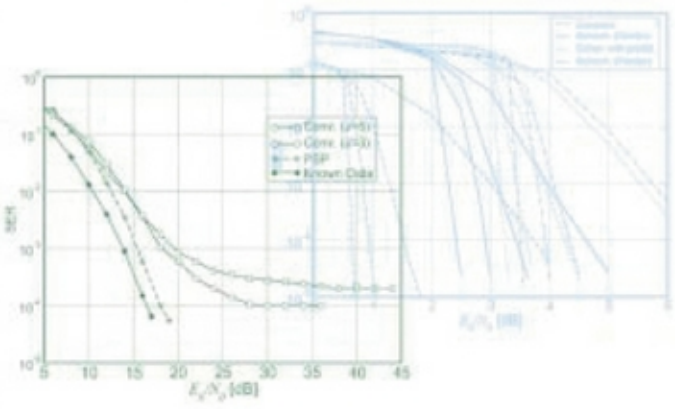
Detection Algorithms

for Wireless Communications

with Applications to Wired and Storage Systems

G. Ferrari | G. Colavolpe | R. Raheli

$$\gamma_k(T_k) = \frac{\mathrm{E}_{\theta_0^k}\{p(\boldsymbol{r}_{k-N}^k | T_k, \theta_0^k)\}}{\mathrm{E}_{\theta_0^{k-1}}\{p(\boldsymbol{r}_{k-N}^{k-1} | S_k, \theta_0^{k-1})\}} P\{a_k\}$$



WILEY

Detection Algorithms for Wireless Communications

With Applications to Wired and Storage Systems

Gianluigi Ferrari, Giulio Colavolpe and Riccardo Raheli
All of
University of Parma
Italy

John Wiley & Sons, Ltd

Detection Algorithms for Wireless Communications

To my parents, Ezio and Ester, for letting me detect my path
Gianluigi Ferrari

To my wife Laura
Giulio Colavolpe

To Annapaola, Enrica and Alberto
Riccardo Raheli

Detection Algorithms for Wireless Communications

With Applications to Wired and Storage Systems

Gianluigi Ferrari, Giulio Colavolpe and Riccardo Raheli
All of
University of Parma
Italy

John Wiley & Sons, Ltd

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-471-85828-1

Typeset by the author using LaTex software.
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Contents

Preface

This book presents, in a unitary and novel perspective, some of the research work the authors have carried out over the last decade, along with several collaborators and students. The roots of this book can be traced back to the design of adaptive sequence detection algorithms for channels with parametric uncertainty. The explosion of turbo codes and iterative decoding around the middle of the Nineties has motivated the design of iterative (turbo and graph-based) detection algorithms.

This book aims at providing the reader with a *unified perspective* on the design of detection algorithms for wireless communications. What does this statement really mean? It has become clear to us, in recent years, that most of the proposed detection algorithms evolve from a simple idea, which can be described as *finite-memory* detection and synthesized by a simple *metric*. This unique metric is the key ingredient to derive:

- sequence detection algorithms based on the Viterbi algorithm;
- symbol detection algorithms based on the forward-backward algorithm;
- graph-based detection algorithms based on the sum-product algorithm.

Although simple, and probably familiar to several researchers working in this area, to the best of our knowledge a unified approach to the design of detection algorithms, based on a single metric, has never been proposed clearly in the literature. This book tries to address this lack, by giving a comprehensive treatment, with several examples of application.

This book should, however, be interpreted by the reader as a starting point, rather than a purely tutorial work. In fact, we believe that the proposed simple unifying idea can find many applications beyond those explored in this book. We would like to mention a single (and significant) example. In current and future wireless communication systems, it will be more and more important to support high data-rate transmissions. Multiple-input multiple-output systems, based on the use of multiple antennas, have received significant interest from the research community over the

last years. All the detection algorithms presented in this book apply to single-input single-output systems. The reader is therefore invited to entertain herself/himself by trying to extend these algorithms to multiple-input multiple-output communication scenarios.

A final comment is related to the subtitle: "With Applications to Wired and Storage Systems." As the reader will see, most of the examples presented in this book are related to wireless communication systems. However, several of the proposed communication scenarios apply also to storage and wired systems: for example, proper inter-symbol interference channels may characterize several storage systems. Moreover, the proposed approach is general and, therefore, suitable for application to scenarios different from those considered explicitly. Again, the reader is invited to use the tools proposed in this book and apply them to solve her/his own communication problems.

As an extra resource we have set up a companion website for our book containing a solutions manual and a sample chapter. Also, for those wishing to use this material for lecturing purposes, electronic versions of most of the figures from our book are available. Please go to the following URL and take a look: ftp://ftp.wiley.co.uk /pub/books/ferrari.

Acknowledgements

This book would have never been possible without the help of several people across the years of research, pain and happiness at the University of Parma (and not only). Since it is impossible to thank explicitly all of them, we would like to take this opportunity to "unitarily" thank all of them, guaranteeing that their help has never been forgotten, but it is well remembered. In particular, we first wish to thank many students at the University of Parma, who have indirectly contributed to this book with their thesis works.

Although explicit and comprehensive acknowledgments are impossible, several researchers must be explicitly mentioned, for particularly significant and important contributions. We would like to thank Prof. Achilleas Anastasopoulos (University of Michigan, Ann Arbor, USA), whose collaboration led to some results presented in Chapter 4. Moreover, we would also like to thank him for kindly proof-reading the entire manuscript. Many thanks go to Prof. Keith M. Chugg (University of Southern California, Los Angeles, USA) and Dr. Phunsak Thiennviboon (TrellisWare Technologies Inc., San Diego, USA), for the collaboration (while Gianluigi Ferrari was visiting the University of Southern California in 2000-2001) from which some results presented in Chapter 4 come from. The invaluable collaboration of Prof. Giuseppe Caire (Institut Eurécom, Sophia Antipolis, France) in part of Chapter 5 is acknowledged. This collaboration started while Giulio Colavolpe was visiting the Institut Eurécom in 2000 and has continued and consolidated in the following years. Thanks to Dr. Alberto Ginesi and Dr. Riccardo De Gaudenzi (ESA-ESTEC, Noordwijk, The Netherlands) for their appreciation and encouragement on some results in Chapter 5. Alan Barbieri (PhD student, University of Parma, Italy) and Gianpietro Germi, are also acknowledged for their contribution to part of Chapter 5. Going backward in time, we would like to acknowledge the contribution of Prof. Andreas Polydoros (University of Athens, Greece) for a long research collaboration which lead to per-survivor processing, described in Chapter 3. We would also like to acknowledge the contribution of Prof. Piero Castoldi (Scuola Superiore Sant'Anna, Pisa, Italy) to some of the results described in Chapter 3 and the Appendix. Finally, we wish to ac-

knowledge the encouragement and support through the years of our senior colleagues Prof. Giorgio Picchi (University of Parma, Italy) and Prof. Giancarlo Prati (Scuola Superiore Sant'Anna, Pisa, Italy), who first appreciated our scientific achievements.

Besides research collaborators, several people have helped in the process of editing the manuscript. Among them, we would like to thank Alessandra De Conti, who read very carefully the entire manuscript, providing extremely valuable comments on the English style. Luca Consolini (PhD student, University of Parma, Italy) is also thanked for reading the entire manuscript and providing comments. We wish also to thank Annuccia Babayan, for proof-reading parts of the manuscript, and Michele Franceschini (PhD student, University of Parma, Italy), for providing useful comments and helping significantly in the editing process. Prof. Enrico Forestieri (Scuola Superiore Sant'Anna, Pisa, Italy) is finally acknowledged for using his vast knowledge of LaTeX to solve a few (unsolvable to us) editing problems and make the manuscript more compliant with the requests of the Publisher.

Last, but not least, we heartly wish to thank several people at John Wiley and Sons Ltd, who made the realization of this book possible. First of all, we are greatly indebted to our Development Editor, Sarah Hinton, who first believed in this project and promoted it. Our gratitude goes also to our Publishing Editor, Mark Hammond, who supported the project along its entire realization. Finally, the Project Editor, Dan Gill, is thanked for managing very efficiently the editorial phase of the manuscript and the Copyeditor, Helen Heyes, is also thanked for providing extremely detailed corrections to the final version of the manuscript.

List of Figures

List of Tables

1

Wireless Communication Systems

1.1 Introduction

In the era of Information and Communication Technology, we are all familiar with the concept of digital transmission of information. The adjective "digital" encompasses the fact that the transmitted message is one out of a finite number of possible messages. In more practical terms, a digital message is composed of a sequence of digits, or *symbols*, belonging to a finite *alphabet*. This digital model of information arises for a number of reasons. There exist information *sources* which inherently generate digital messages. A written text or a computer memory are simple examples of sources of digital information. There also exist, however, information sources that inherently generate *analog* messages, such as voice and sounds. These analog messages can be approximated with the desired degree of accuracy in terms of digital messages by means of sampling and quantization.

The goal of any digital transmission system is the delivery of a sequence of digital information symbols to a *destination*. The transmission of a digital information symbol, or transmission *act*, takes place by selecting an analog signal, or *waveform*, out of a finite set of possible signals and sending this waveform through a transmission medium, or *channel*, which connects the *transmitter* to a *receiver* located at the destination.

Wireless communications will have an increasing importance for future applications, where connectivity will be required everywhere and anyhow. In particular, the large flow of information coming from the optical communication backbone calls for wireless systems able to support large information transmission capacity. It is

Detection Algorithms for Wireless Communications - G. Ferrari, G. Colavolpe and R. Raheli
 ISBN: 0-470-85828-1

becoming more and more important to push to the limit wireless access and distribution, since larger and larger quantities of information need to be carried. For instance, in third generation (3G) mobile communication systems [1], multimedia data will be transferred and this will call for effective and efficient wireless communication systems. In particular, increased signal processing capabilities for lower cost will allow the deployment of advanced *detection algorithms* in the receivers of wireless terminals. Several wireless communication systems are currently operational, and a few significant examples will be summarized in the following.

In passing through a wireless channel, the transmitted waveform is altered in several possible ways. A primary alteration is due to thermal noise which is superimposed on the transmitted waveform and effectively modeled as an additive Gaussian random process. There is, however, a plethora of other possible modifications which may be inflicted upon the signal by the transmission *link*. In radio transmission, the amplitude of the received signal, i.e., the signal at the output of the channel, is a function of the link attenuation, which in turn may depend on the physical link geometry, such as the distance between transmitter and receiver antennas and their heights, but it may also depend on the simultaneous presence of several propagation paths, or *multipath*, which may contribute constructively or destructively. As a result, the (possibly time-varying) signal amplitude may not be perfectly known by the receiver, which must adopt suitable countermeasures. In bandpass transmission, similar considerations hold for the phase of the received signal, which is usually unknown at the receiver. When the amount of variability of multipath phenomena over space or time is significant, the received signal may be characterized by unknown amplitude and phase, which collectively manifest themselves as multiplicative noise termed *fading*.

In general terms, the transformation inflicted by a channel upon the transmitted waveforms must be suitably modeled. This modeling is needed in order to enable the definition of an effective *strategy* to be undertaken by the receiver in order to decide which message was actually transmitted on the basis of the signal actually received. This decision process, or *detection*, should obviously result in a minimal number of decision errors. By adopting a random model for both the information source and the transmission channel, the detection strategy can be optimized according to a meaningful criterion, for example to minimize the probability of decision error. A receiver operating in accordance with such a strategy is said to be *optimal*.

There are several important applications in which random channel models are useful for devising optimal receivers characterized by affordable implementation complexity, among which is the classical additive white Gaussian noise (AWGN) channel. There are, however, applications where the implementation complexity of an optimal receiver becomes prohibitive from a practical viewpoint. The simple presence of a random *time-invariant* channel parameter, such as a phase rotation or a mul-

tiplicative fading coefficient inflicted upon the received signal, may significantly increase the complexity of the optimal receiver. *Time-varying* random channel parameters may also be responsible for the intractable complexity of an optimal receiver. In present-day communication systems, examples of such time-varying random channel models arise to describe the effects of fading due to multiple transmission paths, phase noise due to the inherent instabilities of up- and down-conversion oscillators, or Doppler shift encountered in low- and medium-earth satellite communications.

It might be sometimes necessary to model some channel parameters as unknown deterministic quantities. This modeling assumption has the desirable consequence of reducing the receiver complexity, by means of a *decomposition* approach. In this case, the receiver could be designed as the concatenation of a detection block, devised under the idealizing assumption of perfect knowledge of these deterministic channel parameters, and one or more estimation blocks devoted to the acquisition of the necessary information about the channel parameters. Nonetheless, this modeling assumption has the disadvantage that the concept of optimality, in terms of minimal error probability, vanishes because an optimal detection strategy can only be defined for specific known cases of deterministic channel parameters [2].

In the rest of the book, the focus will be mainly on channels characterized by *stochastic parameters*. In particular, detection algorithms will be designed taking into account a statistical description of the parameters, e.g., by means of their joint probability density function (pdf). This type of approach is usually referred to as Bayesian. Since taking into account the stochastic nature of the transmission channel may lead to a significant increase of the complexity of the detection algorithm, suitable complexity reduction techniques will also be considered. It is also important to underline that, for the design of a detection algorithm, it is sometimes expedient to consider a simplified channel model, with respect to the effective transmission channel.

In this initial chapter, we will also present a brief overview of the concepts of *modulation* and *coding*, providing a few significant examples for wireless communication scenarios. The analysis is by no means complete, since it goes beyond the scope of this book. However, the proposed set of examples should provide the reader with a basic understanding of the concepts and schemes which will be considered in the following chapters.

Modulation can be, as a first instance, separated by coding. It is somehow possible to state that modulation comes conceptually before coding, in the sense that it represents the way in which a signal needs to be *shaped* before undergoing channel transmission. In particular, in the remainder of the book we will focus on *digital* modulations, where the information to be transmitted belongs to a suitable set of finite cardinality. In this case, considering a suitable set of orthonormal functions

it is possible to represent a modulated signal as a point in a constellation of finite cardinality.

After the breakthrough brought by the mathematical theory of communication developed by Shannon [3], it has been clear that coding the information to be transmitted can make the communication more efficient and reduce the probability of erroneous reception at the receiver side. In particular, the main application areas regard *source coding* and *channel coding* [4–7]. While in the first case the main principle is that of compressing the source information still guaranteeing complete recovery, the main idea in the second case (of interest in this book) consists of adding redundancy to the generated information in order to increase the protection against channel impairments.

While modulation and coding can be considered separate as a first instance, seminal work in the Seventies [8] culminated with the clear description of trellis coded modulation (TCM) [9], which showed that modulation and coding should be regarded as two faces of the same coin. This is intuitive, since once the information to be transmitted is suitably coded, the way in which this coded information is actually transmitted over the channel is strictly related to the considered modulation. In the remainder of this chapter, we will review some basic concepts, relevant for the detection algorithms for wireless communications derived in the following chapters, such as the *capacity* of a transmission channel and some interesting types of modulation and coding formats. Note that the focus, rather than on standard coding and modulation techniques, will rather lie on interesting schemes to which the detection algorithms presented in the following chapters can be suitably applied.

1.2 Overview of Wireless Communication Systems

There are various examples of wireless communication systems. We consider here a simple summary of a few communication systems for which the detection algorithms discussed in the remainder of this book might have significant impact.

- *Broadband radio access networks* are gaining more and more interest, since radio traffic is increasingly based on multimedia applications, requiring significant transmission capabilities. Point to multipoint radio access is an important example of this radio communication paradigm.

- *Cellular radio networks* represent the current structure devoted to personal wireless communications [10]. They are based on the subdivision of territory into cells, with a base station placed in each cell and acting as a "reservoir" for transmissions coming from mobile terminals in the cell.

- *Satellite communication systems* are widely used to guarantee ubiquitous connectivity and to allow transfer of information between distant locations [11]. In particular, mobile and satellite communication systems could merge, in the near future, to guarantee ubiquitous coverage.

- *Wireless local area networks* (such as IEEE 802.11 [12]) represent a new communication paradigm which is under continuous development. Its applications are currently limited to multiple access in limited areas, such as university campuses, airports, and buildings. However, novel scenarios with extended coverage are emerging.

