes—involves two decoding processes, one for each constituent code, as shown in Figure 2.2. The two decoders are soft-input soft-output (SISO) modules in a cascade that implement the *a posteriori* probability (APP) or maximum MAP strategy. They iteratively exchange *extrinsic information*, meaning that the soft information output from one decoder is used as the *a priori* soft input information for the second one. This process continues until a reliable decision on the bits is reached.

The SISO module is a device with four ports that takes in two sequences of

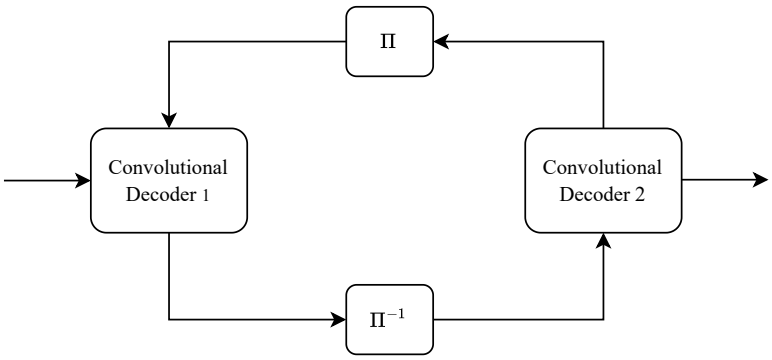


Figure 2.2: Block diagram of a turbo decoder.

likelihoods. These correspond to the information bits ($P(u, I)$) and the coded bits ($P(c, I)$) and are used, together with their knowledge of the trellis representation of the code to obtain the updated versions of the input likelihoods on the information and coded bits, denoted as $P(u, O)$ and $P(c, O)$, respectively. In the context of turbo decoding, $P(u, O)$ and $P(c, O)$ are referred to as *extrinsic information*, which represents the information on a bit given by the collective knowledge of all the bits excepts itself. A sketch of the SISO module can be found in Figure 2.3. The intro-

Figure 2.3: The soft-input soft-output module

duction of turbo codes and iterative decoding has led to significant advancements in communication theory and practical applications. Iterative decoding has revolutionized receiver architectures by enabling the exchange of soft information between decoding processes. This breakthrough has paved the way for the development of more sophisticated error correction techniques and advanced modulation schemes. In addition, the turbo principle, on which turbo codes are based, was found to be useful not only for decoding problems but also for several other applications such as detection and equalization [38]. As communication systems evolve, turbo codes and iterative decoding remain crucial technologies that drive further advancements in data transmission and reception.

2.4.1 Turbo Equalization

Turbo equalization is a powerful technique that builds upon the principles of turbo decoding, and it is widely used in signal processing and communications systems. It operates iteratively, performing both equalization and decoding simultaneously. The concept of turbo equalization was first introduced in 1995 by the authors in [39], which inspired a wave of research in this field. Since then, many researchers have explored various algorithms that leverage turbo decoding approaches to solve challenging problems in signal processing and communications [40]–[42]. A basic end-to-end communication architecture is summarized in Figure 2.4 that sketches the key constituent of a transmitter and a receiver necessary for turbo equalization. In transmission, the input information stream is encoded introducing redundancy bits to allow error correction at the receiver side. After encoding, the coded bits are permuted by the interleaver to spread the coded bits to minimize the damage caused by bursts of error at the receiver end. Finally, the coded bits are mapped into modulation symbols and transmitted through the channel. At the receiver

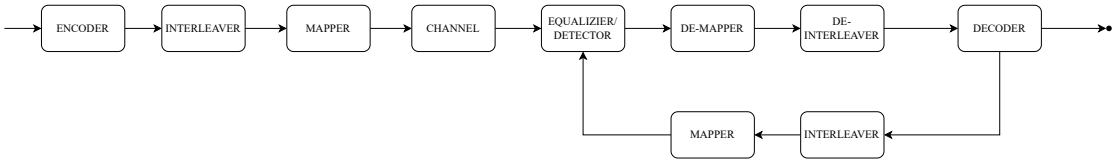


Figure 2.4: End-to-end communication architecture for the turbo equalizer.

side, the turbo equalizer performs joint equalization and decoding iteratively, using the soft information from both the equalizer and the extrinsic information from the turbo decoder, before proper permutation and mapping (see Figure 2.4). The final output of the turbo equalizer is an estimate of the transmitted information bits. The process of exchanging likelihoods (beliefs) between the two algorithms (equalizer/detector and decoder) is also referred to as "belief propagation" or "message passing".

2.5 Iterative Decoding Based on Factor Graphs

Graphical representations have played a key role in understanding the structure of error-correcting codes and designing efficient decoding algorithms. The trellis diagram is an example of such a representation, which naturally describes the Viterbi decoding algorithm [43]. Later on, in 1981, the Tanner graph [44] was introduced to describe and comprehend the structure of the low-density parity-check (LDPC) codes, which were originally proposed by Gallager [44], and other linear codes. Tanner suggested using a bipartite graph, which is a type of graph that has nodes divided into two separate groups. The graph's edges connect nodes from

different groups but not those within the same group. The bipartite graph structure represents the parity-check equations of the code, where one group of nodes represents variables, and the other group represents equations. An edge between a variable node and a parity-check node indicates that the variable is part of the corresponding equation. In 1995 and 1996, the authors of [45], [46] made progress in using iterative detection and decoding. They recognized that algorithms such as the Viterbi, BCJR [23], and decoding of turbo and LDPC codes were instances of a message-passing algorithm operating in a generalized Tanner graph [47]. All these algorithms encounter a marginalization problem that needs to be solved. For instance, implementing the MAP symbol (e.g. the BCJR algorithm) or sequence (e.g. the Viterbi algorithm) detection strategy requires solving this marginalization problem. It became evident that the complexity of solving a marginalization problem could be significantly reduced by properly applying the distributive law to the factorization of a global function. Factorization can be visualized using a bipartite graph, the factor graph, which consists of variable nodes (representing the variables) and the factor nodes (representing the functions or constraints) connected through edges to show which variable is an argument of which function. This representation helps to understand the relationships between variables and functions better.

The message-passing algorithms offer an efficient approach to solving the marginalization problem. At the same time, the factor graph framework provides an environment to take advantage of instances where the distributive law can be applied [48]. In the message-passing algorithm approach, the messages exchanged in the factor graph represent marginal probabilities of subsets of variables or factors in the graph. These partial marginals are then combined to derive the overall marginal distribution of the global function. The SPA is one of the most widely used message-passing techniques for computing local functions associated with a global function. Moreover, when the factor graph is cycle-free, it allows for the exact computation of marginal functions. In [16], the authors realized that message-passing algorithms on a factor graph could be used in signal processing. Following their work, several studies adopted the factor graph and SPA framework to develop different algorithms. Examples are the works in [5], [49]–[51].

2.5.1 Factorization in Graphical Models: An Example

This section provides an example of how to represent a factorization using the factor graph and how the SPA operates within it. Assume that some function $g(x_1, x_2, x_3)$ factorizes as

$$g(x_1, x_2, x_3) = f_1(x_1) f_2(x_1, x_2, x_3) f_3(x_2, x_3) \tag{2.14}$$

$$= \prod_i f_i(X_i)\,, \tag{2.15}$$

with $f_i(X_i)$ being a function of the argument X_i, a subset of the set of variables $\{x_1, x_2, x_3\}$. The factorization in (2.14) is represented by the cycle-free factor graph in Figure 2.5. In this graph, each variable in (2.14) is associated with a variable node (circle), and each function is associated with a factor node (square). A variable node is connected to a factor node with an edge when the variable of the variable node is included in the function represented by that factor. The SPA computes,

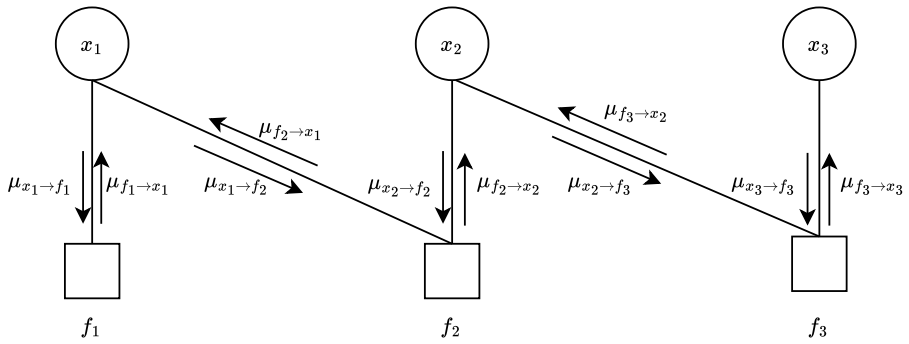


Figure 2.5: Factor graph representing (2.14)

at each node, a message based on the incoming messages it receives and its local information. For nodes representing factors, this involves computing the product of incoming messages and the local function associated with the node. Once the message is computed, the node transmits it along its outgoing edges, but it excludes any information received on the same edge to prevent redundancy. The algorithm begins at the leaf nodes of the graph, which have only one connection. If the leaf node represents a factor, it transmits a message representing that factor. If it means a variable, it transmits a placeholder message known as a "unit" function. The other nodes wait until they receive the messages from all but one of its incoming edges before computing the outgoing message. In the example shown in Figure 2.5, the computation of the messages exchanged along the factor graph follows these steps.

1.
$$\mu_{f_1\to x_1} = \sum_{\sim\{x_1\}} f_1(x_1)$$

$$\mu_{x_3\to f_3} = 1$$

2.
$$\mu_{x_1\to f_2} = \mu_{f_1\to x_1}$$
$$\mu_{f_3\to x_2} = \sum_{\sim\{x_2\}} \mu_{x_3\to f_3} f_3(x_2, x_3)$$

3.
$$\mu_{f_2 \to x_2} = \sum_{\sim\{x_2\}} \mu_{x_1 \to f_2}$$
$$\mu_{x_2 \to f_2} = \mu_{f_3 \to x_2}$$

4.
$$\mu_{x_2 \to f_3} = \mu_{f_2 \to x_1}$$
$$\mu_{f_2 \to x_1} = \sum_{\sim\{x_2\}} \mu_{x_2 \to f_2} f_2(x_1, x_2, x_3)$$

5.
$$\mu_{f_3 \to x_3} = \sum_{\sim\{x_3\}} \mu_{x_2 \to f_3} f_3(x_2, x_3)$$
$$\mu_{x_1 \to f_1 = \mu_{f_2 \to x_1}}$$

The operation $\sum_{\sim\{x_i\}}$ is called "not-sum" or "summary" and it is defined as

$$h_i(x_i) = \sum_{\sim\{x_i\}} h(x_1, x_2, ..., x_n) \tag{2.16}$$
$$= \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} ... \sum_{x_{i-1} \in C_{i-1}} \sum_{x_{i+1} \in C_{i+1}} ... \sum_{x_n \in C_n} h(x_1, x_2, ..., x_n)\,, \tag{2.17}$$

with x_j taking values on C_j, and $h_i(x_i)$ being the i-th marginal associated to the function $h(x_1, x_2, ..., x_n)$. In this example, it is clear that the exchanged messages $(\mu_{f_i \to x_j},\ \mu_{x_j \to f_i})$ represent marginals of subsets of the global function. This iterative process continues until convergence, with each node transmitting messages and updating its local computations. Through this systematic message-passing approach, the sum-product algorithm efficiently computes marginal probabilities and facilitates various probabilistic inference tasks within graphical models.

2.6 MAP Symbol Detection

Before the development of turbo codes and iterative decoding, the most commonly used strategy for detecting symbols was the MAP sequence detection strategy. This strategy was preferred over MAP symbol detection due to its lower complexity. The MAP symbol detection strategy focused on minimizing the bit error probability rather than the sequence error probability. However, the MAP symbol detection algorithms provide the APPs on the transmitted symbols, the desired output for iterative decoding. This makes it a favorable detector choice in iterative schemes.

Consider a sequence of code symbols that are linearly modulated and transmitted over an additive white Gaussian noise (AWGN) channel. The discrete-time base-band equation describing the received signal is

$$y_k = c_k + w_k \,, \tag{2.18}$$

where the symbol c_k belongs to a complex M-ary constellation, and w_k is an i.i.d. complex Gaussian random variable with zero mean and variance σ^2 per component. To detect the information bits given the noisy observable in (2.18), one can adopt the MAP symbol detection strategy that can be expressed as

$$\hat{a}_i = \underset{a_i}{\operatorname{argmax}}\, p(a_i|\mathbf{y}) \,, \tag{2.19}$$

where $\mathbf{y}$ is the vector containing all the received sample, a_i is the i-th information bit, and $\hat{a}_i$ the one detected. The *a posteriori* probability mass function (pmf) of the i-th information bit given the received signal can be obtained by marginalizing the joint posterior probability distribution function (pdf) $p(\mathbf{a}|\mathbf{y})$ with respect to all a_j with $j \neq i$ as

$$p(a_i|\mathbf{y}) = \sum_{\sim a_i} p(\mathbf{a}|\mathbf{y}) \,. \tag{2.20}$$

In the case of the channel model in (2.18), given the encoding function ϕ_c, the joint posterior pdf factorizes as

$$p(\mathbf{a}|\mathbf{y}) \propto P(\mathbf{a}) p(\mathbf{y}|\mathbf{a}) \tag{2.21}$$

$$\propto I(\mathbf{c} = \phi_c(\mathbf{a})) \prod_{k=0}^{K-1} p(y_k|c_k) \,, \tag{2.22}$$

where $I(\mathbf{c} = \phi_c(\mathbf{a}))$ is the code indicator function, equal to 1 when $\mathbf{c}$ is the codeword corresponding to $\mathbf{a}$, and to zero otherwise, and

$$p(y_k|c_k) = \mathcal{N}(y_k; c_k, \sigma^2) \tag{2.23}$$

is a Gaussian distribution of the variable y_k. In this thesis, the notation $y_k \sim \mathcal{N}(y_k; c_k, \sigma^2)$ is adopted for representing a Gaussian distribution in the variable y_k having mean value c_k and variance σ^2. A factor graph can visualize the factorization in (2.22). Applying the SPA to the factor graph representing the factorization of the joint posterior pdf enables the exact computation of the MAP symbol detection algorithm or an approximation in the presence of cycles in the factor graph [5].

Chapter 3

Advanced Detection Algorithms for Carrier Recovery

This chapter addresses the challenge of carrier and data recovery affected by phase noise alone or by a combination of phase noise and frequency uncertainties. One well-known technique for carrier tracking is the PLL [30]. The PLL's capabilities vary based on the filter order: it can track phase variations (first-order or higher), phase variations along with a constant frequency offset (second-order or higher), and both phase and frequency variations (third-order or higher). While the PLL is widely used across various applications, its effectiveness is generally limited to scenarios with favorable propagation conditions. In more challenging environments, such as those characterized by highly varying noise dynamics, the PLL may struggle or even fail to maintain accurate synchronization [3]. In contrast, the Kalman filter is a widely recognized algorithm for estimating phase and frequency in channels with phase noise and frequency drift. Although it gained popularity in different fields [52]–[54], its adoption within the synchronization community has been limited, partly due to its complexity and computational demands [28]. Nevertheless, in the context of this thesis, the Kalman filter——and, by extension, the Kalman smoother——is considered a benchmark for accurate phase, frequency, and frequency drift estimation. As it will be explained in Chapter 4, the Kalman filter can be derived as a specific instance of the SPA, highlighting its relevance within the factor graph framework.

This chapter focuses on advanced synchronization algorithms developed within the factor graph and SPA framework. Specifically, it explores a family of algorithms designed to separate the tasks of detection and synchronization from decoding, allowing for greater adaptability and resilience in complex propagation conditions. It also reviews several state-of-the-art contributions that serve as "milestones" in the evolution of detection algorithms within this framework, providing insights into their development and application. The chapter is divided into two main sections: Section 3.1 outlines the algorithms designed to address detection and tracking in the

presence of phase noise. Section 3.2 concentrates instead on the available solutions for channels affected by both phase and frequency uncertainties.

3.1 Phase Noise Tracking

Consider the transmission of a sequence of code symbols that are linearly modulated and transmitted over an AWGN channel affected by phase noise. As already introduced in Section 2.1, the discrete-time base-band received signal is

$$y_k = c_k e^{j\theta_k} + w_k \,, \tag{3.1}$$

where $\{c_k\}$ is the sequence of transmitted symbols belonging to a M-ary complex-valued constellation and w_k are i.i.d. complex Gaussian random variables with a variance σ^2 per component. The phase noise sample θ_k is modeled as a Wiener process (see Section 2.2.1), and it is assumed that the phase varies slowly compared to the symbol time. The discrete-time equation in (3.1) is just an approximation of the actual received signal and it assumes that the phase noise process does not present high variations during a symbol time.

The algorithms discussed in the following are designed using the factor graph and SPA framework, enabling the derivation of MAP symbol detection strategies. The APPs necessary to implement the MAP symbol detection (see Section 2.6) for the channel in (3.1), can be obtained from the marginalization of the following joint posterior pdf

$$p(\mathbf{c}, \boldsymbol{\theta}|\mathbf{y}) \;\propto\; \prod_{k=0}^{K-1} P(c_k) p(y_k|c_k, \theta_k) p(\theta_0) \prod_{k=1}^{K-1} p_\Delta(\theta_k - \theta_{k-1}) \,, \tag{3.2}$$

where the first phase noise sample θ_0 is distributed according to a uniform distribution in $[0,2\pi)$, thus $p(\theta_0) = \frac{1}{2\pi}$. The probability $P(c_k)$ represents the likelihood that the symbol c_k assumes the value of one of the M symbols in the constellation. The factorization in (3.2) can be visualized in the factor graph in Figure 3.1. Applying the SPA to the factor graph in Figure 3.1 involves the following steps:

- For each time instant k, where $k = 0, 1, \ldots, K-1$, compute the forward and backward messages as

$$\alpha_k(\theta_k) = \int_0^{2\pi} \alpha_{k-1}(\theta_{k-1}) \sum_{c_{k-1}} P(c_{k-1})\, p(y_{k-1} \mid c_{k-1}, \theta_{k-1})\, p_\Delta(\theta_k - \theta_{k-1})\, d\theta_{k-1}, \tag{3.3}$$

$$\beta_k(\theta_k) = \int_0^{2\pi} \beta_{k+1}(\theta_{k+1}) \sum_{c_{k+1}} P(c_{k+1})\, p(y_{k+1} \mid c_{k+1}, \theta_{k+1})\, p_\Delta(\theta_{k+1} - \theta_k)\, d\theta_{k+1}. \tag{3.4}$$

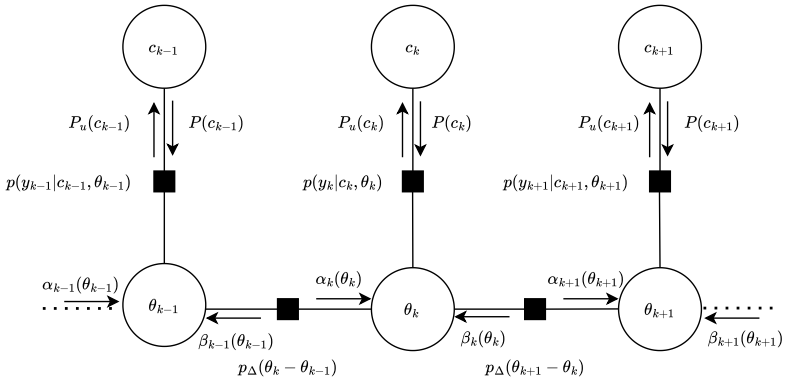


Figure 3.1: Factor graph representing the factorization in (3.2).

Here, $\alpha_k(\theta_k)$ and $\beta_k(\theta_k)$ denote the forward and backward messages, respectively. The forward recursion proceeds from the initial θ_0 to the last variable node θ_{K-1}, while the backward recursion operates in the opposite direction.

- After computing the forward and backward messages, perform the final step to compute the marginal probabilities. For each time instant k, where $k = 0, 1, \ldots, K-1$, calculate

$$P_u(c_k) = \int_0^{2\pi} \alpha_k(\theta_k)\, \beta_k(\theta_k)\, p(y_k \mid c_k, \theta_k)\, d\theta_k. \tag{3.5}$$

This step involves combining the forward and backward messages with the likelihood function to obtain the updated probability $P_u(c_k)$ for each transmitted symbol c_k.

The exact computation of (3.3), (3.4), and (3.5) requires evaluating integrals involving continuous probability distributions. In practice, this is often computationally challenging. As a result, various studies have proposed approximations to simplify the computation of these integrals and the involved pdfs. The canonical distribution method is a commonly used approach to handle continuous random variables. This method restricts the probability density functions to "canonical" families (e.g., the exponential family) [17].

From (3.3), it is evident that the forward recursion arises from a first marginalization over the symbol c_{k-1} (specifically, $\sum_{c_{k-1}} P(c_{k-1})p(y_{k-1}|c_{k-1},\theta_{k-1})$) followed by a second marginalization over the previous phase sample θ_{k-1}. The marginalization over the symbol can be performed in either an exact manner, where it is treated as a mixture (as explained in Section 3.1.3), or in an approximate form (for

example, as a univariate Gaussian, as discussed in Section 3.1.2). Regardless of the approach, the resulting recursion relies solely on the phase. Consequently, only the phase memory propagates throughout the entire factor graph.

In contrast, the contribution of the channel observation is instantaneous, meaning that the dependence on the symbol c_k is memoryless. This distinction is critical as it illustrates that recursive estimation captures long-term dependencies through the phase, while the symbol information influences the likelihood computation only in an instantaneous, memoryless way. The same conclusion applies to the backward recursion as well.

When calculating the log-likelihood over the symbols in (3.5), the product term within the integral, $\alpha_k(\theta_k)\beta_k(\theta_k)$, represents the posterior probability of the phase given the observations, denoted as $p(\theta_k|\mathbf{y})$. This product encloses the entire phase estimation, integrating past and future information about θ_k.

Although the algorithms presented in this chapter, as well as those throughout this thesis, are primarily framed within the context of joint detection and estimation, they can also be viewed from a synchronization perspective. In this alternative interpretation, these algorithms respond to some input a priori information about the phase by providing output likelihoods on the phase (or phase and frequency). Subsequently, these likelihoods can be used to compute symbol likelihoods using some approximation.

The upcoming sections will review some major algorithms that adopt the factor graph and SPA framework for joint detection and tracking of phase noise. These algorithms offer different approaches for approximating the SPA messages, leading to various methods for solving the integrals in (3.3), (3.4), and (3.5). The aim is to make the computations more tractable while maintaining high accuracy.

3.1.1 Discrete Phase BCJR

The discrete phase BCJR (dp-BCJR) algorithm is an adaptation of the classic BCJR algorithm [55], specifically designed to handle scenarios where the channel phase can be represented as a discrete set of values. In this method, the canonical distribution is modeled as a weighted sum of impulses, and the channel phase is assumed to take on discrete values

$$\theta_k \in \left(0, \frac{2\pi}{L}, ..., \frac{2\pi(L-1)}{L}\right) .$$

According to the dp-BCJR, the solution to (3.3), (3.4), and (3.5) is obtained by discretizing the phase values into L levels and turning the integrals into a discrete sum. For each of the L phase values, let θ_k^l represent the l-th phase value at time instant k. The forward and backward messages, $\alpha_k(\theta_k^l)$ and $\beta_k(\theta_k^l)$, respectively, are computed as

$$\alpha_k\left(\theta_k^{(l)}\right) \propto \sum_{\theta_{k-1}^{(l')},c_{k-1}} \alpha_{k-1}(\theta_{k-1}^{(l')})P(c_{k-1})p(y_{k-1}|c_{k-1},\theta_{k-1}^{(l')})p_\Delta(\theta_k^{(l)}-\theta_{k-1}^{(l')}) \tag{3.6}$$
$$\propto \sum_{\theta_{k-1}^{(l')},c_{k-1}} \alpha_{k-1}(\theta_{k-1}^{(l')})P(c_{k-1})e^{-\frac{|c_{k-1}|^2}{2\sigma^2}}$$
$$\times \exp\left\{\Re\left[\frac{y_{k-1}c_{k-1}^*}{\sigma^2}e^{-j\theta_{k-1}^{(l')}}\right]\right\}\exp\left\{-\frac{(\theta_k^{(l)}-\theta_{k-1}^{(l')})^2}{2\sigma_\Delta^2}\right\}$$

$$\beta_k(\theta_k^{(l)}) \propto \sum_{\theta_{k+1}^{(l')},c_{k+1}} \beta_{k+1}(\theta_{k+1}^{(l')})P(c_{k+1})p(y_{k+1}|c_{k+1},\theta_{k+1}^{(l')})p_\Delta(\theta_{k+1}^{(l')}-\theta_k^{(l)}) \tag{3.7}$$
$$\propto \sum_{\theta_{k+1}^{l'},c_{k+1}} \beta_{k+1}(\theta_{k+1}^{(l')})P(c_{k+1})e^{-\frac{|c_{k+1}|^2}{2\sigma^2}}$$
$$\times \exp\left\{\Re\left[\frac{y_{k+1}c_{k+1}^*}{\sigma^2}e^{-j\theta_{k+1}^{(l')}}\right]\right\}\exp\left\{-\frac{(\theta_{k+1}^{(l')}-\theta_k^{(l)})^2}{2\sigma_\Delta^2}\right\}.$$

Once the forward and backward recursions are computed, the completion reads

$$P_u(c_k) \propto \sum_{\theta_k^{(l)}} \alpha_k(\theta_k^{(l)})\beta_k(\theta_k^{(l)})e^{-\frac{|c_k|^2}{2\sigma^2}}\exp\left\{\Re\left[\frac{y_k c_k^*}{\sigma^2}e^{j\theta_k^{(l)}}\right]\right\}. \tag{3.8}$$

The rules for updating the SPA messages in this discrete representation are the same as those in the original BCJR algorithm. As the number of discretization levels L increases, the dp-BCJR algorithm gradually approaches the performance of the exact SPA. However, this improvement comes at the cost of increased computational complexity. Specifically, the complexity of the algorithm scales as $\mathcal{O}(KML^2)$. Therefore, the trade-off between accuracy and computational demand is a crucial factor to consider when applying the dp-BCJR algorithm. This is especially true when high-order modulations are adopted and large values of L are required, highlighting the need for low-complexity algorithms for detection and estimation.

3.1.2 Tikhonov Algorithm

The work in [5] was the first in a series of studies to develop low-complexity Bayesian detection algorithms in the factor graph and SPA framework to deal with channels affected by strong phase noise. The Tikhonov algorithm is an iterative detection and decoding algorithm that operates on a factor graph representing the joint posterior pdf in (3.2). As suggested by its name, this algorithm uses the

Tikhonov distribution as a canonical distribution to approximate the forward and backward SPA messages along the factor graph and solve (3.3), (3.4), and (3.5). The Tikhonov pdf is defined as

$$t(\theta; z) \triangleq \frac{1}{2\pi I_0(|z|)} \exp\left\{\Re\left[ze^{j\theta}\right]\right\}, \tag{3.9}$$

or, equivalently as

$$t(\theta; z) = \frac{1}{2\pi I_0(|z|)} \exp\left\{\kappa \cos(\theta - \mu)\right\}, \tag{3.10}$$

with $\mu = \angle z$ the mean value and $\kappa = |z|$ the concentration (inverse of the dispersion).

Defining the Gaussian mixture in (3.3) and (3.4) as

$$p_d(y_k|\theta_k) = \sum_{c_k} P(c_k) p(y_k|c_k, \theta_k), \tag{3.11}$$

with $p_d(y_k|\theta_k)$ being the posterior channel probability defined as

$$p_d(y_k|\theta_k) = \sum_{c_k} P(c_k) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{|y_k - c_k e^{j\theta_k}|^2}{2\sigma^2}\right\} \tag{3.12}$$

$$\propto \sum_{c_k} P(c_k) \exp\left\{-\frac{|c_k|^2}{2\sigma^2} + \Re\left[\frac{y_k c_k^*}{\sigma^2} e^{-\theta_k}\right]\right\}, \tag{3.13}$$

the authors of [5] approximate $p_d(y_k|\theta_k)$ by the nearest Gaussian in terms of the Kullback-Leibler distance [48]. This reduces to

$$p_d(y_k|\theta_k) \approx \mathcal{N}\left(y_k; a_k e^{j\theta_k}, \frac{1}{2}(2\sigma^2 + b_k - |a_k|^2)\right) \tag{3.14}$$

$$\propto \exp\left\{2\frac{\Re\left[y_k a_k^* e^{-j\theta_k}\right]}{2\sigma^2 + b_k - |a_k|^2}\right\}, \tag{3.15}$$

with

$$a_k = \sum_{i=0}^{M-1} P(c_k = x_i) x_i \qquad b_k = \sum_{i=0}^{M-1} P(c_k = x_i)|x_i|^2 \quad . \tag{3.16}$$

In (3.15), it can be observed that the variable θ_k follows a Tikhonov distribution with a complex parameter of $2\frac{y_k a_k^*}{2\sigma^2 + b_k - |a_k|^2}$. When the a priori on the symbol $P(c_k)$ is uniform, it indicates that no information is received from the channel, as the complex coefficient associated with the Tikhonov pdf in (3.15) is zero. This is why

the algorithm requires a bootstrap using pilot symbols inserted into the transmitted sequence. By plugging (3.15) into (3.3), the forward recursion becomes

$$\alpha_k(\theta_k) \propto \int_0^{2\pi} \alpha_{k-1}(\theta_{k-1}) \exp\left\{2\frac{\Re\left[y_{k-1}a_{k-1}^* e^{-\theta_{k-1}}\right]}{2\sigma^2 + b_{k-1} - |a_{k-1}|^2}\right\} p_\Delta(\theta_k - \theta_{k-1})\, d\theta_{k-1} \quad (3.17)$$

$$\propto \int_0^{2\pi} \exp\left\{\left(z_{k-1}^f + 2\frac{y_{k-1}a_{k-1}^*}{2\sigma^2 + b_{k-1} - |a_{k-1}|^2}\right) e^{j\theta_{k-1}}\right\} p_\Delta(\theta_k - \theta_{k-1})\, d\theta_{k-1}\,, \quad (3.18)$$

where the forward messages have been approximated as Tikhonov pdfs:

$$\alpha_k(\theta_k) \propto \exp\left\{\Re\left[z_k^f e^{-j\theta_k}\right]\right\}. \quad (3.19)$$

Being $p_\Delta(\theta_k - \theta_{k-1})$, by definition, Gaussian distributed, treating the forward (and backward) recursion as a sequence of Tikhonov pdfs is an approximation that improves the closer $p_\Delta(\theta_k - \theta_{k-1})$ is to a delta function ($\sigma_\Delta \to 0$).