- *Personal area networks* are currently becoming more and more popular, with multiple diverse applications. The network communication system nicknamed "bluetooth" is a de facto standard which is developing to guarantee the intercommunication between digital devices in very limited areas [13]. This is the case, for example, for communication among electronic appliances (computers, video cameras, stereo systems) within a room.

- *Mobile ad hoc wireless networks* are emerging as a new communication paradigm which could have numerous applications in the near future [14, 15]. In particular, the fundamental idea behind this type of network communication is the fact that radio communication should be supported by a nonhierarchical architecture as long as the spatial distribution of the communicating terminals is sufficiently dense.

1.3 Wireless Channel Models

In general, radio communications are affected by *large (time) scale* path loss (e.g., free space propagation model) and *small scale* phenomena [16]. This book will focus on small scale phenomena. In the following, we will consider a few significant examples of communication channels which are relevant for the analysis of wireless communication systems. Various combinations of the proposed channel models may also be of interest. The collection of examples is by no means complete, but the aim of the book is to develop a general framework for the design of detection algorithms to be used for transmissions over wireless channels. The interested reader is invited to use the tools presented in the following chapters to derive new detection strategies for other channel models not considered explicitly in this book. Almost all considered models will correspond to the base-band equivalent of the effective transmission channels [17]. Moreover, the derived detection algorithms will almost al-

ways stem from discrete-time equivalent versions of the transmission channels—the equivalence, in terms of statistical sufficiency, of the obtained discrete-time observables will always be guaranteed, possibly by means of oversampling. Appendix A is devoted to a characterization of sampling as a means to derive sufficient statistics.

1.3.1 Additive White Gaussian Noise Channel

The basic communication channel is the AWGN channel, shown in Figure 1.1(a). In particular, the transmitted signal $s(t)$ is corrupted by the additive noise $n(t)$, yielding the received signal $r(t)$, which can be written as follows:

$$r(t) = s(t) + n(t). \tag{1.1}$$

The noise process $n(t)$ is usually associated with the ubiquitous thermal noise at the input of the receiver [16–18]. In particular, the assumption of AWGN means that $n(t)$ is a circularly symmetric Gaussian process with constant power spectral density.

1.3.2 Frequency Nonselective Fading Channel

In this case, the transmitted signal is distorted by multiplicative noise $f(t)$ modeled as a *circularly symmetric* complex Gaussian random process, yielding the following expression for the received signal:

$$r(t) = f(t)s(t) + n(t). \tag{1.2}$$

We recall that a circularly symmetric Gaussian complex random variable is characterized by independent and identically distributed (iid) Gaussian real and imaginary components [19]. If $f(t)$ is zero mean, then the fading amplitude $|f(t)|$ and phase $\arg\{f(t)\}$ are Rayleigh and uniformly distributed, respectively. In the frequency domain, this channel can be interpreted as a time-varying frequency nonselective (*flat*) linear filter. Figure 1.1(b) shows this channel model. If the bandwidth of $f(t)$ is significantly narrower than that of the transmitted signal, its time variations can be neglected and a *slow fading* model results. As an extension of this model, in the case of a Rice fading channel, $f(t)$ has nonzero mean.

In order to describe the "memory" introduced by the channel it is expedient to consider the autocorrelation function,[1] usually characterized by the Clarke model [20, 21]. The Clarke model assumes a fixed transmitter with a vertically polarized and isotropic antenna [16], and uniform scattering around the mobile terminal—for this

[1]This is meaningful in the case of circularly symmetric Gaussian fading with zero mean.

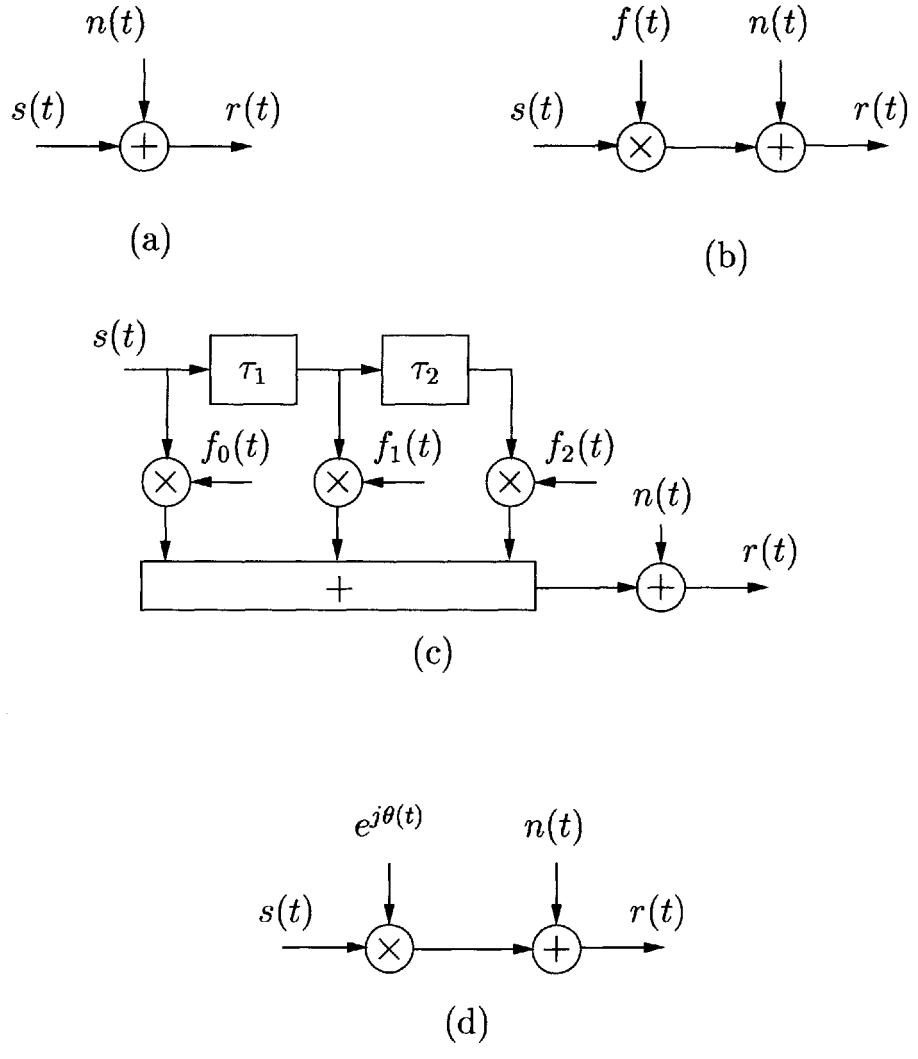


Figure 1.1: Examples of wireless channel models.

reason, this model is usually referred to as the *isotropic scattering model.* In particular, the statistical description of the received signal depends on (uniform) scattering phenomena experienced by the transmitted waveform: under the assumption of the absence of a direct line-of-sight component, the envelope of the received signal has a Rayleigh distribution. In [22], a spectral characterization of Clarke's model is stud-

ied. The power spectral density of the received signal can be written as follows:

$$S_C(f) = \begin{cases} \frac{1}{\pi f_D \sqrt{1-\left(\frac{f-f_c}{f_D}\right)^2}} & |f - f_c| \leq f_D \\ 0 & |f - f_c| > f_D \end{cases} \tag{1.3}$$

where f_D is the maximum Doppler shift and f_c is the carrier frequency. Symbol rate sampling may not be accurate in the case of large Doppler spread, in which case oversampling, as shown in Appendix A, becomes necessary. In particular, the auto-covariance function can be written as $R_C(\tau) = J_0(2\pi f_D \tau)$, where $J_0(\cdot)$ is the Bessel function of order zero [23]. For large values of τ the Bessel function approaches zero, so that one can conclude that two fading samples temporally distant from each other are uncorrelated. This idea of uncorrelatedness between observations distant in time will be one of the basic ideas behind all detection strategies proposed in the rest of the book. In fact, owing to the intuitive idea that observations far from the one at hand cannot carry too much useful information, a suitable *finite memory condition* (FMC) can often be considered without significant performance degradation.

1.3.3 Frequency Selective Fading Channel

The effect of this channel can be described by assuming that the transmitted signal is distorted by a time-varying tapped-delay-line filter with tap delays $\{\tau_\ell\}$ and weight processes $\{f_\ell(t)\}$, each modeled as in Section 1.3.2. In other words, indicating with L the number of taps in the equivalent tapped-delay-line model, the received signal can be expressed as follows:

$$r(t) = \sum_{\ell=0}^{L-1} f_\ell(t) s(t - \tau'_\ell) + n(t) \tag{1.4}$$

where

$$\tau'_\ell \triangleq \sum_{\ell=1}^{l} \tau_i. \tag{1.5}$$

Note that we are implicitly assuming that $\tau_0 = 0$. In the frequency , this channel can be interpreted as a time-varying frequency selective (dispersive) linear filter. Figure 1.1(c) shows a block diagram of this channel model with $L = 3$ taps. In general, the i-th tap corresponds to a dominant propagation path with delay $\tau_1 + \tau_2 + \ldots + \tau_i$.

The frequency selective fading channel can then be interpreted as a time-variant *inter-symbol interference* (ISI) channel. *En passant*, we remark that ISI channels

represent a very important communication channel model, which can arise for various reasons (e.g., mismatch between transmit and receive filters). In particular, a wired communication channel or a magnetic storage channel are usually modeled as ISI channels. We point out that although the book's main topic is wireless communications, many of the proposed detection algorithms—in particular those related to ISI channels—can also be applied to wired communications and magnetic or optical storage systems. Simplified models for fading channels are often desirable to allow a more efficient derivation of detection algorithms. As an example, Gaussian quadrature rule (GQR) channel models can be used [24]. From an implementation viewpoint, the applicability of these models is limited at high transmission data rates: in fact, in this regime, the channel becomes slowly varying, and selectivity arises only in the frequency domain.

1.3.4 Phase-Uncertain Channel: Channel with Phase and Frequency Instabilities

In this case, the communication channel introduces phase noise, generating phase rotation in the transmitted signal. We assume that the phase of the transmitted signal is rotated by a stochastic process $\theta(t)$ in order to model various instabilities of up- and down-conversion oscillators or a possibly time-varying Doppler shift (e.g., typical of low- and medium-orbit satellite systems). The received signal assumes then the following expression:

$$r(t) = e^{j\theta(t)}s(t) + n(t). \tag{1.6}$$

Figure 1.1(d) shows a block diagram of this channel model. Depending on the assumptions on $\theta(t)$, the following special cases can be distinguished:

1. a *phase noncoherent channel* can be modeled by assuming that $\theta(t) = \theta$ is a uniformly distributed random variable (i.e., constant with respect to time);

2. a *frequency offset* (or *Doppler shift*) *channel* is obtained for $\theta(t) = 2\pi\nu t$, where ν is the Doppler rate;

3. a *phase noisy channel* is modeled by assuming $\theta(t)$ is a Wiener random process [19].

Combinations of these special cases are also possible. Note that considering the channel phase as a random variable, i.e., considering a noncoherent channel, can also be used as a useful assumption to derive detection algorithms. This channel

model will often be considered in the remainder of the book—a channel with a single stochastic parameter (for instance the phase) represents indeed a very attractive study case.

In this case as well, we want to initially explore the correlation characteristics among observations at different epochs, in order to motivate the approach, based on an FMC, proposed to perform detection of a signal transmitted over a phase-noisy channel. In particular, in the presence of linear modulation and in the absence of frequency offset and ISI, an equivalent discrete-time version, relative to sampling instants corresponding to multiples of the symbol period, of the received signal in (1.6) becomes:

$$r_k = c_k e^{j\theta_k} + n_k \tag{1.7}$$

where c_k is the transmitted (and possibly coded) symbol, $\{\theta_k\}$ is the discrete-time channel phase process, and $\{n_k\}$ is the additive noise process. A very common model for the phase process $\{\theta_k\}$ is a discrete-time Wiener process [19] described by the following recursion

$$\theta_k = \theta_{k-1} + \Delta_k \tag{1.8}$$

where $\{\Delta_k\}$ are iid Gaussian random variables with zero mean and variance σ_Δ^2. In this case, it can be shown (see Problem 1.2) that the autocovariance of the process $\{e^{j\theta_k}\}$ is

$$R_\theta(n) = \mathrm{E}\left\{e^{j\theta_{k+n}} e^{-j\theta_k}\right\} = \exp\left(-\frac{|n|\sigma_\Delta^2}{2}\right). \tag{1.9}$$

It is immediately realized that for large values of n, i.e., considering observations temporally distant from each other, the autocorrelation is very low. This motivates a finite memory detection approach, which exploits the idea that, in order to derive an explicit (or implicit) estimate of the channel parameter at epoch k, it is sufficient to consider a limited ensemble of observations at instants close to k.

Several other models may be of interest in practical applications, among which are channels affected by like-signal (or co-channel) interference and nonlinear channels modeling the saturation effects of high-power amplifiers [25, 26].

1.4 Demodulation, Detection, and Parameter Estimation

The received signal is the *observation* used by the receiver for decoding the transmitted information sequence. In Figure 1.2, a single block is used to represent a

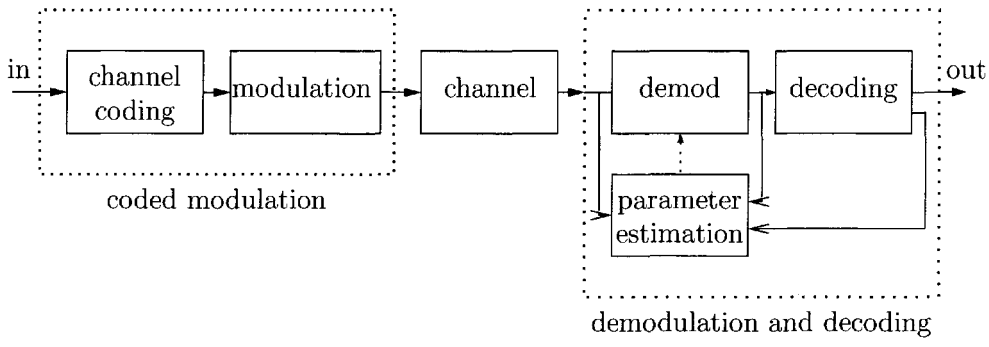


Figure 1.2: Classical model of a communication system.

combined demodulation and decoding process. Inside this block, the subblocks indicate a possible decomposition of the various receiver functions. This decomposition deserves attention because it is a traditional and conceptually straightforward choice in the fact that basic receiver functionalities are implemented by corresponding subsystems. Specifically, a demodulation (or detection) block performs the inverse of the modulation function under the assumption that the necessary channel parameters are known to a certain accuracy. These parameters are estimated by a specific block which observes the available information, namely the received and demodulated signals and, possibly, the decoded data. Finally, the decoding function is performed on the basis of the demodulated (or detected) signal.

It is important to point out that the separation between these receiver functions, though traditional, is somehow limiting. In fact, the advent of coding techniques such as TCM and continuous phase modulation (CPM) [27–30], and decoding/detection techniques such as iterative decoding/detection techniques [31–33], has changed the way of approaching system design. In particular, the receiver functions of demodulation, decoding, and parameter estimation tend to be *combined* in contemporary wireless communication systems. In other words, a detection algorithm can be designed by embedding parameter estimation in the decoding process. This idea, which is the core of the principle of *per-survivor processing* (PSP) [34], will be used diffusely in the remainder of the book and in the derived detection algorithms. Moreover, the receiver functions indicated in Figure 1.2—i.e., demodulation, decoding, and parameter estimation—should not be seen as isolated. The basic principle of iterative detection makes it clear, indeed, that the exchange of information along with the decoding/detection process can significantly enhance the receiver performance.

1.5 Information Theoretic Limits

After a clear modeling of a communication system, Shannon went on and introduced the concept of *capacity* of a single-input single-output communication channel [3, 35]. A possible definition for the channel capacity is the maximum of the mutual information between the input and the output of the transmission channel. Although this idea is intuitive (a channel can reliably carry a maximum information flow which ultimately depends on the channel itself), quantifying it with precise formulas in specific cases was one of Shannon's breakthroughs of paramount importance.

1.5.1 Additive White Gaussian Noise Channel

In the case of an AWGN channel, the work of Shannon [3] shows that the channel capacity can be written as follows:

$$C_{\text{AWGN}} = \log_2(1 + \Xi) \quad \text{[bits/s/Hz]} \tag{1.10}$$

where Ξ represents the signal-to-noise ratio (SNR), defined as $P_{\text{s}}/P_{\text{n}}$, where P_{s} is the received signal power and P_{n} is the AWGN noise power. In particular, the capacity C_{AWGN} of a generic single-input single-output channel can also be interpreted in the following way, with profound implications in the research effort of the coding community after the work of Shannon. For any coding rate $r_{\text{c}} < C_{\text{AWGN}}$, there exists an (n, k) (where k is the length of the input information word, n is the length of the codeword and $r_{\text{c}} \triangleq k/n$) code, with suitably long codeword length n, such that the error probability, with maximum likelihood (ML) decoding, can approach zero as closely as desired.

1.5.2 Frequency Nonselective Fading Channel

The computation of the capacity of a channel affected by fading is, in a general case where the fading is time-varying, extremely difficult. By relaxing some of the assumptions regarding the fading correlation characteristics, it is possible, however, to derive a closed-form expression for the channel capacity.

Considering an equivalent discrete-time communication model with linear coded modulation at the transmitter side and absence of ISI, the observable at time epoch k can be written as follows:

$$r_k = f_k x_k + n_k \tag{1.11}$$

where f_k is a discrete-time Rayleigh fading coefficient and x_k is the transmitted symbol.[2] If the process $\{f_k\}$ is given by a sequence of iid variables (i.e., the channel is memoryless or, equivalently, ideal channel interleaving is considered) and perfect channel state information (CSI) is given to the receiver, it is possible to show that the capacity is [36, 37]

$$\begin{aligned} C_{\text{Rayleigh-i}} &= \mathbf{E}_{f_k}\left[\ln\left(1+\Xi|f_k|^2\right)\right] \\ &= -\exp\left(\frac{1}{\Xi}\right) E_i\left(-\frac{1}{\Xi}\right) \quad \text{[nats/s/Hz]} \end{aligned} \tag{1.12}$$

where $\Xi = P_{\text{s}}/P_{\text{n}}$, with P_{s} corresponding to the power of the transmitted symbol x_k and P_{n} corresponding to the AWGN noise power, and

$$E_i(x) \triangleq \int_{-\infty}^{x} \frac{e^t}{t}\,\mathrm{d}t. \tag{1.13}$$

The capacity of the channel is reached when x_k is circularly symmetric complex Gaussian with zero mean and variance P_{n}. The extension of this result to the case where no CSI is available at the receiver is the subject of [38], whereas the case where CSI is available at both receiver and transmitter is considered in [39, 40]. An excellent overview of information-theoretic and communication-theoretic aspects of fading channels can be found in [41].

The capacity of the single-input single-output Rayleigh fading channel with CSI at the receiver can be interpreted as a particular case of the general result, relative to multiple-input multiple-output Gaussian channels, obtained in [42]. The analysis of the capacity of a multiantenna channel in the absence of CSI at the receiver, as opposed to the scenario considered in [42], is the subject of [43]. Note that the expression of the capacity in (1.12) can also be interpreted as a special case (relative to a single-input single-output channel) of the *perfect-knowledge upper bound* given in [43].

This book will mainly consider single-input single-output communication systems. However, we point out that the proposed general approach to performing detection over channels with memory can be extended to a more general situation where multiple antennas are used.

[2]Note that in the remainder of the book the symbol c_k will often be used to refer to coded/modulated symbols. In this case, the transmitted symbol x_k is not necessarily constrained to belong to a constellation with a fi nite number of points.

1.5.3 Phase-Uncertain Channel

While it is possible to compute the channel capacity in closed form in the case of simple channels, in several cases, corresponding to possible wireless channels, this is not possible, but the capacity can rather be bounded or numerically evaluated. As an example, in the case of transmission over a channel introducing a time-invariant random phase rotation, i.e., a noncoherent channel, the observation at the receiver can be written as

$$r_k = x_k e^{j\theta} + n_k \tag{1.14}$$

where x_k is the transmitted symbol, θ is a time-invariant random phase rotation introduced by the channel, and n_k is a complex AWGN sample with variance P_{n}. The input to the channel is modeled as a vector $\boldsymbol{x} \triangleq (x_1, x_2, \ldots, x_N)^{\mathrm{T}}$ of N complex symbols and the output is a vector $\boldsymbol{r} \triangleq (r_1, r_2, \ldots, r_N)^{\mathrm{T}}$. Based on these definitions, the channel capacity, indicated as C_{nc}, can be lower bounded as follows [44]:

$$\begin{aligned} C_{\mathrm{nc}} \;\geq\; & C_{\mathrm{AWGN}} - 2\,\Xi \log_2 e \\ & + \frac{1}{N} \int_{\mathbb{C}^N} \int_{\mathbb{C}^N} p(\boldsymbol{r}|\boldsymbol{x}) p(\boldsymbol{x}) \log_2 \mathrm{I}_0\!\left(\frac{|\boldsymbol{r}^{\mathrm{T}} \boldsymbol{x}^*|}{P_{\mathrm{n}}} \right) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{r} \quad \text{[bits/s/Hz]} \end{aligned} \tag{1.15}$$

where Ξ represents the SNR per information symbol and is defined as follows:

$$\Xi \triangleq \frac{\mathrm{E}\{||\boldsymbol{x}||^2\}}{\mathrm{E}\{||\boldsymbol{x}||^2\}} = \frac{\mathrm{E}\{||\boldsymbol{x}||^2\}}{2NP_{\mathrm{n}}} \tag{1.16}$$

where $||\boldsymbol{x}||^2 \triangleq |x_1|^2 + \ldots + |x_N|^2$ and $||\boldsymbol{r}||^2 \triangleq |r_1|^2 + \ldots + |r_N|^2$. The integral in (1.15) may be numerically computed and the result is shown in Figure 1.3 for various values of the parameter N. It is immediately recognized that for relatively few observations ($N = 30$) the capacity of this channel approaches that of an AWGN channel. This is in agreement with the experience which says that the limit represented by the performance of an ideal coherent receiver may be reached, for constant or slowly varying channel phase and continuous transmissions, with practical pseudocoherent[3] or noncoherent schemes [45–47]. This also motivates the study of detection strategies for this channel in the following chapters, since there is a clear performance benchmark which can be theoretically achieved. Note that the capacity of a blockwise noncoherent channel was also studied, in special cases, in [48–50].

[3] In a *pseudocoherent* receiver for noncoherent communications, a phase estimate is first generated, and then used as if it were the correct phase.

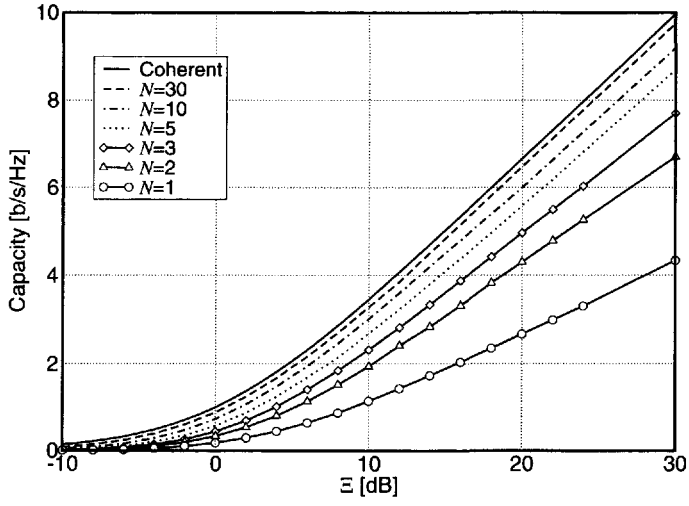


Figure 1.3: Lower bound on the noncoherent channel capacity for different values of N. Reproduced from [44] with permission of John Wiley & Sons.

1.6 Coding and Modulation

We briefly outline some basic concepts relative to coding and modulation, relevant for transmissions over wireless channels.

1.6.1 Block and Convolutional Coding

Since the early work of Shannon proved the existence of error correction codes guaranteeing error-free communication for code rates lower than the capacity, many coding techniques have been developed to fulfill this prediction. In particular, the idea of channel coding is that of introducing redundancy to protect against the noise introduced by the channel.

A binary *block code* is such that an information word, i.e., a sequence of information bits, is mapped into a codeword, i.e., another sequence of bits in number larger than the number of information bits. The ensemble of the codewords can be depicted in a Hamming space of suitable dimension, where each codeword lies on a particular point of this discrete coordinate system. The performance of the block code depends ultimately on the distribution of the codewords in this space, and on the relative distances between them. In this case, the *Hamming distance* between two codewords is defined as the number of bits in which the two codewords differ. A compact parameter describing the asymptotic performance, for large SNR, is the *minimum* (or *free*)

distance between any possible pair of codewords [4, 7].

A *convolutional code* can be instead considered as generated by a convolutional encoder, where this term indicates that the output sequence can be seen as the *convolution* of the input information sequence with the impulse response of the encoder [7, 51–53]. In detail, a convolutional encoder can be usually represented as a time-invariant finite state machine (FSM), with state μ_k belonging to a set of finite cardinality. In the case of a convolutional code with rate k/n, this FSM is characterized by two basic *time-invariant* functions:[4]

- a *next-state* function $\mathrm{ns}(\mu_m, \boldsymbol{a}_m)$, whose output represents the state μ_{m+1} at epoch $m+1$ to which the encoder evolves from state μ_m at epoch m upon the arrival of a suitable sequence of k information symbols, concisely indicated as $\boldsymbol{a}_m$;
- an *output* function $\mathrm{o}(\mu_m, \boldsymbol{a}_m)$, whose output represents a vector $\boldsymbol{c}_m$ of n coded bits.

Since a convolutional encoder can be described as an FSM, a useful and simple description of its evolution is obtained by considering a suitable *trellis* diagram. The structure of the branches connecting trellis states at consecutive epochs is determined by the code. In the case of a convolutional code, it is possible to show (the idea is intuitive but a rigorous proof is not trivial) that the performance depends on the "weight," defined as the number of 1s among the coded bits, of the shortest trellis path diverging from the all-0 path [56].

Note that the symbols n and k used for a convolutional code have a different meaning compared to the case when they are used for a block code. In fact, in the case of a block code, k indicates the entire length of the information message, which is then transformed, through a linear matrix operation, in a coded sequence with n symbols. In the case of convolutional coding, the information symbols enter, in groups of k, into the convolutional encoder, which generates n corresponding coded symbols. The two categories of code, block and convolutional, are comparable in terms of coding rate $r_c = k/n$. For more details, see many of the available books on coding [4–7].

1.6.2 Linear Modulation Without Memory

Several modulation formats for digital transmission over radio channels have been studied. In particular, we refer the interested reader to the ample literature available

[4] As shown in [54, 55], a block encoding process can be described over a *time-varying* trellis diagram, in which case the next-state and output functions depend on the particular epoch.

on the subject (see [17] and references therein). In the remainder of this book, the following main linear modulation formats without memory will be considered:

- *Phase Shift Keying* (PSK). In this case, the information to be transmitted is embedded in the phase of the transmitted signal.
- *Quadrature Amplitude Modulation* (QAM). In this case, the information to be transmitted is embedded in the amplitude and phase of the signal to be transmitted.

Other types of modulation, such as CPM, are not linear and represent an example of modulation with memory. In this sense, they can be considered as coded modulation. Hence, this type of modulation is considered in the following subsection.

1.6.3 Combined Coding and Modulation

The idea that coding should be combined with modulation is *intuitive* and *very subtle*. It is intuitive, since the generated information is processed through the concatenation between coder and modulator, so it seems natural that they constitute, combined, the transmitter. However, the impact of this combination in terms of spectral and energy efficiency is not immediately understood. As previously mentioned, the first clear intuition is due to Massey in 1974 [8], but the idea was clearly formalized in the 1982 milestone paper by Ungerboeck [9], where the concept and principles of TCM are presented [57]. We summarize below a few interesting coded modulation formats. Some of them will be considered in the following chapters. Other formats, recently proposed, represent possible attractive formats to which the proposed detection algorithms could be extended.

- *Trellis coded modulation.* The designed codes are binary trellis codes, whose output is mapped into modulated symbols of suitable dimensions. Increasing the cardinality of the output modulation leads to a degradation of the performance, in terms of bit error rate (BER). However, careful assignment of the trellis transitions to coded symbols determines an increase of the minimum *Euclidean* distance—which represents the natural extension, for modulated signals, of the Hamming distance for binary signals—in such a way that the overall performance may improve. The concept of TCM has been extended to *multidimensional* TCM [57].
- *Continuous phase modulation.* The characteristics of this signaling technique were first clearly formalized by Aulin and Sundberg in the early Eighties [27–29]. The information is embedded in the transmitted signal phase, which has

to be continuous. The continuity of the phase signal determines a very compact power spectral density. Moreover, this modulation is very attractive, since the envelope of the modulated signal is constant. These reasons motivated the choice of a CPM signal, more precisely Gaussian minimum shift keying (GMSK), as the modulation format for the pan-European Group Special Mobile (GSM) communication system [58]. It has been shown that a CPM signal can be decomposed into the sum of several pulse amplitude modulated (PAM) signals [59, 60]. This suggests a receiver structure based on a bank of filters matched to the linear components of the CPM signal. It was also shown that any CPM modulator can be decomposed into the sequence of a continuous phase encoder (CPE), given by a recursive rate $1/n$ binary encoder, and a memoryless mapper (MM) [61]. Therefore, CPM can be considered as a special case of coded linear modulation. Instead of considering a bank of filters relative to the PAM decomposition of the CPM signal, it is possible to simply consider a single filter matched to the MM and then consider a decoder relative to the recursive CPE.

- *Multilevel coding* (MC). Originally introduced in the Seventies [62], the idea is that of protecting each bit of a high-order constellation symbol through the use of (block) codes. This coded modulation scheme is very attractive, since it allows one to approach the ultimate performance limits predicted by Shannon. The design rules for the various subcodes of a multilevel code have been recently listed and described in a clear way [63]. These rules, which basically make coding for each bit independent of the final high-order modulation format, show somehow that the Ungerboeck paradigm (which states that coding and modulation should be designed jointly to maximize the communication system performance) can be practically overcome.

- *Bit interleaved coded modulation* (BICM). This coded modulation, first introduced in [64] and studied in detail in [65], is very effective for transmission over fading channels. The key idea is that of considering channel interleaving on the coded bits before mapping these bits on suitable high-order modulated symbols. In this case as well, coding and modulation can be designed separately, even though they are eventually combined.

1.7 Approaching Shannon Limits: Turbo Codes and Low Density Parity Check Codes

Originally introduced in 1993 by Berrou, Glavieux, and Thitmajshima [33, 66], the ideas of *turbo coding*, and above all *iterative decoding*, represent the major breakthrough in communication and information theory of (at least) the last decade [67]. The main contribution of this work is to clearly indicate that *concatenated* codes, originally introduced in [68], with particular structures, can be decoded in a simple, suboptimal way entailing minor performance loss with respect to an optimal symbol or sequence maximum *a posteriori* (MAP) decoding. A classical turbo code is constituted by a parallel concatenation of two component convolutional codes. At the receiver side, there are two soft-input soft-output (SISO) modules, corresponding to the two component codes, which exchange information refining their decisions. Provided that the SNR at the input of the receiver is above a critical threshold, the iterative process converges.

The idea of iterative decoding opened a "highway" in terms of research activity [69–72]. Various blocks of a communication system can "talk" to each other in order to refine the quality of knowledge they hold about suitable quantities of the communication system under consideration. In particular, an immediate extension of the original work in [33, 66] consists of observing that the channel can be considered as a particular block: this led to what is referred to as "turbo-equalization" or, more correctly, "turbo-detection" [73, 74].