Finally, to solve (3.18), it can be recalled that, for a Tikhonov pdf $t(\theta, z)$, when the concentration parameter ($|z|$) is sufficiently large, it can be approximated by a Gaussian distribution [56]. If this is the case, the Gaussian distribution will have mean value $\angle z$ and variance $1/|z|$. In our case,

$$\exp\left\{\Re\left[\left(z_{k-1}^f + 2\frac{y_{k-1}a_{k-1}^*}{2\sigma^2 + b_{k-1} - |a_{k-1}|^2}\right) e^{-j\theta_{k-1}}\right]\right\} \approx \exp\left\{-\frac{(\theta_{k-1} - \angle\hat{z}_{k-1}^f)^2}{2/|\hat{z}_{k-1}^f|}\right\}$$

where $\hat{z}_{k-1}^f = z_{k-1}^f + 2\frac{y_{k-1}a_{k-1}^*}{2\sigma^2 + b_{k-1} - |a_{k-1}|^2}$. Plugging the resulting pdf in (3.18) leaves us with the convolution between two Gaussian distributions

$$\alpha_k(\theta_k) \propto \int_0^{2\pi} \exp\left\{-\frac{(\theta_{k-1} - \angle\hat{z}_{k-1}^f)^2}{2/|\hat{z}_{k-1}^f|}\right\} \exp\left\{-\frac{(\theta_k - \theta_{k-1})^2}{2\sigma_\Delta^2}\right\} d\theta_{k-1} \quad (3.20)$$

$$\approx \exp\left\{\Re\left[z_k^f e^{-j\theta_k}\right]\right\}, \quad (3.21)$$

which, in turn, can be seen as a Tikhonov distribution with parameter

$$z_k^f = \hat{z}_{k-1}^f \gamma(\sigma_\Delta^2, |\hat{z}_{k-1}^f|)\,,$$

where $\gamma(x, y) = \frac{1}{1+xy}$.

The derivation for the backward recursion follows the same steps outlined for the forward. Similarly, the backward messages are approximated as a sequence of Tikhonov distributions

$$\beta_k(\theta_k) \propto \exp\left\{\Re\left[z_k^b e^{-j\theta_k}\right]\right\}, \quad (3.22)$$

where the complex coefficients can be computed recursively as

$$z_k^b = \hat{z}_{k+1}^b \frac{1}{1+\sigma_\Delta^2 |\hat{z}_{k+1}^b|},$$

with $\hat{z}_{k+1}^b = z_{k+1}^b + 2\frac{y_{k+1}a_{k+1}^*}{2\sigma^2+b_{k+1}-|a_{k+1}|^2}$.

To compute the extrinsic information of the k-th symbol given the received signal vector $\mathbf{y} = \{y_0, y_1, ..., y_{K-1}\}$ ($p(c_k|\mathbf{y})$) the forward and backward messages are plugged into (3.5), obtaining

$$\begin{aligned} P_u(c_k) &\propto \int_0^{2\pi} \exp\left\{\Re[z_k^f e^{-j\theta_k}]\right\} \exp\left\{\Re[z_k^b e^{-j\theta_k}]\right\} \exp\left\{\Re\left[\frac{y_k c_k^*}{\sigma^2} e^{-j\theta_k}\right]\right\} d\theta_k \\ &\propto \int_0^{2\pi} \exp\left\{\Re\left[\left(z_k^f + z_k^b + \frac{y_k c_k^*}{\sigma^2}\right) e^{-j\theta_k}\right]\right\} d\theta_k \\ &\propto I_0\left(\left|z_k^f + z_k^b + \frac{y_k c_k^*}{\sigma^2}\right|\right). \end{aligned} \tag{3.23}$$

The Tikhonov algorithm heavily relies on an iterative process with the decoder to achieve good performance. In this process, the decoder takes the likelihood on the symbols $P_u(c_k)$ as input and returns updated likelihoods, which are then used by the detector as a priori information. This iteration continues until convergence.

All symbols have a uniform a priori during the first iteration except for the pilot symbols. As the detector and decoder exchange information, the Tikhonov algorithm can utilize channel information, the reliability of which is determined by the decoder's output, to enhance tracking performance.

The Tikhonov algorithm performs better when the pilot symbols are spread throughout the sequence rather than concentrated in bursts. Concentrating the pilot symbols in bursts would result in very precise phase prediction at the pilot locations but could lead to deviations from the actual phase track between bursts of pilots. For this reason, the authors in [5] have proposed a modification to make the algorithm resilient to bursts of pilots in the case the application demands such conditions [32]. Considering the forward recursion, they propose evaluating the following product in (3.3)

$$\alpha_k(\theta_k) \sum_{i=0}^{M-1} P(c_k = x_i) p(y_k|x_i, \theta_k)$$

and identifying, for all the possible symbols in the constellation, whether there is a dominant term. This is determined by the following inequality

$$\ln P(c_k = x_i) + \left|z_k^f + \frac{y_k x_i^*}{\sigma^2}\right| > \delta + \ln P(c_k = x_j) \left|z_k^f + \frac{y_k x_j^*}{\sigma^2}\right| \text{ with } i \neq j\,. \tag{3.24}$$

The threshold δ is a real-valued parameter tuned by numerical simulations for the specific scenario. If a symbol x_i satisfies the above inequality, the parameters a_k

and b_k in (3.15) are set as

$$a_k = x_i \ , \ b_k = |x_i|^2 \, ,$$

otherwise the definitions in (3.16) are adopted. The same process is followed for the backward recursion as well. This modification in the original algorithm ensures that a decision is made when there is high reliability on the possible symbol transmitted, instead of waiting to update the estimate until the next burst of pilot arrives.

3.1.3 Tikhonov Mixtures Algorithm

The authors in [14] identified that the Tikhonov algorithm (Section 3.1.2) was overly dependent on the quality of likelihood outputs from the decoder, requiring several iterations to produce accurate information. To address this limitation and enable robust performance in severe phase noise conditions without excessive reliance on the decoder, they proposed in [14] an alternative method using Tikhonov mixtures to approximate forward and backward recursions, effectively capturing multiple phase trajectories. These mixtures have an adaptive order that changes for each symbol, although a modification that limits the maximum order was proposed to reduce complexity. To cope with possible cycle slips due to the limited order, the authors propose to introduce a cycle slip recovery mechanism by keeping track, along the message passing, of the probability that a cycle slip has occurred based on the rejected trajectories. Additionally, they introduced a reduction algorithm based on the Kullback-Leibler divergence to reduce the mixture while maintaining the divergence upper-bounded.

In order to solve the integrals in (3.3) and (3.4), [14] approximates the forward and backward messages by

$$\alpha_k(\theta_k) \triangleq \sum_{i=0}^{N_k^f - 1} w_k^{f,i} t_k^{f,i}(\theta_k; z_k^{f,i}) \tag{3.25}$$

$$\beta_k(\theta_k) \triangleq \sum_{i=0}^{N_k^b - 1} w_k^{b,i} t_k^{b,i}(\theta_k; z_k^{b,i}) \, , \tag{3.26}$$

with N_k^f and N_k^b being the k-th forward and backward mixture orders, respectively.

Considering the forward recursion, by substituting (3.25) in (3.3) it is obtained

$$\begin{aligned}\alpha_k(\theta_k) \propto & \sum_{i=0}^{N_{k-1}^f-1}\sum_{j=0}^{M-1} w_{k-1}^{f,i} P(c_{k-1}=x_j)\int_0^{2\pi} t_{k-1}^{f,i}(\theta_{k-1};z_{k-1}^{f,i}) \\ & \times p(y_{k-1}|x_j,\theta_{k-1}) p_\Delta(\theta_k-\theta_{k-1})\,d\theta_{k-1} \qquad (3.27)\\ \propto & \sum_{i=0}^{N_{k-1}^f-1}\sum_{j=0}^{M-1} w_{k-1}^{f,i} P(c_{k-1}=x_j)\exp\left\{-\frac{|x_j|^2}{2\sigma^2}\right\}\frac{I_0\left(\left|\hat{z}_k^{f,i,j}\right|\right)}{I_0\left(\left|z_{k-1}^{f,i}\right|\right)} \\ & \times \int_0^{2\pi} t_{k-1}^{f,i,j}\left(\theta_{k-1};\hat{z}_k^{f,i,j}\right) p_\Delta(\theta_k-\theta_{k-1})\,d\theta_{k-1} \qquad (3.28)\\ \propto & \sum_{i=0}^{N_{k-1}^f-1}\sum_{j=0}^{M-1} w_{k-1}^{f,i} P(c_{k-1}=x_j)\exp\left\{-\frac{|x_j|^2}{2\sigma^2}\right\}\frac{I_0\left(\left|\hat{z}_k^{f,i,j}\right|\right)}{I_0\left(\left|z_{k-1}^{f,i}\right|\right)} \\ & \times t_k^{f,i,j}\left(\theta_k;\hat{z}_k^{f,i,j}\gamma\left(\sigma_\Delta^2,\left|\hat{z}_k^{f,i,j}\right|\right)\right)\,. \end{aligned}$$

An interesting analogy can be observed between the mixture approach and the modification proposed in Section 3.1.2 to address bursts of pilots. As proven by (3.27), the past estimation $\alpha_{k-1}(\theta_{k-1})$ is multiplied by the channel information $p(y_{k-1}|x_j,\theta_{k-1})$ constituting what can be referred to as a *prediction step*. Likewise, in (3.24), this prediction is used to decide on the transmitted symbol.

Finally, the forward and backward recursions read

$$\alpha_k(\theta_k) \propto \sum_{i=0}^{N_{k-1}^f-1}\sum_{j=0}^{M-1} w_{k-1}^{f,i,j}\lambda_{k-1}^{f,i,j}\frac{\exp\left\{\Re\left[\hat{z}_k^{f,i,j}\gamma(\sigma_\Delta^2,\left|\hat{z}_k^{f,i,j}\right|)e^{-j\theta}\right]\right\}}{2\pi I_0\left(\left|\hat{z}_k^{f,i,j}\right|\right)} \qquad (3.29)$$

$$\beta_k(\theta_k) \propto \sum_{i=0}^{N_{k+1}^b-1}\sum_{j=0}^{M-1} w_{k+1}^{b,i,j}\lambda_{k+1}^{b,i,j}\frac{\exp\left\{\Re\left[\hat{z}_k^{b,i,j}\gamma(\sigma_\Delta^2,\left|\hat{z}_k^{b,i,j}\right|)e^{-j\theta}\right]\right\}}{2\pi I_0\left(\left|\hat{z}_k^{b,i,j}\right|\right)}\,, \qquad (3.30)$$

where

$$\begin{aligned}\hat{z}_k^{f,i,j} &= z_{k-1}^{f,i}+\frac{y_{k-1}x_j^*}{\sigma^2}\,,\\ \hat{z}_k^{b,i,j} &= z_{k+1}^{b,i}+\frac{y_{k+1}x_j^*}{\sigma^2}\,,\\ \lambda_{k-1}^{f,i,j} &= P(c_{k-1}=x_j)\exp\left\{-\frac{|x_j|^2}{2\sigma^2}\right\}\frac{I_0\left(\left|\hat{z}_k^{f,i,j}\right|\right)}{I_0\left(\left|z_{k-1}^{f,i}\right|\right)}\,,\\ \lambda_{k+1}^{b,i,j} &= P(c_{k+1}=x_j)\exp\left\{-\frac{|x_j|^2}{2\sigma^2}\right\}\frac{I_0\left(\left|\hat{z}_k^{b,i,j}\right|\right)}{I_0\left(\left|z_{k+1}^{b,i}\right|\right)}\,.\end{aligned}$$

The resultant Tikhonov mixtures in (3.29) and (3.30) have order $N_{k-1}^{f}M$ and $N_{k+1}^{b}M$, respectively. However, since the mixture order would increase exponentially with time, it is necessary to reduce it at each time instant. The reduction algorithm proposed in [14] (see Algorithm 2) operates by first identifying the term in the mixture that has the highest weight, and then calculating the Kullback-Leilbler distance between that maximum weight component and all the other components. Components whose Kullback-Leibler distance from the maximum-weight component falls below a specified threshold ϵ are merged, using circular mean and variance matching [14]. If the order of the reduced mixture is equal to or below the target order, the algorithm stops; otherwise, it repeats until the target order is reached. Throughout the reduction algorithm, a probability that a cycle slip has occurred is computed by considering the mixture components that have not been merged because they exceeded the maximum desired distance. If so, clearly the probability that a cycle slip has occurred increases.

The purpose of accounting for the probability of cycle slips is to ensure that, when a cycle slip occurs, the resulting phase estimation from the algorithm is recognized as unreliable. In such cases, a more accurate estimation must be obtained, and this is where the pilot symbols play a crucial role. If the Tikhonov algorithm was exploiting pilot symbols to give a bootstrap to the algorithm, now they are required to prevent the phase tracking from deteriorating when the probability of a cycle slip occurring is high. In such cases, the algorithm relies only on the pilot symbols. When a pilot symbol is transmitted, the forward ($\alpha_{k-1}(\theta_{k-1})$) and backward ($\beta_{k+1}(\theta_{k+1})$) messages used to solve the integral (3.3) and (3.4) are replaced by

$$\hat{\alpha}_{k-1}(\theta_{k-1}) = \Phi_{k-1}^{f}\alpha_{k-1}(\theta_{k-1}) + \left(1-\Phi_{k-1}^{f}\right)\frac{1}{2\pi},$$

$$\hat{\beta}_{k+1}(\theta_{k+1}) = \Phi_{k+1}^{b}\beta_{k+1}(\theta_{k+1}) + \left(1-\Phi_{k+1}^{b}\right)\frac{1}{2\pi},$$

where Φ_{k-1}^{f} and Φ_{k+1}^{b} represent the probability that a cycle slip has not occurred in the $(k-1)$-th forward and $(k+1)$-th backward recursion. By doing so, whenever the probability that a cycle slip occurs is high, the algorithm relies on the information coming from the pilot symbol rather than the previous messages. The introduced probabilities $\hat{\alpha}_k(\theta_k)$ and $\hat{\beta}_k(\theta_k)$ are mixtures with an order increase by one compared to the original ones.

Considering the cycle-slip recovery algorithm, the probabilities $P_u(c_k)$ are computed as

$$P_u(c_k) \propto \int_0^{2\pi} \hat{\alpha}_k(\theta_k)\hat{\beta}_k(\theta_k)p(y_k, c_k, \theta_k)\, d\theta_k \tag{3.31}$$

$$\begin{aligned}
&\propto \sum_{i=0}^{N_k^f-1}\sum_{j=0}^{N_k^b-1} \frac{w_k^{f,i}\Phi_k^f w_k^{b,j}\Phi_k^b}{I_0\left(\left|z_k^{f,i}\right|\right) I_0\left(\left|z_k^{b,j}\right|\right)} \int_0^{2\pi} \exp\left\{\Re\left[\left(z_k^{f,i}+z_k^{b,j}+u_k\right)e^{-j\theta_k}\right]\right\} d\theta_k \\
&+ \frac{\left(1-\Phi_k^f\right) w_k^{b,j}\Phi_k^b}{I_0\left(\left|z_k^{b,j}\right|\right)} \int_0^{2\pi} \exp\left\{\Re\left[\left(z_k^{b,j}+u_k\right)e^{-j\theta_k}\right]\right\} d\theta_k \\
&+ \frac{w_k^{f,i}\Phi_k^f \left(1-\Phi_k^b\right)}{I_0\left(\left|z_k^{f,i}\right|\right)} \int_0^{2\pi} \exp\left\{\Re\left[\left(z_k^{f,i}+u_k\right)e^{-j\theta_k}\right]\right\} d\theta_k \\
&+ \left(1-\Phi_k^f\right)\left(1-\Phi_k^b\right) \int_0^{2\pi} \exp\left\{\Re\left[u_k e^{-j\theta_k}\right]\right\} d\theta_k
\end{aligned} \tag{3.32}$$

$$\begin{aligned}
&\propto \sum_{i=0}^{N_k^f-1}\sum_{j=0}^{N_k^b-1} \frac{w_k^{f,i}\Phi_k^f w_k^{b,j}\Phi_k^b}{I_0\left(\left|z_k^{f,i}\right|\right) I_0\left(\left|z_k^{b,j}\right|\right)} I_0\left(\left|z_k^{f,i}+z_k^{b,j}+u_k\right|\right) + \\
&\frac{\left(1-\Phi_k^f\right) w_k^{b,j}\Phi_k^b}{I_0\left(\left|z_k^{b,j}\right|\right)} I_0\left(\left|z_k^{b,j}+u_k\right|\right) + \frac{w_k^{f,i}\Phi_k^f \left(1-\Phi_k^b\right)}{I_0\left(\left|z_k^{f,i}\right|\right)} I_0\left(\left|z_k^{f,i}+u_k\right|\right) + \\
&\left(1-\Phi_k^f\right)\left(1-\Phi_k^b\right) I_0\left(|u_k|\right),
\end{aligned} \tag{3.33}$$

with $u_k = \frac{y_k c_k^*}{\sigma^2}$.

When using Tikhonov mixtures, it is important to note that the symbol c_k is not marginalized before computing the recursion. This approach allows for the full utilization of the channel information, extending beyond just the pilot symbols. However, unlike the Tikhonov algorithm in Section 3.1.2, this approach updates the forward and backward messages at each time step, thereby increasing the reliability of previous estimations relative to channel observations, particularly at moderate SNR levels. As a result, if the algorithm experiences a cycle slip, the presence of pilot symbols alone may not be sufficient to restore accurate tracking. To limit the number of mixture components and improve robustness, the cycle slip recovery algorithm proposed in [14] is crucial.

3.1.4 Expectation Propagation

The last discussed algorithm is based on expectation propagation (EP), falling in between the Tikhonov (Section 3.1.2) and the Tikhonov mixtures approach in Section 3.1.3. Even though the EP framework has been widely adopted [33], [57], [58], the work in [59] will be taken as a reference. Here, the authors offer an effective

solution in the EP framework for channels affected by strong phase noise. The study in [59] adopts the factor graph and SPA framework described in Section 3.1. The first difference with the Tikhonov algorithm lies in how the posterior channel probability $p_d(y_k|\theta_k)$ is treated. According to EP, the message from the channel observation is combined with the existing belief on the target variable, which is based on information from the rest of the factor graph. The channel probability $p_d(\theta_k)$ is, by definition,

$$p_d(\theta_k) = \sum_{i=0}^{M-1} w_k^i t(\theta_k, z_k^i)\,, \tag{3.34}$$

where

$$w_k^i = P(c_k = x_i) e^{-\frac{|x_i|^2}{2\sigma^2}} I_0\left(\left|z_k^i\right|\right)\,,$$

and x_i belongs to a complex M-ary constellation, and $z_k^i = \frac{y_k x_i^*}{\sigma^2}$. The EP framework approximates the entire marginal of each latent variable. This means to find an approximation for $p(\theta_k)$, the entire marginal of θ_k,

$$p(\theta_k) = p_d(\theta_k)\alpha_k(\theta_k)\beta_k(\theta_k)\,,$$

by projecting it into the Tikhonov family. In mathematical formulas,

$$p^{EP}(\theta_k) \approx \underset{t(\theta_k;x)}{\operatorname{argmin}}\, KL\left[p_d(\theta_k)\alpha_k(\theta_k)\beta_k(\theta_k)||t(\theta_k;x)\right]\,, \tag{3.35}$$

where $\alpha_k(\theta_k)$ and $\beta_k(\theta_k)$ are defined as in (3.19) and (3.22), respectively. Therefore, the EP approximation of the marginal probability $p(\theta_k)$, denoted as $p^{EP}(\theta_k)$ in (3.35), is by definition, a Tikhonov distribution. To characterize the probability $p^{EP}(\theta_k)$, the authors of [59] makes use of the moment matching, thus obtaining

$$\frac{I_1\left(\left|z_k^{EP}\right|\right)}{I_0\left(\left|z_k^{EP}\right|\right)} e^{j\angle z_k^{EP}} \approx \sum_{i=0}^{M-1} \overline{\lambda}_k^i \frac{I_1\left(\left|z_k^i + z_k^f + z_k^b\right|\right)}{I_0\left(\left|z_k^i + z_k^f + z_k^b\right|\right)} e^{j\angle(z_k^i + z_k^f + z_k^b)}\,, \tag{3.36}$$

where

$$\lambda_k^i = \frac{I_0\left(\left|z_k^i + z_k^f + z_k^b\right|\right)}{I_0\left(\left|z_k^f\right|\right) I_0\left(\left|z_k^b\right|\right)} P(x_i) e^{-\frac{|x_i|^2}{2\sigma^2}}\,,$$

and $\overline{\lambda}_k^i$ is defined such that $\sum_{i=0}^{M-1} \overline{\lambda}_k^i = 1$. To avoid numerical instabilities, the authors suggest approximating $I_1(x)/I_0(x) \approx x$, thus obtaining

$$z_k^{EP} \approx \sum_{i=0}^{M-1} \overline{\lambda}_k^i \left(z_k^i + z_k^f + z_k^b\right)\,, \tag{3.37}$$

The authors also propose incorporating a "cycle slip detector", which monitors the concentration (absolute value) of the coefficients z_k^{EP} for sudden drops. When such a drop occurs, the algorithm discards the corresponding message ($\alpha_k(\theta_k)$ or $\beta_k(\theta_k)$) and relies only on the message that is not discarded.

Comparing the EP algorithm with the Tikhonov mixture algorithm in Section 3.1.3, it can be realized that both maximize the use of available channel information. In doing so, they must restrict the marginal distribution $p(\theta_k)$ to a specific form. In this case, the marginal distribution is limited to the Tikhonov distribution, whereas in Section 3.1.3 it is constrained to a reduced-order Tikhonov mixture. These limitations are necessary to ensure the algorithm's complexity remains manageable, but they also increase the chances of cycle slip events.

3.2 Phase Noise and Carrier Frequency Tracking

Many communication systems experience misalignment in frequency offset, which can degrade system performance if not recovered. Examples include communications between Earth and satellites, which require robust algorithms to ensure reliable links under harsh propagation environments.

In the literature, the algorithms designed to address the joint detection and synchronization of time-varying phase and frequency offset are quite limited compared to those available for phase noise tracking only. In particular, to the best of the authors' knowledge, only the authors in [22] propose a solution in the factor graph and SPA framework for a channel affected by phase noise and a constant frequency offset. Their algorithm was found to be resilient to "slow" variations of the frequency offset, making it a good candidate for satellites in low Earth orbit (LEO).

3.2.1 Tikhonov Algorithm for Phase Noise and Frequency Offset

In [22], the authors present an algorithm that can cope with strong phase noise and carrier frequency offsets. This algorithm, developed using the factor graph and SPA framework, is based on Tikhonov distributions. It modifies the algorithm described in Section 3.1.2 by considering multiple tracking hypotheses related to potential frequency offsets. Consider the channel model in (3.1) and a phase noise modeled as a Wiener process with a constant frequency offset

$$\theta_k = \theta_{k-1} + \phi + \delta_k \,, \tag{3.38}$$

where the phase offset $\phi = 2\pi\nu T$ represents the contribution of the frequency offset ν, where T is the symbol time. The frequency offset is assumed to be uniformly distributed within the range $[-\nu_0, \nu_0]$, and remains constant throughout the entire

codeword duration. Additionally, to satisfy the Nyquist criterion, it should hold that $\frac{-1}{2T} < \nu < \frac{1}{2T}$, which translates to $-\pi < \phi < \pi$. The computation of the APPs necessary to implement the MAP symbol detection strategy can be obtained by applying the SPA to the factor graph representing the joint posterior probability

$$p(\mathbf{c}, \boldsymbol{\theta}, \phi) \propto \prod_{k=0}^{K-1} P(c_k) p(y_k|c_k\theta_k) p(\theta_0) \prod_{k=1}^{K-1} p(\theta_k|\theta_{k-1}, \phi) p(\phi)\,. \tag{3.39}$$

The factorization in (3.39) can be visualized in the factor graph shown in Figure 3.2. In this graph, the variable node $\boldsymbol{\mu_k} = [\theta_k, \phi]$ represents the state of the detector at time k, with the variable ϕ stretched to every variable node to prevent short cycles. Assuming the knowledge of the variable ϕ, the probability $p(\theta_k|\theta_{k-1}, \phi)$ is defined

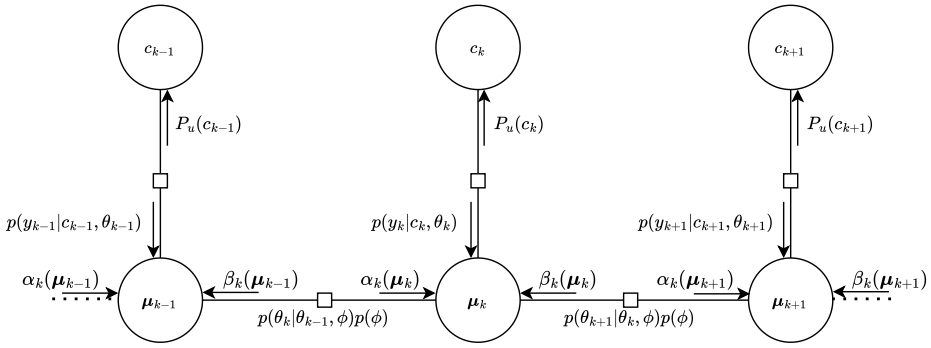


Figure 3.2: Factor graph representing the factorization in (3.39).

as

$$p(\theta_k|\theta_{k-1}, \phi) \triangleq \mathcal{N}(\theta_k - \theta_{k-1} - \phi; 0, \sigma_\Delta^2)\,, \tag{3.40}$$

while (3.11) and (3.15) hold for the posterior channel probability in (3.39). To derive the forward and backward messages, the authors adopt the canonical distribution approach, as already done in [5], and approximate the variable ϕ as a discrete random variable taking L possible values $\{\phi^l\}_{l=0}^{L-1}$ within the range $[-\phi_0, \phi_0]$. Since ϕ is represented as a discrete random variable, it is characterized by a pmf. As a result, in the definition of forward and backward messages, expressed as the a posteriori probabilities of $\boldsymbol{\mu}_k$ given the past and future observable, respectively,

$$\alpha_k(\boldsymbol{\mu}_k) \triangleq p(\theta_k, \phi|\boldsymbol{y}_0^{k-1}) = p(\phi|\boldsymbol{y}_0^{k-1}) p(\theta_k|\phi, \boldsymbol{y}_0^{k-1})\,, \tag{3.41}$$

and

$$,\beta_k(\boldsymbol{\mu}_k) \triangleq p(\theta_k, \phi|\boldsymbol{y}_{k+1}^{K-1}) = p(\phi|\boldsymbol{y}_{k+1}^{K-1}) p(\theta_k|\phi, \boldsymbol{y}_{k+1}^{K-1})\,, \tag{3.42}$$

the pdfs $p(\phi|\boldsymbol{y}_0^{k-1})$ and $p(\phi|\boldsymbol{y}_{k+1}^{K-1})$ are replaced with the corresponding pmfs, denoted as

$$P(\phi = \phi^l|\boldsymbol{y}_0^{k-1}) = \gamma_{f,k}^l\,,$$

$$P(\phi = \phi^l | \boldsymbol{y}_{k+1}^{K-1}) = \gamma_{b,k}^l \,.$$

The forward and backward messages in (3.41) and (3.42) are approximated as L Tikhonov distributions weighted by the probability $\gamma_{f,k}^l$ and $\gamma_{b,k}^l$, respectively. Therefore,

$$p(\theta_k | \phi = \phi^l, \boldsymbol{y}_0^{k-1}) \approx t(z_{f,k}^l; \theta_k) \,, \tag{3.43}$$

$$p(\theta_k | \phi = \phi^l, \boldsymbol{y}_{k+1}^{K-1}) \approx t(z_{b,k}^l; \theta_k) \,, \tag{3.44}$$

and the forward recursion can be computed as

$$\begin{aligned}
\alpha_k(\theta_k, \phi^l) &\propto \int_0^{2\pi} \alpha_k(\theta_{k-1}, \phi^l) p_d(y_{k-1}|\theta_{k-1}) p(\theta_k|\theta_{k-1}, \phi^l) d\theta_{k-1} && (3.45)\\
&\propto \gamma_{f,k-1}^l \int_0^{2\pi} t(z_{f,k-1}^l; \theta_{k-1}) t\left(u_{k-1}; \theta_{k-1}\right) \mathcal{N}(\theta_k - \theta_{k-1} - \phi^l; 0, \sigma_\Delta^2) d\theta_{k-1} \\
&\propto \gamma_{f,k-1}^l \frac{I_0(|\hat{z}_{f,k-1}^l|)}{I_0(|z_{f,k-1}^l|) I_0(|u_{k-1}|)} \int_0^{2\pi} t\left(\hat{z}_{f,k-1}^l; \theta_{k-1}\right) \\
&\times \mathcal{N}(\theta_k - \theta_{k-1} - \phi^l; 0, \sigma_\Delta^2) d\theta_{k-1} \\
&\approx \gamma_{f,k-1}^l \frac{I_0(|\hat{z}_{f,k-1}^l|)}{I_0(|z_{f,k-1}^l|) I_0(|u_{k-1}|)} \mathcal{N}\left(\theta_k; \angle \hat{z}_{f,k-1}^l - \phi^l, \sigma_\Delta^2 + \frac{1}{|\hat{z}_{f,k-1}^l|}\right) && (3.46)\\
&\approx \gamma_{f,k-1}^l \frac{I_0(|\hat{z}_{f,k-1}^l|)}{I_0(|z_{f,k-1}^l|) I_0(|u_{k-1}|)} t\left(\hat{z}_{f,k-1}^l e^{j\phi^l} \gamma\left(\sigma_\Delta^2, \left|\hat{z}_{f,k-1}^l\right|\right); \theta_k\right) \,, && (3.47)
\end{aligned}$$

where $u_k = \frac{2y_k a_k^*}{(2\sigma^2 + b_k^2 - |a_k|^2)}$ and $\hat{z}_{f,k}^l = z_{f,k}^l + u_k$. (3.46) approximates the Tikhonov distribution as a Gaussian to compute the convolution. Finally, (3.47) goes back to the Tikhonov domain. By comparison of the expressions in (3.47) and in (3.41), the forward messages can be recursively computed as

$$\gamma_{f,k}^l = \gamma_{f,k-1}^l \frac{I_0(|\hat{z}_{f,k-1}^l|)}{I_0(|z_{f,k-1}^l|) I_0(|u_{k-1}|)} \,, \tag{3.48}$$

$$z_{f,k}^l = \hat{z}_{f,k-1}^l e^{j\phi^l} \gamma\left(\sigma_\Delta^2, \left|\hat{z}_{f,k-1}^l\right|\right) \,. \tag{3.49}$$

Similarly, the backward messages can be computed as

$$\gamma_{b,k}^l = \gamma_{b,k+1}^l \frac{I_0(|\hat{z}_{b,k+1}^l|)}{I_0(|z_{b,k+1}^l|) I_0(|u_{k+1}|)} \,, \tag{3.50}$$

$$z_{b,k}^l = \hat{z}_{b,k+1}^l e^{j\phi^{-l}} \gamma\left(\sigma_\Delta^2, \left|\hat{z}_{b,k+1}^l\right|\right) \,, \tag{3.51}$$

with $\hat{z}_{b,k}^{l} = z_{b,k}^{l} + u_k$. Once the recursions have been computed, the probability of the symbols c_k can be found as

$$P(c_k) \propto \sum_{l=0}^{L-1} \int_0^{2\pi} \alpha_k(\theta_k, \phi^l)\beta_k(\theta_k, \phi^l)p(y_k|c_k, \theta_k)d\theta_k \tag{3.52}$$

$$\propto \sum_{l=0}^{L-1} \gamma_{f,k}^{l}\gamma_{b,k}^{l} \frac{I_0\left(\left|z_{f,k}^{l} + z_{b,k}^{l} + \frac{y_k c_k^*}{\sigma^2}\right|\right)}{I_0\left(\left|z_{f,k}^{l}\right|\right) I_0\left(\left|z_{b,k}^{l}\right|\right)} \,. \tag{3.53}$$

The authors propose replacing $I_0(x)$ with e^x to reduce the computational complexity of the algorithm.

The Tikhonov algorithm for phase and frequency offset acts like L parallel Tikhonov algorithms (see Section 3.1.2), each addressing a different hypothesis of the carrier frequency offset.

To manage complexity, the authors suggest using a large value of L (e.g. $L = 11$) only during the first iteration. For subsequent iterations, L is decreased to 3, as this significantly affects the overall complexity. It is important to note that the algorithm cannot accurately estimate the frequency offset after the first iteration. Multiple iterations with the decoder are necessary to refine the estimate and enhance accuracy.

Chapter 4

Extending PLLs: from Gaussian to Tikhonov-Based Models

This chapter focuses on the connection between advanced synchronization algorithms and the commonly known PLL. While the relationship between the Kalman filter and the PLL—both relying on the Gaussian distribution—is well established, the concept of a PLL that is based on the Tikhonov distribution has not received as much attention. This chapter first derives the Kalman filter within the factor graph and SPA framework. Building on the concepts presented in [14], a new family of PLLs based on the Tikhonov distribution is introduced. An analysis will thoroughly compare the performance and characteristics of the Tikhonov-PLL with those of the Kalman filter. Although this chapter may seem to disrupt the flow of the discussion, it is crucial for a deeper understanding of subsequent chapters. The following chapters will consistently reference the foundational concepts of the Kalman filter and the trade-offs between the Gaussian and the Tikhonov distribution. The content of this chapter is derived from the work presented in [60].

4.1 Kalman Filter and PLLs: a Factor Graph Perspective

Consider a pure carrier tone on the Wiener phase noise channel that, despite its simplicity, is still representative of actual oscillators [61]. The samples of the received signal normalized and in complex base-band are defined as

$$y_k = e^{j\theta_k} + w_k \,, \tag{4.1}$$

where w_k is AWGN with variance $\sigma^2 = N_0/2PT$ per component, N_0 is the noise power spectral density, P is the carrier power, T is the sampling time, and θ_k is the k-th phase noise sample. The phase noise process is defined according to (2.5). By identifying with $\boldsymbol{y}_k$ the vector that collects all samples $y_0, .., y_k$, up to k-th

instant, the minimum mean square error (MSE) Bayesian estimator is computed as $\hat{\theta}_k = \mathrm{E}\left[\theta_k|\boldsymbol{y}_k\right]$. This requires the pdf $p(\theta_k|\boldsymbol{y}_k)$ that can be derived using the factor graphs and SPA [16]. Given the vectors $\boldsymbol{\theta}$ and $\boldsymbol{y}$ containing all samples of the phase and received signal, the posterior probability factorizes as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \quad \propto \quad p(\theta_0) \prod_{k\geq 0} p(y_k|\theta_k) \prod_{k\geq 1} p(\theta_k|\theta_{k-1})\,, \tag{4.2}$$

represented by the factor graph in Figure 4.1. Assuming a symmetrical and non-

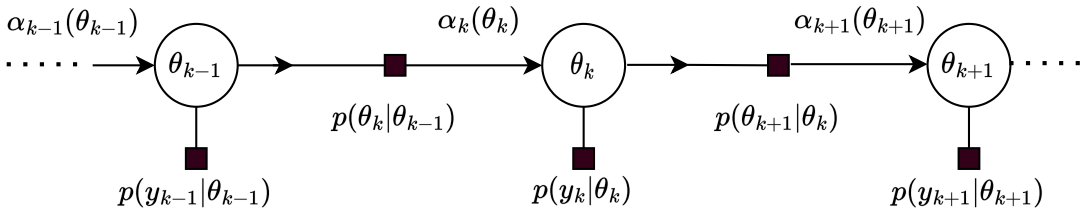


Figure 4.1: Factor graph corresponding to (4.2).

skewed distribution for the forward messages $\alpha_k(\theta_k)$, the MSE estimator can be derived using the SPA as

$$\hat{\theta}_k = \arg\max_{\theta_k} \alpha_k(\theta_k) p(y_k|\theta_k)\,, \tag{4.3}$$

where $\forall k\ \alpha_k(\theta)$ can be recursively computed as

$$\alpha_k(\theta_k) \propto \int_0^{2\pi} \alpha_{k-1}(\theta_{k-1}) p(y_{k-1}|\theta_{k-1}) p(\theta_k|\theta_{k-1})\, d\theta_{k-1}\,. \tag{4.4}$$

It is not possible to solve (4.3) and (4.4) in a closed form. However, by approximating the messages with distributions belonging to the exponential family, a closed-form solution can be found. For instance, the well-known Kalman filter is derived using the Gaussian distribution. Modeling $p(y_k|\theta_k)$ as $\mathcal{N}\left(\theta_k, \angle y_k, \sigma^2\right)$ and $\alpha_k(\theta_k)$ as $\mathcal{N}\left(\theta_k, \mu_k, \sigma_k^2\right)$, the recursion in (4.4) generates a sequence of Gaussian pdfs. Their mean value and variance can be recursively computed as

$$\mu_{k+1} \quad = \quad \mu_k + \beta_k \left(\angle y_k - \mu_k\right) \tag{4.5}$$

$$\sigma_{k+1}^2 \quad = \quad \frac{1}{\left(1/\sigma_k^2 + 1/\sigma^2\right)} + \sigma_\Delta^2\,, \tag{4.6}$$

where β_k is known as the *Kalman gain* [19] and is defined as

$$\beta_k = \frac{\sigma_k^2}{\sigma_k^2 + \sigma^2}\,. \tag{4.7}$$

Thus, the MSE estimator in (4.3) is found as

$$\hat{\theta}_k = \mu_{k+1}\,, \tag{4.8}$$

and can be implemented using the digital filtering scheme in Figure 4.2a. In this figure, the first block is responsible for extracting the input's argument, which is necessary to linearize the model—an essential step for implementing the Kalman filter. More precisely, what is referred to as the Kalman filter, in this context, is an extended Kalman filter, a variant that linearizes the nonlinear input before applying the standard Kalman filter equations [28]. As the figure shows, the phase estimate $\hat{\theta}_k = \mu_{k+1}$ is updated based on the phase error $\angle y_k - \mu_k$. The recursive calculations in (4.5) and (4.6) corresponds to the *time-update equations*[1] [19].

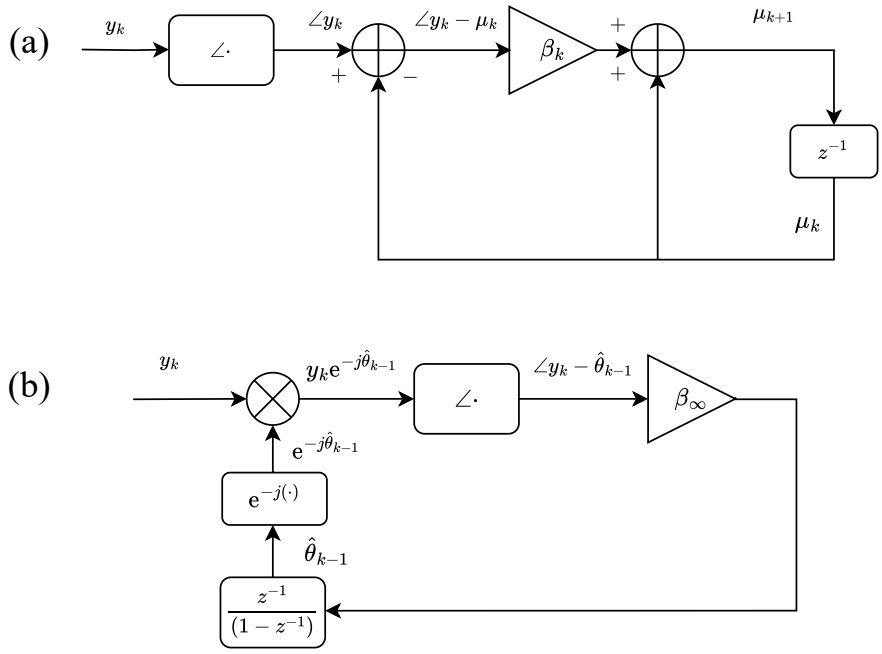


Figure 4.2: Block diagrams of the digital a) Kalman filter and b) PLL.

The Kalman filter steady-state variance satisfies its Riccati equation [62], i.e. $\sigma_k^2 \to \sigma_\infty^2$ as $k \to \infty$. As a result, the gain of the Kalman filter tends to a steady-state fixed value (denoted as β_∞ as $k \to \infty$). Substituting (4.8) into (4.5), $\hat{\theta}_k$ is obtained as

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \beta_\infty(\angle y_k - \hat{\theta}_{k-1})\,, \tag{4.9}$$

[1] According to the classical Kalman filtering notation [28], μ_k represents both the updated state estimation ($\hat{x}_{k|k}$) and the predicted state ($\hat{x}_{k|k-1}$). On the other hand, σ_k^2 corresponds to the variance associated with the predicted state ($\sigma_{k|k-1}^2$), while the variance of the update state estimation ($\sigma_{k|k}^2$) is represented by the term $(1/\sigma_k^2 + 1/\sigma^2)^{-1}$ in (4.6).

where one can recognize the equation of a digital PLL of the first type [30] having loop gain β_∞. The recursive phase estimation in (4.9) can be implemented according to the filtering scheme in Figure 4.2b, which is equivalent to the Kalman filter in Figure 4.2a.

More generally, it can be shown that performing the SPA on the factor graph in Figure 4.1 using an N-th order Gaussian distribution leads to a Kalman filter [16]. In steady-state conditions, when $N \leq 3$, this Kalman filter becomes equivalent to a PLL of the N-th type [63]. However, there is a significant difference in the way the two schemes operate. In the standard PLL, the observable $y_k e^{-j\hat{\theta}_{k-1}}$ is used for carrier tracking [30]. On the other hand, the Kalman filter uses the most current estimate $\hat{\theta}_k = \mu_{k+1}$ to compensate the signal directly. Thus, the PLL effectively operates like a Kalman filter that provides MSE delayed by one sample, resulting in reduced tracking performance.

4.2 Tikhonov Bayesian Estimator and PLL

Section 4.1 defined the well-known concept of Kalman-PLL duality (see [64], [65]) from the perspective of factor graphs. This section introduces a new, generalizable model of PLL. The Tikhonov distribution, another well-known example of the exponential family, is considered to describe the messages flowing along the factor graph in Figure 4.1. $\alpha_k(\theta_k)$ is modeled as a circular distribution in $\theta_k \in [0,2\pi)$ that reads

$$\alpha_k(\theta_k) \propto e^{\Re[z_k e^{-j\theta_k}]},$$

where z_k is a complex-value parameter. Its angle $\angle z_k$ determines the mode of the pdf, while $1/|z_k|$ represents its dispersion. On the other hand, $p(y_k|\theta_k)$ is by definition $t(\frac{y_k}{\sigma^2};\theta_k)$.

The work in [5] demonstrated that the recursion in (4.4) can be approximated as a sequence of Tikhonov pdfs with z_k recursively computed as

$$z_{k+1} = \gamma\left(\sigma_\Delta^2, z_k + \frac{y_k}{\sigma^2}\right), \tag{4.10}$$

where $\gamma(x_1, x_2) = \frac{x_2}{1+x_1|x_2|}$. Thus, the MSE is found to be

$$\begin{aligned} \hat{\theta}_k &= \angle\left(z_k + y_k/\sigma^2\right) \\ &= \angle z_{k+1}\,. \end{aligned} \tag{4.11}$$

Such an estimator can be implemented with the digital filtering scheme in Figure 4.3.

To better understand the behavior of the phase transition in (4.11), Figure 4.4 provides a vector representation of the equation. Consider the scenario in Figure 4.4a, where the magnitude of the channel coefficient y_k/σ^2 is greater than the

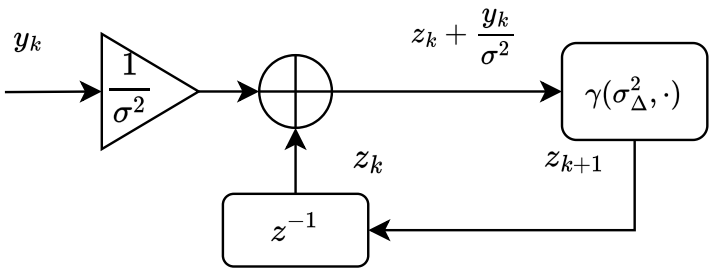
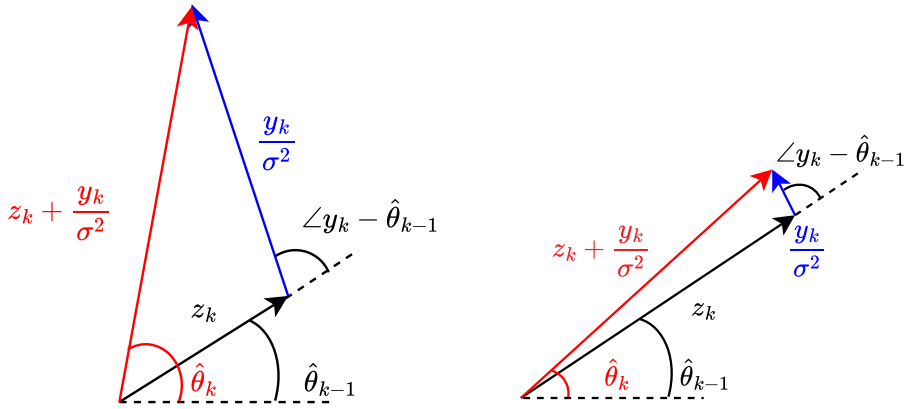


Figure 4.3: Block diagram of the digital PLL based on the Tikhonov distribution.

forward coefficient z_k. This situation usually occurs in high SNR scenarios, where the past estimate is less reliable than the channel information. In these cases, the phase estimation, represented by the argument of $z_k + y_k/\sigma^2$ (equal to that of z_{k+1}), tends to closely follow the error identified as $\angle y_k - \hat{\theta}_{k-1}$. However, in situations where the magnitude of the forward coefficient dominates over that of the channel term, such as in the scenario depicted in Figure 4.4b, the predicted phase will be heavily influenced by the past estimations. This is because the past estimations are more reliable than the channel information.



(a) (b)

Figure 4.4: Vector representation of (4.11).

It will now be proven that the MSE derived in (4.11) has an equivalency to

first-type PLLs, as shown for Kalman filtering. By expanding the terms in (4.11),

$$\hat{\theta}_k = \angle\left(|z_k|e^{j\hat{\theta}_{k-1}} + \frac{|y_k|}{\sigma^2}e^{j\angle y_k}\right) \tag{4.12}$$

$$= \hat{\theta}_{k-1} + \angle\left(|z_k| + \frac{|y_k|}{\sigma^2}e^{j(\angle y_k - \hat{\theta}_{k-1})}\right) \tag{4.13}$$

$$\approx \hat{\theta}_{k-1} + \frac{|y_k|/\sigma^2}{|z_k| + |y_k|/\sigma^2}(\angle y_k - \hat{\theta}_{k-1}) \tag{4.14}$$

$$\approx \hat{\theta}_{k-1} + \beta_k(\angle y_k - \hat{\theta}_{k-1})\,, \tag{4.15}$$

where (4.14) is derived by using a linear approximation of the angle (ratio of imaginary and real part) and assuming that the difference $\angle y_k - \hat{\theta}_{k-1}$ is small. Finally, in (4.15), β_k is redefined, with an abuse of notation, as

$$\beta_k = \frac{1/|z_k|}{1/|z_k| + \sigma^2/|y_k|}\,. \tag{4.16}$$

A preliminary comparison of (4.15) and (4.9) reveals the similarity between the two estimators. This similarity is expected in scenarios characterized by reliable estimation, such as those with high SNR. In such contexts, the Tikhonov distribution, characterized by its small dispersion, closely approximates a Gaussian distribution [56]. Consequently, the Tikhonov can be viewed as a Gaussian with variance $1/|z_k|$. However, when comparing (4.7) with (4.16), it can be observed the introduction of time dependence on the variance of the measurement ($\sigma_k^2/|y_k|$). Although not demonstrated here, numerical results showed that the gain in (4.16) does not converge to a steady-state value, but its average $\mathbb{E}[\beta_k]$ does. Additionally, for high SNR, it holds that $\mathbb{E}[\beta_k] \to \beta_\infty$, and $1/|z_k| \to \sigma_\infty^2$. Thus, despite the fluctuations of β_k in (4.16) around its mean value, the scheme presented in Figure 4.3 introduces a new concept of a first-type digital PLL.

4.3 Key Algorithmic Features

In high SNR scenarios, mathematical derivations from Sections 4.1 and 4.2 suggest that the Kalman and Tikhonov PLLs offer comparable performance. However, having support within the range $(0,2\pi]$, the Tikhonov pdf aligns perfectly with the nature of phase distributions. This results in slightly improved phase estimation in scenarios with strong phase noise and has implications in scenarios where data modulate the transmitted carrier. For instance, consider the case of a binary modulation that directly modulates the carrier in (4.1) with symbols $x_k \in \{\pm 1\}$. For demodulation, it is beneficial to perform a "hard" decision on the phase for

the Kalman filter by compensating the received signal using its estimation. Log-likelihood ratios (LLRs) are then computed, by definition, as

$$\lambda_k = \log\left[\frac{p(\tilde{y}_k|x_k=1)}{p(\tilde{y}_k|x_k=-1)}\right],$$

where $\tilde{y}_k = y_k e^{-j\hat{\theta}_k}$, and the distribution $p(\tilde{y}_k|x_k = \pm 1) \sim \mathcal{N}(\tilde{y}_k, x_k = \pm 1, \sigma^2)$. In contrast, the Tikhonov PLL performs a "soft" approach as described in [5], computing LLRs as

$$\lambda_k = \log\left[\frac{I_0\left(\left|z_k + \frac{y_k}{\sigma^2}\right|\right)}{I_0\left(\left|z_k - \frac{y_k}{\sigma^2}\right|\right)}\right],$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind and order 0. The Tikhonov-based "soft" approach yields better code error rate (CER) performance than the Kalman method. However, it would be possible to explore a hybrid approach that integrates the Kalman phase recursion based on Gaussian distributions with the LLR computation using the soft Tikhonov approach. Another advantage of the Tikhonov PLL scheme is its extensive literature support. Following the work in [5], a series of studies have produced advanced forward & backward algorithms based on the Tikhonov pdf to enhance tracking performance and resilience in the absence of pilots. These algorithms could be used to derive equivalent Tikhonov-based PLLs. For example, the algorithm described in [14] proposes a Tikhonov mixture approximation of SPA messages. The authors demonstrate that the algorithm effectively represents a set of PLLs with a controller determining the number of tracking loops necessary to maintain tracking on all trajectories. In a different approach, the authors in [59] reduce the Tikhonov mixture order to one per variable node, enabling its implementation using a standard PLL scheme. Likewise, the decision-aided variant of the Tikhonov algorithm, as discussed in [5], can be implemented using a conventional PLL that makes direct decisions on the symbols based on a threshold for selected symbols (see Section 3.1.2). Potential areas for further research could include establishing the equivalency with higher-order PLLs in the presence of N-dimensional pdfs, for enabling also frequency shift and rate estimation, as well as expanding the investigation to other pdfs belonging to the exponential family.

4.4 Performance Comparison

This section compares the performance of the PLL architectures discussed earlier. These include the classical (first-type) PLL, the Kalman filter (denoted as KF), the Tikhonov PLL (referred to as TK-PLL), and a delayed Kalman filter (labeled as delayed KF). The latter corresponds to the KF producing a phase estimate with a one-sample delay. As discussed in Section 4.1, the delayed KF represents

the maximum performance achievable by classical PLLs, which have a fixed gain that needs to be tuned depending on the scenario.

Consider the case of a pure carrier transmission through a channel affected by phase noise as in (4.1). Figure 4.5 compares the phase jitter over PT/N_0 for a channel affected by phase noise with a standard deviation of σ_Δ of 0.1 and 6 deg. The figure includes multiple curves corresponding to different loop gain values for the classical PLL. When $\sigma_\Delta = 6$ deg, the figure reveals that the TK-PLL outperforms classical PLLs, which are lower-bounded by the KF delayed by one sample. This improvement is due to the variable loop gain inherent in the KF and TK-PLL schemes, as opposed to the constant gain of classical PLLs[2]. The KF and the TK-PLL exhibit a linear decrease in jitter as PT/N_0 increases, with the TK-PLL showing slightly lower jitter at low SNR. When σ_Δ is small, the jitter of the TK-PLL and the delayed version of the KF coincide, except for low SNR values.

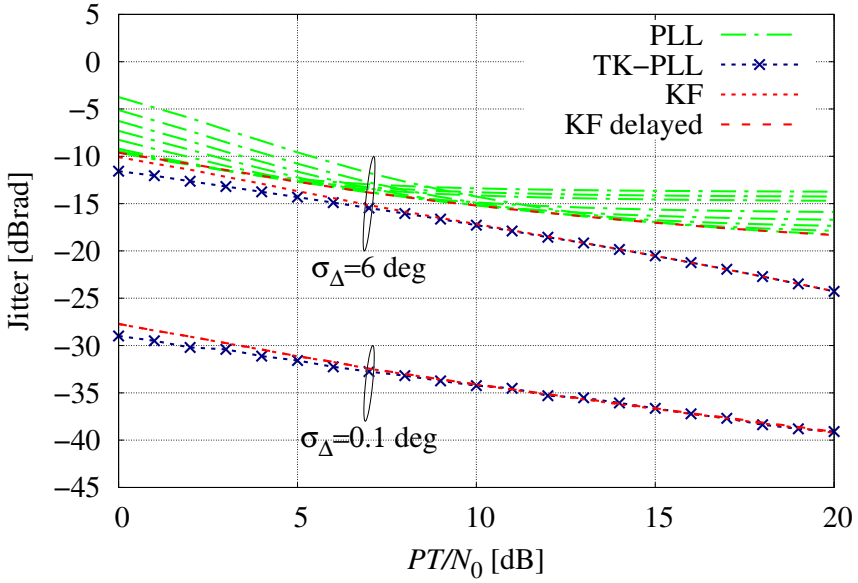


Figure 4.5: Jitter performance comparison with phase noise σ_Δ of 0.1 and 6 deg.

Figure 4.6 shows the time-evolution of both the KF gain, β_k, and the TK-PLL mean value, $\mathbb{E}[\beta_k]$, with a phase noise of $\sigma_\Delta = 6$ deg and PT/N_0 values of 0 and 10 dB. As expected, the KF gain β_k rapidly converges to its steady-state value β_∞. On the other hand, as discussed in Section 4.2, $\mathbb{E}[\beta_k]$ tends to β_∞ only in high-SNR scenarios. This can be observed at a PT/N_0 of 10 dB in Figure 4.6. However, when PT/N_0 decreases to 0 dB, $\mathbb{E}[\beta_k]$ converges to a larger value than the Kalman

[2]The benefits of adaptive loop gain in reducing phase jitter variance have been demonstrated in prior studies, particularly in the context of QAM carrier phase synchronization [66].

steady-state gain β_∞. This suggests that, on average, the TK-PLL tends to keep a larger loop bandwidth for more effective phase noise tracking.

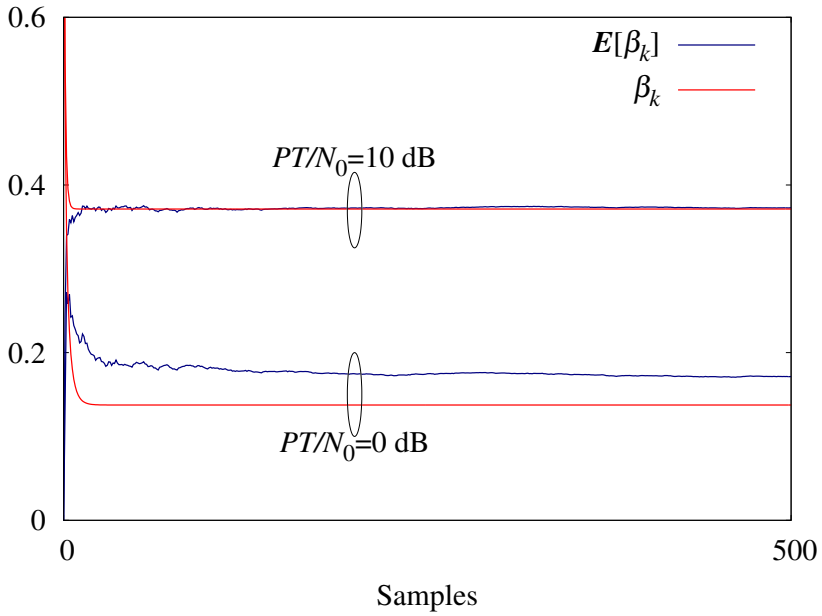


Figure 4.6: Evolution of the loop gain β_k (KF) and $\mathbb{E}[\beta_k]$ (TK-PLL) as function of time, with phase nosie $\sigma_\Delta = 6$ deg, for a PT/N_0 of 0 and 10 dB.

Finally, consider the transmission of a carrier modulated by binary data as discussed in the previous section. A short LDPC code having a code rate of 1/2 and a block length of 256 [67] is considered, as well as binary phase-shift keying (BPSK) modulation, which is typical of low-rate applications where severe phase noise may occur. To facilitate the demodulation and tracking, the pilot symbols are distributed along the transmitted sequence (interspersed), with a frequency of one pilot for every 20 symbols. Both the KF and the TK-PLL adopt a data-aided approach where an update in the phase estimate only occurs in the presence of pilots. Figure 4.7 compares the CER of the two schemes when the phase noise process has standard deviation σ_Δ of 1.5 and 6 deg. As the figure shows, the performance of the TK-PLL and the KF algorithms is nearly identical for a small phase noise of σ_Δ of 1.5 deg. However, the TK-PLL algorithm performs better with a higher value of phase noise standard deviation (6 deg). The results in Figures 4.5 and 4.6 revealed identical phase tracking capabilities for both algorithms. Hence, the disparity in CER between the two algorithms under $\sigma_\Delta = 6$ deg can be attributed to differences in their LLR computation. For small values of σ_Δ, $I_0(\cdot)$ behaves similarly to an exponential function. This explains the minimal differences observed between the two CER curves when $\sigma_\Delta = 1.5$ deg. However, as the phase noise standard deviation increases, the behavior of $I_0(\cdot)$ and an exponential function gradually start to

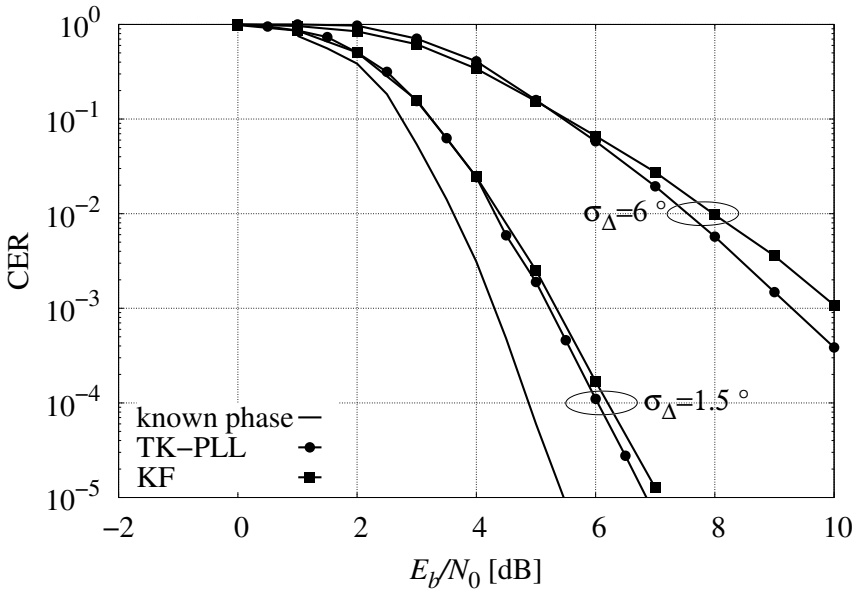


Figure 4.7: CER performance of the TK-PLL and the KF schemes for a phase noise characterized by a σ_Δ of 1.5 and 6 deg.

differ. This can cause the LLRs of the KF to be less precise representations of the true LLRs. Differently, the $I_0(\cdot)$ function provides a better approximation of the LLRs, which is reflected in the performance improvement observed in Figure 4.7. The TK-PLL's performance could be further improved using more advanced algorithmic features such as in [5], [14], [59] discussed in Section 4.3. Moreover, both the KF and the TK-PLL performance could be improved by utilizing a non-linear model that matches the non-linear observation equation in (4.1) [19], [68]–[70].

Chapter 5

Oversampling Techniques for Phase Noise Channels

In scenarios where low data rate signals are received with low SNRs and are subject to rapidly changing dynamics, achieving reliable detection and synchronization is particularly challenging. The low data rate results in relatively slow signal dynamics variations within each symbol period. However, the highly varying noise dynamics introduce complex challenges, as significant variations can occur within a single symbol duration. These rapid fluctuations at low SNR make the synchronization task challenging. This chapter examines the potential of oversampling techniques to enhance the performance of detection and synchronization algorithms, particularly under strong phase noise channels. By capturing multiple samples per symbol, oversampling offers finer temporal resolution, which can mitigate the effects of phase noise. Section 5.1 introduces an algorithm in the BCJR form that operates on samples rather than symbols. Section 5.2 presents a reduced-complexity alternative based on the Tikhonov distribution.

5.1 Multi-sample dp-BCJR

Section 3.1 described different solutions to the problem of phase noise tracking and signal detection. The symbol-level discrete-time channel model in (3.1) assumes that the phase process changes slowly compared to the signal dynamics. This assumption is valid when the phase variation over symbol time is less than 6 deg [26]. However, if the variations are faster, the discrete-time model no longer fits, and receivers based on this model would experience information loss [71]. To address this issue, Ghozlan and Kramer ([72], [73]) proposed oversampling to obtain a sufficient statistic. Based on their findings, a detection algorithm in the BCJR [55] form is derived using the factor graph and SPA framework. This section is based on the content of the work in [74].

5.1.1 Channel Model

Consider the transmission of a linearly modulated signal $x(t)$ over an AWGN channel, affected by phase noise. The received signal has a complex base-band expression

$$r(t) = x(t)e^{j\theta(t)} + w(t)\,, \tag{5.1}$$

where $w(t)$ is complex-valued white Gaussian noise with spectral density N_0 and $\theta(t)$ is the phase noise. The transmitted signal $x(t)$ reads

$$x(t) = \sum_k c_k p(t - kT)\,,$$

being $p(t)$ the shaping pulse, T the symbol time, and $\{c_k\}$ the sequence of information symbols belonging to an $M-$ary complex-valued constellation. Without loss of generality, consider that $p(t)$ has unitary energy, satisfies the Nyquist criterion (no inter-symbol interference, ISI), and that the constellation symbols have energy $\mathbb{E}\{|c_k|^2\} = E_s$. In addition, the Wiener model is adopted to describe the phase noise process.