The characteristics of turbo codes, and more generally of concatenated (either serially or in parallel) codes, have been widely considered in the literature. It has been pointed out that these codes show remarkable performance for an SNR larger than a value in correspondence with which the BER curve has a characteristic "knee:" more precisely, the BER curve is very steep above the knee value and exhibits a waterfall behavior, whereas for lower values of the BER the curve flattens and a characteristic "floor" appears [70]. This is due to the fact that the minimum distance of a turbo code is relatively small. Hence, various researchers tried to understand what the characteristic of a turbo code is, in terms of the distance spectrum, which determines such performance. This characteristic was soon recognized as the *spectral thinning*, i.e., a distance spectrum where the multiplicities of the low weight codewords are limited [75]. In other words, at low SNR the whole distance spectrum, and not only the minimum distance, is relevant. In [76], it is shown that, in the case of linear block codes, maximization of the minimum distance is expected to guarantee optimal performance above the channel cut-off rate. It is still an open question to understand which "aspect" of the distance spectrum is significant in this SNR region (i.e., below

the cut-off rate rate SNR).

The success of iterative decoding, based on the exchange of soft information between *blocks*, revived the research activity on low density parity check (LDPC) codes. In particular, in 1962 Gallager, besides inventing these codes, suggested clearly in his PhD thesis a suboptimal iterative processing, related to the codes' graphical structure, which enables one to decode them [31, 32]. At that time, he could not verify the validity of his intuition, due to limited computing processing possibilities. This was first done in [77, 78]. The idea of iterative decoding for particular block codes with a sparse parity check matrix was clearly formalized in terms of a *message passing* algorithm over a bipartite graph, where there are two groups of nodes: variable nodes and check nodes [79, 80]. This graph representation of block codes, proposed in the Eighties by Tanner[5] [81], has been rediscovered in recent years. The idea of message passing algorithms for block codes can be interpreted as the equivalent of the idea of iterative decoding for concatenated convolutional codes: *nodes* (instead of blocks) exchange information to refine it. More recently, parallel or serially concatenated codes that can be decoded by using a message passing algorithm operating on the Tanner graph of the overall code, have been proposed [82].

Simulations of iterative decoding schemes might be very intensive. Simple techniques have been introduced to analytically estimate their performance. For example, an analysis based on *density evolution* associates the logarithmic likelihood ratio (LLR) generated by a SISO module [83] with a suitable pdf and tracks the evolution of this pdf for increasing input SNR and number of decoding iterations [84, 85]. Another analytical technique is based on the use of *extrinsic information transfer* (EXIT) charts [86, 87]. Intuitively, the EXIT chart of a SISO module can be considered as its characteristic "amplification" curve.

1.8 Space Time Coding

This class of codes can be used in communication systems with multiple transmitter and receiver antennas. The main idea behind the design of these codes is that of exploiting, besides multiple dimensions in the signal space, *spatial diversity* (especially at the transmitter side) obtained by using multiple antennas. The attention of the scientific community to this type of codes was boosted by theoretical work on the capacity of multiple-antenna systems [42, 88] in the late Nineties. This theoretical work has clearly shown that the capacity of a multiple-input multiple-output channel is significantly larger than the capacity of a single-input single-output channel.

[5]For this reason, this type of graph is usually referred to as a "Tanner" graph.

Space time codes, clearly introduced in [89], represent an efficient way to approach the capacity of multiple-input multiple-output channels. In particular, they exploit in an efficient way the *transmit diversity*—receiver diversity, especially in multipath fading channels, had been long exploited [90]. In particular, both *trellis* space time codes [89] and *block* space time codes [91] were proposed. In both cases, the codes must respect precise rules in order for the spatio-temporal diversity of the multiple-input multiple-output system to be effectively exploited.

1.9 Summary

In this chapter, we have considered an overview of *wireless communication systems*, in order to provide the reader with a few basic concepts which represents a theoretical background for the material presented in this book. After a brief overview of wireless communication systems, we have described a few *channel models* which will be frequently used in the following chapters, namely additive white Gaussian noise channels, frequency nonselective and selective fading channels, and phase-uncertain channels. The concepts of *demodulation*, *detection* and *parameter estimation* have been introduced, and considerations about the *capacity* of the considered channels have been made. The relation between *channel coding and modulation* has been analyzed, culminating with the description of turbo codes and LDPC codes, which have recently allowed one to "approach" the ultimate limits predicted by Shannon in 1948. The chapter terminates with a discussion about space time codes: while all the detection algorithms in the book will be derived for single-input single-output channels, they can be extended to multiple-input multiple-output channels in a standard fashion.

1.10 Problems

Problem 1.1: Show that the power spectral density of the fading process according to Clarke's model has the expression given by (1.3) [20].

Problem 1.2: Considering a phase-uncertain communication system with linear modulation and absence of ISI, show that if the phase process is Wiener with equivalent discrete-time phase model given by (1.8), then the autocovariance can be written as in (1.9). Compare your solutions with that proposed in Chapter 3.

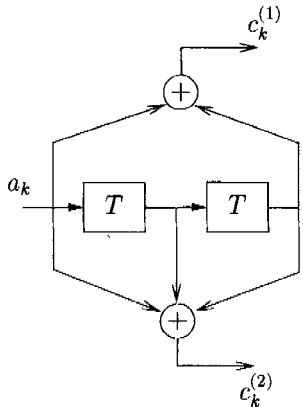


Figure 1.4: Rate-1/2 convolutional encoder with generators $G_1 = 5$ and $G_2 = 7$.

Problem 1.3: With the help of classic information theory books (e.g., [35]), derive expression (1.10) of the capacity of an AWGN channel.

Problem 1.4: Derive the lower bound (1.15) for the capacity of a noncoherent channel. Compare your derivation with the derivation in [44].

Problem 1.5: Consider a rate-1/2 convolutional encoder with generators (in octal notation) $G_1 = 5$ and $G_2 = 7$, pictured in Figure 1.4: for each information symbol a_k, two coded symbols $(c_k^{(1)}, c_k^{(2)})$ are generated.

A. Determine the minimum Euclidean distance of the code.

B. Characterize the encoder as an FSM: define a state μ_k and determine the "next-state" function $\mathrm{ns}(\mu_k, a_k)$ and the "output" function $\mathrm{o}(\mu_k, a_k)$.

Problem 1.6: Consider the (6,3) binary linear block characterized by the following parity check matrix:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

A. Design the time-varying trellis diagram associated with this block code [54]. Compare your solution with that in [79, Example 2].

B. In this case, the "next-state" and "output" functions are time-varying. Determine suitable functions $\mathrm{ns}_k(\mu_k, a_k)$ and $\mathrm{o}_k(\mu_k, a_k)$.

Problem 1.7: Consider the serial concatenation of a time-invariant encoder and a linear modulator characterized by a causal pulse with duration equal to three symbol periods. Assume that the encoder can be described as an FSM with state μ_k and "next-state" and "output" functions given by $\mathrm{ns}(\mu_k, a_k)$ and $\mathrm{o}(\mu_k, a_k)$, respectively. The serially concatenated system can be seen as an overall FSM. Define a state for the latter FSM and determine suitable "next-state" and "output" functions to describe its evolutions.

Problem 1.8: Assume that complex information symbols $\{a_k\}$, belonging to a given constellation, are mapped into coded symbols $\{c_k\}$ according to a *differential encoding* rule, i.e., $c_k = c_{k-1}a_k$, with c_0 given. Define a suitable state for the differential encoder, and determine "next-state" and "output" functions to describe its evolution.

Problem 1.9: You have been commissioned by a telecommunication company to design a 4-state and rate-2/3 TCM code with 8-PSK [57]. Denote by $a_k^{(1)}$ and $a_k^{(2)}$ the two bits at the input of the encoder.

A. Design the code by determining the transitions in a section of a trellis diagram with state $\mu_k = (a_{k-1}^{(2)}, a_{k-2}^{(2)})$.

B. Show that there are parallel transitions in the trellis diagram, and compute the minimum distance $d_{\min}$ of this TCM code.

C. Design the code by determining the transitions in a section of a trellis diagram with state $\mu'_k = (a_{k-1}^{(2)}, a_{k-1}^{(1)})$.

D. Show that in the trellis diagram with state μ'_k there are no parallel transitions and compute the minimum distance $d'_{\min}$ in this case.

Problem 1.10: Consider a binary CPM signal with $h = 1/2$ and 2REC pulse [30].

A. Design the tree phase of this coded modulation (see [28–30]).

B. Compute the number of states necessary to describe this signal.

C. According to the *decomposition approach* proposed in [61], each CPM modulator can be equivalently implemented as a serial concatenation of a rate-$1/n$ binary recursive continuous phase encoder (CPE) followed by a memoryless mapper (MM). Describe the relation between the FSM model of the CPE with the tree phase.

2

A General Approach to Statistical Detection for Channels with Memory

2.1 Introduction

Wireless channels, as seen in Chapter 1, can be of various types, and a plethora of detection strategies can be derived, as shown by the vast existing literature on the subject. Each detection strategy is based on a specific channel model and on suitable assumptions. It is, however, interesting to derive a *general* approach, applicable to many communication scenarios, for the design of detection algorithms. More precisely, finding a general framework could ultimately provide significant intuition for the design of new detection algorithms, beyond those proposed in this book.

We consider in this chapter a preliminary general approach to *statistical detection* for *channels with memory*. The main characteristics of the proposed approach can be summarized as follows.

- The transmission channel models hereafter considered will be mainly characterized by *stochastic parameters*. Most of the proposed detection strategies will be derived by assuming that knowledge of the statistical distribution of the channel parameters is available at the receiver.

- Stochastic channels are important examples of *channels with memory*. A simple and unified definition of the *memory* of the channel is not immediate. An intuitive description could be based, as a first attempt, on the assumption that channel observations at different time epochs are correlated with each other.

Detection Algorithms for Wireless Communications - G. Ferrari, G. Colavolpe and R. Raheli
 ISBN: 0-470-85828-1

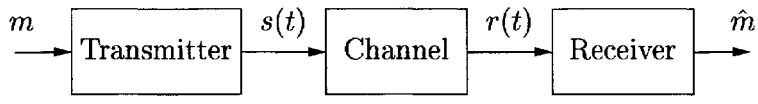


Figure 2.1: M-ary signaling and detection.

This is the case, for example, for a phase noncoherent channel or a frequency nonselective fading channel.

In the remainder of this book, the central idea of the proposed unified approach to the design of detection algorithms is based on the fact that the channel memory, even if theoretically unlimited in several cases (such as in the case of time-invariant stochastic channels), can be assumed as finite for practical purposes. This *finite memory condition* (FMC) is shown to lead to a variety of solutions that appear in the literature and allows one to unify them in a simple way.

2.2 Statistical Detection Theory

In this section, we review the basics of statistical detection theory [2,92]. We consider the transmission of an information message, belonging to a finite alphabet of possible messages, by means of a signal waveform extracted by a corresponding set of possible known signals in a one-to-one correspondence with the information messages. An information message may be thought of as a single information symbol transmitted in a single transmission act or a finite sequence of information symbols transmitted in a corresponding sequence of transmission acts. This distinction, viewed as purely formal in this section, is worth mentioning because practical transmission of information takes place almost solely in the latter form.

As shown in Figure 2.1, let m be an information message drawn from an alphabet of M possible messages denoted by $\{m_i\}_{i=1}^{M}$. The notation m denotes a generic random message (i.e., a random variable), whereas m_i denotes the i-th element of the alphabet (i.e., a possible realization of this random variable). Message m_i is transmitted by means of a signal $s_i(t)$ belonging to a set of M possible signals $\{s_i(t)\}_{i=1}^{M}$, rigidly associated with the messages by the transmitter. Hence, the notation $s(t)$ denotes a random process with sample functions $\{s_i(t)\}_{i=1}^{M}$. The received signal $r(t)$ is observed by the receiver, which decides in favor of a message $\hat{m} \in \{m_i\}_{i=1}^{M}$. If $\hat{m} \neq m$ the detected and transmitted messages differ, and a detection error takes place.

Adopting a probabilistic model of the transmission system, we may use the error

Table 2.1: *A posteriori* probabilities of the information messages.

$\hat{m}$	$P\{m = \hat{m} \mid r(t) = \rho(t)\}$
m_1	$P\{m = m_1 \mid r(t) = \rho(t)\}$
m_2	$P\{m = m_2 \mid r(t) = \rho(t)\}$
m_3	$P\{m = m_3 \mid r(t) = \rho(t)\}$
$\vdots$	$\vdots$
m_M	$P\{m = m_M \mid r(t) = \rho(t)\}$

probability as an optimality criterion and derive a *detection strategy* minimizing this probability. To this purpose we may assign *a priori* probabilities $P\{m = m_i\}$ to the possible messages. As a consequence, the received signal is a random process because it depends on the random transmitted signal and the random alteration introduced by the channel, which at least introduces thermal noise. Finally, the detected message is a random variable as well, because it is deterministically related to the random received signal.[1]

Denoting by $\{m \neq \hat{m}\}$ an error event, we seek to minimize the error probability $P\{m \neq \hat{m}\}$ based on the observation of the received signal $r(t)$. Equivalently, we could maximize the probability of correct detection $P(m = \hat{m}) = 1 - P\{m \neq \hat{m}\}$. The *average* probability of a correct decision can be expressed, in terms of the conditional probability of a correct decision given the observation, as

$$P\{m = \hat{m}\} = \mathrm{E}\{P\{m = \hat{m} \mid r(t)\}\} \tag{2.1}$$

in which $\mathrm{E}\{\cdot\}$ denotes statistical expectation and $P\{m = \hat{m} \mid r(t)\}$ is a random variable because it depends on the realization of the random process $r(t)$.

In order to derive a decision rule, or detection strategy, which maximizes $P\{m = \hat{m}\}$, let us denote by $\rho(t)$ the observed realization of the received signal. Given this realization, a decision in favor of message m_i, i.e., $\hat{m} = m_i$, yields a conditional probability of correct detection

$$P\{m = \hat{m} \mid r(t) = \rho(t)\} = P\{m = m_i \mid r(t) = \rho(t)\}\,. \tag{2.2}$$

The conditional probability of correct detection for a specific decision equals the *a posteriori* probability (APP) of the corresponding information message given the observation $\{r(t) = \rho(t)\}$. Table 2.1 shows these probabilities for any possible decision. As these probabilities are positive quantities for any $\rho(t)$, the maximization

[1] We are implicitly assuming a deterministic detection strategy.

of the average probability of correct detection (2.1) is achieved by maximizing the APP of the possible messages for any given observation. Hence, the optimal detection strategy can be realized by computing the right column of Table 2.1 for the given signal observation and deciding in favor of the row corresponding to the maximum entry. The resulting optimal detection strategy operates in accordance with the maximum *a posteriori* (MAP) criterion and can be formally expressed as

$$\hat{m} = \underset{m_i}{\operatorname{argmax}} P\{m = m_i | r(t) = \rho(t)\} \tag{2.3}$$

where "argmax" denotes the argument which maximizes the following function with respect to the variable of interest, i.e., m_i. For notational conciseness, this formulation can be abbreviated as

$$\hat{m} = \underset{m_i}{\operatorname{argmax}} P\{m_i | r(t)\} \tag{2.4}$$

where the event m_i denotes the transmission of the i-th information message and $r(t)$ denotes the signal observation.

We remark that the described approach is general, in the sense that the considered messages are still unspecified. In particular, should the message be the entire transmitted information sequence, the considered optimality criterion would lead to MAP *sequence detection*. On the other hand, if the considered message is an information symbol, then the considered optimality criterion leads to MAP *symbol detection*.

The computation of the APPs in (2.4) can be approached through a process of *discretization* of the involved signals. This process entails the representation of a continuous-time signal by means of a sequence of discrete-time variables. A discrete representation of the received signal is obtained in terms of the components of its orthogonal projection onto an orthonormal basis spanning a suitable signal space [92]. As shown in Appendix A, discretization leads to negligible loss of information, possibly considering oversampling.

In a number of important applications, this *relevant* signal space has finite dimensionality and the resulting discrete representation of a signal can be geometrically interpreted as a finite-dimensional *image* vector. This occurs whenever the channel introduces time-invariant perturbations on the M-ary signals at its input, besides additive white Gaussian noise (AWGN). As described in Section 2.1, an interesting case is obtained when these perturbations are of a stochastic nature, provided that they can be described in terms of a finite number of random parameters. Examples of channel models of this type are all those depicted in Figure 1.1, under the assumption of time-invariance, i.e., including the cases with known static dispersion, flat slow fading, dispersive slow fading, and phase noncoherence.