5.1.2 Auxiliary Channel with Oversampling

Starting from the continuous-time channel in (5.1), the symbol-time discrete channel in (3.1) is obtained under the assumption that $\{y_k\}$ are the samples at the output of a matched filter (MF), as shown in Figure 5.1a. If $\theta(t)$ changes

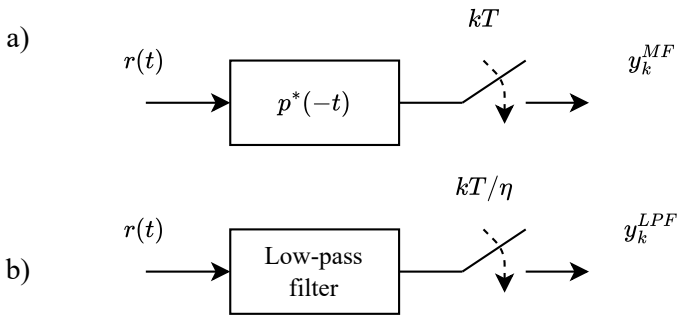


Figure 5.1: Receiver block diagram when a) using a matched filter and b) a low-pass filter with oversampling.

slowly w.r.t. the symbol time, the observable at the MF output is approximately $y_k^{MF} \approx y_k$. This approximation will no longer be valid when the phase noise variance σ_Δ^2 is "large".

A channel model derived using the oversampling method described in [75] is considered instead. The signal $r(t)$ is input to an ideal low-pass filter (LPF) with

a bandwidth equal to η/T, with η being the oversampling factor, as shown in Figure 5.1b. The output of the LPF is then sampled at intervals of T/η, resulting in the sequence $\{y_k^{LPF}\}$. The phase variation between two adjacent samples is σ_Δ^2/η, and it decreases as η increases. With a sufficiently large oversampling factor chosen, the notation is slightly abused to write

$$y_k^{LPF} \approx y_k = \sum_i c_i p_{k-i\eta} e^{j\theta_k} + w_k \,, \tag{5.2}$$

where $p_k = p(kT/\eta)$, $\theta_k = \theta(kT/\eta)$, and w_k has variance $N_0\eta$, thus η times larger rather than in (3.1).

The design of receivers based on a mismatched channel is an instance of *mismatched decoding* [76], [77]. More specifically, being $\boldsymbol{y}\,, \boldsymbol{c}$ the vectors collecting the observable and transmitted symbols, if the receiver is based on an *auxiliary channel* [78] with pdf $q(\boldsymbol{y}|\boldsymbol{c})$ (different from the actual channel law), it will have an achievable information rate (AIR, [79]) equal to

$$I_{\text{AIR}} = \lim_{N\to\infty} \frac{1}{N} \left[\mathbb{E}\left\{-\log_2 q(\boldsymbol{y})\right\} + \mathbb{E}\left\{\log_2 q(\boldsymbol{y}|\boldsymbol{c})\right\}\right] ,$$

where N is the number of transmitted symbols, $q(\boldsymbol{y}) = \sum_{\boldsymbol{c}} q(\boldsymbol{y}|\boldsymbol{c})P(\boldsymbol{c})$, and the expectations are computed w.r.t. the actual channel statistics. The AIR is defined as the highest of all achievable rates and it is upper-bounded as

$$I_{\text{AIR}} \leq I\left(\boldsymbol{c};\boldsymbol{r}\right) ,$$

where $I\left(\boldsymbol{c};\boldsymbol{r}\right)$ is the information rate of the actual channel. This holds for both the information rates I_{AIR}^{MF} and I_{AIR}^{LPF}, achievable when adopting the auxiliary channels in (3.1) and (5.2), respectively. However, when using an oversampling factor $\eta \to \infty$, y_k^{LPF} tends to be a sufficient statistic for the channel output, and thus $I_{\text{AIR}}^{LPF} \to I\left(\boldsymbol{c};\boldsymbol{r}\right)$. A detailed analysis of the AIR when using an oversampled auxiliary channel was done in [72], showing that I_{AIR}^{LPF} outperforms I_{AIR}^{MF}. In light of this, a detection algorithm based on oversampling is derived in the next section.

5.1.3 Derivation of Multi-sample dp-BCJR

For the discrete-time auxiliary channel in (3.1), the authors in [5] derived the dp-BCJR algorithm described in Section 3.1.1. A similar approach is adopted by defining an equivalent algorithm that operates on the oversampled observable.

In its most general form, the algorithm derivation for the channel model in (5.2) is quite complex as it needs a BCJR algorithm that can handle both ISI and the phase noise model. However, by adopting time-limited shaping pulses, the problem becomes more tractable. Therefore, any shaping pulse with duration $\leq T$ can be considered by including the coefficients $p_k = p(kT/\eta)$ into (5.3). Without loss

of generality, a squared shaping pulse is considered [1] and the channel equation simplifies to

$$y_k = c_{\lfloor k/\eta \rfloor} e^{j\theta_k} + w_k \,. \tag{5.3}$$

Using the factor graphs SPA framework [16], a dp-BCJR algorithm for the channel model in (5.3) is derived.

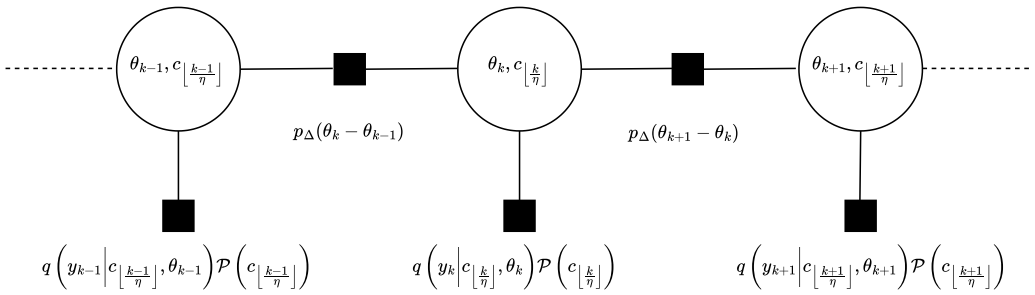


Figure 5.2: Factor graph representing the factorization in (5.4).

The APP pdf $q(\boldsymbol{c}, \boldsymbol{\theta}|\boldsymbol{y})$ can be factorized as

$$\begin{aligned} q(\boldsymbol{c}, \boldsymbol{\theta}|\boldsymbol{y}) &\propto q(\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{\theta}) P(\boldsymbol{c}) p(\boldsymbol{\theta}) \\ &= \left(\prod_{k=0}^{K\eta-1} q(y_k|c_{\lfloor k/\eta \rfloor}, \theta_k) \mathcal{P}(c_{\lfloor k/\eta \rfloor}) \right) p(\boldsymbol{\theta}) \,, \end{aligned} \tag{5.4}$$

where $q(y_k|c_{\lfloor k/\eta \rfloor}, \theta_k)$ is a Gaussian pdf with average $c_{\lfloor k/\eta \rfloor} e^{j\theta_k}$ and variance $N_0\eta$. The term $\mathcal{P}(c_{\lfloor k/\eta \rfloor})$ is instead an indicator function that takes values

$$\mathcal{P}(c_{\lfloor k/\eta \rfloor}) = \begin{cases} P(c_{\lfloor k/\eta \rfloor}) & \text{if } (k+1) \text{ is multiple of } \eta \\ 1 & \text{otherwise} \end{cases} \tag{5.5}$$

being $P(c_{\lfloor k/\eta \rfloor})$ the probability of the information symbol $c_{\lfloor k/\eta \rfloor}$. In turn, the probability of the phase is factorized as

$$\begin{aligned} p(\boldsymbol{\theta}) &= p(\theta_0) \prod_{k=1}^{K\eta-1} p(\theta_k|\theta_{k-1}) \\ &= \frac{1}{2\pi} \prod_{k=1}^{K\eta-1} p_\Delta(\theta_k - \theta_{k-1}) \,, \end{aligned} \tag{5.6}$$

[1] For instance, in [80], it is recommended to adopt a residual carrier modulation scheme with a square-wave subcarrier for transmitting low data rates in deep space missions.

where $p_\Delta(\theta)$ is defined as in (2.8). By plugging (5.6) into (5.4), the factorization can be visualized in the factor graph of Figure 5.2. In this graph, the variables couple $(\theta_k, c_{\lfloor k/\eta \rfloor})$ are clustered into a single variable node. Clustering is a common technique used to eliminate the short cycles in the factor graph that would impact the SPA convergence [16].

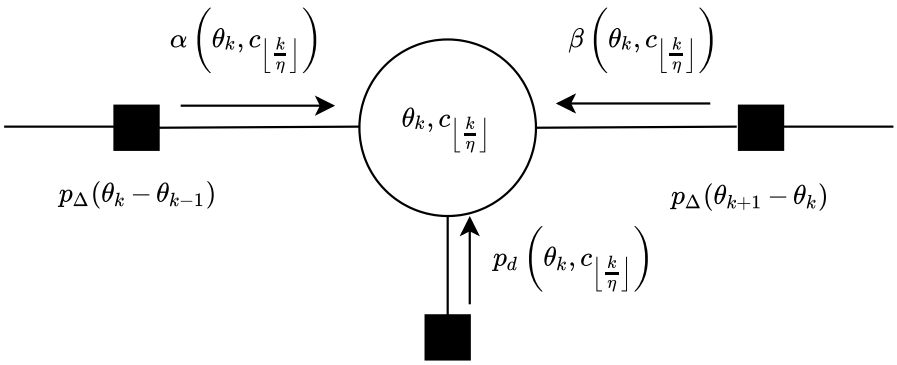


Figure 5.3: In-depth view of the messages passing through a single variable node.

Defining $\alpha(\cdot)$, $\beta(\cdot)$, and $p_d(\cdot)$ as the messages shown in Figure 5.3, the forward recursion can be derived as

$$\alpha(\theta_k, c_{\lfloor k/\eta \rfloor}) = \sum_{c_{\lfloor (k-1)/\eta \rfloor}} \int_0^{2\pi} \alpha(\theta_{k-1}, c_{\lfloor (k-1)/\eta \rfloor}) p_d(\theta_{k-1}, c_{\lfloor (k-1)/\eta \rfloor}) p_\Delta(\theta_k - \theta_{k-1}) \, \mathrm{d}\theta_{k-1} \tag{5.7}$$

when k is multiple of η, and,

$$\alpha(\theta_k, c_{\lfloor k/\eta \rfloor}) = \int_0^{2\pi} \alpha(\theta_{k-1}, c_{\lfloor (k-1)/\eta \rfloor}) p_d(\theta_{k-1}, c_{\lfloor (k-1)/\eta \rfloor}) p_\Delta(\theta_k - \theta_{k-1}) \, \mathrm{d}\theta_{k-1} \tag{5.8}$$

otherwise, where

$$p_d(\theta_k, c_{\lfloor k/\eta \rfloor}) = q(y_k | c_{\lfloor k/\eta \rfloor}, \theta_k) \mathcal{P}(c_{\lfloor k/\eta \rfloor}) \, .$$

Similarly, the backward recursion can be derived as

$$\beta(\theta_k, c_{\lfloor k/\eta \rfloor}) = \sum_{c_{\lfloor (k+1)/\eta \rfloor}} \int_0^{2\pi} \beta(\theta_{k+1}, c_{\lfloor (k+1)/\eta \rfloor}) p_d(\theta_{k+1}, c_{\lfloor (k+1)/\eta \rfloor}) p_\Delta(\theta_{k+1} - \theta_k) \mathrm{d}\theta_{k+1} \tag{5.9}$$

when $k+1$ is multiple of η, and

$$\beta(\theta_k, c_{\lfloor k/\eta \rfloor}) = \int_0^{2\pi} \beta(\theta_{k+1}, c_{\lfloor (k+1)/\eta \rfloor}) p_d(\theta_{k+1}, c_{\lfloor (k+1)/\eta \rfloor}) p_\Delta(\theta_{k+1} - \theta_k) \mathrm{d}\theta_{k+1} \tag{5.10}$$

otherwise. The k-th forward and backward messages, as defined in (5.7) and (5.9), require clarification. (5.7) and (5.9) deal with the transition between two adjacent symbols in the factor graph. When calculating the forward or backward messages input to the first sample of a new symbol, the previous symbol will be marginalized out. As a result, in the case of (5.7) and (5.9), the probability density functions will depend on the sole variable θ_k.

Finally, the maximum a posteriori probability on the information symbols can be derived as $q(\boldsymbol{y}|c_k)P(c_k)$, being $q(\boldsymbol{y}|c_k)$ the extrinsic information computed as

$$q(\boldsymbol{y}|c_k) = \int_0^{2\pi} \alpha(\theta, c_k) q(y_k|c_k, \theta) \beta(\theta, c_k) \mathrm{d}\theta \,. \tag{5.11}$$

For the sake of clarity, $\theta = \theta_{(k+1)\eta-1}$ is defined in (5.11) for $k = 0, ..., K-1$ and, with a slight abuse of notation, $\beta(\theta)$ is referred to as $\beta(\theta, c_k)$. The exact computation of (5.7)–(5.11) is impractical, since it involves continuous pdfs. However, as already described in Section 3.1.1, the integral can be solved by considering the phase as a discrete random variable taking on L values between [0,2π). Following this approach, a multi-sample dp-BCJR algorithm is obtained. The algorithm can be used for iterative detection and decoding by setting the probabilities in (5.5) with the a priori information on the symbols, and using the extrinsic information in (5.11) as input to the decoder (see Section 3.1.1).

Differently from what is shown in Section 3.1.1, the complexity in this case scales as $\mathcal{O}(KML^2\eta)$. Even though the complexity is η times larger than that of the classical dp-BCJR, it can be still tractable with a standard workstation for practical cases. However, in many applications where there are not enough computational resources, it would be desirable to limit the complexity. For this reason, the next section introduces a reduced-complexity version of the multi-sample dp-BCJR algorithm.

5.2 Multi-sample Tikhonov

This section proposes a reduced-complexity version of the algorithm in Section 5.1 based on the oversampling concept. Similar to [5], the goal is to reduce the algorithm's complexity by utilizing the Tikhonov distribution.

5.2.1 Derivation of Multi-sample Tikhonov

Starting from the channel model in (5.3), the factor graph and SPA framework is applied as outlined in Section 5.1. However, this time, instead of approximating the integrals with a discrete sum over L discrete levels of θ_k, the forward and backward messages are constrained as weighted Tikhonov distributions

$$\alpha(\theta_k, c_{\lfloor k/\eta \rfloor}) \triangleq \lambda_k^f t(\theta_k; z_k^f) \,, \tag{5.12}$$

$$\beta(\theta_k, c_{\lfloor k/\eta \rfloor}) \triangleq \lambda_k^b t(\theta_k; z_k^b)\,, \tag{5.13}$$

and solve the integrals in (5.7), (5.8), (5.9), and (5.10) accordingly.

Similar to Section 5.1, two cases are handled for both the forward and backward messages. Focusing on the forward recursion, the first case occurs when the time instant k is a multiple of the oversampling factor η. In this scenario, by plugging (5.12) into (5.7) the forward reads

$$\begin{aligned}\alpha(\theta_k, c_{\lfloor k/\eta \rfloor}) &\propto \int_0^{2\pi} \sum_{c_{\lfloor (k-1)/\eta \rfloor}} P(c_{\lfloor (k-1)/\eta \rfloor}) e^{-\frac{|c_{\lfloor (k-1)/\eta \rfloor}|^2}{2\sigma^2}} \exp\left\{\left[u_{k-1} e^{-j\theta_{k-1}}\right]\right\} \\ &\quad \times \lambda_{k-1}^f t^f(\theta_{k-1}; z_{k-1}^f) p_\Delta(\theta_k - \theta_{k-1}) d\theta_{k-1} \qquad (5.14) \\ &\propto \int_0^{2\pi} \sum_{c_{\lfloor (k-1)/\eta \rfloor}} w_{k-1}^f t(\theta_{k-1}; \hat{z}_{k-1}^f) p_\Delta(\theta_k - \theta_{k-1}) d\theta_{k-1}\,, \qquad (5.15)\end{aligned}$$

where

$$\begin{aligned} u_k &= \frac{y_k c_{\lfloor k/\eta \rfloor}^*}{\sigma^2}\,, \\ \hat{z}_{k-1}^f &= z_{k-1}^f + u_{k-1}\,, \\ w_{k-1}^f &= \lambda_{k-1}^f \frac{I_0\left(\left|\hat{z}_{k-1}^f\right|\right)}{I_0\left(\left|z_{k-1}^f\right|\right)} e^{-\frac{|c_{\lfloor (k-1)/\eta \rfloor}|^2}{2\sigma^2}}\,. \end{aligned}$$

It can be observed that the term inside the summation over $c_{\lfloor (k-1)/\eta \rfloor}$ in (5.15) is a Tikhonov mixture of order M. To limit complexity, the Tikhonov mixture is reduced according to the circular mean and variance matching (CMVM) presented in [14]. It results that the sequence of the forward Tikhonov coefficients can be recursively computed as

$$z_k^f = \hat{z}_{k-1}^{f,m} \gamma\left(\sigma_\Delta^2, \left|\hat{z}_{k-1}^{f,m}\right|\right)\,, \tag{5.16}$$

where $\gamma(x_1, x_2) = \frac{1}{1+x_1 x_2}$, and according to [14], $\hat{z}_{k-1}^{f,m}$ is derived as

$$t(\theta_k; \hat{z}_{k-1}^{f,m}) = CMVM\left(\sum_{c_{\lfloor (k-1)/\eta \rfloor}} w_{k-1}^f t(\theta_{k-1}; \hat{z}_{k-1}^f)\right)\,. \tag{5.17}$$

After marginalizing over the symbol, the resulting Tikhonov distribution depends only on the phase.

The second case occurs instead when k is not multiple of η, and by plugging (5.12) into (5.8), the forward reads

$$\begin{aligned}\alpha(\theta_k, c_{\lfloor k/\eta \rfloor}) &\propto \frac{I_0\left(\left|z_{k-1}^f + u_{k-1}\right|\right)}{I_0\left(\left|z_{k-1}^f\right|\right)} \int_0^{2\pi} \lambda_{k-1}^f t^f(\theta_{k-1}; z_{k-1}^f + u_{k-1}) p_\Delta(\theta_k - \theta_{k-1}) d\theta_{k-1} \\ &\propto \lambda_k^f t^f(\theta_k; z_k^f)\,, \qquad (5.18)\end{aligned}$$

where

$$z_k^f = \left(z_{k-1}^f + u_{k-1}\right) \gamma\left(\sigma_\Delta^2, \left|z_{k-1}^f + u_{k-1}\right|\right) , \tag{5.19}$$

and

$$\lambda_k^f = \lambda_{k-1}^f \frac{I_0\left(\left|z_{k-1}^f + u_{k-1}\right|\right)}{I_0\left(\left|z_{k-1}^f\right|\right)} . \tag{5.20}$$

For the backward recursion, the complex coefficients can be calculated recursively as

$$z_k^b = \hat{z}_{k+1}^{b,m} \gamma\left(\sigma_\Delta^2, \left|\hat{z}_{k+1}^{b,m}\right|\right) , \tag{5.21}$$

where $\hat{z}_{k+1}^{b,m}$ results from

$$t(\theta_{k+1}, \hat{z}_{k+1}^{b,m}) = CMCV\left(\sum_{c_{\lfloor (k+1)/\eta \rfloor}} w_{k+1}^b t(\theta_{k+1}, \hat{z}_{k+1}^b)\right) ,$$

with

$$\hat{z}_{k+1}^b = z_{k+1}^b + u_{k+1} ,$$

$$w_{k+1}^b = \lambda_{k+1}^b \frac{I_0\left(\left|\hat{z}_{k+1}^b\right|\right)}{I_0\left(\left|z_{k+1}^b\right|\right)} e^{-\frac{|c_{\lfloor (k+1)/\eta \rfloor}|^2}{2\sigma^2}} ,$$

when $k+1$ is a multiple of η, and

$$z_k^b = \left(z_{k+1}^b + u_{k+1}\right) \gamma\left(\sigma_\Delta^2, \left|z_{k+1}^b + u_{k+1}\right|\right) , \tag{5.22}$$

$$\lambda_k^b = \lambda_{k+1}^b \frac{I_0\left(\left|z_{k+1}^b + u_{k+1}\right|\right)}{I_0\left(\left|z_{k+1}^b\right|\right)} , \tag{5.23}$$

otherwise. As it was pointed out in Section 5.1, after the marginalization of the symbol, the resulting message (forward or backward) will be the sole function of the phase variable.

Finally, using the detection algorithm in an iterative detection and decoding scheme, the extrinsic information can be calculated according to (5.11) as

$$\begin{aligned} q(\mathbf{y}|c_k) &\propto \int_0^{2\pi} \alpha(\theta, c_k) q(y_k|c_k) \beta(\theta) d\theta \\ &\propto \frac{\lambda^f I_0\left(\left|z^f + z^b + \frac{y_k c_k^*}{\sigma^2}\right|\right)}{I_0\left(|z^f|\right) I_0\left(|z^b|\right)} , \end{aligned} \tag{5.24}$$

where for simplicity it is defined $\theta = \theta_{(k+1)\eta-1}$, $\lambda^f = \lambda_{(k+1)\eta-1}^f$, $z^f = z_{(k+1)\eta-1}^f$, and $z^b = z_{(k+1)\eta-1}^b$, with $k = 0, ..., K-1$.

The multi-sample Tikhonov algorithm's complexity scales down as $\mathcal{O}(KM\eta)$, which is L^2 times lower than the multi-sample dp BCJR of Section 5.1, and L/η times lower than the classical dp BCJR of Section 3.1.1.

5.3 Performance Comparison under Phase Noise Channel

This section compares the extrinsic information transfer (EXIT) chart [81] of the following algorithms:

- dp-BCJR (Section 3.1.1),
- multi-sample dp-BCJR (Section 5.1),
- Tikhonov (Section 3.1.2),
- multi-sample Tikhonov (Section 5.2),

First, consider the dp-BCJR and its multi-sampled version, the multi-sample BCJR. Using the Monte Carlo method shown in [78], the AIR I_{AIR}^{LPF} (representative of the AIR of the multi-sample dp-BCJR) is derived as a function of the SNR E_s/N_0. Figure 5.4 shows the example of a quaternary phase-shift keying (QPSK) and 16-PSK, over a phase noise channel with $\sigma_\Delta = 28$ deg when using $L = 32$ and $\eta = 4$. In all cases, 1 pilot for every 20 symbols is considered. The figure also illustrates

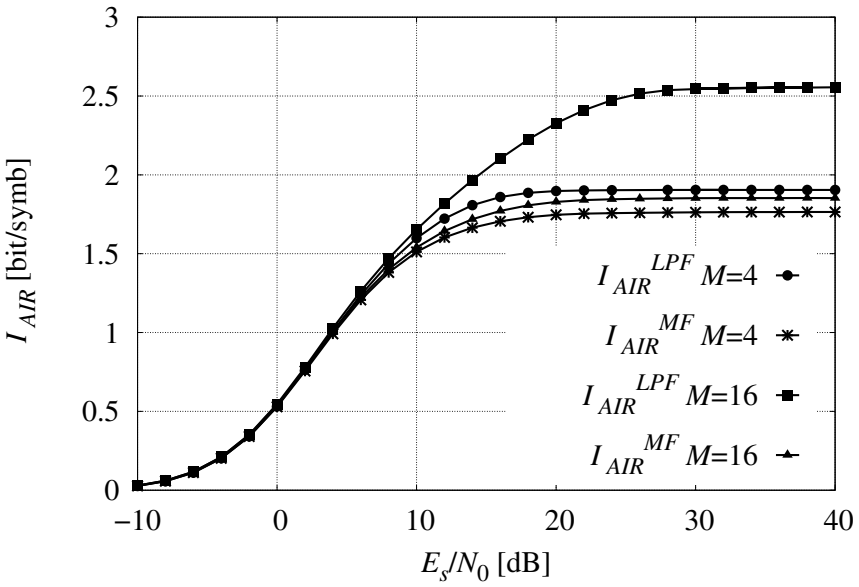


Figure 5.4: Achievable information rates for QPSK (M=4) and 16-PSK (M=16) constellation, with $\sigma_\Delta = 28$ deg, $\eta = 4$, and $L = 32$.

I_{AIR}^{MF} for $L = 32$, which represents the AIR of the classical dp-BCJR. As demonstrated in [72], the multi-sample dp-BCJR exhibits a higher AIR as the E_s/N_0

increases, meaning that performance improves when phase noise becomes the main limiting factor. The gain further improves as the constellation cardinality M grows, especially at high σ_Δ values. This trend is particularly noticeable for M-PSK modulation. On the other hand, when using constellations that are more robust to phase noise, such as quadrature amplitude modulation (QAM), the advantages of the multi-sample dp-BCJR tend to diminish. For example, Figure 5.5 shows the case of a 16-QAM over a phase noise channel with $\sigma_\Delta = 28$ deg. In this scenario, the gain of the multi-sample dp-BCJR over the classical version is approximately $\sim 14\%$ in the saturation region, compared to around $\sim 38\%$ for 16-PSK. Thus, the use of the multi-sample dp-BCJR should always be carefully balanced between complexity and target information rate. As thoroughly detailed in [74], an analysis

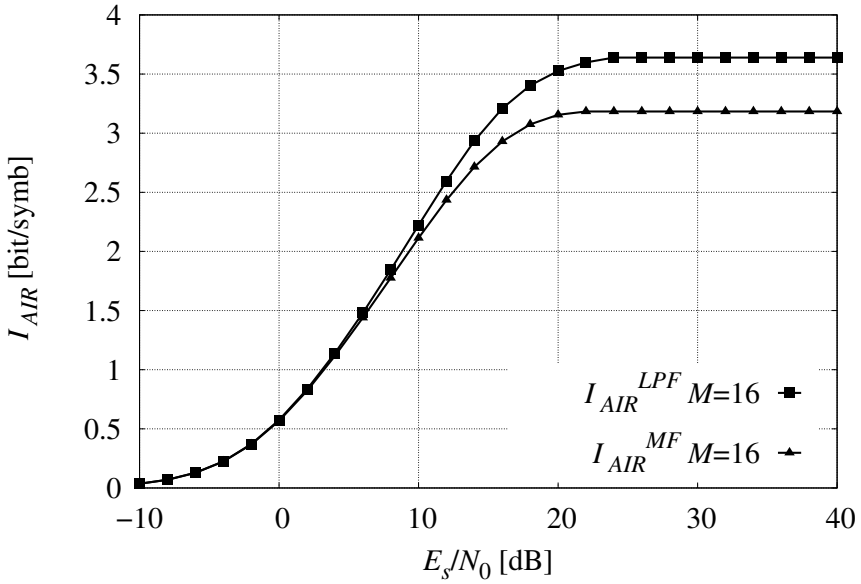


Figure 5.5: Achievable information rates for 16-QAM (M=16) constellation with $\sigma_\Delta = 28$ deg, $\eta = 4$, and $L = 32$.

of iterative detection and decoding schemes using the extrinsic information transfer (EXIT) charts of the two decoders with LDPC codes of block lengths of 4096 and 64800, and a rate of 1/2 [32], revealed that these codes are not suitable for strong phase noise channels. Consequently, there is a need to redesign the error correction codes for effective use in such environments. For further details, please refer to [74].

For the oversampling technique to benefit the dp-BCJR algorithm, the phase noise standard deviation, σ_Δ, must be sufficiently high (greater than 10 deg), as well as the modulation cardinality. This requirement limits the practical applications of the techniques, especially considering the additional complexity that would need to be addressed on top of the inherent complexity of the standard dp-BCJR.

A more interesting topic for discussion is the multi-sample Tikhonov algorithm in comparison to the classical Tikhonov and dp-BCJR. The Tikhonov, multi-sample Tikhonov, and dp-BCJR (with $L = 32$ discretization levels) are simulated over the channel described in equation (5.1), using a phase noise standard deviation of $\sigma_\Delta = 6$ deg and a squared shaping pulse $p(t)$ with $\eta = 4$, across various modulation formats: PSK, QAM, and amplitude phase-shift keying (APSK). The performance is evaluated using EXIT charts. Figure 5.6 presents the results for 16−PSK mod-

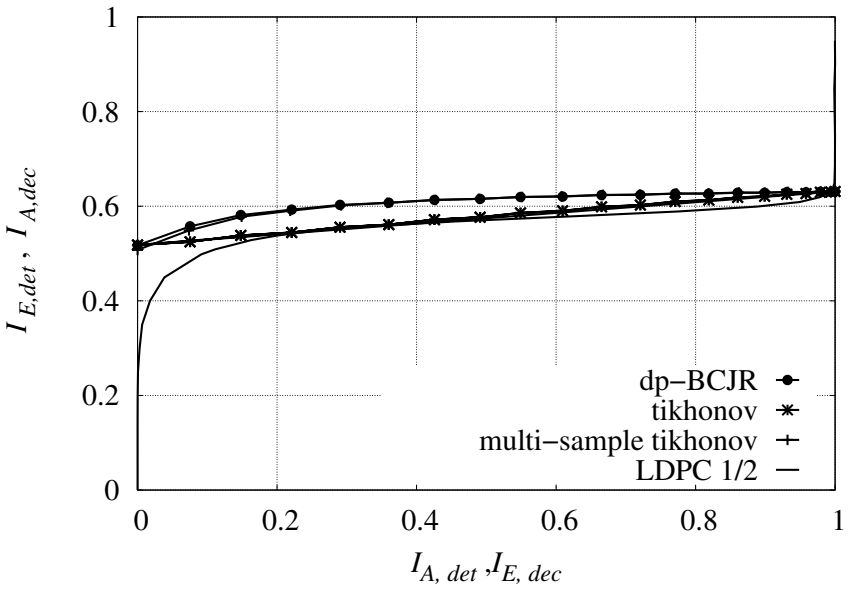


Figure 5.6: EXIT chart for 16−PSK at $E_s/N_0 = 10$ dB with $\sigma_\Delta = 6$ deg, for multi-sample Tikhonov ($\eta = 4$), Tikhonov, dp-BCJR ($L = 32$), and an LDPC code with rate 1/2.

ulation at E_s/N_0 of 10 dB, Figure 5.7 for 64−APSK modulation at $E_s/N_0 = 13.5$ dB, and finally, Figure 5.8 for 128−QAM modulation at E_s/N_0 of 15.5 dB. The three figures demonstrate the superiority of the multi-sample Tikhonov algorithm compared to the standard Tikhonov, as expected. Remarkably, the multi-sample Tikhonov achieves performance comparable to the dp-BCJR independently of the modulation format adopted. This finding is significant because it not only shows performance equivalent to an algorithm known to be optimal (to the extent of the number of discretization levels) but also does so with considerably lower complexity. The multi-sample Tikhonov allows us to achieve optimal results with a complexity scale down of L/η. In practical scenarios, depending on the SNR and the phase noise level, practical values of L for the dp-BCJR are typically 16 or 32 [72]. Optimal performance of the multi-sample Tikhonov is observed when $\eta = 4$, indicating that the complexity can be reduced by at least a factor 4 or 8. This reduction in

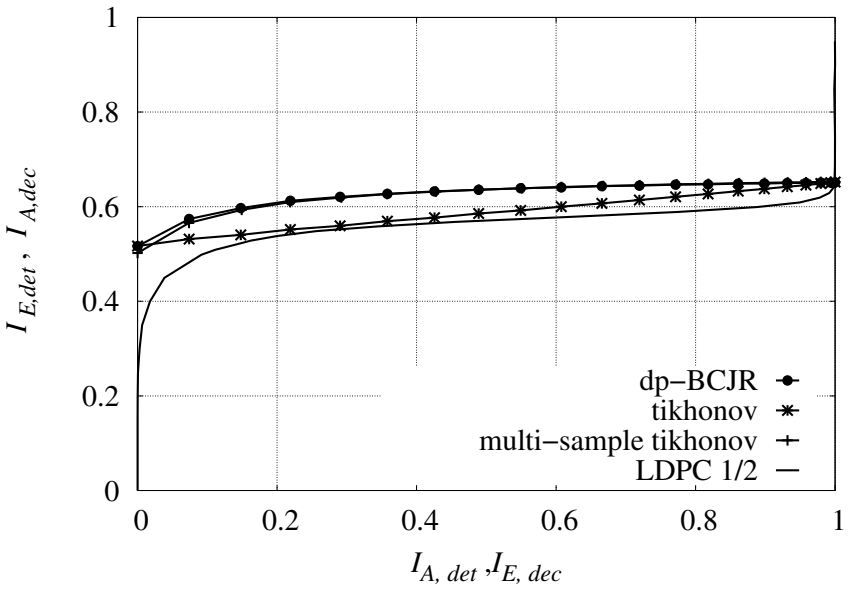


Figure 5.7: EXIT chart for 64−APSK at $E_s/N_0 = 13.5$ dB with $\sigma_\Delta = 6$ deg, for multi-sample Tikhonov ($\eta = 4$), Tikhonov, dp-BCJR ($L = 32$), and an LDPC code with rate 1/2.

complexity makes the multi-sample Tikhonov particularly appealing for real-time applications, where computational efficiency is crucial.

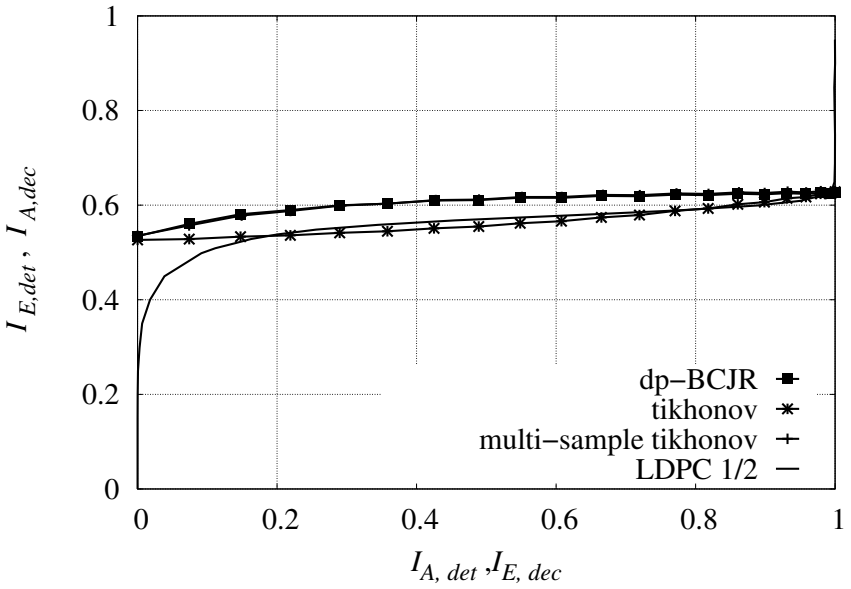


Figure 5.8: EXIT chart for 128−QAM at $E_s/N_0 = 15.5$ dB with $\sigma_\Delta = 6$ deg, for multi-sample Tikhonov, Tikhonov ($\eta = 4$), dp-BCJR ($L = 32$), and an LDPC code with rate 1/2.

Chapter 6

Phase Noise and Time-Varying Frequency Offset: The Case of Deep Space Communications

Soon, deep space missions, such as Ice Giants [82], Vigil [83], and ODINUS [84] will target very distant regions in outer space. Reliable communication in both the Earth-to-Space and Space-to-Earth direction is essential for the success of every space mission. The Earth-to-space link, in particular, is critical as it is used to transmit TCs that are vital for the spacecraft and the mission itself. Achieving successful communication over such large distances with limited available power translates into a data rate reduction. For future deep space missions, the data rate could be as low as a few bits per second or even lower. The current satellite receiver used for deep-space communication employs high-rate codes and classical loop architectures, such as the PLL, which work well under favorable propagation conditions. However, these methods result in sub-optimal receiver performance in the presence of common impairments in deep-space communication links, such as low SNRs, significant Doppler impact, and phase noise. Initially designed for higher SNRs, the current receiver struggles to maintain reliability under these challenging conditions. Although LDPC codes are essential for enabling low TC data rates, they operate at low SNRs that are beyond the capabilities of the existing receiver architecture. As a result, this leads to reduced performance and reliability at lower data rates. Due to the sub-optimal design of the synchronization and coding architecture, a lower bound of approximately 7.8125 bps is imposed on the data rate. This chapter is based on a study commissioned by the European Space Research and Technology Centre, which aims to study and develop advanced detection algorithms suitable for low-data rate signal reception in harsh propagation environments.

6.1 System Scenario

The CCSDS recommends the adoption of residual carrier modulations for TC transmission in deep space missions [80]. The advantage is that the residual carrier acts as a permanent pilot that can aid synchronization, especially at low SNRs. The complex base-band representation of a residual carrier modulation can be expressed as a phase modulation given by

$$r(t) = \exp\{jm_x x(t)\}\,, \tag{6.1}$$

where m_x is the modulation index. The modulating signal is defined as

$$x(t) = \sum_k a_k p(t - kT)\sin(2\pi f_x t)\,, \tag{6.2}$$

with f_x the sub-carrier frequency of 8 or, usually, 16 kHz [80], and T the symbol time. According to the standard, the TC signal is encapsulated with a start and tail sequence of known bits to aid synchronization [67]. In (6.2), $p(t)$ is the non-return-to-zero (NRZ) shaping pulse, and a_k is the k-th information symbol belonging to the BPSK constellation. The reader who is not familiar with the concept of residual carrier modulation can refer to [85], which demonstrates how residual carrier modulation can be approximated using an unmodulated carrier plus a linear modulation. Following the initial work in [3], which identified the carrier and sub-carrier tracking algorithms to constrain the current receiver architecture, advanced designs of the satellite receiver are investigated. The transmission of the signal $r(t)$ through an AWGN channel affected by phase noise and Doppler dynamics is considered. First, a carrier recovery algorithm designed to outperform classical PLLs is investigated, focusing on enhancing overall carrier recovery performance. Following this, a sub-carrier tracking algorithm that operates under the assumption of perfect carrier recovery is designed. This assumption is valid when the phase noise on the carrier is accurately recovered. The need for a sub-carrier independent tracking algorithm arises because the carrier and sub-carrier are not generated coherently at the ground station [86]. Additionally, ideal timing synchronization is assumed, since it is easy to achieve in the low rate scenario, as shown in [3].

6.2 Carrier Tracking

Consider the transmission of a carrier signal over an AWGN channel affected by phase noise, Doppler shift, and Doppler rate. Assuming a phase noise process "slow" enough w.r.t the sampling time T_s, the discrete-time base-band received signal reads

$$y_k = e^{j\theta_k} + w_k\,, \tag{6.3}$$

where w_k are i.i.d. complex Gaussian random variable having zero mean and variance σ^2. The discrete phase noise process evolves according to the model in (2.9). The Kalman filter is adopted to improve the carrier tracking performance, exceeding the traditional PLL [28]. The derivation of the algorithm within the factor graph and SPA framework has been presented in Section 4.1 for the sole phase tracking. In this scenario, it is also necessary to extend it to the frequency and frequency drift estimation. Only the final algorithm equations for the forward recursion are provided to the reader to avoid redundancy in the derivation.

The algorithm derivation is based on the channel model presented in Section 2.2.2. Being the state vector containing the phase, frequency, and frequency drift to be estimated

$$\mathbf{\Phi}_k = \begin{bmatrix} \theta_k & f_k & \dot{f}_k \end{bmatrix}^\top ,$$

the forward message, entering the k-th variable node, is defined as a multivariate Gaussian distribution of the variable $\mathbf{\Phi}_k$ as

$$\alpha_k(\mathbf{\Phi}_k) \triangleq \mathcal{N}\left(\mathbf{\Phi}_k; \boldsymbol{\mu}_k, \underline{\mathbf{\Sigma}}_k\right) . \tag{6.4}$$

First, the transition matrix is defined as

$$\underline{\mathbf{H}} = \begin{bmatrix} 1 & 2\pi T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} , \tag{6.5}$$

with $\underline{\mathbf{B}}$ a 3×3 diagonal matrix having σ_Δ^2, σ_f^2, and $\sigma_{\dot{f}}^2$ in the main diagonal and zeroes elsewhere (see Section 2.2.2). The recursive computation of the mean vector and inverse covariance matrix follows the equations

$$\boldsymbol{\mu}_k = \underline{\mathbf{H}} \hat{\boldsymbol{\mu}}_k , \tag{6.6}$$

$$\underline{\mathbf{\Sigma}}_k^{-1} = \underline{\mathbf{H}}^{-\top} \left(\hat{\underline{\mathbf{\Sigma}}}_k + \left(\underline{\mathbf{H}}^\top \underline{\mathbf{B}}^\dagger \underline{\mathbf{H}} \right)^\dagger \right)^{-1} \underline{\mathbf{H}}^{-1} , \tag{6.7}$$

where

$$\hat{\underline{\mathbf{\Sigma}}}_k = \left(\underline{\mathbf{\Sigma}}_{k-1} + \sigma^2 \mathbf{e}_1 \mathbf{e}_1^\top \right)^{-1} , \tag{6.8}$$

and

$$\hat{\boldsymbol{\mu}}_k = \boldsymbol{\mu}_{k-1} + \hat{\underline{\mathbf{\Sigma}}}_k (1/\sigma^2) \mathbf{e}_1 \mathbf{e}_1^\top (\angle y_k \mathbf{e}_1 - \boldsymbol{\mu}_{k-1}) . \tag{6.9}$$

In (6.9), the well-known formula for the mean value obtained from the product of two multivariate Gaussian distributions is rearranged to account for the fact that the mean values represent phases. As a result, any differences or sums involving these phases must be wrapped within the interval $[0, 2\pi)$. The standard basis vector $\mathbf{e}_1 \in \mathcal{R}^3$ is necessary to combine multivariate Gaussian distributions and the channel probability law. Originally, in the Kalman filter in Section 4.1, the

channel probability is treated as a univariate Gaussian. Here, it is considered a degenerate multivariate Gaussian pdf. Additionally, the use of the Moore-Penrose inverse, identified by † in (6.7), is justified by the possibility of setting one or two elements of the diagonal of $\underline{\mathbf{B}}$ to 0. If the reader is unfamiliar with operations regarding Gaussian distribution, additional details can be found in Appendices A and B. Finally, the MSE estimator is found as

$$\hat{\theta}_k = \mathbf{e}_1^\top \boldsymbol{\mu}_{k+1} \,, \tag{6.10}$$

where in (6.10), the vector $\mathbf{e}_1$ is used to isolate the element of the estimation vector $\boldsymbol{\mu}_{k+1}$ relative to the variable θ_k.

6.2.1 Carrier Tracking: Results and Observations

This section presents the results of the Kalman-based carrier algorithm compared to the classical PLL. Our focus is on evaluating performance using phase jitter as a metric. First, the performance of the Kalman-based algorithm, identified as KF in the results, is compared with that of a PLL in a scenario where the AWGN channel is affected by phase noise.

Figure 6.1 illustrates the difference between an adaptive bandwidth algorithm (KF) and a fixed bandwidth algorithm (PLL) for a bit rate (R_b) of 7.8125 bps. Here, the considered bit rate directly translates into a specific value of σ_Δ as outlined in [4] and shown in Table 6.1. The figure shows the KF carrier jitter that

Bit Rate R_b [bps]	σ_Δ [deg]
7.8125	0.8938
3.9062	1.2641
1.9531	1.7877
0.9765	2.5282
0.4882	3.5754
0.2441	5.0564
0.1220	7.1509

Table 6.1: Phase noise standard deviation of the discrete Wiener model at different bit rates.

linearly decreases with $C/N_0/B_L$ - the carrier-to-noise ratio over the loop bandwidth B_L. On the other hand, due to the fixed loop bandwidth (6 Hz in this case), the PLL exhibits a floor in its carrier jitter at -23 dB. This result is in line with the discussion already addressed in Chapter 4.

Table 6.2 compares the residual carrier jitter of the KF and the PLL at different C/N_0 values, each corresponding to different bit rates. The results demonstrate a

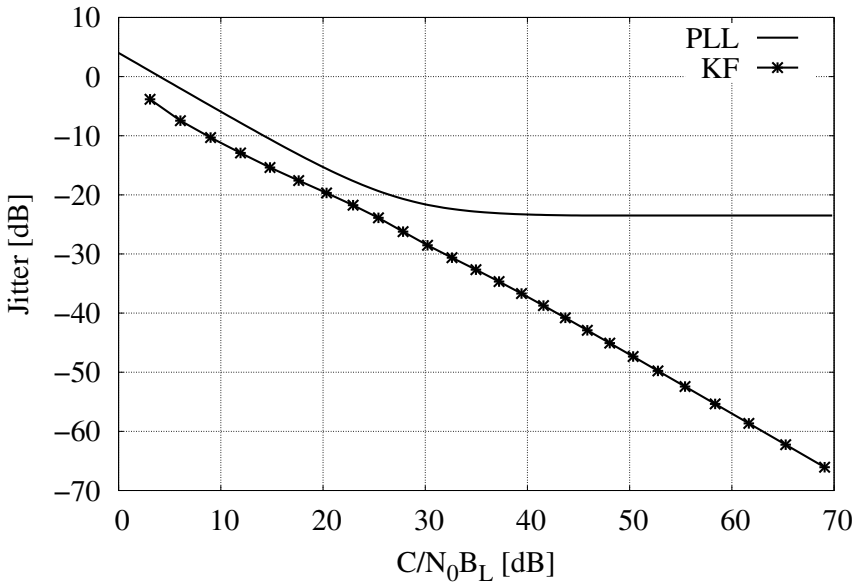


Figure 6.1: Carrier jitter for a first-order PLL with a loop bandwidth of 6 Hz compared to a first-order KF (adaptive in bandwidth) when R_b =7.8125 bps.

Bit rate [bps]	C/N_0[dBHz]	KF Jitter [dB]	PLL Jitter [dB]
7.8125	21.3	-23.3	-13
3.9062	18.3	-21.8	-10
1.9531	15.3	-20.2	-7.2
0.9765	12.3	-18.6	-4.3
0.4882	9.3	-17.1	-1.3
0.2441	6.3	-15.4	1.7
0.122	3.3	-13.5	4.7

Table 6.2: Residual carrier jitter for different bit rates: KF vs PLL.

consistent advantage of the Kalman over the PLL across all C/N_0. For instance, at the highest bit rate of 7.8125 bps and C/N_0 of 21.3 dBHz, the KF achieves a carrier jitter of -23.3 dB while the PLL exhibits a significantly higher jitter of -13 dB. This pattern persists across lower bit rates, with the KF maintaining a lower jitter value in all cases. The PLL's performance degrades more quickly due to its fixed loop bandwidth, which was specifically tuned for the highest bit rate (7.8125 bps).

A carrier frequency sweep characterized by an 8 Hz frequency shift and a 30 Hz/s frequency drift is considered to further analyze the Kalman filter's performance in tracking phase and frequency variations. This scenario is simulated according to

the model in (2.10), with phase noise added as specified in Table 6.1. At the receiver side, a sampling rate $R_s = 20$ sps is used to satisfy the Nyquist criterion. The results presented in Table 6.3 show the third-order KF's carrier jitter for different C/N_0. Comparing Tables 6.2 and 6.3, the KF's performance slightly deteriorates with an

Bit rate [bps]	C/N_0[dBHz]	KF Carrier Jitter [dB]
7.8125	21.3	-22.6
3.9062	18.3	-21.1
1.9531	15.3	-19.4
0.9765	12.3	-17.6
0.4882	9.3	-15.5
0.2441	6.3	-13.1
0.122	3.3	-10.3

Table 6.3: Phase jitter corresponding to a frequency shift of 8 Hz and a frequency drift of 30 Hz/s, and a phase noise according to Table 6.1, with $R_s = 20$ sps using a third-order KF.

average degradation of about 1 dB at bit rates down to $R_b = 0.4882$ bps. However, as the bit rate decreases further, the loss in performance becomes more pronounced, indicating the increased challenge of tracking phase variations at lower bit rates under significant frequency drifts. Finally, it is important to emphasize that adopting a third-order KF to track Doppler dynamics, in addition to phase noise affecting the TC signal, requires careful tuning of the three algorithmic parameters: σ_Δ, σ_f, and $\sigma_{\dot{f}}$ (see Section 6.2). This tuning process is not straightforward and can be a time-consuming process. Given these challenges, future work should investigate the development of adaptive algorithms, such as the adaptive Kalman filter, which can autonomously adjust these parameters based on the specific characteristics of the scenario [87], [88].

6.3 Sub-Carrier Tracking

Assuming perfect carrier recovery and ideal timing synchronization, the transmission of a sequence of linearly modulated complex binary symbols $\{c_k\}$ is considered, each transmitted at time kT. The symbols, shaped with a unitary energy squared pulse, go through an AWGN channel affected by Doppler dynamics. Assuming a phase noise process "slow" enough w.r.t the symbol time, the discrete-time sub-carrier received signal coincides with (3.1). The phase process evolves according

to a modified version of (2.9). In particular,

$$\theta_k = \theta_{k-1} + 2\pi f_{k-1} T \tag{6.11a}$$
$$f_k = f_{k-1} + \varepsilon_k \,, \tag{6.11b}$$

with ε_k a real i.i.d. Gaussian random variable with variance σ_F^2. For convenience, $\Psi_k = 2\pi f_k T$, and $\sigma_\Psi^2 = \sigma_F^2 4\pi^2 T^2$. The proposed model assumes that, given perfect carrier recovery, the phase noise on the sub-carrier is negligible. Therefore, (6.11a) is considered deterministic prior knowledge of the frequency. Our goal is to estimate the sequence of transmitted symbols $\{c_k\}$ adopting the SPA to derive a MAP detection strategy detection algorithm (see Section 2.6). This can be achieved from the marginalization of $p(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\Psi}|\boldsymbol{y})$ w.r.t $\boldsymbol{\theta}$, $\boldsymbol{\Psi}$, and all symbols c_i with $i \neq k$. In formulas,

$$\begin{aligned} p(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\Psi}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\Psi}) P(\boldsymbol{c}) p(\boldsymbol{\theta}|\boldsymbol{\Psi}) p(\boldsymbol{\Psi}) \\ &\propto \prod_{k=0}^{K-1} p(y_k|c_k, \theta_k) P(c_k) p(\theta_0) \prod_{k=1}^{K-1} p(\theta_k|\theta_{k-1}, \Psi_{k-1}) p(\Psi_k|\Psi_{k-1}) \,. \end{aligned} \tag{6.12}$$

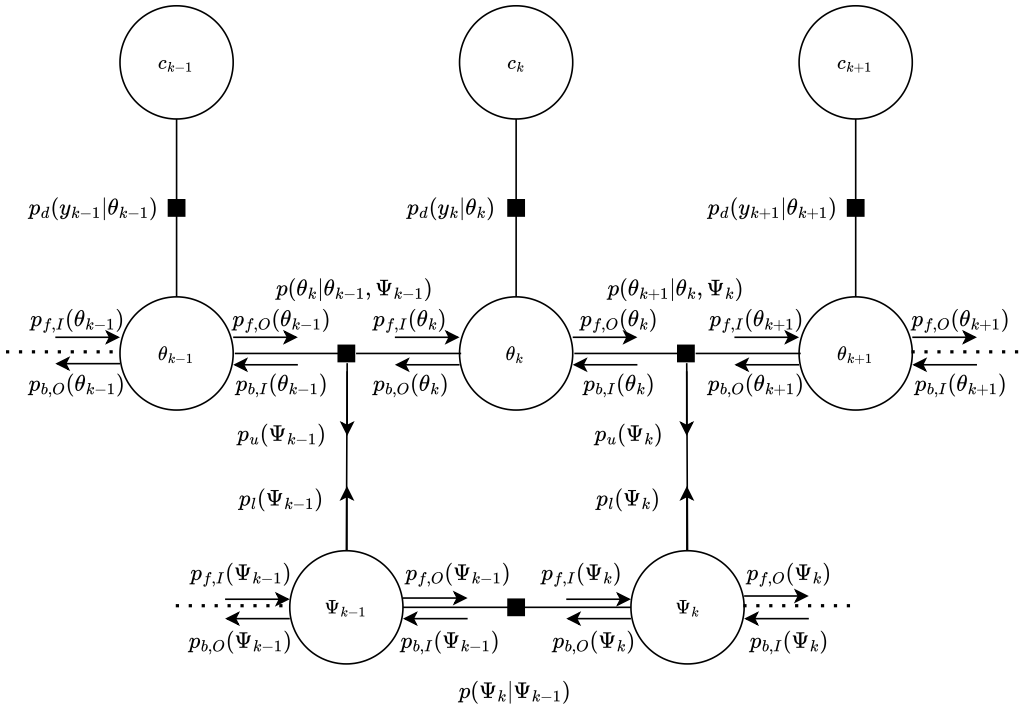


Figure 6.2: Factor graph representing the factorization in (6.12).

The factorization of the joint probability $p(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\Psi}|\boldsymbol{y})$ in (6.12) is visualized in the factor graph in Figure 6.2. In the figure, the transition probability distribution $p(\Psi_k|\Psi_{k-1})$ is defined as

$$p(\Psi_k|\Psi_{k-1}) \triangleq \mathcal{N}\left(\Psi_k - \Psi_{k-1}; 0, \sigma_\Psi^2\right) ,$$

where σ_Ψ denotes the standard deviation regulating the variability transitioning from Ψ_{k-1} to Ψ_k. As a consequence, the probability $p(\theta_k|\theta_{k-1}, \Psi_{k-1})$ is a Dirac delta

$$p(\theta_k|\theta_{k-1}, \Psi_{k-1}) = \delta(\theta_k - \theta_{k-1} - \Psi_{k-1}) ,$$

different from 0 only when $\theta_k = \theta_{k-1} + \Psi_{k-1}$. The posterior probability $p_d(y_k|\theta_k)$ is defined as in (3.15). Figure 6.2 represents the forward and backward message flows input and output to the two sets of variable $\{\theta_k\}_{k=0}^{K-1}$ and $\{\Psi_k\}_{k=0}^{K-1}$, and represent each recursion in the factor graph as a sequence of Tikhonov pdfs. The input and output forward and backward recursions for the variable θ_k and Ψ_k are defined as

$$p_{x,y}(\theta_k) \triangleq \exp\left\{\Re[z_{x,y,\theta_k} e^{-j\theta_k}]\right\} ,$$

and

$$p_{x,y}(\Psi_k) \triangleq \exp\left\{\Re[z_{x,y,\Psi_k} e^{-j\Psi_k}]\right\} ,$$

respectively. The first subscript "x" identifies either the forward ("f") or backward ("b") message direction, and the second subscript "y" specifies whether it is an input ("I") or output ("O") message.

The message

$$p_u(\Psi_k) \triangleq \exp\left\{\Re[z_{u,k} e^{-j\Psi_k}]\right\}$$

represents the a priori probability associated with the variable Ψ_k, with the subscript "u" indicating the use of information messages from the upper portion of the graph. On the other hand,

$$p_l(\Psi_k) \triangleq \exp\left\{\Re[z_{l,k} e^{-j\Psi_k}]\right\}$$

is the a posteriori probability related to variable Ψ_k, and the subscript "l" indicates the use of information from the lower section of the graph.

According to the SPA, the message $p_{f,I}(\theta_k)$ can be computed as

$$p_{f,I}(\theta_k) \propto \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} p_{f,O}(\theta_{k-1}) p_l(\Psi_{k-1}) p(\theta_k|\theta_{k-1}, \Psi_{k-1}) d\theta_{k-1} d\Psi_{k-1} \tag{6.13}$$

$$\propto \int_{-\pi}^{\pi} p_{f,O}(\theta_{k-1}) p_l(\theta_k - \theta_{k-1}) d\theta_{k-1} \tag{6.14}$$

$$\propto \exp\left\{\Re[z_{f,I,\theta_k} e^{-j\theta_k}]\right\} , \tag{6.15}$$

where in (6.14), the sampling property of the Dirac delta function is applied. As a result, $p_l(\Psi_{k-1})$ transforms into $p_l(\theta_k - \theta_{k-1})$, leading to a convolution between

$p_{f,O}(\theta_{k-1})$ and $p_l(\theta_k - \theta_{k-1})$. This convolution is solved in the Gaussian domain, and approximated by a Tikhonov pdf in (6.15) [5]. Therefore, the complex parameters associated with $p_{f,I}(\theta_k)$ can be recursively computed as

$$z_{f,I,\theta_k} = z_{f,O,\theta_{k-1}} z_{l,k} \frac{1}{|z_{f,O,\theta_{k-1}}| + |z_{l,k}|} . \tag{6.16}$$

The forward output message, defined as

$$p_{f,O}(\theta_k) \propto p_{f,I}(\theta_k) p_d(y_k|\theta_k) , \tag{6.17}$$

is characterized by a complex parameter

$$z_{f,O,\theta_k} = z_{f,I,\theta_k} + z_{d,k} ,$$

where $z_{d,k} = |y_k \alpha_k^* / (2\sigma^2 + \beta_k - |\alpha_k|^2)| e^{j\angle y_k \alpha_k^*}$.

Similarly, the sequence of backward complex coefficients of the input and output messages can be recursively computed as

$$z_{b,I,\theta_k} = z_{b,O,\theta_{k+1}} z_{l,k+1}^* \frac{1}{|z_{b,O,\theta_{k+1}}| + |z_{l,k+1}|} , \tag{6.18}$$

and

$$z_{b,O,\theta_k} = z_{b,I,\theta_k} + z_{d,k} ,$$

respectively.

Having derived the forward and backward recursions on the phase, the a priori on the frequency can be computed as

$$\begin{aligned} p_u(\Psi_k) &\propto \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} p_{f,O}(\theta_k) p_{b,O}(\theta_{k+1}) p(\theta_{k+1}|\theta_k, \Psi_k) d\theta_k d\theta_{k+1} \\ &\propto \int_{-\pi}^{\pi} p_{f,O}(\theta_{k+1} - \Psi_k) p_{b,O}(\theta_{k+1}) d\theta_{k+1} \\ &\propto \exp\left\{ \Re[z_{u,k} e^{-j\Psi_k}] \right\} , \end{aligned}$$

with

$$z_{u,k} = z_{f,O,\theta_k}^* z_{b,O,\theta_{k+1}} \frac{1}{|z_{f,O,\theta_k}| + |z_{b,O,\theta_{k+1}}|} . \tag{6.19}$$

Focusing on the lower part of the graph, the input and output forward and backward recursions on the frequency variable can be obtained by computing the following complex coefficients

$$z_{f,I,\Psi_k} = z_{f,O,\Psi_{k-1}} \frac{1}{|z_{f,O,\Psi_{k-1}}| \sigma_\Psi^2 + 1} ,$$

$$z_{f,O,\Psi_k} = z_{f,I,\Psi_k} + z_{u,k} ,$$

$$z_{b,I,\Psi_k} = z_{b,O,\Psi_{k+1}} \frac{1}{|z_{b,O,\Psi_{k+1}}|\sigma_\Psi^2 + 1}\,,$$

and

$$z_{b,O,\Psi_k} = z_{b,I,\Psi_k} + z_{u,k}\,.$$

Finally, it is possible to derive

$$p_l(\Psi_k) \propto p_{f,I}(\Psi_k) p_{b,I}(\Psi_k)\,,$$

with

$$z_{l,k} = z_{f,I,\Psi_k} + z_{b,I,\Psi_k}\,.$$

Additionally, in the case of bursts of pilot symbols, foreseen by the system scenario [67], the modification suggested in [5] and reported in Section 3.1.2 is adopted.

6.3.1 Tikhonov-based Sub-Carrier Algorithm: Scheduling

Since the factor graph in Figure 6.2 has cycles, the choice of the scheduling algorithm, defining the order in which the messages are exchanged through the factor graph, is important. Consider two scheduling algorithms: flooding and serial, described by the following pseudo-code. The constituent algorithms are defined as follows

Algorithm 1 Forward and backward output from θ_k

for k $= 0$ to $K-1$ **do**
$z_{f,O,\theta_k} = z_{f,I,\theta_k} + z_{d,k}$
$z_{b,O,\theta_{K-1-k}} = z_{b,I,\theta_{K-1-k}} + z_{d,K-1-k}$
end for

Algorithm 2 *A priori* on Ψ_k

for k $= 0$ to $K-2$ **do**
$z_{u,k} = z^*_{f,O,\theta_k} z_{b,O,\theta_{k+1}} \frac{1}{|z_{f,O,\theta_k}|+|z_{b,O,\theta_{k+1}}|}$
end for

Algorithm 3 Forward and backward on Ψ_k

for k $= 0$ to $K-2$ **do**
$z_{f,O,\Psi_k} = z_{f,I,\Psi_k} + z_{u,k}$
$z_{f,I,\Psi_{k+1}} = z_{f,O,\Psi_k} \frac{1}{|z_{f,O,\Psi_k}|\sigma_\Psi^2+1}$
$z_{b,O,\Psi_{K-1-k}} = z_{b,I,\Psi_{K-1-k}} + z_{u,K-1-k}$
$z_{b,I,\Psi_{K-1-k-1}} = z_{b,O,\Psi_{K-1-k}} \frac{1}{|z_{b,O,\Psi_{K-1-k}}|\sigma_\Psi^2+1}$
end for

Algorithm 4 *A posteriori* on Ψ_k

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

$z_{l,k} = z_{f,I,\Psi_k} + z_{b,I,\Psi_k}$

end for

Algorithm 5 Forward and backward input to θ_k

for k = 1 to $K-1$ **do**

$z_{f,I,\theta_k} = z_{f,O,\theta_{k-1}} z_{l,k-1} \frac{1}{|z_{f,O,\theta_{k-1}}|+|z_{l,k-1}|}$

$z_{b,I,\theta_{K-1-k}} = z_{b,I,\theta_{K-k}} z^*_{l,K-k} \frac{1}{|z_{b,I,\theta_{K-k}}|+z_{l,K-k}}$

end for

Algorithm 6 Forward and backward output from θ_k

for k = 0 to $K-1$ **do**

$z_{f,O,\theta_k} = z_{f,I,\theta_k} + z_{d,k}$

$z_{f,I,\theta_{k+1}} = z_{f,O,\theta_k} z_{l,k} \frac{1}{|z_{f,O,\theta_k}|+|z_{l,k}|}$

$z_{b,O,\theta_{K-1-k}} = z_{b,I,\theta_{K-1-k}} + z_{d,K-1-k}$

$z_{b,I,\theta_{K-1-k-1}} = z_{b,I,\theta_{K-1-k}} z^*_{l,K-1-k} \frac{1}{|z_{b,I,\theta_{K-1-k}}|+z_{l,K-1-k}}$

end for

The flooding schedule consists of Algorithms 1 through 5 in sequential order. In contrast, the serial schedule comprises Algorithms 1 through 4, followed by Algorithm 6.