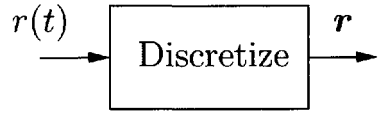


Figure 2.2: Discretization of the received signal.

It is worth noting that all that is needed for optimal detection are the components of the received signal in the relevant signal space. In fact, the resulting image vector is only representative of the orthogonal projection of the received signal onto this signal space. The received signal itself cannot in general be reconstructed from its image.[2] Nonetheless, the received signal image vector encompasses all relevant information for optimal detection, that is it is a *sufficient statistic*. Figure 2.2 shows this information-preserving discretization function applied to the received signal $r(t)$, yielding the image vector $\boldsymbol{r}$. The block in Figure 2.2 can be thought of as the receiver *front-end*.

Assuming a sufficient statistic $\boldsymbol{r}$ is obtained by discretization, an optimal detection strategy equivalent to (2.4) can be formulated in terms of the APPs of the information messages given the observed image vector $\boldsymbol{r}$, as follows

$$\begin{aligned} \hat{m} &= \underset{m_i}{\operatorname{argmax}} P\{m_i|\boldsymbol{r}\} \\ &= \underset{m_i}{\operatorname{argmax}} p(\boldsymbol{r}|m_i)P\{m_i\} \end{aligned} \tag{2.5}$$

where $p(\boldsymbol{r}|m_i)$ is the conditional probability density function (pdf) of the observation vector given the transmission of the i-th message and $P\{m_i\}$ is the *a priori* probability of this message. The second line in (2.5) is immediately obtained by applying the Bayes rule and discarding irrelevant factors, i.e.:

$$P\{m_i|\boldsymbol{r}\} = \frac{p(\boldsymbol{r}|m_i)P\{m_i\}}{p(\boldsymbol{r})} \sim p(\boldsymbol{r}|m_i)P\{m_i\} \tag{2.6}$$

in which $p(\boldsymbol{r})$ is the unconditional pdf of the observation vector and the symbol $\sim$ denotes a monotonic relationship with respect to the argument of interest (m_i in this case). We remark at this point the significant impact of the Bayes rule in the derivation of the proposed detection algorithms. Ultimately, for all detection algorithms derived in the following chapters, the starting point will be given by the application of this rule.

[2]The ubiquitous thermal noise is a suffi cient reason for the inherent nonrepresentability or infi nite dimensionality of the received signal; for example, AWGN is not representable due to its infi nite variance.

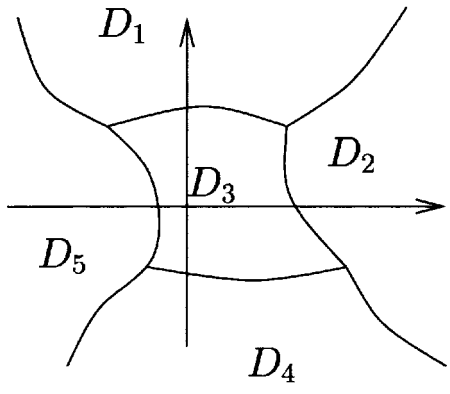


Figure 2.3: Decision regions.

According to (2.5), the MAP detection strategy allows a pleasing geometrical interpretation. In fact, assuming that K denotes the dimension of the observation vector $\boldsymbol{r}$, each point in the observation vector space yields a decision for message m_i which maximizes the APPs $P\{m_i|\boldsymbol{r}\}$. Hence, this space can be partitioned into M *decision regions*, as pictorially exemplified in Figure 2.3 in a simple bi-dimensional case. When the observation vector falls into region D_i, the receiver makes a decision for message m_i. Generalizations of this basic idea to the space of stochastic channel parameters have recently led to an interesting approach to perform polynomial-complexity *optimal* detection over particular channels [93].

The formulation of the MAP detection strategy (2.5) emphasizes that an optimal receiver needs only statistical characterizations of: (i) the cascade of transmitter and channel, i.e., the conditional pdfs $p(\boldsymbol{r}|m_i)$, and (ii) the information source, i.e., the *a priori* message probabilities $P\{m_i\}$. While the latter quantities $P\{m_i\}$ are needed *a priori* in any detection algorithm, the key part of the derivation of a detection algorithm consists of the computation of the conditional pdf $p(\boldsymbol{r}|m_i)$. In fact, the quality and efficiency of a detection algorithm depends on finding a clever way to compute these pdfs at the received point $\boldsymbol{r}$.

When all the *a priori* message probabilities are equal, the factors $P\{m_i\}$ in (2.5) are irrelevant and can be discarded. In this case, the MAP detection strategy is equivalent to the maximum likelihood (ML) detection strategy, a *logical* detection procedure useful when the *a priori* message probabilities are unknown.[3] For this reason, the terminology MAP and ML is used interchangeably when these probabilities are

[3]We remark that the notion of optimality as previously introduced is not defined in the absence of complete statistical information.

known and equal, perhaps inappropriately, despite the fact that strictly speaking the terminology ML refers to the case where *a priori* message probabilities are unknown.

A general condition for the *irrelevance* of signal components in the detection problem can be derived as follows. Let the observation vector be the concatenation of two subvectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. The MAP detection strategy (2.5) can be formulated in terms of the conditional joint pdf of these vectors

$$\begin{aligned} \hat{m} &= \underset{m_i}{\operatorname{argmax}}\, p(\boldsymbol{r}_1, \boldsymbol{r}_2 | m_i) P\{m_i\} \\ &= \underset{m_i}{\operatorname{argmax}}\, p(\boldsymbol{r}_1 | m_i) p(\boldsymbol{r}_2 | \boldsymbol{r}_1, m_i) P\{m_i\} \end{aligned} \tag{2.7}$$

where the chain factorization rule has been employed in the second line. If the following irrelevance condition is met

$$p(\boldsymbol{r}_2 | \boldsymbol{r}_1, m_i) = p(\boldsymbol{r}_2 | \boldsymbol{r}_1) \qquad \forall i \tag{2.8}$$

vector $\boldsymbol{r}_2$ is independent of m_i, conditionally on $\boldsymbol{r}_1$, and can be discarded because it is irrelevant to the decision (see Problem 2.12).

The MAP detection strategy formulated in (2.5) is very general and depends on the conditional pdf of the observation. Considering specific applications, the statistics of this function can be used and more explicit formulations of the strategy obtained. A particularly important case is that of M-ary signaling in an AWGN channel, where the receiver has perfect knowledge of the signal waveforms $\{s_i(t)\}_{i=1}^{M}$. In this case, the received signal has the expression

$$r(t) = s(t) + n(t) \tag{2.9}$$

where $n(t)$ is a white circular complex Gaussian noise process. The relevant signal space for this detection problem is the space *spanned* by the signal waveforms $\{s_i(t)\}_{i=1}^{M}$. Denoting by $\{\varphi_i(t)\}_{i=1}^{D_S}$ an orthonormal basis for this space, where D_S is the space dimension, a sufficient statistic is given by the components of the orthogonal projection of the received signal onto this space. Hence, the k-th element of this discrete observable $\boldsymbol{r}$ is

$$r_k = \int_0^T r(t) \varphi_k^*(t)\, \mathrm{d}t \tag{2.10}$$

where T denotes the signaling interval[4] and $(\cdot)^*$ denotes complex conjugation.

[4]This implicitly assumes that $\{s_i(t)\}$ have duration equal to T.

Given the transmitted waveform $s_i(t)$, the discrete observable is a conditional Gaussian random vector, due to thermal noise. As the projection of white noise onto any orthonormal basis has independent components with equal variance, the components of $\boldsymbol{r}$ have equal variance. Therefore, conditionally on the transmission of message m_i, the observation vector is $\boldsymbol{r} = \boldsymbol{s}_i + \boldsymbol{n}$, where $\boldsymbol{s}_i$ is the image of the i-th signal waveform and $\boldsymbol{n}$ is a Gaussian random vector with independent zero-mean equal-variance elements. We conclude that the MAP strategy entails the maximization of the following quantity:

$$\begin{aligned} p(\boldsymbol{r}|m_i)P\{m_i\} &= \frac{1}{(\pi 2\sigma^2)^{D_S}} \exp\left(-\frac{1}{2\sigma^2}||\boldsymbol{r}-\boldsymbol{s}_i||^2\right) P\{m_i\} \\ &\sim -||\boldsymbol{r}-\boldsymbol{s}_i||^2 + 2\sigma^2 \ln P\{m_i\} \\ &\sim \Re\left\{\boldsymbol{r}^{\mathrm{T}}\boldsymbol{s}_i^*\right\} - \frac{1}{2}||\boldsymbol{s}_i||^2 + \sigma^2 \ln P\{m_i\} \\ &= \Re\left\{\int_0^T r(t)s_i^*(t)\,\mathrm{d}t\right\} - \frac{1}{2}\int_0^T |s_i(t)|^2\,\mathrm{d}t + \sigma^2 \ln P\{m_i\} \end{aligned} \tag{2.11}$$

where σ^2 is the variance of the real and imaginary components of the complex noise samples, $(\cdot)^{\mathrm{T}}$ denotes vector or matrix transposition and $\Re(\cdot)$ denotes the real part of a complex number—a direct time-domain formulation is given in the last line.

2.3 Transmission Systems with Memory

The transmission of a significant amount of information is normally achieved by periodically repeating M-ary signaling and detection acts. In a repetition, or *signaling*, interval an M-ary information symbol or, equivalently, $\log_2 M$ information bits are transmitted. If proper conditions are verified these subsequent signaling acts do not influence each other and optimal detection can be realized based on independent signal observations. In this case, the transmission system is *memoryless*.

There may exist situations, however, in which the received signal in the "current" transmission act is affected by "past" and, possibly, "future" transmission acts. More generally, the detection process may benefit from the observation of the received signal over more than one signaling interval. Specifically, this may occur because of the following reasons.

1. Channel coding is employed for error protection. In this case, the *codewords* are sequences of code symbols characterized by the fact that not all sequences

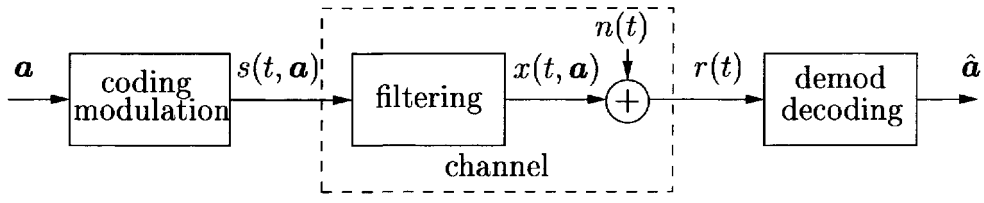


Figure 2.4: Transmission system.

are possible codewords. This means that the encoder operates introducing *redundancy* in the code sequence.

2. The transmission channel is dispersive. Because of that, the received signal is affected by inter-symbol interference (ISI) and in a given signaling interval it may depend on a group of transmitted symbols.

3. The model of the transmission channel includes stochastic parameters, such as a phase rotation or a complex fading weight. The received signal may therefore depend on the entire sequence of previous symbols.

4. The channel additive Gaussian noise is colored, i.e., its power spectral density is not constant. In this case, the detection process may benefit from the observation of the received signal over several signaling intervals because, implicitly, it may predict the realization of the correlated noise process.

These situations are commonly characterized by the fact that an optimal detection process of each information symbol requires the observation of the received signal over several signaling intervals, e.g., beyond the signaling interval of the information symbol being detected. As a consequence, the transmission system is inherently characterized by *memory*.

A general model of a transmission system with memory is shown in Figure 2.4. An information symbol belonging to an M-ary alphabet is denoted as a_k, where the index k now refers to the discrete-time instant. We assume that K information symbols are transmitted successively from time 0 to time $K-1$. A sequence of information symbols in the discrete time interval (k_1, k_2) is denoted in vector notation as

$$\boldsymbol{a}_{k_1}^{k_2} = (a_{k_2}, \ldots, a_{k_1+1}, a_{k_1})^{\mathrm{T}}. \tag{2.12}$$

For brevity, the entire information sequence is denoted as $\boldsymbol{a} = \boldsymbol{a}_0^{K-1}$. This sequence is input to the encoder and modulator. The coded and modulated signal is denoted

as $s(t, \boldsymbol{a})$ to emphasize its dependence on the information sequence. The channel is viewed as a noiseless filter (possibly stochastic) with output signal $x(t, \boldsymbol{a})$, rendered noisy by the addition of thermal noise $n(t)$. The received signal $r(t)$ is observed by the demodulation and decoding block which outputs the decision sequence $\hat{\boldsymbol{a}}$. We remark that the indexing used in the current notation refers to time, instead of the element of the message alphabet as in Section 2.2.

The encoder/modulator block in Figure 2.4 is a generic system which evolves, upon receiving at its input the information sequence $\boldsymbol{a}$, through a sequence of states $\{\mu_0, \mu_1, \ldots\}$. In many realistic cases, the encoder/modulator can be described as a time-invariant finite state machine (FSM) (e.g., trellis-coded or continuous phase modulation). The state μ_k belongs in this case to a set of finite cardinality and a *time-invariant* "next-state" function $\mathrm{ns}(\cdot, \cdot)$ describes the evolution of the system as

$$\mu_{k+1} = \mathrm{ns}(\mu_k, a_k). \tag{2.13}$$

The evolution of the encoder/modulator can therefore be described through a trellis diagram, in which there are M exiting branches (in correspondence with M different information symbols) from each state. Hereafter, the initial state μ_0 of the FSM will be assumed known.

The received signal can be expressed as

$$r(t) = x(t, \boldsymbol{a}) + n(t). \tag{2.14}$$

By means of a *discretization process*, the received signal $r(t)$ can be converted into an equivalent discrete-time sequence $\boldsymbol{r}$ [94]. Let us consider a discrete observable $\boldsymbol{r}$ with dimensionality equal to that of the information sequence $\boldsymbol{a}$. As a consequence, there is one observable r_k per information symbol a_k or, formally, $\boldsymbol{r} = \boldsymbol{r}_0^{K-1}$, with a notation similar to that used for the information sequence. For the sake of simplicity, this formulation does not account for *border effects* due to slight differences in the dimension of $\boldsymbol{r}$ and $\boldsymbol{a}$. For example, in the case of ISI channels characterized by an equivalent discrete-time model with more than one tap, the number of observables $\{r_k\}$ could be larger than the number of effectively transmitted information symbols $\{a_k\}$. The extension of the proposed formulation to account for this situation is straightforward.

The considered model is based on a sampling rate of one sample per symbol, which is practically sufficient in many cases. In a more general setting, there may be two or more elements of $\boldsymbol{r}$ per information symbol a_k, e.g., when a convolutional code or a time-varying channel is considered. These cases can be encompassed by the proposed formulation interpreting each observation r_k as a *vector* of suitable (small) dimensionality. The choice of a unitary sampling rate is expedient to make the following derivation clear.

Under a *causality condition*, the received signal in the signaling interval $kT < t < (k+1)T$ can be written as

$$r(t) = x(t, \boldsymbol{a}) + n(t) = x(t, \boldsymbol{a}_0^k) + n(t) \tag{2.15}$$

i.e., it depends on the information sequence up to epoch k. In the equivalent discrete-time model the continuous causality condition in (2.15) becomes

$$p(\boldsymbol{r}_0^k|\boldsymbol{a}) = p(\boldsymbol{r}_0^k|\boldsymbol{a}_0^k). \tag{2.16}$$

2.3.1 Causality and Finite Memory

In this subsection, we introduce key conditions for the derivation of a general class of detection algorithms applicable to channels with memory. Note that some of the proposed conditions or part of the derivation might seem to come out, at this point in the book, without a clear motivation. The true reason is the fact that after trying to derive various specific algorithms one can realize that there are a few unifying conditions, which are applicable in many situations and encompass the derivation of several algorithms. Hence, the reader should not pretend to understand immediately *why* the conditions in the following represent the foundation of the unified detection approach described in this book. The reader should instead try to understand these conditions through an *iterative* learning process: first, he/she should read these conditions; then he/she should study the algorithms proposed in the following chapters; finally, he/she should go back to these conditions to convince himself/herself that they are the basic necessary conditions.