6.3.2 Sub-Carrier Tracking: Results and Observations

This section discusses the performance of the sub-carrier tracking algorithm. The algorithm is initially tested using the two scheduling methods (flooding and serial, as described in Section 6.3.1) in the presence of AWGN, both in the uncoded and coded scenarios. For the latter, the LDPC (64,128) coding [67] is adopted and includes interleaving and de-interleaving to address possible error bursts. The simulation results were obtained by counting a minimum of 100 bit errors per simulation point for the BER, and a minimum of 100 frame errors per simulation point for the FER. The results for the uncoded case are presented in Figure 6.3. In the figure, both the serial and flooding scheduling algorithms perfectly match the theoretical BER. In the uncoded scenario, iterations are typically not required due to the absence of a decoder. However, in the case of flooding scheduling alone, the algorithm is left to iterate with itself due to the latency in updating messages throughout the entire factor graph. This delay is not a characteristic of the serial scheduling, therefore only one iteration of the serial algorithm is performed.

Then, the sub-carrier algorithm is tested under AWGN conditions, incorporating

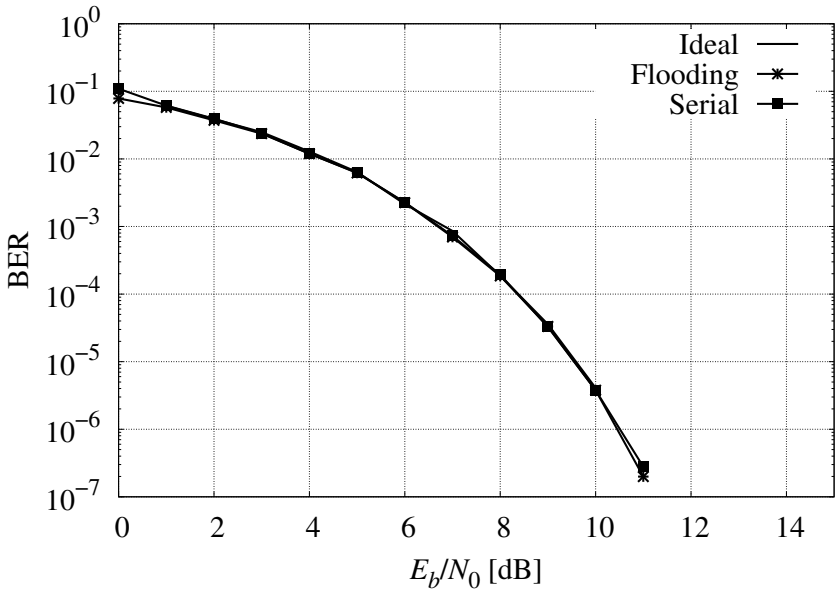


Figure 6.3: BER in the uncoded scenario adopting flooding (100 iterations) and serial scheduling using 10 pilots every 100.

the LDPC (64,128). The findings are reported in Figure 6.4. Even though both FER curves demonstrate results consistent with the theoretical FER, it is important to note that the serial algorithm does not undergo iterations with the decoder in this case. This is because an increase in the number of iterations appears to impact performance negatively. Therefore, for the context of Figure 6.4, the serial algorithm only runs once while the decoder iterates autonomously.

Due to the anomalous behavior observed in the serial scheduler, only the flooding schedule is adopted to evaluate the performance under the channel described in (6.11) at various bit rates ($R_b = 1/T$) with $\sigma_F = 0.06$ mHz/s. The relationship between bit rate and frequency variations is critical; as the bit rate decreases, the level of frequency variations tends to increase, thus impacting system performance. Given this context, the results presented in Figure 6.5 illustrate that the flooding scheduler maintains resilience at bit rates as low as 0.48 bps. However, at lower bit rates, the performance is drastically impacted, highlighting the limitations of the flooding scheduler in maintaining effective communication under such conditions.

6.3.3 Stability Analysis

This section analyzes the algorithm's behavior at each iteration and compares the two schedulers. By considering a constant frequency shift, the performance is

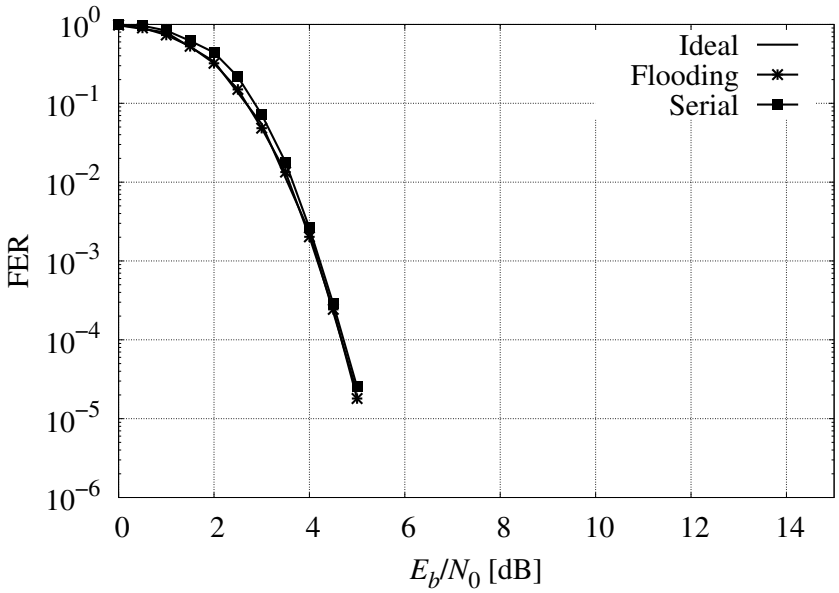


Figure 6.4: FER using LDPC (64,128) adopting flooding (100 total iterations) and serial scheduling (1 instance and 100 decoder iterations) using 10 pilots every 100.

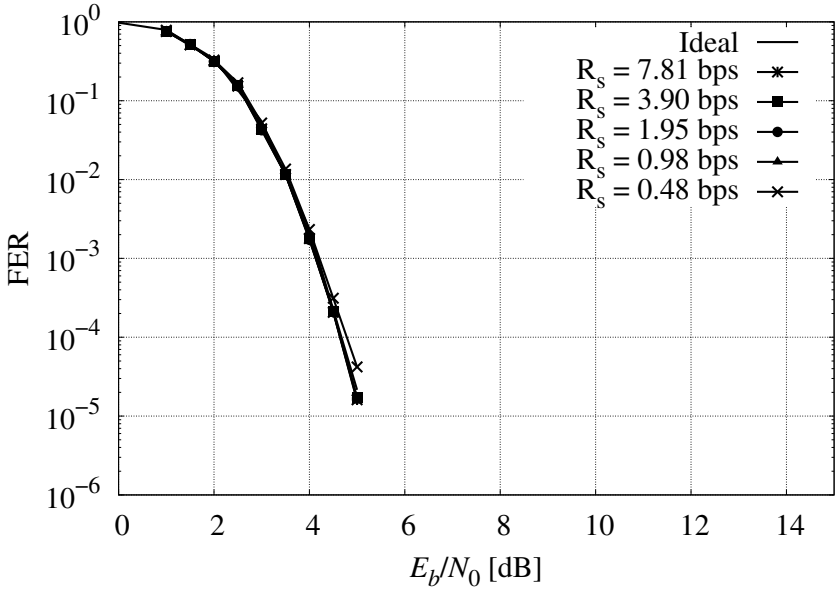
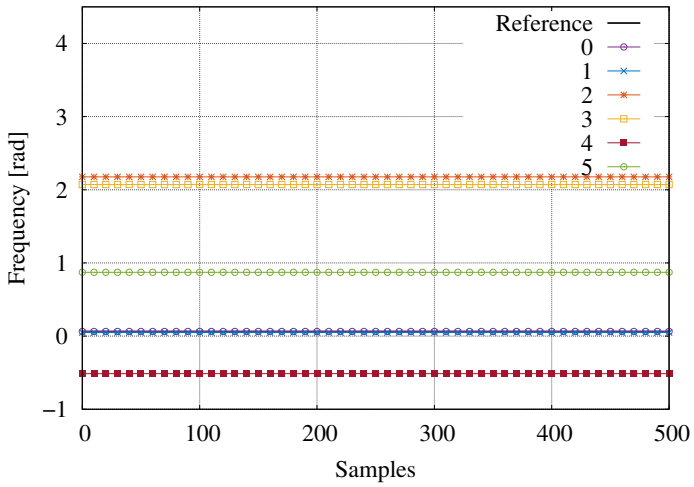


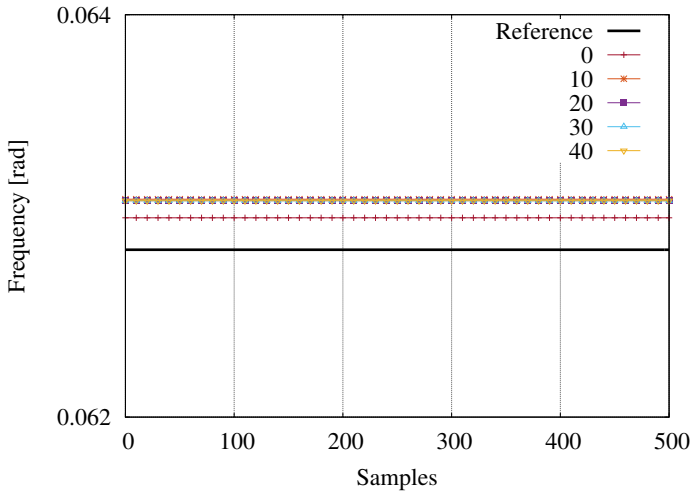
Figure 6.5: FER using LDPC (64, 128) with flooding scheduling (100 total iterations) and $\sigma_F = 0.06$ mHz/s.

analyzed, and revealed that accurate frequency estimates are obtained after the first

iteration using serial scheduling. However, these estimates gradually deteriorated with subsequent iterations. Figure 6.6 shows the frequency estimation performance of the two scheduling algorithms when dealing with a constant frequency offset of 6.3×10^{-2} rad. For a closer look at the erratic behavior of the serial scheduler, Figure 6.6a displays the estimated frequency after the first six iterations. The figure indicates that the estimate aligns with the real frequency at iteration "0", but then starts deviating and jumping to different values. In comparison, Figure 6.6b demonstrates the behavior of the flooding scheduler, which remains relatively stable.



(a) Serial



(b) Flooding

Figure 6.6: Frequency estimation of the serial and flooding schedulers for a constant frequency shift of 6.3×10^{-2} rad.

To describe this behavior mathematically, let us consider a scenario with the following assumptions:

- Constant frequency offset.
- After the first iteration, the frequency estimates remain constant across all time steps k.
- The initial frequency estimates in this first iteration contain an error, denoted as ϵ, which can be made arbitrarily small.

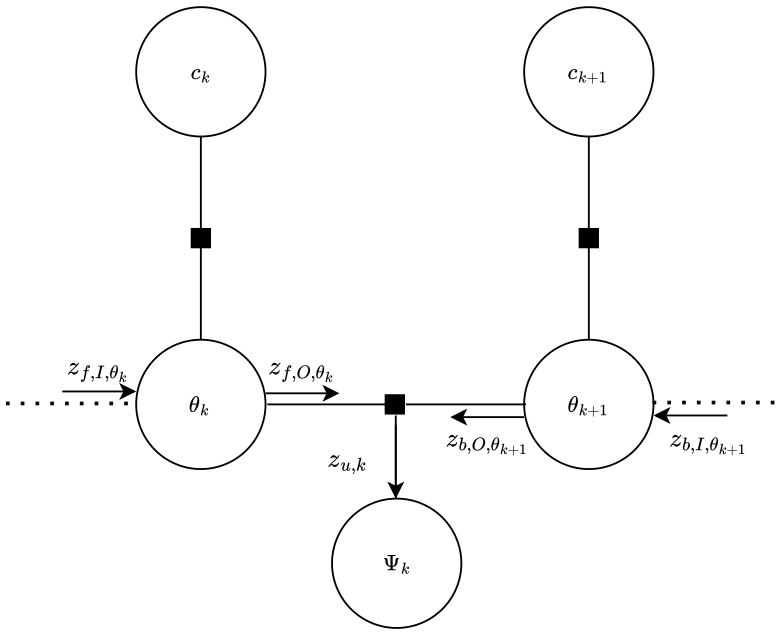


Figure 6.7: Zoom of a section of the factor graph in Figrue 6.2.

Focusing on the segment of the factor graph shown in Figure 6.7, and assuming for simplicity that the symbols c_k and c_{k+1} in the figure are unknown (i.e. not pilot symbols), the phase (arg) of $z_{u,k}$ is computed at the second iteration, following the expression in (6.19), as

$$\begin{aligned}
\arg(z_{u,k}) &= \arg(z_{b,O,\theta_{k+1}}) - \arg(z_{f,O,\theta_k}) && (6.20)\\
&= \arg(z_{b,I,\theta_{k+1}}) - \arg(z_{f,I,\theta_k}) && (6.21)\\
&= \arg(z_{b,O,\theta_{k+2}}) - \arg(z_{l,k+1} + \epsilon) - \arg(z_{f,O,\theta_{k-1}}) - \arg(z_{l,k-1} + \epsilon) && (6.22)\\
&= \hat{\theta}_{k+2} - \hat{\Psi}_{k+1} - \epsilon - \hat{\theta}_{k-1} - \hat{\Psi}_{k-1} - \epsilon && (6.23)\\
&= \hat{\theta}_{k+1} - \hat{\theta}_k - 2\epsilon\,, && (6.24)
\end{aligned}$$

where to pass from (6.20) to (6.21), it is assumed that $z_{d,k}$ and $z_{d,k+1}$ are equal to 0, given that the symbols are unknown. To go from (6.21) to (6.22), the definitions provided in (6.16) and (6.18) are used. Therefore, (6.24) proves that the mean value of the a priori probabilities on Ψ_k, for all k, shows an increasing error with each iteration.

In conclusion, two main issues associated with each scheduling approach are identified. In serial mode, the primary concern is the algorithm's stability which can significantly impact its performance. On the other hand, flooding scheduling introduces a different challenge – an inherent delay in the algorithm, requiring numerous iterations before updating all the messages in the factor graph. This delay can be problematic, particularly when the algorithm aims to track highly dynamic variations. However, combining the flooding scheduler with the modifications proposed for bursts of pilot data (see Section 3.1.2) results in a direct-decision algorithm that maintains stability and exhibits the best performance.

Chapter 7

Phase Noise and Frequency Offset Tracking using Gaussian Mixtures

In the literature, there are only a few algorithms that address joint detection and synchronization under phase and frequency uncertainties. Section 3.2.1 discussed the work in [22], which effectively performs joint detection and decoding as part of an iterative decoding scheme. The authors also demonstrated that the algorithm is resilient to slow time-varying frequency offsets. This is particularly relevant in scenarios with high baud rates, such as Earth observation satellite missions in LEO [89]. However, in scenarios with lower baud rates and harsher Doppler rates, such as deep space missions, the frequency offset variations are significantly higher, compromising the algorithm's effectiveness [3].

The authors in [3] propose Kalman-based synchronization techniques to replace the standard PLL for carrier synchronization in challenging environments characterized by low SNRs and data rates. Kalman filtering [19] is an optimal Bayesian estimator widely used to predict the state of a system using its noisy observations. Expanding on the work in [20], the authors in [21] utilize a bank of extended Kalman filters to develop a nonlinear Bayesian estimator for phase and frequency tracking. However, this work is limited to the case of analog demodulation of phase and frequency-modulated signals.

This chapter derives a joint synchronization and detection algorithm using two-dimensional Gaussian mixtures in the factor graph and SPA framework. Approximating the messages exchanged in the factor graph with Gaussian mixtures makes tracking multiple hypotheses of the phase noise process possible. The choice of selecting Gaussian distributions rather than Tikhonov is related to the specific problem, which involves tracking both the phase and the frequency.

7.1 Detector Equivalent Channel Model

Consider the transmission of a sequence of linearly modulated symbols $\{c_k\}$ belonging to an M-ary complex-valued constellation set $\mathcal{S}$, and shaped with a unitary energy squared pulse satisfying the Nyquist criterion. The sequence $\{c_k\}$, including pilot symbols, is transmitted over an AWGN channel affected by time-varying phase and frequency offset. Assuming the phase process is slow enough compared to the symbol time T, the discrete-time complex base-band received signal is described by (3.1). For the derivation of the detection algorithm, the discrete-time phase noise process is assumed to evolve as

$$\theta_k = \theta_{k-1} + \nu_{k-1} + \delta_k \tag{7.1a}$$
$$\nu_k = \nu_{k-1} + \epsilon_k \tag{7.1b}$$

with δ_k and ϵ_k i.i.d real Gaussian random variables with zero mean and variance σ_Δ^2 and σ_F^2, respectively. The set of equations in (7.1) represent a well-established model commonly employed in the derivation of PLL and PLL-like algorithms [27]–[29] (see Section 2.2.2). For convenience, $\nu_k \triangleq 2\pi f_k T$ is defined, with f_k the instantaneous frequency offset at time kT that is uniformly distributed within the range $\left[-\frac{1}{2T}, \frac{1}{2T}\right]$. Here, σ_Δ^2 and σ_F^2 correspond to the phase noise and frequency offset variances over the reference symbol time T, respectively. Since σ_F characterizes the variability in the frequency offset, it can also be interpreted as a measure of frequency drift. The evolution of the phase and frequency processes is assumed to be independent, as they originate from different sources. The phase noise arises from imperfections in the local oscillator, while the frequency drift is primarily due to the Doppler effect.

Instead of dealing with individual equations, (7.1a) and (7.1b) are combined into a unified state-space model. The state vector $\mathbf{\Phi}_k = \begin{bmatrix}\theta_k & \nu_k\end{bmatrix}^\top$ is defined as the tracking vector, and the state-space model as

$$\mathbf{\Phi}_k = \underline{\mathbf{H}}\mathbf{\Phi}_{k-1} + \boldsymbol{\eta}_k\,, \tag{7.2}$$

where $\underline{\mathbf{H}} = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$ is the state transition matrix, and $\boldsymbol{\eta}_k = \begin{bmatrix}\delta_k & \epsilon_k\end{bmatrix}^\top$ is the process noise vector having zero mean and covariance matrix $\underline{\mathbf{B}} = \begin{bmatrix}\sigma_\Delta^2 & 0\\ 0 & \sigma_F^2\end{bmatrix}$. The proposed equivalent channel model has been adopted to design an algorithm that can exploit the Markovian properties of the model. Although the algorithm is tailored to this model, it remains adaptable to different channel models.

7.2 Derivation of Gaussian Mixture Algorithm

To derive a joint detection and estimation algorithm according to the optimal MAP detection strategy, the a posteriori pmf of the k-th symbol given the observation sequence $\mathbf{y} = \{y_0, y_1, ..., y_{K-1}\}$, $p(c_k|\mathbf{y})$, can be obtained by marginalizing $p(\boldsymbol{c}, \boldsymbol{\Phi}|\boldsymbol{y})$ w.r.t the sequence of state vectors $\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_0, \boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_{K-1}\}$ and all symbols c_i with $i \neq k$. Assuming the transmitted symbols to be uniform i.i.d., the joint posterior pdf can be factorized as follows

$$\begin{aligned} p(\boldsymbol{c}, \boldsymbol{\Phi}|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{c}, \boldsymbol{\Phi}) P(\boldsymbol{c}) p(\boldsymbol{\Phi}) \\ &\propto \prod_{k=0}^{K-1} P(c_k) p(y_k|c_k, \theta_k) \prod_{k=1}^{K-1} p(\boldsymbol{\Phi}_k|\boldsymbol{\Phi}_{k-1}) \,, \end{aligned} \tag{7.3}$$

where the constant term $p(\boldsymbol{\Phi}_0)$ is neglected, and with $P(c_k)$ being the probability of the k-th information symbol. The channel law probability is given by

$$p(y_k|c_k, \theta_k) \propto e^{-\frac{|c_k|^2}{2\sigma^2}} \exp\left\{\Re\left[\frac{y_k c_k^*}{\sigma^2} e^{-j\theta_k}\right]\right\} \tag{7.4}$$

$$\approx e^{-\frac{|c_k|^2}{2\sigma^2}} \mathcal{N}\left(\theta_k; \angle\frac{y_k c_k^*}{\sigma^2}, \frac{\sigma^2}{|y_k c_k^*|}\right) , \tag{7.5}$$

where passing from (7.4) to (7.5), the Tikhonov distribution is approximated with a Gaussian pdf [56]. The factorization of the joint probability $p(\boldsymbol{c}, \boldsymbol{\Phi}_k|\boldsymbol{y})$ in (7.3) is represented by the cycle-free factor graph in Figure 7.1[1].

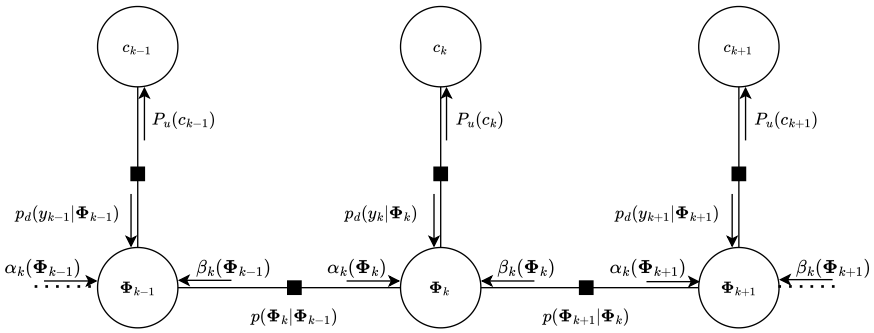


Figure 7.1: Factor graph representing the factorization in (7.3).

[1]The choice to cluster the phase and frequency variables into a single variable node allowed for the construction of a cycle-free factor graph, differently from the algorithm in Section 6.3.

In the figure, the message $p_d()$ is defined as

$$p_d(y_k|\mathbf{\Phi}_k) \triangleq \sum_{j=0}^{M-1} \lambda_k^j p(y_k|c_k = x_j, \theta_k) \tag{7.6}$$

$$= \sum_{j=0}^{M-1} \lambda_k^j \mathcal{N}\left(\mathbf{\Phi}_k; \boldsymbol{\gamma}_k^j, \underline{\mathbf{\Gamma}}_k^j\right) , \tag{7.7}$$

where in (7.7), it is considered as a degenerate bivariate Gaussian mixture of the variable $\mathbf{\Phi}_k = \begin{bmatrix}\theta_k & 0\end{bmatrix}^\top$ characterized by weight

$$\lambda_k^j = P(c_k = x_j) e^{-\frac{|x_j|^2}{2\sigma^2}} ,$$

mean

$$\boldsymbol{\gamma}_k^j = \angle u_k^j \mathbf{e}_1 ,$$

and inverse covariance matrix

$$\left(\underline{\mathbf{\Gamma}}_k^j\right)^{-1} = |u_k^j| \mathbf{e}_1 \mathbf{e}_1^\top ,$$

with $u_k^j = \frac{y_k x_j^*}{\sigma^2}$, and $x_j \in \mathcal{S}$. The probability of the state evolution, $p(\mathbf{\Phi}_k|\mathbf{\Phi}_{k-1})$ is, by definition, a bivariate Gaussian distribution

$$p(\mathbf{\Phi}_k|\mathbf{\Phi}_{k-1}) \triangleq \mathcal{N}\left(\mathbf{\Phi}_k - \underline{\mathbf{H}}\mathbf{\Phi}_{k-1}; \mathbf{0}, \underline{\mathbf{B}}\right) . \tag{7.8}$$

7.2.1 Forward and Backward Recursions

The marginalization of the joint posterior pdf $p(\mathbf{c}, \mathbf{\Phi}|\mathbf{y})$ can be achieved by applying the SPA to the factor graph in Figure 7.1, which results in the following forward and backward recursions

$$\alpha_k(\mathbf{\Phi}_k) \propto \int_{\mathbf{\Phi}_{k-1}} \alpha_{k-1}(\mathbf{\Phi}_{k-1}) p_d(y_{k-1}|\mathbf{\Phi}_{k-1}) p(\mathbf{\Phi}_k|\mathbf{\Phi}_{k-1})\, d\mathbf{\Phi}_{k-1} , \tag{7.9}$$

and

$$\beta_k(\mathbf{\Phi}_k) \propto \int_{\mathbf{\Phi}_{k+1}} \beta_{k+1}(\mathbf{\Phi}_{k+1}) p_d(y_{k+1}|\mathbf{\Phi}_{k+1}) p(\mathbf{\Phi}_{k+1}|\mathbf{\Phi}_k)\, d\mathbf{\Phi}_{k+1} , \tag{7.10}$$

respectively. However, solving (7.9) and (7.10) in closed-form is not feasible due to the integration and computation of continuous pdfs. The canonical distribution approach suggested in [17] is adopted, and the SPA messages are approximated as bivariate Gaussian mixtures of order N,

$$\alpha_k(\mathbf{\Phi}_k) \triangleq \sum_{i=0}^{N-1} w_k^{f,i} \mathcal{N}\left(\mathbf{\Phi}_k; \boldsymbol{\mu}_k^{f,i}, \underline{\mathbf{\Sigma}}_k^{f,i}\right) , \tag{7.11}$$

$$\beta_k(\mathbf{\Phi}_k) \triangleq \sum_{i=0}^{N-1} w_k^{b,i} \mathcal{N}\left(\mathbf{\Phi}_k; \boldsymbol{\mu}_k^{b,i}, \underline{\mathbf{\Sigma}}_k^{b,i}\right) . \tag{7.12}$$

In the following, the discussion focuses on the forward recursion. The extension to the backward recursion is straightforward and will not be discussed in detail.

To solve (7.9), the product given by

$$\alpha_{k-1}(\mathbf{\Phi}_{k-1}) p_d(y_{k-1}|\mathbf{\Phi}_{k-1}) \propto \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \hat{w}_{k-1}^{f,ij} \mathcal{N}\left(\mathbf{\Phi}_{k-1}; \boldsymbol{\rho}_{k-1}^{f,ij}, \underline{\boldsymbol{P}}_{k-1}^{f,ij}\right) , \tag{7.13}$$

is computed. It results in a bivariate Gaussian mixture characterized by the following parameters

$$\underline{\boldsymbol{P}}_{k-1}^{f,ij} = \left(\left(\underline{\mathbf{\Sigma}}_{k-1}^{f,i}\right)^{-1} + |u_{k-1}^j| \mathbf{e}_1 \mathbf{e}_1^\top\right)^{-1} ,$$

$$\boldsymbol{\rho}_{k-1}^{f,ij} = \boldsymbol{\mu}_{k-1}^{f,i} + \underline{\boldsymbol{P}}_{k-1}^{f,ij} |u_{k-1}^j| \mathbf{e}_1 \mathbf{e}_1^\top \left(\boldsymbol{\gamma}_{k-1}^j - \boldsymbol{\mu}_{k-1}^{f,i}\right) , \tag{7.14}$$

$$\hat{w}_{k-1}^{f,ij} = \left(\frac{\left|\underline{\mathbf{P}}_{k-1}^{f,ij}\right|}{\left|\underline{\mathbf{\Sigma}}_{k-1}^{f,i}\right|}\right)^{1/2} \lambda_{k-1}^j w_{k-1}^{f,i} e^{-\frac{1}{2}\zeta_{k-1}^{f,ij}} . \tag{7.15}$$

Here, the standard basis vector $\mathbf{e}_1 \in \mathbf{R}^2$. The mean vector in (7.14) has been rearranged from the common result in (A.8) of Appendix A. This adjustment is necessary to account for the representation of mean values as phases. It is important to ensure that any sum or difference between two phase values remains within the $(0, 2\pi]$ range. The scaling factor in (7.15) arises from the product of two bivariate Gaussian distributions, as detailed in (A.7) of Appendix A. Given that $p_d(y_{k-1}|\mathbf{\Phi}_{k-1})$ is a degenerate bivariate Gaussian, the term $\zeta_{k-1}^{f,ij}$ here is defined as

$$\zeta_{k-1}^{f,ij} = \left(\angle u_{k-1}^j - \mathbf{e}_1^\top \boldsymbol{\mu}_{k-1}^{f,i}\right)^2 \mathbf{e}_1^\top \left(\underline{\mathbf{\Gamma}}_{k-1}^j + \underline{\mathbf{\Sigma}}_{k-1}^{f,i}\right)^{-1} \mathbf{e}_1 . \tag{7.16}$$

This equation corresponds to considering those parameters in (A.6) that depend solely on the phase variable.

By plugging (7.13) into (7.9), and solving the convolution as detailed in Appendix B, the forward recursion reads

$$\alpha_k(\mathbf{\Phi}_k) \propto \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \hat{w}_k^{f,ij} \mathcal{N}\left(\mathbf{\Phi}_k; \hat{\boldsymbol{\mu}}_k^{f,ij}, \hat{\underline{\mathbf{\Sigma}}}_k^{f,ij}\right) , \tag{7.17}$$

where

$$\hat{\boldsymbol{\mu}}_k^{f,ij} = \underline{\mathbf{H}} \boldsymbol{\rho}_{k-1}^{f,ij} ,$$

$$\hat{\underline{\mathbf{\Sigma}}}_k^{f,ij} = \left(\underline{\mathbf{H}}^{-T} \left(\underline{\boldsymbol{P}}_{k-1}^{f,ij} + \tilde{\underline{\mathbf{B}}}^{-1}\right)^{-1} \underline{\mathbf{H}}^{-1}\right)^{-1} ,$$

with $\underline{\tilde{\mathbf{B}}}$ defined as in (B.2).

Equivalently, solving the integral in (7.10), the backward message $\beta_k(\mathbf{\Phi}_k)$ results in the following bivariate Gaussian mixture

$$\beta_k(\mathbf{\Phi}_k) \propto \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \hat{w}_k^{b,ij} \mathcal{N}\left(\mathbf{\Phi}_k; \hat{\boldsymbol{\mu}}_k^{b,ij}, \hat{\underline{\mathbf{\Sigma}}}_k^{b,ij}\right), \tag{7.18}$$

characterized by the following parameters

$$\hat{\boldsymbol{\mu}}_k^{b,ij} = \underline{\mathbf{H}}^{-1} \boldsymbol{\rho}_{k+1}^{b,ij},$$

$$\hat{\underline{\mathbf{\Sigma}}}_k^{b,ij} = \left(\underline{\mathbf{H}}^T \left(\underline{\boldsymbol{P}}_{k+1}^{b,ij} + \underline{\mathbf{B}}\right)^{-1} \underline{\mathbf{H}}\right)^{-1},$$

$$\hat{w}_k^{b,i,j} = \left(\frac{\left|\underline{\mathbf{P}}_{k+1}^{b,ij}\right|}{\left|\underline{\mathbf{\Sigma}}_{k+1}^{b,i}\right|}\right)^{1/2} \lambda_{k+1}^{j} w_{k+1}^{b,i} e^{-\frac{1}{2}\zeta_{k+1}^{b,ij}},$$

where

$$\underline{\boldsymbol{P}}_{k+1}^{b,ij} = \left(\left(\underline{\mathbf{\Sigma}}_{k+1}^{b,i}\right)^{-1} + |u_{k+1}^{j}| \mathbf{e}_1 \mathbf{e}_1^{\top}\right)^{-1},$$

$$\boldsymbol{\rho}_{k+1}^{b,ij} = \boldsymbol{\mu}_{k+1}^{b,i} + \underline{\boldsymbol{P}}_{k+1}^{b,ij} |u_{k+1}^{j}| \mathbf{e}_1 \mathbf{e}_1^{\top} \left(\boldsymbol{\gamma}_{k+1}^{j} - \boldsymbol{\mu}_{k+1}^{b,i}\right),$$

$$\zeta_k^{b,ij} = \left(\angle u_{k+1}^{j} - \mathbf{e}_1^{\top} \boldsymbol{\mu}_k^{b,i}\right)^2 \mathbf{e}_1^{\top} \left(\underline{\mathbf{\Gamma}}_k^{j} + \underline{\mathbf{\Sigma}}_k^{b,i}\right)^{-1} \mathbf{e}_1 .$$

The forward and backward messages in (7.17) and (7.18) are mixtures with order M times larger than the one at the previous time step. If no action is taken, at each time step the order of the mixtures will increase by a factor M each time. The next section describes a mixture reduction algorithm to reduce the mixture order back to the original (N).