Taking into account the above considerations, it is possible to show that the following FMC constitutes the basic assumption to derive many detection algorithms:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \tag{2.17}$$

where C is a suitable *finite memory parameter* and μ_{k-C} represents the state, at epoch $k-C$, of the encoder/modulator. The finite memory condition (2.17) is an extension of the *folding condition* introduced in [95], obtained by accounting for the encoder/modulator state μ_k. For the algorithm derivation proposed in the following chapters, it is very useful to consider other conditions which are obtained as consequences of causality and finite memory conditions. In particular, the following two results will be very useful.

1. The first useful consequence is that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \qquad \forall D \geq C. \tag{2.18}$$

Proof. In fact,

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}) = \sum_{\boldsymbol{a}_0^{k-D-1}} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \mu_{k-D}) \cdot P\{\boldsymbol{a}_0^{k-D-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}\} \quad (2.19)$$

where the notation $\boldsymbol{a}_0^{k-D-1}$ at the bottom of the sum operator indicates that the sum is considered over all possible information sequences $\boldsymbol{a}_0^{k-D-1}$. If a sequence $\boldsymbol{a}_0^{k-D-1}$, given μ_0, is incompatible with μ_{k-D}, then $P\{\boldsymbol{a}_0^{k-D-1}|\mu_{k-D}\} = 0$. Hence, for any sequence $\boldsymbol{a}_0^{k-D-1}$ compatible with μ_{k-D}, one can write:

$$\begin{aligned} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \mu_{k-D}) &= p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) \\ &= p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \end{aligned} \quad (2.20)$$

where the last equality is based on the FMC (2.17). Hence, (2.19) becomes:

$$\begin{aligned} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}) &= \sum_{\boldsymbol{a}_0^{k-D-1}} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \\ &\quad \cdot P\{\boldsymbol{a}_0^{k-D-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}\} \\ &= p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \\ &\quad \cdot \sum_{\boldsymbol{a}_0^{k-D-1}} P\{\boldsymbol{a}_0^{k-D-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}\}. \end{aligned} \quad (2.21)$$

Since

$$\sum_{\boldsymbol{a}_0^{k-D-1}} P\{\boldsymbol{a}_0^{k-D-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}\} = 1 \quad (2.22)$$

from (2.21) it follows that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-D}^k, \mu_{k-D}) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \quad (2.23)$$

which completes the proof. □

2. The second useful consequence of causality and finite memory conditions is that

$$p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^k) = p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^k, \mu_{k-C+1}). \quad (2.24)$$

Proof. In fact,

$$p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^k) = \sum_{\boldsymbol{a}_{k+1}^{K-1}} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^{K-1}) P\{\boldsymbol{a}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^k\}. \quad (2.25)$$

Based on the independence between information symbols and on the system causality assumption, it follows that

$$P\{\boldsymbol{a}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^k\} = P\{\boldsymbol{a}_{k+1}^{K-1}\}. \quad (2.26)$$

Applying the chain rule, one can immediately write:

$$p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^{K-1}) = \prod_{i=k+1}^{K-1} p(r_i|\boldsymbol{r}_0^{i-1}, \boldsymbol{a}_0^{K-1}) \quad (2.27)$$

where, based on the causality and finite memory conditions, each term in (2.27) can be expressed as

$$p(r_i|\boldsymbol{r}_0^{i-1}, \boldsymbol{a}_0^{K-1}) = p(r_i|\boldsymbol{r}_0^{i-1}, \boldsymbol{a}_{k-C+1}^{K-1}, \mu_{k-C+1}). \quad (2.28)$$

Using (2.28) in (2.27) we may write:

$$\begin{aligned} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^{K-1}) &= \prod_{i=k+1}^{K-1} p(r_i|\boldsymbol{r}_0^{i-1}, \boldsymbol{a}_{k-C+1}^{K-1}, \mu_{k-C+1}) \\ &= p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^{K-1}, \mu_{k-C+1}). \end{aligned} \quad (2.29)$$

Finally, (2.25) becomes:

$$\begin{aligned} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_0^k) &= \sum_{\boldsymbol{a}_{k+1}^{K-1}} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^{K-1}, \mu_{k-C+1}) P\{\boldsymbol{a}_{k+1}^{K-1}\} \\ &= p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^k, \mu_{k-C+1}) \end{aligned} \quad (2.30)$$

which completes the proof. □

The encoder/modulator block in Figure 2.4 can be often decomposed into a cascade of an encoder and a memoryless mapper. In this case, the causality and finite memory conditions imply analogous relations between the observation sequence $\boldsymbol{r}$ and *code* sequence $\boldsymbol{c} = \boldsymbol{c}_0^{K-1}$, where c_k is a generic code symbol. These conditions can accordingly be formulated as follows:

$$p(\boldsymbol{r}_0^k|\boldsymbol{c}) = p(\boldsymbol{r}_0^k|\boldsymbol{c}_0^k) \quad (2.31)$$

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_{k-C}^k). \quad (2.32)$$

These conditions, however, involve only the transmission channel and do not imply (2.16) and (2.17) in general. A case of interest may be that of a linear block code followed by a memoryless modulator. In particular, a linear block code is not guaranteed to be causal,[5] so that channel causality (2.31) does not imply system causality (2.16).

In the case of a linear block code, a trellis representation is possible, but the trellis is time-variant both in terms of states and branches [54,55]. In this case, the evolution of the encoder/modulator over a trellis diagram should be described by a *time-variant* next-state function $\mathrm{ns}_k(\cdot,\cdot)$. One can immediately conclude that a *Tanner graph* [81] (or other suitable graphical) representation for a linear block code—where the parity checks determine the structure of the graph—can be more appealing, especially if the parity check equations involve a few symbols (as in the case of low density parity check, LDPC, codes) [78].

A significant example where causality and finite memory conditions strictly hold is given by transmission over channels with ISI, possibly encompassing a *nonlinearity with finite memory*. The pdf at the right-hand side of (2.17) simplifies, by dropping the conditioning observables, to

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^{k}, \mu_{k-C}) \;=\; p(r_k|\boldsymbol{a}_{k-C}^{k}, \mu_{k-C}) \tag{2.33}$$

where in this case the finite memory parameter C equals the channel dispersion. While it will be shown in much more detail in the following chapters, we now remark that the pdf in (2.33) can be directly used both in a Viterbi algorithm (VA) and a forward backward (FB) algorithm. A similar property holds for (2.32): this is of interest, for example, in the case of transmission of linear block codes over ISI channels. Hence, a sum-product (SP) algorithm based on the same basic metric can also be applied (see Chapter 5 for more details).

2.3.2 Stochastic Channels: Channels with Infinite Memory

While in Section 2.3.1 the case of detection for channels with strictly finite memory has been considered, it is important to underline that this is rarely observed in the case of wireless communication channels.

In the case of a channel affected by stochastic uncertainty, the observations $\{r_k\}$ are *dependent*, so that the channel memory may not be finite. A very general parametric model for the observation r_k is the following:

$$r_k = g(\boldsymbol{a}_{k-L}^{k}, \mu_{k-L}, \boldsymbol{\theta}_0^{k}) + n_k \tag{2.34}$$

[5] *Block-wise* causality must be indeed satisfi ed.

where L is an integer, $\boldsymbol{\theta}_0^k$ is a sequence of stochastic parameters independent of $\boldsymbol{a}$, and n_k is an additive noise sample.[6] Under this channel model, the following *conditional Markov property* (CMP)

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) \tag{2.35}$$

where N is the order of Markovianity, is sufficient to guarantee an FMC. It is possible to show that (2.35) implies the following:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \tag{2.36}$$

where the finite memory parameter is $C = N + L$. In fact, based on (2.35), one can write

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) \;=\; p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) = \frac{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_0^k)}{p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{a}_0^k)}. \tag{2.37}$$

The conditional pdf in the numerator of (2.37) can be expressed, by applying the total probability theorem, as follows:

$$p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_0^k) \;=\; \underbrace{\int \cdots \int}_{\boldsymbol{\theta}_0^k} p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_0^k, \boldsymbol{\theta}_0^k) p(\boldsymbol{\theta}_0^k)\, \mathrm{d}\boldsymbol{\theta}_0^k. \tag{2.38}$$

Owing to the considered observation model (2.34), it is immediately concluded that

$$p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_0^k, \boldsymbol{\theta}_0^k) = p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C}, \boldsymbol{\theta}_0^k) \tag{2.39}$$

where $C = N + L$. As the stochastic parameters are independent from the information symbols, the second pdf inside the integral in (2.38) can be equivalently expressed as $p(\boldsymbol{\theta}_0^k|\boldsymbol{a}_{k-L-C}^k, \mu_{k-L-C})$. Finally, the integral (2.38) becomes:

$$\begin{aligned} p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_0^k) \;&=\; \underbrace{\int \cdots \int}_{\boldsymbol{\theta}_0^k} p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C}, \boldsymbol{\theta}_0^k)\, p(\boldsymbol{\theta}_0^k)\, \mathrm{d}\boldsymbol{\theta}_0^k \\ &=\; p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C}). \end{aligned} \tag{2.40}$$

[6]The additive noise is not required to be Gaussian for the validity of the following result. We also point out that the symbol θ will be used, in the remainder of the book, to indicate a possible channel phase rotation, besides a generic channel stochastic parameter, as in the subsection at hand. The context should eliminate any ambiguity.

Applying the same reasoning to the denominator of (2.37) (taking also into account the causality of the system), one can conclude that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = \frac{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C})}{p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C})} = p(r_k|\boldsymbol{r}_{k-L-C}^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \quad (2.41)$$

which corresponds to (2.36). One can immediately recognize that (2.36) represents a special case of (2.17). As a consequence, all the derivations in the previous section hold.

A statistical description of the stochastic parameter allows one to compute the basic pdf in (2.36) as follows:

$$\begin{aligned} p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) &= \frac{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C})}{p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C})} \\ &= \frac{\mathrm{E}_{\boldsymbol{\theta}_0^k}\{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{a}_{k-C}^k, \mu_{k-C}, \boldsymbol{\theta}_0^k)\}}{\mathrm{E}_{\boldsymbol{\theta}_0^{k-1}}\{p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}, \boldsymbol{\theta}_0^{k-1})\}}. \end{aligned} \quad (2.42)$$

Unfortunately, the above exact result is limited by the fact that *in realistic scenarios* the CMP (2.35) is seldom exactly met [47, 95]. However, this result suggests a reasonable approach to devising good approximate detection algorithms whenever the conditional observations are asymptotically independent for increasing index difference.

2.4 Overview of Detection Algorithms for Stochastic Channels

As described in Section 2.1, the design of detection algorithms involves many aspects, such as the channel model, the considered modulation/coding format, and possible constraints on the complexity of the algorithm. In particular, a major classification among detection algorithms is the following.

- *Hard-output* detection algorithms generate "hard" decisions on the transmitted message, in the sense that their output is one of the possible messages in the transmission alphabet. The most common of these algorithms is the VA [96, 97], which performs MAP sequence detection. This algorithm will be described in more detail in Chapter 3.

- *Soft-output* detection algorithms, instead of generating hard decisions, compute *reliability values* on the transmitted messages. The most common algorithm is

the FB algorithm [54]. These reliability values can be of various types and they are usually related to the APP of the transmitted messages $P\{m_i|\boldsymbol{r}\}$. In particular, this type of algorithm is generally used to perform (exact or approximated) MAP symbol detection, so that the calculated APP can be written as $P\{a_k|\boldsymbol{r}\}$. In the case of binary information symbols, a commonly considered reliability value is given by the *logarithmic likelihood ratio* (LLR), defined as

$$\mathrm{LLR}_k \triangleq \log \frac{P\{a_k = 1|\boldsymbol{r}\}}{P\{a_k = 0|\boldsymbol{r}\}}. \tag{2.43}$$

One can immediately recognize that an LLR captures within a single quantity the relationship between the APP of a transmitted "1" and that of a transmitted "0." Note that the formulation, based on the use of the LLR, can be also extended to the case of larger information symbol cardinality. In the case of M-ary symbols, $(M-1)$ LLRs are needed: the LLR relative to the m-th symbol, $m = 1, \ldots, M-1$, is given by the logarithm of the ratio between $P\{a_k = m|\boldsymbol{r}\}$ and $P\{a_k = 0|\boldsymbol{r}\}$—in other words, the reference probability still remains the APP of zero symbol, for which the LLR is thus 0.

The transmission model described in Section 2.3 can be viewed as based on a single M^K-ary signaling act or K repetitions of M-ary signaling acts. In the former interpretation, the message is the entire information sequence, whereas in the latter the message is each individual information symbol. According to these interpretations, two MAP detection strategies are obtained. MAP *sequence* detection is optimal in the sense that it minimizes the probability of erroneously detecting the entire sequence, i.e., selecting a sequence not equal to the transmitted one. MAP *symbol* detection minimizes the probability of erroneously detecting each information symbol. Specializing (2.4), we may derive the following general formulations of these detection strategies:

$$\begin{aligned} \hat{\boldsymbol{a}} &= \operatorname*{argmax}_{\boldsymbol{a}} P\{\boldsymbol{a}|\boldsymbol{r}\} && \text{MAP sequence detection} \\ \hat{a}_k &= \operatorname*{argmax}_{a_k} P\{a_k|\boldsymbol{r}\} && \text{MAP symbol detection.} \end{aligned}$$

Although in principle which of these detection strategies is best may depend on the specific application, the difference is mainly of a conceptual nature.[7] In fact, the most frequent erroneous decisions occurring in sequence detection are those characterized by only very few symbol errors. For high signal-to-noise ratios (SNRs), where

[7]Note, however, that the use of symbol detection becomes essential in the case of iterative detection, as will be shown in detail in Chapter 4.

most frequent errors dominate, most sequence errors entail only very few symbol errors, typically one in uncoded systems or the minimum allowed number in coded systems. As a consequence, the symbol error rates attained by receivers designed according to these two strategies tend to become equal for increasing SNR.

The receiver complexity necessary to implement these detection strategies may be significantly different. Sequence detection is often simpler to implement thanks to the well-known and celebrated VA. Symbol detection, however, has an important inherent feature which has attracted much attention in recent years, namely it generates the APP of each information symbol. This APP can be regarded as *soft* information about a symbol, in the sense that besides symbol decisions, taken by selecting the symbol with maximum APP, they provide information about the reliability of these decisions. If the APP of a specific symbol is much greater than those of the other symbols, a decision in favor of that symbol is more reliable with respect to the case when that APP is just a little greater than the others. The usefulness of soft information can be immediately understood considering a receiver implementation, as in the "decomposed" block diagram in Figure 1.2. The decoding function can be significantly facilitated, and its performance improved, if the demodulation function outputs soft information about the code sequence, instead of just the code symbol decisions. This soft information is also the basis of *iterative* detection techniques [33, 98]. In this case, in order to compute $P\{a_k|\boldsymbol{r}\}$, one could average (by applying the total probability theorem) over the probabilities of all paths containing a_k. However, this is not efficient. A much more efficient algorithm is the FB algorithm, which will be described in more detail in Chapter 4.

Particular attention must finally be devoted to a soft-output decoding/detection technique which has been developing over the last few years, namely *graph-based* decoding. As mentioned in Section 1.6.1, while a trellis representation is attractive for a convolutional code, in the case of a linear block code a graphical representation is more appealing. For example, one can consider a *factor graph* constituted by two kinds of node: *variable* nodes (relative to the transmitted coded symbols) and *check* nodes (relative to the parity check equations characterizing the linear block code). Such a graphical representation is practical if the number of edges in the considered graph is not too large. This is the case, for instance, for a linear block code such that each parity check equation involves a limited number of coded symbols. Linear block codes of this type are, for example, LDPC codes. In particular, a simple and, under certain conditions, almost optimal decoding algorithm, referred to as a *message passing algorithm*, is based on the exchange of soft information between check and variable nodes.

As will be shown in the following chapters, several of the proposed algorithms will guarantee almost optimal performance. However, the price to pay in terms of

complexity will sometimes be very heavy. In particular, one of the main conclusions which stem from the general framework proposed in this book is that the goal in designing a detection algorithm is not that of approaching[8] the optimal performance of a receiver which perfectly knows the channel (*regardless* of the algorithmic complexity), but rather to approach the ideal performance with *limited* complexity.

2.5 Summary

This Chapter represents the "key" to understanding, with a unitary perspective, the entire book. After a brief discussion about statistical detection theory, the focus moved to transmission systems with memory. A statistical definition of *causality* and *finite-memory* represents the root for the derivation of most of the detection algorithms presented in the following chapters. In fact, these simple techniques lead to a unified approach to *finite memory detection*, which allows one to systematically derive MAP (trellis-based) sequence and (trellis-based and graph-based) symbol detection algorithms. In the case of stochastic channels, which correspond to the reality of wireless communication systems, we have shown that the application of a *conditional Markov property* still leads to finite memory detection. In particular, while not rigorous in most cases, the conditional Markov property practically allows one to describe well a stochastic channel, provided that the order of Markovianity is sufficiently large.

2.6 Problems

Problem 2.1: Consider a 12-cross QAM constellation, obtained from a 16-QAM constellation by removing the four points at the corners of the constellation. Assuming uncoded transmission, determine the decision regions for symbol-by-symbol detection at the receiver side.

Problem 2.2: Consider a 32-ary amplitude PSK (32-APSK) constellation, as shown in Figure 2.5. The three radii r_1, r_2 and r_3 are suitably optimized values. Assuming uncoded transmission, determine the decision regions for symbol-by-symbol decision at the receiver side.

Problem 2.3: [92, Section 4.2] In the transmission schemes considered in Figure 2.6, $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are additive noise vectors independent from each other and

[8]Note that reaching the optimal performance might be basically impossible in some cases.

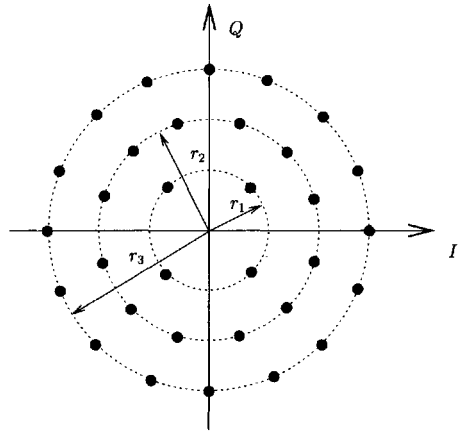


Figure 2.5: Constellation for 32-APSK.

from the discrete-time transmitted signal vector $\boldsymbol{s}$. Determine, in each of the three cases shown in Figure 2.6, if vector $\boldsymbol{r}_2$ is irrelevant in detecting the transmitted signal, i.e., if $\boldsymbol{r}_1$ is a sufficient statistic (see Appendix A).

Problem 2.4: Consider a single-input single-output channel with additive noise whose output is a scalar $r = s + n$, where the noise n is a Gaussian random variable with zero mean and variance σ^2. Assume a binary transmission: $s \in \{s_1, s_2\}$, with $P\{s_1\} = p$. Determine the decision regions depending on p.

Problem 2.5: Consider a bidimensional channel with additive Gaussian noise. The received discrete-time vector can be written as $\boldsymbol{r} = \boldsymbol{s}+\boldsymbol{n}$, where the signal vector $\boldsymbol{s}$ can assume the following values (ternary transmission):

$$\boldsymbol{s}_1 = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \qquad \boldsymbol{s}_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad \boldsymbol{s}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

and the noise vector $\boldsymbol{n}$ has independent components, with zero mean and variance σ^2. Determine the decision regions in the following cases.

A. Equally probable messages.

B. The probabilities of the messages are $P\{m_1\} = 1/9$, $P\{m_2\} = P\{m_3\} = 4/9$, and $\sigma^2 = 1/\ln 2$.

Problem 2.6: The observation vector at the output of a fading channel can be expressed as

$$\boldsymbol{r} = \boldsymbol{s} \odot \boldsymbol{f} + \boldsymbol{n}$$

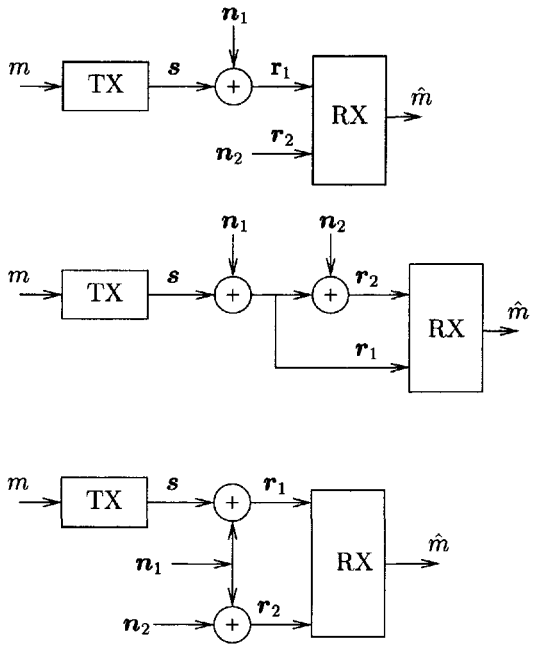


Figure 2.6: Possible communication systems. Reproduced from [92] by permission of John Wiley & Sons.

where $\odot$ denotes an element-by-element product of vectors or matrices. Show that the conditional mean vector and covariance matrix of the observation vector can be written as follows:

$$\begin{aligned} \boldsymbol{\eta}_i &\triangleq \mathrm{E}\{\boldsymbol{r}|m_i\} = \boldsymbol{s}_i \odot \boldsymbol{\eta}_f \\ \boldsymbol{C}_i &\triangleq \mathrm{E}\{\boldsymbol{r}\boldsymbol{r}^{\mathrm{T}}|m_i\} - \boldsymbol{\eta}_i\boldsymbol{\eta}_i^{\mathrm{T}} = (\boldsymbol{s}_i\boldsymbol{s}_i^{\mathrm{T}}) \odot \boldsymbol{C}_f + \boldsymbol{C}_n \end{aligned}$$

where $\boldsymbol{\eta}_f$ and $\boldsymbol{C}_f$ indicate, respectively, the mean vector and covariance matrix of the fading vector $\boldsymbol{f}$, and $\boldsymbol{C}_n$ is the covariance matrix of the additive noise $\boldsymbol{n}$.

Problem 2.7: In a binary transmission system over a fading channel, the observation can be written as:

$$\boldsymbol{r} = \boldsymbol{s} \odot \boldsymbol{f} + \boldsymbol{n}$$

where $\boldsymbol{s} \in \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$, with

$$\boldsymbol{s}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \boldsymbol{s}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \boldsymbol{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \quad \boldsymbol{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}.$$

Assume that the random variables f_1 and f_2 are zero-mean, independent Gaussian variables with equal variance σ_f^2, and that n_1 and n_2 are zero-mean, independent Gaussian variables as well, with equal variance σ_n^2. Derive the MAP decision strategy and the decision regions.

Problem 2.8: A transmission system uses QPSK with the following signals:

$$s_i(t) = \sqrt{2}A\cos(\omega t + \theta_i) \qquad 0 \le t \le T \qquad i = 1, 2, 3, 4$$

where $\omega = 2\pi/T$ and

$$\theta_i = \frac{\pi}{2}(i-1) \qquad i = 1, 2, 3, 4.$$

Derive the structure of the optimal receiver assuming that the signals are equally likely and transmitted over an AWGN channel.

Problem 2.8: Provide an intuitive or physical explanation of the reason why the MAP detection strategy for known signals $\{s_i(t)\}_{i=1}^{M}$ transmitted over an AWGN channel cannot be expressed as an explicit function of the following integral:

$$\int_{T_0} [r(t) - s_i(t)]^2 \, \mathrm{d}t$$

where T_0 is a suitable time interval at least equal to the transmission interval.

Problem 2.9: Consider transmission of independent and identically distributed antipodal binary symbols $\{a_k\}$ belonging to $\{+1, -1\}$. Assume that these information symbols are coded into symbols $\{c_k\}$ through the following differential encoding rule: $c_k = c_{k-1}a_k$. Show that MAP sequence and MAP symbol detection criteria coincide.

Problem 2.10: Consider the transmission of PSK symbols $\{a_k\}$ belonging to the set $\{\exp(j2\pi\frac{k}{M}),\ k = 0, \ldots, M-1\}$ and suppose that they are encoded through the following differential encoding rule: $c_k = c_{k-1}a_k$. Show that the optimal receiver, according to the MAP sequence detection criterion, reduces to a symbol-by-symbol receiver for coded symbols $\{c_k\}$, emitting symbol decisions $\{\hat{c}_k\}$ from which the estimated information symbols are recovered according to $\hat{a}_k = \hat{c}_k\hat{c}_{k-1}^*$.

Problem 2.11: Suppose that the discrete-time observation at the receiver can be written as $r_k = a_k + n_k$. Suppose that a_k is real and n_k is complex. Show that $\Im\{n_k\}$ is irrelevant for any detection strategy.

Problem 2.12: Let the observation vector be the concatenation of two subvectors:

$$\boldsymbol{r}^{\mathrm{T}} = (\boldsymbol{r}_1^{\mathrm{T}}, \boldsymbol{r}_2^{\mathrm{T}})$$

and assume that the following condition is satisfied:

$$p(\boldsymbol{r}_2^{\mathrm{T}} | \boldsymbol{r}_1^{\mathrm{T}}, m_i) = p(\boldsymbol{r}_2^{\mathrm{T}} | \boldsymbol{r}_1^{\mathrm{T}}) \qquad \forall m_i.$$

Show that vector $\boldsymbol{r}_2$ is *irrelevant*, given $\boldsymbol{r}_1$, in the decision problem and can be discarded (Theorem of irrelevance).

Hint: *formulate the MAP detection strategy in terms of the conditional joint pdf of these vectors and use chain factorization.*

Problem 2.13: Consider an M-ary signaling scheme with signal set $\{s_i(t)\}_{i=1}^{M}$. Assuming signal $s_i(t)$ is sent, the received signal at the output of an AWGN phase-noncoherent channel is

$$r(t) = s_i(t)e^{j\theta} + n(t)$$

where θ is uniformly distributed over 2π.

A. Determine a discretization process for the received signal which provides a sufficient statistic for MAP detection.

B. Derive the noncoherent MAP strategy.

C. Give examples of signal sets suitable for noncoherent detection.

Problem 2.14: Prove that in the case of transmission over a channel with strictly finite memory, (2.33) holds.

3

Sequence Detection: Algorithms and Applications

3.1 Introduction

In this chapter, we consider a generic approach to maximum *a posteriori* (MAP) *sequence detection*, with emphasis on detection over stochastic channels. In particular, the principle of MAP sequence detection, equivalent to the maximum likelihood (ML) sequence detection principle for equal *a priori* sequence probabilities, was introduced in the literature long ago [2]. The most celebrated and used MAP sequence detection algorithm, the *Viterbi algorithm* (VA), was introduced by Viterbi in 1967 and represents a very efficient solution for hard-output sequence detection [96, 97].

In the remainder of this chapter the principle of sequence detection will be first introduced and revisited, recalling the main characteristic of a VA and describing some details about one of its possible extensions, namely the *soft-output Viterbi algorithm* (SOVA) [99]. The application of the general approach for detection over channels with memory proposed in Chapter 2 will then be specialized for the case of sequence detection. We will also consider a general discussion on detection and estimation, culminating with the description of the principle of *per-survivor processing* (PSP) [34].

Detection Algorithms for Wireless Communications - G. Ferrari, G. Colavolpe and R. Raheli
 ISBN: 0-470-85828-1

3.2 MAP Sequence Detection Principle

As seen in Chapter 1, the MAP sequence detection strategy can be formulated, referring to the equivalent discrete-time observation $\boldsymbol{r} = \boldsymbol{r}_0^{K-1}$, where K is the transmission length, as

$$\widehat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmax}}\, P\{\boldsymbol{a}|\boldsymbol{r}\}. \tag{3.1}$$

In particular, a brute-force approach to the identification of $\widehat{\boldsymbol{a}}$ consists of the evaluation of $P\{\boldsymbol{a}|\boldsymbol{r}\}$ for all possible information sequences $\boldsymbol{a}$, selecting the sequence which maximizes the *a posteriori* probability. By chain factorization and owing to the independence of the information symbols, the MAP sequence detection strategy in (3.1) can be formulated in terms of the conditional probability density functions (pdfs) $p(\boldsymbol{r}|\boldsymbol{a})$ and the *a priori* sequence probabilities $P(\boldsymbol{a})$:

$$\begin{aligned} P\{\boldsymbol{a}|\boldsymbol{r}\} \sim p(\boldsymbol{r}|\boldsymbol{a})P(\boldsymbol{a}) &= \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a})P\{a_k\} \\ &= \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k)P\{a_k\} \\ &\sim \sum_{k=0}^{K-1} \left[\ln p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) + \ln P\{a_k\}\right] \end{aligned} \tag{3.2}$$

where the symbol $\sim$ indicates that two quantities are monotonically related with respect to the variable of interest (in this case, $\boldsymbol{a}$); a discrete-time version of the system causality (see Chapter 2) is assumed to hold in the second line; and the monotonicity of the logarithm is used in the third line. In particular, if the *a priori* probabilities $\{P\{a_k\}\}$ are equal, the MAP sequence detection criterion coincides with the ML sequence detection criterion.

The maximization of (3.2) over all possible sequences $\{\boldsymbol{a}\}$ can be implemented as a search of a *path* in a *tree* diagram where each branch is in a one-to-one correspondence with an information symbol a_k. Consequently, a path is in a one-to-one correspondence with a *partial* sequence $\boldsymbol{a}_0^k$ up to epoch k. Assign first a *metric* equal to the k-th term of (3.2) to a branch at epoch k associated with symbol a_k. Defining a path metric as the sum of the metrics of the branches forming that path, the MAP sequence detection strategy implements a search for the path with largest metric in this tree diagram.

As assumed in Chapter 1, the encoder/modulator can be described as a finite state machine (FSM) with state μ_k and characterized by the following "next-state" and

"output" functions:

$$\begin{cases} \mathrm{ns}(\mu_k, a_k) = \mu_{k+1} \\ \mathrm{o}(\mu_k, a_k) = c_k. \end{cases} \tag{3.3}$$

Considering a channel with complex additive white Gaussian noise (AWGN) with circular symmetry, i.e., with independent real and imaginary components, each with variance σ^2—a simple and very common example of memoryless channel—the generic observation at epoch k can be written as

$$r_k = c_k + n_k \tag{3.4}$$

where n_k is the AWGN sample. In this case, it follows that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\mu_k, a_k) = \frac{1}{2\pi\sigma^2} e^{-\frac{|r_k - c_k|^2}{2\sigma^2}} \tag{3.5}$$

and (3.2) can be reformulated as

$$\begin{aligned} P\{\boldsymbol{a}|\boldsymbol{r}\} &\sim \sum_{k=0}^{K-1} \left[-\frac{|r_k - c_k|^2}{2\sigma^2} + \ln P\{a_k\} \right] \\ &\sim \sum_{k=0}^{K-1} \left[-|r_k - c_k|^2 + 2\sigma^2 \ln P\{a_k\} \right]. \end{aligned} \tag{3.6}$$

As mentioned above, a brute-force approach to the implementation of the MAP sequence detection criterion would consist of evaluating (3.6) for all possible information sequences, choosing the maximizing sequence. Assuming M-ary information symbols, the complexity of this brute-force approach would be M^K, i.e., exponential with the transmission length. Hence, this implementation of the MAP sequence detection principle is feasible only for short transmission lengths, whereas it becomes highly inefficient for longer transmission lengths. A much more efficient and appealing MAP sequence detection algorithm is the VA, which will be described in the next section.