7.2.2 Mixture Reduction Algorithm

The forward and backward messages in (7.17) and (7.18) are bivariate Gaussian mixtures of order MN. However, to avoid exponential growth in complexity and prevent the order from growing by a factor of M at each symbol time, a mixture reduction algorithm is used to decrease the order from MN to N to match (7.11) and (7.12). Several studies have explored Gaussian mixture reduction algorithms, characterized by different levels of complexity, applications, and performance (see [90]–[92]). The mixture reduction algorithm proposed in [93] is selected due to its low complexity, simplicity, and demonstrated effectiveness. This

method involves dividing the phase interval $[-\pi, \pi)$ into N different sub-intervals and merging those Gaussian mixture components with mean values falling within the same sub-interval.

Clustering:

Starting from MN Gaussian mixture components, the goal is to cluster them into N sets based on their mean values. By denoting the mean of the i-th Gaussian component as $\boldsymbol{\mu}_i$, these components are clustered into N groups based on the condition that a component i falls into group j if:

$$\frac{(2j-N)\pi}{N} \leq \boldsymbol{\mu}_i < \frac{(2j-N+2)\pi}{N}, \quad j = 0, \ldots, N-1\,.$$

Merging:

After grouping the components into N clusters, the components within each cluster are merged. Let L_j represent the number of components in the j-th cluster. For each cluster j, the weight, mean, and covariance of the i-th component are denoted as w_i^j, $\boldsymbol{\mu}_i^j$, and $\underline{\underline{\boldsymbol{\Sigma}}}_i^j$ respectively. The merging process for each cluster j operates as follows

$$\sum_{i=0}^{L_j-1} w_i^j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^j, \underline{\underline{\boldsymbol{\Sigma}}}_i^j) \rightarrow w_m^j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m^j, \underline{\underline{\boldsymbol{\Sigma}}}_m^j)$$

where w_m^j, $\boldsymbol{\mu}_m^j$, and $\underline{\underline{\boldsymbol{\Sigma}}}_m^j$ are computed as

$$w_m^j = \sum_{i=0}^{L_j-1} w_i^j \tag{7.19}$$

$$\boldsymbol{\mu}_m^j = \sum_{i=0}^{L_j-1} \frac{w_i^j}{w_m^j} \boldsymbol{\mu}_i^j \tag{7.20}$$

$$\underline{\underline{\boldsymbol{\Sigma}}}_m^j = \sum_{i=0}^{L_j-1} \frac{w_i^j}{w_m^j} \left(\underline{\underline{\boldsymbol{\Sigma}}}_i^j + (\boldsymbol{\mu}_i^j - \boldsymbol{\mu}_m^j)(\boldsymbol{\mu}_i^j - \boldsymbol{\mu}_m^j)^\top \right)\,. \tag{7.21}$$

After applying the mixture reduction algorithm to the forward and backward messages in (7.17) and (7.18), the reduced Gaussian mixtures of order N in (7.11) and (7.12) are obtained, respectively. Their weights, mean values, and covariance matrices are those defined defined in (7.19), (7.20), and (7.21), respectively.

7.2.3 Computation of Extrinsic Information $P_u(c_k)$

Once the forward and backward messages have been computed, the messages $P_u(c_k)$ in Figure 7.1 can be derived. According to the SPA,

$$P_u(c_k) \int_{\boldsymbol{\Phi}_k} \alpha_k(\boldsymbol{\Phi}_k)\beta_k(\boldsymbol{\Phi}_k)p(y_k|c_k, \boldsymbol{\Phi}_k)\, d\boldsymbol{\Phi}_k\,. \tag{7.22}$$

The sequence of messages $\{P_u(c_k)\}_{k=0}^{K-1}$ represent the extrinsic information on the sequence of symbols $\{c_k\}$, used by the decoder to update the likelihoods of the information bits. Therefore, their accuracy directly affects the reliability of the decoder output. Section 7.2 approximated the channel probability law in (7.4), originally a Tikhonov distribution, as a Gaussian. This choice was made to simplify the computation of the forward and backward recursions. However, maintaining this approximation beyond that point is unnecessary and should be avoided, as its accuracy depends significantly on the SNR. At this stage, it is essential to minimize approximation errors to ensure high precision. Thus, instead of continuing with the Gaussian approximation, the probability density function resulting from the product of the forward and backward messages, $\alpha_k(\mathbf{\Phi}_k)$ and $\beta_k(\mathbf{\Phi}_k)$,is approximated as a bivariate Tikhonov mixture to compute (7.22). This change is justified because, after completing the recursions, the covariance matrices of the forward and backward probabilities become sufficiently small ($< 10^{-1}$ [94]) allowing the Tikhonov approximation to provide a more accurate representation. Therefore, defining the pdf resulting from the product between $\alpha_k(\mathbf{\Phi}_k)$ and $\beta_k(\mathbf{\Phi}_k)$ in (7.22) as

$$p(\mathbf{\Phi}_k|\mathbf{y}) \triangleq \alpha_k(\mathbf{\Phi}_k)\beta_k(\mathbf{\Phi}_k) \tag{7.23}$$

$$\propto \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_k^{ij}\mathcal{N}\left(\mathbf{\Phi}_k; \boldsymbol{\mu}_k^{ij}, \underline{\mathbf{\Sigma}}_k^{ij}\right) , \tag{7.24}$$

with

$$w_k^{ij} = \left(\frac{\left|\underline{\mathbf{\Sigma}}_k^{ij}\right|}{\left|\underline{\mathbf{\Sigma}}_k^{f,i}\right|\left|\underline{\mathbf{\Sigma}}_k^{b,j}\right|}\right)^{1/2} w_k^{f,i}w_k^{b,j}e^{-\frac{1}{2}\zeta_k^{ij}} ,$$

$$\boldsymbol{\mu}_k^{ij} = \boldsymbol{\mu}_k^{f,i} + \underline{\mathbf{\Sigma}}_k^{ij}\left(\underline{\mathbf{\Sigma}}_k^{b,j}\right)^{-1}\left(\boldsymbol{\mu}_k^{b,j} - \boldsymbol{\mu}_k^{f,i}\right) , \tag{7.25}$$

$$\underline{\mathbf{\Sigma}}_k^{ij} = \left(\left(\underline{\mathbf{\Sigma}}_k^{f,i}\right)^{-1} + \left(\underline{\mathbf{\Sigma}}_k^{b,j}\right)^{-1}\right)^{-1} , \tag{7.26}$$

and ζ_k^{ij} defined according to (A.6), the posterior probability $p(\mathbf{\Phi}_k|\mathbf{y})$ is approximated as

$$p(\mathbf{\Phi}_k|\mathbf{y}) \approx \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_k^{ij}\frac{1}{\left(\left|\underline{\mathbf{\Sigma}}_k^{ij}\right|\right)^{1/2}} t\left(\mathbf{\Phi}_k; \boldsymbol{\mu}_k^{ij}, \boldsymbol{\kappa}_k^{ij}\right) , \tag{7.27}$$

as a Tikhonov mixture having a mean value corresponding to (7.25) and coefficients $\boldsymbol{\kappa}_k^{ij} = \{\kappa_k^{1,ij}, \kappa_k^{2,ij}, \kappa_k^{3,ij}\}$ computed as shown in (D.7), (D.8), (D.9) in Appendix D, where instead of $\underline{\mathbf{\Sigma}}$, the covariance $\underline{\mathbf{\Sigma}}_k^{ij}$ is adopted. By plugging (7.4) and (7.27)

into (7.22), $P_u(c_k)$ reads

$$
\begin{aligned}
P_u(c_k) \propto & \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_k^{ij} \frac{1}{\left(\left|\underline{\boldsymbol{\Sigma}}_k^{ij}\right|\right)^{1/2}} \int_0^{2\pi}\int_0^{2\pi} \exp\left\{\Re\left[z_k^{2,ij} e^{-j\nu_k} + \left(z_k^{1,ij} + \frac{y_k c_k^*}{\sigma^2}\right) e^{-j\theta_k}\right.\right. \\
& \left.\left. - z_k^{3,ij} e^{-j(\theta_k-\nu_k)}\right]\right\} d\theta_k d\nu_k \\
\propto & \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_k^{ij} \frac{1}{\left(\left|\underline{\boldsymbol{\Sigma}}_k^{ij}\right|\right)^{1/2}} \int_0^{2\pi} \exp\left\{\Re\left[z_k^{2,ij} e^{-j\nu_k}\right]\right\} \\
& \times \int_0^{2\pi} \exp\left\{\Re\left[\left(z_k^{1,ij} + \frac{y_k c_k^*}{\sigma^2} - z_k^{3,ij} e^{j\nu_k}\right) e^{-j\theta_k}\right]\right\} d\theta_k d\nu_k \\
\propto & \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \frac{w_k^{ij}}{\left(\left|\underline{\boldsymbol{\Sigma}}_k^{ij}\right|\right)^{1/2}} \int_0^{2\pi} \exp\left\{\Re\left[z_k^{2,ij} e^{-j\nu_k}\right]\right\} I_0\left(\left|z_k^{1,ij} + \frac{y_k c_k^*}{\sigma^2} - z_k^{3,ij} e^{j\nu_k}\right|\right) d\nu_k \\
\propto & \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \frac{w_k^{ij}}{\left(\left|\underline{\boldsymbol{\Sigma}}_k^{ij}\right|\right)^{1/2}} \frac{\exp\left\{\left|z_k^{2,ij}\right|\right\}}{\sqrt{2\pi\left|z_k^{2,ij}\right|}} I_0\left(\left|z_k^{1,ij} + \frac{y_k c_k^*}{\sigma^2} - \kappa_k^{3,ij} e^{j\mathbf{e}_1^\top \boldsymbol{\mu}_k^{ij}}\right|\right), \qquad (7.28)
\end{aligned}
$$

where

$$
z_k^{1,ij} = \kappa_k^{1,ij} e^{j\mathbf{e}_1^\top \boldsymbol{\mu}_k^{ij}},
$$

$$
z_k^{2,ij} = \kappa_k^{2,ij} e^{j\mathbf{e}_2^\top \boldsymbol{\mu}_k^{ij}},
$$

$$
z_k^{3,ij} = \kappa_k^{3,ij} e^{j\left(\mathbf{e}_1^\top \boldsymbol{\mu}_k^{ij} - \mathbf{e}_2^\top \boldsymbol{\mu}_k^{ij}\right)}.
$$

In (7.28), the Laplace method is adopted to approximate the integral as detailed in Appendix C. In Chapter 4, the Kalman filter is described as making a hard decision on the symbols by compensating the received signal with the estimated phase. However, deriving the algorithm in the factor graph and SPA framework, makes it possible to obtain a soft-output algorithm and compute the probabilities as demonstrated in this section. As discussed in Section 4.3, an advantage of the Tikhonov over the Gaussian is the ability to compute more accurate soft information. The hybrid approach proposed enables efficient forward and backward recursions in the Gaussian domain while maintaining precise computation of extrinsic information.

7.3 Gaussian Mixtures Performance Evaluation

This section evaluates the performance of the Gaussian Mixture algorithm introduced in this chapter, comparing it to the Kalman Filter algorithm. For a fair comparison, the Kalman filter has been adapted to the phase model outlined in (7.1). Two different mixture orders for the Gaussian mixture algorithm, $N = 1$

and $N = 2$, are considered. For $N = 1$, the optimal pilot ratio was found to be 5 pilots for every 100 symbols. In contrast, for $N = 2$ and the Kalman filter, a ratio of 1 pilot for every 20 symbols was used. All results presented refer to the BPSK modulation, and they are compared to the ideal case where the phase is known, denoted as the "known phase". In the results, the Gaussian mixture algorithm is labeled as "GM", while the Kalman filter is denoted as "KF". The GM algorithm has been designed to operate effectively with just a single iteration of the detector, enabling the decoder to operate independently. In contrast, the algorithm discussed in Section 3.2.1 relies significantly on multiple iterations to refine the initial coarse frequency estimate. For this reason, a comparison between the two would not be fair.

First, the performance of the KF and GM algorithms under the matched phase model presented in (7.1) is assessed. This comparison is based on the metric of *pragmatic capacity* [95], which measures the average achievable information rate $\mathbf{E}[I]$ when practical coding and modulation schemes are employed.

Figure 7.2 illustrates the results with parameters $T = 1$ s, $\sigma_\Delta = 0.3$ deg, and a frequency drift of 10^{-5} Hz/s, corresponding to $\sigma_F = 6.28 \times 10^{-5}$ rad. The

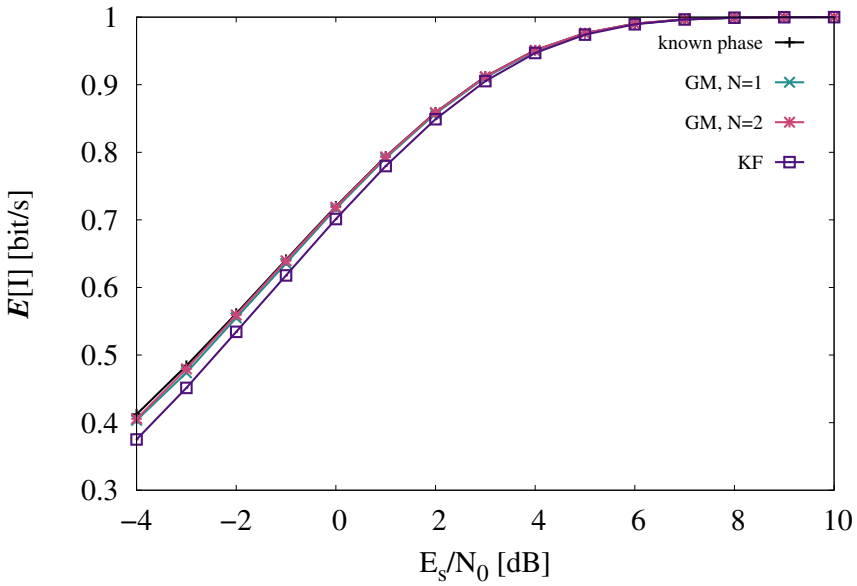


Figure 7.2: Pragmatic capacity for KF and GM operating under (7.1) with $T = 1$ s, $\sigma_\Delta = 0.3$ deg, $\sigma_F = 6.28 \times 10^{-5}$ rad.

figure shows that the GM algorithm slightly outperforms the KF at lower values of E_s/N_0. When the noise level is moderate, increasing N does not translate into a performance improvement, as evidenced by the overlapping of the curves for $N = 1$ and $N = 2$ in Figure 7.2. Moreover, low phase noise values require the parameter

σ_Δ used by the GM algorithm to be adjusted (increased) to prevent cycle slips. For consistency, the same adjustment was applied to the KF algorithm, which also benefited from this tuning. Figure 7.3 presents the results for the same scenario as Figure 7.2 but with a higher phase noise level of $\sigma_\Delta = 3$ deg. In this case, no tuning was required. In Figure 7.3, it can be observed that increasing the mixture

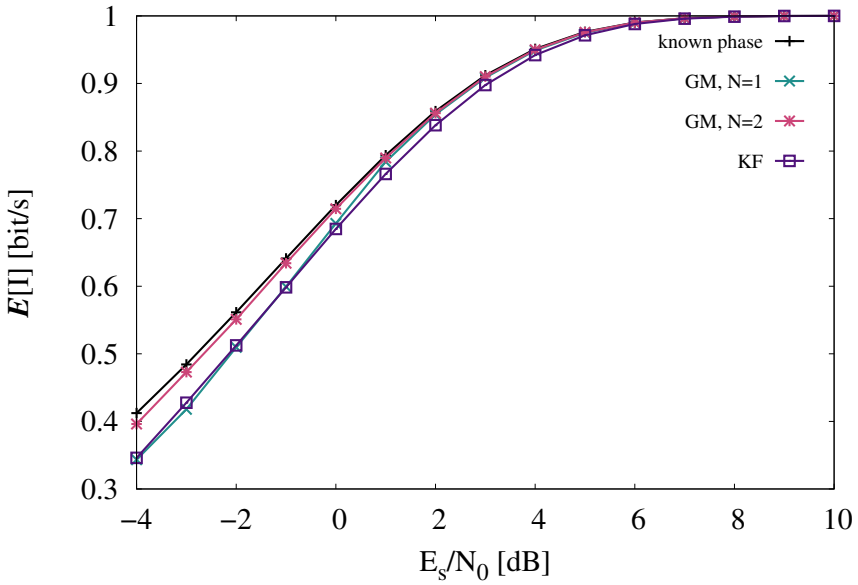


Figure 7.3: Pragmatic capacity for KF and GM operating under (7.1) with $T = 1$ s, $\sigma_\Delta = 3$ deg, $\sigma_F = 6.28 \times 10^{-5}$ rad.

order from $N = 1$ to $N = 2$ leads to performance improvement. The KF achieves performance equivalent to the GM with $N = 1$, although the latter shows slight improvements at high SNR values.

To evaluate how well the algorithms adapt to a mismatched channel, the sinusoidal Doppler model described in Section 2.2.4, combined with the phase noise following the Wiener process (Section 2.2.1) was considered. Figure 7.4 shows the pragmatic capacity for the sinusoidal model characterized by a frequency shift of $f_D = 0.01$ Hz, a frequency drift of $f_R = 10^{-5}$ Hz/s, and a phase noise standard deviation of $\sigma_\Delta = 0.3$ deg. The figure demonstrates that the GM algorithm with $N = 2$ performed nearly as well as the ideal case. However, reducing the number of components to $N = 1$ results in a noticeable, though small, performance gap, particularly at lower SNRs. In contrast, the KF exhibits a significant performance drop due to its limited adaptability to the mismatched sinusoidal model, making it less robust under low-SNR conditions.

Despite the promising results observed in Figures 7.2, 7.3, and 7.4, the GM and the KF algorithms turned out to be vulnerable to cycle slip events. Cycle slips

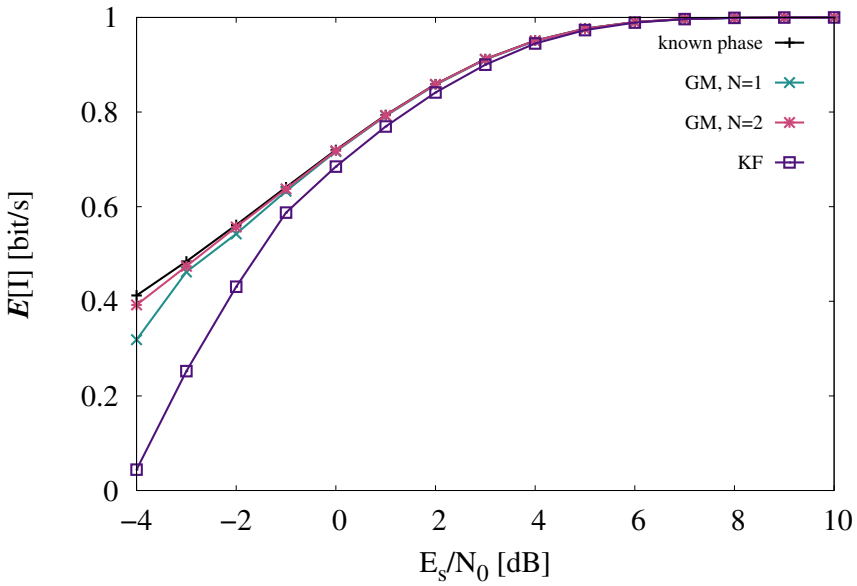


Figure 7.4: Pragmatic capacity for KF and GM operating under (2.11) with $\sigma_\Delta = 0.3$ deg, $f_D = 0.01$ Hz, $f_R = 10^{-5}$ Hz/s.

are a well-known challenge in the literature, leading to a significant degradation in system performance. Previous studies, such as those by the authors of [14] and [59], highlighted the necessity of implementing effective control mechanisms to detect and mitigate these events. From the analysis, this vulnerability is attributed to the Gaussian approximation employed for the channel probability distributions ($p_d()$), by definition Tikhonov distributions. This approximation introduces a deviation from the true channel characteristics, impacting the robustness of the algorithms. As a result, achieving satisfactory BER and FER performance becomes challenging without employing "long" interleavers. Figure 7.5 presents the FER comparison between the GM with $N = 2$ and the KF, both under the same Doppler effect conditions as illustrated in Figure 7.4. In this analysis, an LDPC code with a rate of 1/2 and a block length of 4096 is adopted. For each simulation point in the figure, at least 100 frame errors are recorded. To achieve the desired steepness in the FER curve, which is characteristic of the chosen LDPC code, both algorithms were paired with a row-column interleaver configuration consisting of 800 rows and 4096 columns. The interleaver's purpose was to enhance the resilience to cycle slip events by spreading their impact across a wider range of symbols. As shown in Figure 7.5, the GM curve closely achieves ideal performance, while the KF suffers disruptions caused by cycle slips.

The algorithms discussed in Chapter 3 and those proposed in this thesis have been analyzed with the interspersed pilot insertion strategy. To determine whether

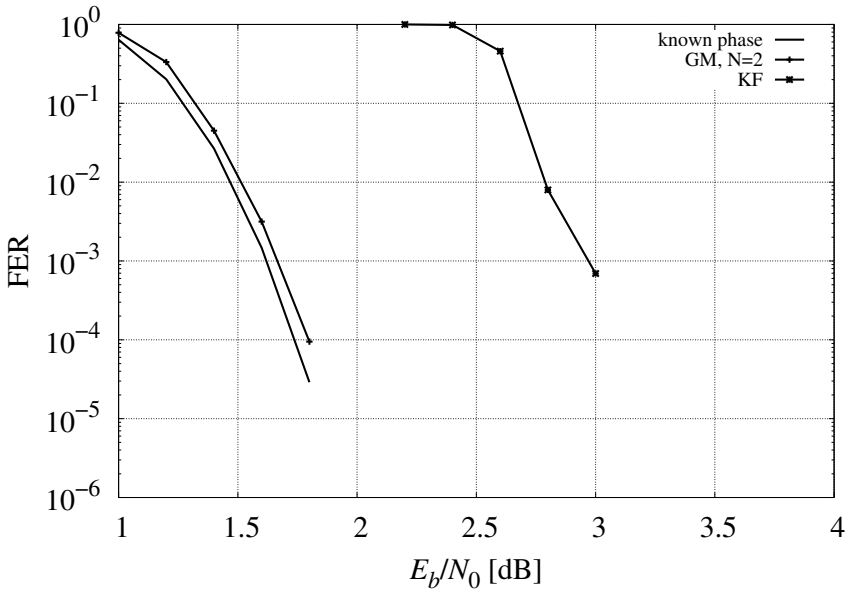


Figure 7.5: GM with $N = 2$, and KF over the sinusoidal Doppler model of (2.11) with $f_D = 0.01$ Hz, and $f_R = 10^{-5}$ Hz/s and $\sigma_\Delta = 0.3$ deg.

an alternative pilot insertion strategy could reduce the GM algorithm's susceptibility to cycle slip events, the superimposed strategy described in Section 2.3 is explored.

In this scenario, the GM algorithm follows the derivation outlined in Section 7.2. The only modification made is to replace the symbol c_k with the linear combination presented in equation (2.13) throughout all relevant algorithm equations. For testing purposes, the sinusoidal Doppler model in 2.2.4 is adopted with a frequency shift of $f_D = 0.01$ Hz, and frequency drift values of 6×10^{-5}, 6×10^{-4}, and 6×10^{-3} Hz/s. The phase noise is also modeled as a Wiener process with $\sigma_\Delta = 0.9$ deg. In this setup, a pilot-to-total power ratio of $\gamma = 10^{-4}$ was adopted, indicating that the pilot symbols are allocated only 0.01% of the total power. This allocation is significantly lower than the interspersed pilot scenario discussed earlier in the thesis, where the pilot symbols accounted for 5% of the total power. Figure 7.6 presents the FER of the GM algorithm obtained using an LDPC code of rate 1/2 and block length 4096. Each simulation point in the figure counts at least 100 frame errors. The figure demonstrates that the GM algorithm exhibits remarkable resilience to cycle slips, closely matching the ideal curve with less than 0.5 dB of loss, even in the most challenging scenarios, and achieving this without the need for an interleaver. Furthermore, in this scenario, the GM algorithm did not gain any advantage in using a mixture order greater than $N = 1$. The computational load of the GM algorithm per code symbol for M-PSK modulation is approximately $\mathcal{O}(M)$. This

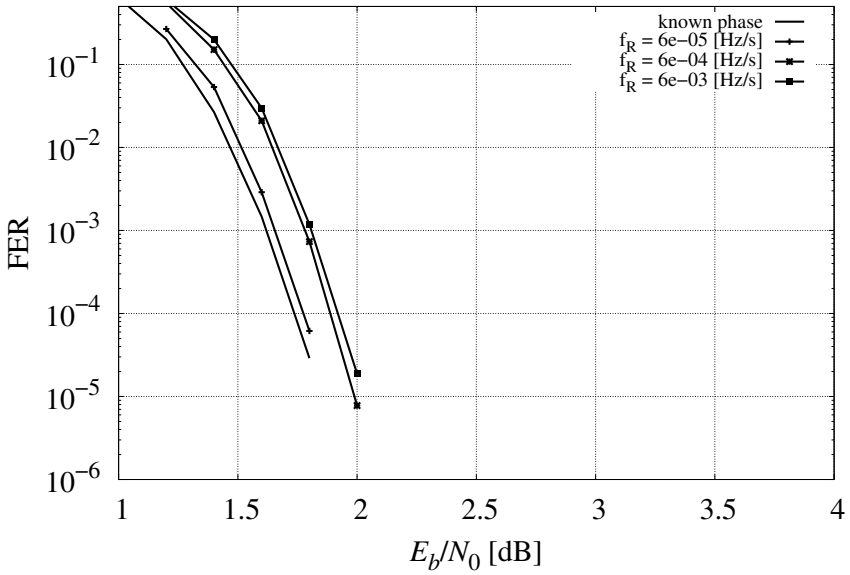


Figure 7.6: FER for GM with $N = 1$ over the sinusoidal Doppler model with $f_D = 0.01$ Hz, and f_R of 6×10^{-5}, 6×10^{-4}, and 6×10^{-3} Hz/s, and $\sigma_\Delta = 0.9$ deg.

complexity arises from the assumption that the number of component mixtures does not exceed the size of the constellation, M.

The algorithm works by dividing the phase interval into M or fewer regions and then clustering the components within each region. Each clustering operation involves a constant computational cost, which makes the overall complexity linearly dependent on M.

Similarly, the KF algorithm also has a complexity of $\mathcal{O}(M)$ per code symbol. However, even though both algorithms share the same asymptotic complexity, their practical performances may differ significantly due to the constant factors hidden within the $\mathcal{O}(M)$ notation. These constants depend on the specific operations performed within each algorithm.

The KF algorithm first merges the mixture into a single dominant component. Once this merging is complete, operations are performed only on this single component, eliminating the need to cluster multiple components. This significantly reduces the computational load per symbol.

Chapter 8

Conclusion

This thesis addressed the challenge of synchronization in coherent communication systems operating over time-selective channels. It recognizes the limitations of conventional methods, such as the PLL, especially in harsh propagation environments characterized by low SNRs and Doppler variations. The thesis explores advanced detection and synchronization techniques based on the factor graph and SPA framework.

After providing an overview of foundational concepts, a review of existing algorithms that utilize the factor graph and SPA framework in phase noise channels and channels affected by frequency uncertainties follows. This review highlights both the advantages and limitations of these algorithms, revealing that the existing literature on joint detection and synchronization of phase and frequency is quite limited.

As an initial contribution, the thesis discusses the duality between the Kalman filter——a Bayesian estimator derived from the SPA—and the PLL. This perspective aims to assist readers who may be unfamiliar with the advanced algorithms presented. Additionally, an alternative version of the PLL is proposed based on a different exponential family: the Tikhonov distribution.

Building upon these insights, novel detection and synchronization algorithms designed for harsh propagation environments are introduced. Focusing on phase noise, the thesis proposes adopting an oversampling technique to enhance the information rate in phase noise channels. A variation of the classical dp-BCJR algorithm operating at the sample level is derived. The resulting algorithm, called multi-sample dp-BCJR, demonstrates superior performance compared to the traditional dp-BCJR in scenarios characterized by high phase noise levels, particularly with high-order modulations like M-PSK, which are more susceptible to phase noise. However, the increased complexity of the multi-sample dp-BCJR relative to the classical dp-BCJR limits its applications to low cardinality modulations.

To address this, a low-complexity alternative to the multi-sample dp-BCJR is proposed while still employing the oversampling technique. This alternative uses

the Tikhonov approximation and leads to the development of the multi-sample Tikhonov algorithm. Remarkably, the multi-sample Tikhonov shows performance comparable to that of the dp-BCJR, which is optimal given a sufficient number of levels of phase discretization. The multi-sample Tikhonov algorithm stands out due to its reduced complexity compared to the classical dp-BCJR.

The thesis then examined the synchronization challenges in channels impacted by phase noise and frequency uncertainties. It presented an introductory case study focused on the ESA deep space transponder, addressing the challenges inherent to future deep space missions. The current deep space transponder relies on traditional synchronization techniques (PLLs), which are inadequate under significantly noisy conditions expected for these missions. The study has been divided into two parts: carrier and sub-carrier synchronization. The feasibility of adopting a Kalman filter was investigated for carrier synchronization to improve the residual phase jitter on the sub-carrier. By doing so, sub-carrier synchronization focused on tracking Doppler frequency shift and frequency drift. For this purpose, an algorithm within the factor graph and SPA framework was designed, utilizing Tikhonov distributions to approximate the exchanged messages. The resulting factor graph, associated with the MAP strategy, was characterized by two sets of variables—one for phase and one for frequency—communicating through message exchanges. Given the short cycles in the factor graph, two scheduling algorithms were proposed: flooding and serial. The serial method was favored due to the latency issues of the flooding scheduler. Unfortunately, the serial scheduler introduced instability when employed in an iterative scheme with the decoder.

As a result, an alternative approach was considered: clustering the frequency and phase variables in a single variable node and adopting bivariate Gaussian mixtures to approximate the SPA messages. The resulting algorithm demonstrated superior performance when compared to the Kalman filter, especially when the mixture order exceeded one, and exhibited greater resilience than the Kalman filter under a mismatched channel. Nevertheless, both algorithms proved to be susceptible to cycle slips and the Gaussian mixture algorithm turned out to benefit from the use of long interleavers for optimal performance. To address these vulnerabilities, future research should explore the implementation of a cycle slip detector, and possibly a cycle slip corrector, to minimize the reliance on long interleavers, which introduce increased latency into the system. In contrast, considering a superimposed pilot insertion strategy, wherein pilot symbols were linearly combined with the information symbols, allowed the Gaussian mixture algorithm to perform effectively with only a single Gaussian component and eliminated the need for any interleaver.

Appendix A

Constant Term in Gaussian Product

In this appendix, we focus on the weight resulting from the product between two bivariate Gaussian distributions. We begin by expressing the weighted bivariate Gaussian in canonical form

$$\exp\left\{C_i + \mathbf{B}_i^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \underline{\mathbf{A}}_i \mathbf{x}\right\} ,$$

where the canonical parameters are defined as follows

$$C_i = \log(w_i) - \frac{1}{2}\left(\log\left(2\pi\left|\underline{\boldsymbol{\Sigma}}_i\right|\right) + \boldsymbol{\mu}_i^\top \underline{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{\mu}_i\right) , \tag{A.1}$$

$$\mathbf{B}_i = \underline{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{\mu}_i , \tag{A.2}$$

$$\underline{\mathbf{A}}_i = \underline{\boldsymbol{\Sigma}}_i^{-1} . \tag{A.3}$$

Here, w_i represents the weight, $\boldsymbol{\mu}_i$ the mean vector, and $\underline{\boldsymbol{\Sigma}}_i$ the covariance matrix.

The product of two weighted bivariate Gaussian distributions, $w_i\mathcal{N}\left(\mathbf{x};\boldsymbol{\mu}_i,\underline{\boldsymbol{\Sigma}}_i\right)$ and $w_j\mathcal{N}\left(\mathbf{x};\boldsymbol{\mu}_j,\underline{\boldsymbol{\Sigma}}_j\right)$, can be expressed as

$$\exp\left\{C_i + C_j - C_{ij}\right\}\exp\left\{C_{ij} + \mathbf{B}_{ij}^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \underline{\mathbf{A}}_{ij}\mathbf{x}\right\} , \tag{A.4}$$

where

$$C_{ij} = \log(w_i w_j) - \frac{1}{2}\left(\mathbf{B}_{ij}^\top \underline{\mathbf{A}}_{ij}^{-1}\mathbf{B}_{ij} - \log\left(2\pi\left|\underline{\mathbf{A}}_{ij}\right|\right)\right) ,$$

$$\underline{\mathbf{A}}_{ij} = \sum_{i=0}^{1} \underline{\mathbf{A}}_i ,$$

$$\mathbf{B}_{ij} = \sum_{i=0}^{1} \mathbf{B}_i .$$

The first term on the right-hand in (A.4) is again a Gaussian function. Expanding the expression using the definition in (A.1), we obtain

$$\exp\{C_i + C_j - C_{ij}\} = \exp\left\{-\frac{1}{2}\left(\log|A_{ij}| + \log|\underline{\boldsymbol{\Sigma}}_i| + \log\left|\underline{\boldsymbol{\Sigma}}_j\right|\right)\right\}\exp\left\{-\frac{1}{2}\zeta_{ij}\right\}.$$

By using the definitions in (A.2) and (A.3), and considering the following identity

$$\left(\underline{\boldsymbol{\Sigma}}_i^{-1} + \underline{\boldsymbol{\Sigma}}_j^{-1}\right)^{-1} = \underline{\boldsymbol{\Sigma}}_i\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\underline{\boldsymbol{\Sigma}}_j = \underline{\boldsymbol{\Sigma}}_j\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\underline{\boldsymbol{\Sigma}}_i,$$

the term ζ_{ij} can be written as

$$\begin{aligned}
\zeta_{ij} = &-\Big(-\boldsymbol{\mu}_i^T\underline{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{\mu}_i - \boldsymbol{\mu}_j^T\underline{\boldsymbol{\Sigma}}_j^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_i^T\underline{\boldsymbol{\Sigma}}_i^{-1}\underline{\boldsymbol{\Sigma}}_j\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i \\
&+\boldsymbol{\mu}_i^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i + \boldsymbol{\mu}_j^T\underline{\boldsymbol{\Sigma}}_j^{-1}\underline{\boldsymbol{\Sigma}}_i\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j\Big) \\
= &-\Big(\boldsymbol{\mu}_i^T\underline{\boldsymbol{\Sigma}}_i^{-1}\left[\underline{\boldsymbol{\Sigma}}_j\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1} - \underline{\mathbf{I}}\right]\boldsymbol{\mu}_i + \boldsymbol{\mu}_j^T\underline{\boldsymbol{\Sigma}}_j^{-1}\left[\underline{\boldsymbol{\Sigma}}_i\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1} - \underline{\mathbf{I}}\right]\boldsymbol{\mu}_j \\
&+\boldsymbol{\mu}_i^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i\Big),
\end{aligned} \tag{A.5}$$

where in (A.5), we gather the common terms together. Let us re-write the identity matrix, $\underline{\mathbf{I}}$ as

$$\underline{\mathbf{I}} = \left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1},$$

and by using simple mathematical manipulations, we obtain

$$\begin{aligned}
\zeta_{ij} = &-\Big(\boldsymbol{\mu}_i^T\underline{\boldsymbol{\Sigma}}_i^{-1}\left[-\underline{\boldsymbol{\Sigma}}_i\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\right]\boldsymbol{\mu}_i + \boldsymbol{\mu}_j^T\underline{\boldsymbol{\Sigma}}_j^{-1}\left[-\underline{\boldsymbol{\Sigma}}_j\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\right] \\
&+\boldsymbol{\mu}_i^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j + \boldsymbol{\mu}_j^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i\Big) \\
= &\Big(\boldsymbol{\mu}_i^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i + \boldsymbol{\mu}_j^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j - \boldsymbol{\mu}_i^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_j \\
&-\boldsymbol{\mu}_j^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\boldsymbol{\mu}_i\Big) \\
= &\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^T\left(\underline{\boldsymbol{\Sigma}}_i + \underline{\boldsymbol{\Sigma}}_j\right)^{-1}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right).
\end{aligned} \tag{A.6}$$

In standard notation, (A.4) translates into

$$w_{ij}\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{ij},\underline{\boldsymbol{\Sigma}}_{ij}) = w_{ij}\frac{1}{\left(2\pi|\underline{\boldsymbol{\Sigma}}_{ij}|\right)^{1/2}}e^{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{ij})^\top\underline{\boldsymbol{\Sigma}}_{ij}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{ij})\right\}},$$

with

$$w_{ij} = e^{-\frac{1}{2}\zeta_{ij}}\frac{w_i w_j}{\left(\left|\underline{\mathbf{A}}_{ij}\right||\underline{\boldsymbol{\Sigma}}_i|\left|\underline{\boldsymbol{\Sigma}}_j\right|\right)^{1/2}}, \tag{A.7}$$

$$\begin{aligned} \underline{\boldsymbol{\Sigma}}_{ij} &= \underline{\mathbf{A}}_{ij}^{-1}\,, \\ \boldsymbol{\mu}_{ij} &= \underline{\mathbf{A}}_{ij}^{-1}\mathbf{B}_{ij}\,, \end{aligned} \tag{A.8}$$

and ζ_{ij} defined as in (A.6).

Appendix B

Convolution of Bivariate Gaussians

Consider the convolution of the following bivariate Gaussian pdfs

$$\begin{aligned}
&\int_{\mathbf{x}} g(\boldsymbol{\mu}, \underline{\boldsymbol{\Sigma}}; \mathbf{x}) g(\mathbf{0}, \underline{\mathbf{B}}; \mathbf{y} - \underline{\mathbf{H}}\mathbf{x}) d\mathbf{x} \\
&\int_{\mathbf{x}} e^{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \underline{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}} e^{\left\{-\frac{1}{2}(\mathbf{y}-\underline{\mathbf{H}}\mathbf{x})^\top \underline{\mathbf{B}}^{-1}(\mathbf{y}-\underline{\mathbf{H}}\mathbf{x})\right\}} d\mathbf{X} \\
&\int_{\mathbf{x}} e^{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \underline{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}} e^{\left\{-\frac{1}{2}(\underline{\mathbf{H}}^{-1}\mathbf{y}-\mathbf{x})^\top \tilde{\underline{\mathbf{B}}}^{-1}(\underline{\mathbf{H}}^{-1}\mathbf{y}-\mathbf{x})\right\}} d\mathbf{X} \\
&e^{\left\{-\frac{1}{2}(\underline{\mathbf{H}}^{-1}\mathbf{y}-\boldsymbol{\mu})^\top \left(\underline{\boldsymbol{\Sigma}}+\tilde{\underline{\mathbf{B}}}\right)^{-1}(\underline{\mathbf{H}}^{-1}\mathbf{y}-\boldsymbol{\mu})\right\}} \\
&e^{\left\{-\frac{1}{2}(\mathbf{y}-\underline{\mathbf{H}}\boldsymbol{\mu})^\top \underline{\mathbf{H}}^{-T}\left(\underline{\boldsymbol{\Sigma}}+\tilde{\underline{\mathbf{B}}}\right)^{-1}\underline{\mathbf{H}}^{-1}(\mathbf{y}-\underline{\mathbf{H}}\boldsymbol{\mu})\right\}}, \qquad \text{(B.1)}
\end{aligned}$$

where we define

$$\tilde{\underline{\mathbf{B}}}^{-1} = \underline{\mathbf{H}}^\top \underline{\mathbf{B}}^{-1} \underline{\mathbf{H}}. \qquad \text{(B.2)}$$

Appendix C

Laplace Approximation Method

The Laplace method provides us with an approximation to solve the following integral

$$I = \int_a^b f(x) e^{-\lambda g(x)} dx \,, \tag{C.1}$$

where $\lambda \gg 1$ is a positive number. According to the Laplace method, the integral in (C.1) can be approximated as

$$I \approx \left(\frac{2\pi}{\lambda}\right)^{1/2} \frac{1}{\sqrt{g''(x_0)}} f(x_0) e^{-\lambda g(x_0)} \,, \tag{C.2}$$

where x_0 is the value that minimizes the function $g(x)$ in (a, b) and for which $f(x)$ is different from 0. In the case of solving the integral necessary for the computation of the soft information on the symbols, the integral we are interested in is the following

$$I(\nu) = \int_0^{2\pi} e^{\kappa \cos(\nu - \mu)} I_0 \left(\left| \left(z_i - z_j e^{j\nu} \right) \right| \right) d\nu \tag{C.3}$$

$$= \int_0^{2\pi} e^{-\kappa g(\nu)} f(\nu) d\nu \,, \tag{C.4}$$

where we defined

$$g(\nu) = -\cos(\nu - \mu) \,, \tag{C.5}$$

and

$$f(\nu) = I_0 \left(\left| \left(z_i - z_j e^{j\nu} \right) \right| \right) \,. \tag{C.6}$$

To apply the Laplace approximation, we first identify the minimum of the function $g(\nu)$

$$\begin{aligned} \nu_0 &= \min g(\nu) \\ &= \min -\cos(\nu - \mu) \\ &= \mu \,, \end{aligned}$$

and obtain

$$I(\nu) \approx \left(\frac{2\pi}{\kappa}\right)^{1/2} I_0\left(\left|\left(z_i - z_j e^{j\mu}\right)\right|\right) e^{\kappa}\,. \tag{C.7}$$

Appendix D

Bivariate Tikhonov-Gaussian Equivalence

In this appendix, we want to show the equivalence between the bivariate Gaussian distribution and the cosine variant of the bivariate Tikhonov distribution. Consider the cosine model representation of a bivariate Tikhonov distribution of the variable $\boldsymbol{\Phi} = [\theta, \nu]^\top$

$$t(\boldsymbol{\Phi}; \boldsymbol{\mu}, \boldsymbol{\kappa}) = c(\boldsymbol{\kappa}) \exp\{\kappa_1 \cos(\theta - \mu_1) + \kappa_2 \cos(\nu - \mu_2) - \kappa_3 \cos(\theta - \mu_1 - \nu + \mu_2)\}\,, \tag{D.1}$$

where μ_1, μ_2, κ_1, κ_2 are the mean values and concentration parameters of θ and ν, respectively. Instead, κ_3 controls the correlation between θ and ν, and $c(\boldsymbol{\kappa})$ is the normalization constant [96]. The definition in (D.1) can be expanded using the Taylor series for the cosine function, and obtain

$$t(\boldsymbol{\mu}, \boldsymbol{\kappa}; \boldsymbol{\Phi}) \approx c(\boldsymbol{\kappa}) \exp\left\{-\frac{1}{2}\left[\kappa_1(\theta - \mu_1)^2 + \kappa_2(\nu - \mu_2)^2 - \kappa_3(\theta - \mu_1 - \nu + \mu_2)^2\right]\right\} \tag{D.2}$$

$$\propto c(\boldsymbol{\kappa}) \exp\left\{-\frac{1}{2}\left[(\theta - \mu_1)^2(\kappa_1 - \kappa_3) + (\nu - \mu_2)^2(\kappa_2 - \kappa_3) + 2(\theta - \mu_1)(\nu - \mu_2)\kappa_3\right]\right\}. \tag{D.3}$$

where we neglected the constant terms. Disregarding the normalization constant $c(\boldsymbol{\kappa})$, Equation (D.3) has the form of a bivariate Gaussian distribution of the variable $\boldsymbol{\Phi} = [\theta, \nu]^\top$

$$\mathcal{N}(\boldsymbol{\Phi}; \boldsymbol{\mu}, \underline{\boldsymbol{\Sigma}}) \propto \exp\left\{-\frac{1}{2}\begin{bmatrix}\theta - \mu_1 \\ \nu - \mu_2\end{bmatrix}^\top \underline{\boldsymbol{\Sigma}}^{-1} \begin{bmatrix}\theta - \mu_1 \\ \nu - \mu_2\end{bmatrix}\right\} \tag{D.4}$$

$$\propto \exp\left\{-\frac{1}{2}\left[(\theta - \mu_1)^2\Sigma_{11} + (\nu - \mu_2)^2\Sigma_{22} + 2(\theta - \mu_1)(\nu - \mu_2)\Sigma_{12}\right]\right\}, \tag{D.5}$$

having mean vector $\boldsymbol{\mu} = [\mu_1, \mu_2]^\top$ and inverse covariance matrix $\underline{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$. By comparison of (D.3) and (D.4), it turns out that the bivariate Tikhonov distribution can be seen as a bivariate Gaussian having mean value $\boldsymbol{\mu}$ and inverse covariance matrix

$$\underline{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \kappa_1 - \kappa_3 & \kappa_3 \\ \kappa_3 & \kappa_2 - \kappa_3 \end{bmatrix} . \tag{D.6}$$

Vice versa, a bivariate Gaussian distribution can be seen as a bivariate Tikhonov distribution with mean value $\boldsymbol{\mu}$ and coefficients

$$\kappa_1 = \Sigma_{11} + \Sigma_{12} , \tag{D.7}$$

$$\kappa_2 = \Sigma_{22} + \Sigma_{12} , \tag{D.8}$$

$$\kappa_3 = \Sigma_{12} . \tag{D.9}$$

Bibliography

[1] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill, 2008.

[2] ECSS-E-ST-50-05C, *Space Engineering - Radio frequency and modulation*, Available at www.ecss.nl, 2011.

[3] J. Vilà-Valls, M. Navarro, P. Closas, and M. Bertinelli, "Synchronization challenges in deep space communications", *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 1, pp. 16–27, 2019.

[4] A. Piemontese, G. Colavolpe, and T. Eriksson, "A new analytical model of phase noise in communication systems", in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 926–931. DOI: 10.1109/WCNC51071.2022.9771968.

[5] G. Colavolpe, A. Barbieri, and G. Caire, "Algorithms for iterative decoding in the presence of strong phase noise", *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 9, pp. 1748–1757, 2005. DOI: 10.1109/JSAC.2005.853813.

[6] G. Colavolpe *et al.*, "Communications over phase-noise channels: A tutorial review", *International Journal of Satellite Communications and Networking*, vol. 32, pp. 167–185, 2014.

[7] A. Chorti and M. Brookes, "On the effect of Voigt profile oscillators on OFDM systems", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 11, pp. 768–772, 2011. DOI: 10.1109/TCSII.2011.2168011.

[8] S. Hinedi, "NASA's next generation all-digital deep space network breadboard receiver", *IEEE transactions on communications*, vol. 41, no. 1, pp. 246–257, 1993.

[9] M. C. Comparini, F. De Tiberis, R. Novello, *et al.*, "Advances in deep-space transponder technology", *Proceedings of the IEEE*, vol. 95, no. 10, pp. 1994–2008, 2007. DOI: 10.1109/JPROC.2007.905078.

[10] P. K. Enge, "The global positioning system: Signals, measurements, and performance", *International Journal of Wireless Information Networks*, vol. 1, pp. 83–105, 1994.

[11] N. Samama, *Global positioning: Technologies and performance*. John Wiley & Sons, 2008.

[12] 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 Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1345–1350, 2005.

[13] Y. Zhu, D. Guo, and M. L. Honig, "A message-passing approach for joint channel estimation, interference mitigation, and decoding", *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 6008–6018, 2009.

[14] S. Shayovitz and D. Raphaeli, "Message passing algorithms for phase noise tracking using Tikhonov mixtures", *IEEE Transactions on Communications*, vol. 64, no. 1, pp. 387–401, 2015.

[15] Q. Shi, N. Wu, H. Wang, and W. Yuan, "Joint channel estimation and decoding in the presence of phase noise over time-selective flat-fading channels", *IET Communications*, vol. 10, no. 5, pp. 577–585, 2016.

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

[17] A. Worthen and W. Stark, "Unified design of iterative receivers using factor graphs", *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 843–849, 2001. DOI: 10.1109/18.910595.

[18] A. Barbieri and G. Colavolpe, "Soft-output decoding of rotationally invariant codes over channels with phase noise", *IEEE Transactions on communications*, vol. 55, no. 11, pp. 2125–2133, 2007.

[19] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[20] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations", *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972. DOI: 10.1109/TAC.1972.1100034.

[21] P. Tam and J. Moore, "A gaussian sum approach to phase and frequency estimation", *IEEE Transactions on Communications*, vol. 25, no. 9, pp. 935–942, 1977.

[22] A. Barbieri, G. Colavolpe, and G. Caire, "Joint iterative detection and decoding in the presence of phase noise and frequency offset", *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 171–179, 2007.

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

[24] M. Patzold and F. Laue, "Statistical properties of Jakes' fading channel simulator", in *VTC '98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No.98CH36151)*, vol. 2, 1998, 712–718 vol.2. DOI: 10.1109/VETEC.1998.683675.

[25] A. Coulson, A. Williamson, and R. Vaughan, "A statistical basis for lognormal shadowing effects in multipath fading channels", *IEEE Transactions on Communications*, vol. 46, no. 4, pp. 494–502, 1998. DOI: 10.1109/26.664305.

[26] A. Piemontese, G. Colavolpe, and T. Eriksson, "Phase noise in communication systems: from measures to models", *arXiv preprint arXiv:2104.07264*, 2021.

[27] A. Patapoutian, "On phase-locked loops and Kalman filters", *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 670–672, 1999.

[28] J. Vila-Valls, P. Closas, M. Navarro, and C. Fernandez-Prades, "Are PLLs dead? a tutorial on Kalman filter-based techniques for digital carrier synchronization", *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 7, pp. 28–45, 2017.

[29] S. Golestan, J. M. Guerrero, and J. C. Vasquez, "Steady-state linear Kalman filter-based PLLs for power applications: a second look", *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9795–9800, 2018.

[30] F. M. Gardner, *Phaselock techniques*. John Wiley & Sons, 2005.

[31] CCSDS 130.11-G-2, Summary of definition and performance, Available at http://public.ccsds.org/, May 2023.

[32] ETSI, "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)", ETSI, Technical Specification EN 302 307, 2009.

[33] 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)*, 2013.

[34] P. Hoeher and F. Tufvesson, "Channel estimation with superimposed pilot sequence", in *Seamless Interconnection for Universal Services. Global Telecommunications Conference. GLOBECOM'99.(Cat. No. 99CH37042)*, IEEE, vol. 4, 1999, pp. 2162–2166.

[35] T. Cui and C. Tellambura, "OFDM channel estimation and data detection with superimposed pilots", *European Transactions on Telecommunications*, vol. 22, no. 3, pp. 125–136, 2011.

[36] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1", in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, 1993, 1064–1070 vol.2. DOI: 10.1109/ICC.1993.397441.

[37] F. Mo, S. Kwatra, and J. Kim, "Analysis of puncturing pattern for high rate turbo codes", in *MILCOM 1999. IEEE Military Communications. Conference Proceedings (Cat. No. 99CH36341)*, IEEE, vol. 1, 1999, pp. 547–550.

[38] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art", in *Proc. International Symposium on Turbo Codes and Related Topics*, 1997, pp. 1–11.

[39] C. Douillard, M. Jézéquel, C. Berrou, *et al.*, "Iterative correction of intersymbol interference: turbo-equalization", *European transactions on telecommunications*, vol. 6, no. 5, pp. 507–511, 1995.

[40] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA", *IEEE Transactions on communications*, vol. 47, no. 7, pp. 1046–1061, 1999.

[41] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenated trellis coded modulation with iterative decoding", in *Proceedings of IEEE International Symposium on Information Theory*, IEEE, 1997, p. 8.

[42] G. Ferrari, G. Colavolpe, R. Raheli, *et al.*, "Noncoherent iterative decoding of spectrally efficient coded modulations", in *ANNALES DES TÉLÉCOMMUNICATIONS*, vol. 56, 2001, pp. 409–421.

[43] G. D. Forney, "The Viterbi algorithm", *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[44] R. Tanner, "A recursive approach to low complexity codes", *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981. DOI: 10.1109/TIT.1981.1056404.

[45] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs", *European Transactions on telecommunications*, vol. 6, no. 5, pp. 513–525, 1995.

[46] N. Wiberg, "Codes and decoding on general graphs", 1996.

[47] H.-A. Loeliger, "An introduction to factor graphs", *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[48] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge university press, 2008.

[49] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels", *IEEE Transactions on Communications*, vol. 53, no. 5, pp. 818–825, 2005.

[50] M. Kobayashi, J. Boutros, and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation", *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 8, pp. 1450–1460, 2001. DOI: 10.1109/49.942508.

[51] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis", *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1772–1793, 2002. DOI: 10.1109/TIT.2002.1013125.

[52] J. Vilà-Valls, P. Closas, and C. Fernández-Prades, "Advanced KF-based methods for GNSS carrier tracking and ionospheric scintillation mitigation", in *2015 IEEE Aerospace Conference*, 2015, pp. 1–10. DOI: 10.1109/AERO.2015.7118930.

[53] J.-H. Won, T. Pany, and B. Eissfeller, "Characteristics of Kalman filters for GNSS signal tracking loop", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3671–3681, 2012. DOI: 10.1109/TAES.2012.6324756.

[54] J. Vilà-Valls, J.-M. Brossier, and L. Ros, "Oversampled phase tracking in digital communications with large excess bandwidth", *Signal Processing*, vol. 90, no. 3, pp. 821–833, 2010, ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2009.08.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168409003752.

[55] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)", *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974. DOI: 10.1109/TIT.1974.1055186.

[56] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical distributions*. John Wiley & Sons, 2011.

[57] T. P. Minka, "Expectation propagation for approximate Bayesian inference", *arXiv preprint arXiv:1301.2294*, 2013.

[58] E. Conti, A. Piemontese, G. Colavolpe, and A. Vannucci, "Expectation propagation for flat-fading channels", *IEEE Wireless Communications Letters*, 2023.

[59] G. Colavolpe, E. Conti, A. Piemontese, and A. Vannucci, "The difficult road of expectation propagation towards phase noise detection", in *Proc. IEEE Intern. Conf. Commun.*, Rome, Italy, Jun. 2023.

[60] B. Ripani, A. Modenini, and G. Montorsi, "Digital PLLs for phase noise channels: a concept based on the Tikhonov distribution", *IEEE Signal Processing Letters*, 2024.

[61] A. Piemontese, G. Colavolpe, and T. Eriksson, "Discrete-time models and performance of phase noise channels", *IEEE Open Journal of the Communications Society*, 2024.

[62] N. Assimakis and M. Adam, "Kalman filter Riccati equation for the prediction, estimation, and smoothing error covariance matrices", *International Scholarly Research Notices*, vol. 2013, no. 1, p. 249 594, 2013.

[63] H. Shu, E. P. Simon, and L. Ros, "Third-order Kalman filter: Tuning and steady-state performance", *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1082–1085, 2013.

[64] G. S. Christiansen, "Modeling of PRML timing loop as a Kalman filter", in *1994 IEEE GLOBECOM. Communications: The Global Bridge*, IEEE, vol. 2, 1994, pp. 1157–1161.

[65] P. F. Driessen, "DPLL bit synchronizer with rapid acquisition using adaptive Kalman filtering techniques", *IEEE Transactions on Communications*, vol. 42, no. 9, pp. 2673–2675, 1994.

[66] T. Oren and D. Raphaeli, "A new suppressed carrier synchronizer with reduced phase jitter for QAM systems", in *21st IEEE Convention of the Electrical and Electronic Engineers in Israel. Proceedings (Cat. No. 00EX377)*, IEEE, 2000, pp. 443–447.

[67] CCSDS 231.0-B-4, TC synchronization and channel coding, Available at http://public.ccsds.org/, Jul. 2021.

[68] C. Fernández-Prades and J. Vilà-Valls, "Bayesian nonlinear filtering using quadrature and cubature rules applied to sensor data fusion for positioning", in *2010 IEEE international conference on communications*, IEEE, 2010, pp. 1–5.

[69] M. L. Psiaki and H. Jung, "Extended Kalman filter methods for tracking weak GPS signals", in *Proceedings of the 15th international technical meeting of the satellite division of the Institute of Navigation (ION GPS 2002)*, 2002, pp. 2539–2553.

[70] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation", in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, Ieee, 2000, pp. 153–158.

[71] M. Martalò, C. Tripodi, and R. Raheli, "On the information rate of phase noise-limited communications", in *Proc. Information Theory and Applications Workshop*, San Diego, CA, USA, 2013, pp. 1–7.

[72] H. Ghozlan and G. Kramer, "Multi-sample receivers increase information rates for Wiener phase noise channels", in *Proc. IEEE Global Telecommun. Conf.*, Atlanta, GA, U.S.A., Dec. 2013, pp. 1897–1902.

[73] H. Ghozlan and G. Kramer, "Phase modulation for discrete-time Wiener phase noise channels with oversampling at high SNR", in *Proc. IEEE International Symposium on Information Theory*, Honolulu, HI, USA, 2014, pp. 1554–1557.

[74] B. Ripani, A. Modenini, and G. Montorsi, "A multi-sample discrete-phase BCJR algorithm for phase noise channels", in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, IEEE, 2022, pp. 6164–6169.

[75] H. Meyr, M. Oerder, and A. Polydoros, "On sampling rate, analog prefiltering, and sufficient statistics for digital receivers", *IEEE Trans. Commun.*, vol. 42, pp. 3208–3214, Dec. 1994.

[76] N. Merhav, G. Kaplan, A. Lapidoth, and S. Shamai, "On information rates for mismatched decoders", *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1953–1967, Nov. 1994.

[77] A. Ganti, A. Lapidoth, and I. E. Telatar, "Mismatched decoding revisited: general alphabets, channels with memory, and the wide-band limit", *IEEE Trans. Inform. Theory*, vol. 46, no. 7, pp. 2315–2328, Nov. 2000.

[78] D. M. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng, "Simulation-based computation of information rates for channels with memory", *IEEE Trans. Inform. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.

[79] A. Modenini, "Advanced transceivers for spectrally-efficient communications", Ph.D. dissertation, University of Parma, Parma, Italy, 2014.

[80] CCSDS 401.0-B-30, Radio frequency and modulation systems, part 1, Available at http://public.ccsds.org/, Feb. 2020.

[81] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes", *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[82] European Space Agency, "CDF study report - Ice Giants", European Space Agency, Tech. Rep., 2019, Available at https://sci.esa.int/web/future-missions-department/-/61307-cdf-study-report-ice-giants.

[83] M. Palomba and J.-P. Luntama, "Vigil: ESA space weather mission in L5", *44th COSPAR Scientific Assembly. Held 16-24 July*, vol. 44, p. 3544, 2022.

[84] D. Turrini, R. Politi, R. Peron, *et al.*, "The ODINUS Mission Concept-The Scientific Case for a Mission to the Ice Giant Planets with Twin Spacecraft to Unveil the History of our Solar System", *arXiv preprint arXiv:1402.2472*, 2014.

[85] A. Modenini and B. Ripani, "A tutorial on the tracking, telemetry, and command (TT&C) for space missions", *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1510–1542, 2023.

[86] M. Baldi, M. Bertinelli, F. Chiaraluce, *et al.*, "State-of-the-art space mission telecommand receivers", *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 6, pp. 4–15, 2017. DOI: 10.1109/MAES.2017.160079.

[87] R. Mehra, "On the identification of variances and adaptive Kalman filtering", *IEEE Transactions on automatic control*, vol. 15, no. 2, pp. 175–184, 1970.

[88] J. Vila-Valls, P. Closas, and C. Fernández-Prades, "On the identifiability of noise statistics and adaptive KF design for robust GNSS carrier tracking", in *2015 IEEE Aerospace Conference*, IEEE, 2015, pp. 1–10.

[89] I. Ali, N. Al-Dhahir, and J. Hershey, "Doppler characterization for LEO satellites", *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 309–313, 1998. DOI: 10.1109/26.662636.

[90] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at Gaussian mixture reduction algorithms", in *14th International Conference on Information Fusion*, 2011, pp. 1–8.

[91] A. D'Ortenzio, C. Manes, and U. Orguner, "A model selection criterion for the mixture reduction problem based on the Kullback - Leibler divergence", in *2022 25th International Conference on Information Fusion (FUSION)*, 2022, pp. 1–8. DOI: 10.23919/FUSION49751.2022.9841270.

[92] D. Schieferdecker and M. F. Huber, "Gaussian mixture reduction via clustering", in *2009 12th International Conference on Information Fusion*, 2009, pp. 1536–1543.

[93] F. Lehmann, "A Gaussian sum approach to blind carrier phase estimation and data detection in turbo coded transmissions", *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2619–2632, 2009. DOI: 10.1109/TCOMM.2009.09.080028.

[94] R. Krishnan, G. Colavolpe, A. Graell i Amat, and T. Eriksson, "Algorithms for joint phase estimation and decoding for mimo systems in the presence of phase noise and quasi-static fading channels", *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3360–3375, 2015. DOI: 10.1109/TSP.2015.2420533.

[95] S. Benedetto, G. Montorsi, A. Perotti, and A. Tarable, "A pragmatic approach to coded continuous-phase modulation", in *2007 Information Theory and Applications Workshop*, 2007, pp. 36–40. DOI: 10.1109/ITA.2007.4357559.

[96] K. V. Mardia, C. C. Taylor, and G. K. Subramaniam, "Protein bioinformatics and mixtures of bivariate von mises distributions for angular data", *Biometrics*, vol. 63, no. 2, pp. 505–512, Nov. 2006, ISSN: 0006-341X. DOI: 10.1111/j.1541-0420.2006.00682.x. eprint: https://academic.oup.com/biometrics/article-pdf/63/2/505/52398953/biometrics_63\

_2_505.pdf. [Online]. Available: https://doi.org/10.1111/j.1541-0420.2006.00682.x.