3.3 Viterbi Algorithm

In the case of a strictly finite memory channel, MAP sequence detection can be formulated as indicated in (3.6), possibly by redefining symbol c_k and the underlying FSM model. In particular, the optimal tree diagram can be folded into a *trellis diagram*, where the possible states at each epoch are given by all possible values of the

state μ_k of the encoder/modulator FSM. In the remainder of this chapter the number of states of the encoder/modulator FSM will be indicated by S_c. Denoting by $t_k \triangleq (\mu_k, a_k)$ a transition in the trellis diagram, a *branch metric* associated with this transition can be defined as follows:

$$\lambda_k(\mu_k, a_k) = \lambda_k(t_k) \triangleq \ln p(r_k|\mu_k, a_k) + \ln P\{a_k\}. \tag{3.7}$$

Upon the definition of the branch metric (3.7), the *a posteriori* sequence probability can be written as follows:

$$P\{\boldsymbol{a}|\boldsymbol{r}\} \sim \sum_{k=0}^{K-1} \lambda_k(\mu_k, a_k). \tag{3.8}$$

Without entering into the details (the interested reader can find plenty of literature regarding the VA [17,56]), the implementation principle of the VA is that of associating to each state μ_n a *partial path metric* relative to the corresponding path originating from a known state μ_0, at epoch $k = 0$, and terminating into μ_n. This partial path metric, indicated by $\Lambda_n(\mu_n)$, can be written as follows:

$$\Lambda_n(\mu_n) \triangleq \sum_{k=0}^{n-1} \lambda_k(t_k) \qquad 0 \leq n \leq K. \tag{3.9}$$

Obviously, $P\{\boldsymbol{a}|\boldsymbol{r}\} = \Lambda_K(\mu_K)$. Based on the trellis representation of the underlying FSM, the partial metrics, associated with the trellis states, can be computed recursively. For the sake of simplicity, we consider binary information symbols, i.e., $M = 2$. A pictorial description of a step of the VA is shown in Figure 3.1. The path metrics associated with states $\mu_k^{(1)}$ and $\mu_k^{(2)}$ are indicated as $\Lambda_k(\mu_k^{(1)})$ and $\Lambda_k(\mu_k^{(2)})$, respectively. The VA associates with state μ_{k+1} (the common ending state of both transitions $t_k^{(1)} = (\mu_k^{(1)}, a_k^{(1)})$ and $t_k^{(2)} = (\mu_k^{(2)}, a_k^{(2)})$) the following path metric:

$$\Lambda_{k+1}(\mu_{k+1}) = \max\left\{\Lambda_k(\mu_k^{(1)}) + \lambda_k(\mu_k^{(1)}, a_k^{(1)}), \Lambda_k(\mu_k^{(2)}) + \lambda_k(\mu_k^{(2)}, a_k^{(2)})\right\}. \tag{3.10}$$

In this sense, the basic operation of the VA is defined as an *add-compare-select* (ACS) operation, since: (i) the path metrics associated with the two starting states are summed with the branch metrics of the two branches entering into the common final state; (ii) the obtained partial path metrics are compared; and (iii) the largest path metric is selected as *the* path metric associated with μ_{k+1}.

The evaluation of path metrics as indicated above guarantees that, at each state and epoch, the path terminating in that state is, among all entering paths, the one to

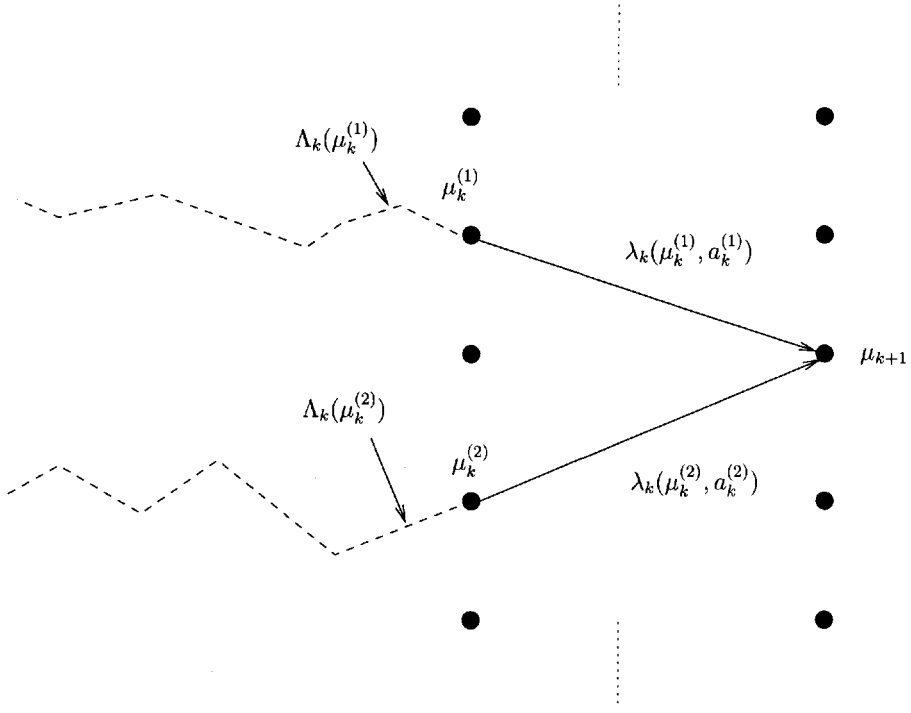


Figure 3.1: Add-compare-select operation in a VA.

which the largest metric is associated. At any epoch, the S_c paths with the largest possible path metrics are therefore tracked. Consequently, in correspondence to the final trellis section at epoch K, the largest path metric among those associated with the final states is such that the corresponding information sequence satisfies the MAP sequence detection criterion in (3.1).

Even if the complexity of the VA is proportional to KS_c (i.e., the dependence on the transmission length is linear and not exponential anymore), the delay would still be unacceptable in the case of long transmission lengths, since one should wait for the transmission of the entire information sequence before being able to output the sequence of symbols satisfying the MAP sequence detection criterion. An appealing feature of the VA, however, consists of the fact that the surviving paths at each state at a given epoch k converge backwards at epoch $k - D$, where D is related to the memory of the underlying FSM. The probability of convergence rapidly approaches 1 for increasing values of D. Hence, at epoch k, it is possible to emit the decision $\widehat{a}_{k-D}$ relative to the final MAP information sequence. In other words, by assuming that consecutive observations are sequentially available, the latency corresponds to

only D symbol intervals.

3.4 Soft-Output Viterbi Algorithm

While the VA represents an efficient algorithm to determine the information sequence satisfying the MAP sequence criterion, it does not provide any measure of the *likelihood* of the estimated sequence. Considering a trellis section at a generic epoch, given that the trellis is *regular*, in the case of binary information symbols there are two branches entering each state, as indicated in Figure 3.1. In this case, an heuristic and reasonable approach to associating a likelihood to the information symbol relative to the winning entering path consists of estimating such a likelihood as the absolute value of the difference between the path metrics of the two entering paths. In fact, the larger the difference between the two path metrics, the more reliable the winning path will be. This is the basic principle of the SOVA.

In particular, as will be made clearer in Chapter 4, the SOVA does not produce soft outputs by considering all paths in the trellis diagram as in the case of the forward backward (FB) algorithm, but only two paths: the maximum likelihood path and its strongest competitor. We do not describe further the SOVA here, since it will be considered in Chapter 4 in the context of iterative detection.

3.5 Finite Memory Sequence Detection

As seen in Section 3.2, the formulation of the MAP sequence detection principle requires, according to (3.2), a search over a *tree*. It was shown that in the case of a memoryless channel the computation of (3.2) can be carried out over a trellis diagram, by using the VA. This conclusion holds also in the case of a channel with strictly finite memory, provided that the state is properly redefined accounting for modulator/encoder and channel memories.

If the causality condition and the finite memory condition (FMC) introduced in Chapter 2 hold, the independence between information symbols and the chain factor-

ization rule allows one to derive the following:

$$\begin{aligned} P\{\boldsymbol{a}|\boldsymbol{r}\} &\sim p(\boldsymbol{r}|\boldsymbol{a})P\{\boldsymbol{a}\} \\ &= \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a})P\{a_k\} \\ &= \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k)P\{a_k\} \\ &= \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C})P\{a_k\}. \end{aligned} \tag{3.11}$$

Note that the last step in (3.11), where the FMC is applied, holds if $k \geq C$, i.e., in a regime situation for the VA. In the initial transient, for $k < C$, (3.11) holds assuming that negative indexes are replaced by 0. In the logarithmic domain, one can write:

$$\ln P\{\boldsymbol{a}|\boldsymbol{r}\} \sim \sum_{k=0}^{K-1} \left[\ln p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) + \ln P\{a_k\}\right]. \tag{3.12}$$

Defining the augmented trellis state and transition as follows:

$$S_k \triangleq (\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}) \tag{3.13}$$

$$T_k \triangleq (S_k, a_k) = (\boldsymbol{a}_{k-C}^k, \mu_{k-C}) \tag{3.14}$$

the k-th term in (3.12) can be considered as a branch metric, i.e., one can define:

$$\lambda_k(T_k) \triangleq \ln p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) + \ln P\{a_k\}. \tag{3.15}$$

At this point, one can immediately recognize that the MAP sequence detection principle can be implemented with a VA running over a trellis with state S_k and branch metric (3.15). The state S_k is augmented with respect to the state μ_k of the encoder/modulator. This corresponds to considering *combined* detection and decoding. The next-state function $\mathrm{NS}(\cdot,\cdot)$, describing the transitions in the augmented trellis (and, consequently, the operations of the VA), can be straightforwardly expressed as follows:

$$S_{k+1} = \mathrm{NS}(S_k, a_k) = (\boldsymbol{a}_{k-C+1}^k, \mathrm{ns}(\mu_{k-C}, a_{k-C})). \tag{3.16}$$

In Chapter 2, we showed that under the general stochastic channel model (2.34), if the conditional Markov property (CMP) (2.35) holds, the branch metric (3.15) can be further simplified as follows:

$$\lambda_k(T_k) \triangleq \ln p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) + \ln P\{a_k\} \tag{3.17}$$

where N is the order of Markovianity, and the finite memory parameter can be written as $C = N + L$, where L is an integer characterizing the expression of the observation (e.g., it can take into account the presence of inter-symbol interference, ISI). In particular, a generic model for the observation at the output of a stochastic channel could be the following:

$$r_k = g_{\text{ch}}(\boldsymbol{\theta}_0^k, \boldsymbol{a}_{k-L}^k, \mu_{k-L}) + n_k \tag{3.18}$$

where $g_{\text{ch}}(\cdot,\cdot,\cdot)$ is a known deterministic function, $\{\theta_k\}$ is the sequence of (possibly vectorial) channel stochastic parameters, and $\{n_k\}$ is the additive noise process. Note that the first term of the branch metric (3.17) is the logarithm of a conditional pdf in which the conditioning part is given by a window of observations of length N. Assuming that, at the receiver, the statistics of the stochastic channel parameters are available, the branch metric (3.17) can be written as follows:

$$\begin{aligned}\lambda_k(T_k) &= \ln \mathrm{E}_{\boldsymbol{\theta}_0^k}\left\{p(\boldsymbol{r}_{k-N}^k | T_k, \boldsymbol{\theta}_0^k)\right\} - \ln \mathrm{E}_{\boldsymbol{\theta}_0^{k-1}}\left\{p(\boldsymbol{r}_{k-N}^{k-1} | S_k, \boldsymbol{\theta}_0^{k-1})\right\} \\ &\quad + \ln P\{a_k\}.\end{aligned} \tag{3.19}$$

Although important from a conceptual viewpoint, expression (3.19) for the branch metric is not operative because of its dependence from a sequence of k stochastic parameters. In a special (although very important) case, the observation can be written as follows:

$$r_k = g_{\text{ch}}(\theta_k, \boldsymbol{a}_{k-L}^k, \mu_{k-L}) + n_k. \tag{3.20}$$

In this case, (3.19) specializes to the following:

$$\begin{aligned}\lambda_k(T_k) &= \ln \mathrm{E}_{\boldsymbol{\theta}_{k-N}^k}\left\{p(\boldsymbol{r}_{k-N}^k | T_k, \boldsymbol{\theta}_{k-N}^k)\right\} - \ln \mathrm{E}_{\boldsymbol{\theta}_{k-N}^{k-1}}\left\{p(\boldsymbol{r}_{k-N}^{k-1} | S_k, \boldsymbol{\theta}_{k-N}^{k-1})\right\} \\ &\quad + \ln P\{a_k\}.\end{aligned} \tag{3.21}$$

The observation model (3.20) suggests also another generic approach for the computation of the branch metric in this case. In fact, one could consider a *suitable* estimate[1] $\widehat{\theta}_k$ based on the previous observations. A general expression for such an estimate is

$$\widehat{\theta}_k = \widehat{\theta}_k(\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}) = \widehat{\theta}_k(S_k). \tag{3.22}$$

Note that while the computation of the branch metric according to (3.21) is, given statistical knowledge of the channel stochastic parameters, *exact*, the computation of

[1]The criterion for the computation of $\widehat{\theta}_k$ could be any "reasonable" criterion.

the branch metric using an estimate as in (3.22) may not be exact—for example, in the case of frequency nonselective fading channels for which the estimate is a minimum mean square error (MMSE) estimate based on linear prediction, the obtained metric is exact. In order to clarify the two proposed approaches for the computation of the branch metric (3.17), two interesting examples are considered.

3.5.1 Inter-Symbol Interference Channel

As an important specific application, we consider a communication system where a linearly modulated and possibly encoded signal is transmitted over a static dispersive AWGN channel, i.e., an ISI channel. Being a static channel, its deterministic response can be measured or estimated with the desired accuracy; hence, we assume it is perfectly known at the receiver. After discretization, normally achieved by filtering and sampling (see the Appendix for more details), this channel can be modeled by the following mapping from the information sequence $\{a_k\}$ to the discrete observable:

$$r_k = \sum_{l=0}^{L} f_l c_{k-l} + n_k \tag{3.23}$$

where: $\{f_l\}_{l=0}^{L}$ denotes the overall *white noise discrete equivalent* of the ISI channel [100]; $\{c_k\}$ is a code sequence at the output of a channel encoder; and $\{n_k\}$ is an independent and identically distributed (iid) Gaussian noise sequence with variance σ^2 per component. All quantities in (3.23) may be complex in order to model the discrete-time equivalent of a bandpass system.[2] The code sequence is in a one-to-one correspondence with the information sequence according to the encoder/modulator FSM structure characterized by (3.3).

For the system model in (3.23), the conditional pdf of the observation at epoch k has the following form:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{a}_0^k). \tag{3.24}$$

In fact, given the information sequence $\boldsymbol{a}_0^k$, the coded sequence is specified; in particular the most recent $L+1$ code symbols in (3.23) are uniquely determined and the conditional pdf of r_k in the right-hand side of (3.24) is equal to that of the noise sample n_k shifted by the signal component, i.e., the summation in (3.23). Defining an augmented system state in the usual manner, i.e.:

$$S_k = (a_{k-1}, a_{k-2}, \ldots, a_{k-C}, \mu_{k-C}) \tag{3.25}$$

[2] In this case, the noise sequence is circularly symmetric.

where $C \geq L$, the dependence of the conditional observation on the information sequence can be formulated as a dependence on the current information symbol a_k and the system state S_k, i.e.:

$$p(r_k|\boldsymbol{a}_0^k) = p(r_k|a_k, S_k) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|r_k - x(a_k, S_k)|^2}{2\sigma^2}\right) \tag{3.26}$$

where the notation

$$x(a_k, S_k) \triangleq \sum_{l=0}^{L} f_l c_{k-l} \tag{3.27}$$

has been introduced. The assumed FSM model is a finite memory system by definition and complies with (2.17) by a suitable choice of the total memory C, which may exceed L to account for the code memory.

Based on (3.23), we can now easily derive the branch metric (3.15) in the form

$$\begin{aligned} \lambda_k(a_k, S_k) &= \ln p(r_k|a_k, S_k) + \ln P\{a_k\} \\ &\sim -|r_k - x(a_k, S_k)|^2 + 2\sigma^2 \ln P\{a_k\}. \end{aligned} \tag{3.28}$$

In the special case of equally likely information symbols, the *a priori* probabilities are irrelevant and the following branch metric is sufficient:

$$\lambda_k(a_k, S_k) \sim -|r_k - x(a_k, S_k)|^2 \,. \tag{3.29}$$

3.5.2 Flat Slow Fading Channel

As a second important application, let us consider transmission of a linearly modulated, possibly encoded, signal over a flat slowly fading channel. After discretization, the observation model is

$$r_k = f c_k + n_k \tag{3.30}$$

where f is a circular complex Gaussian random variable, c_k is the code symbol and n_k is an AWGN noise sample. This discrete channel model is a special case of (3.23), in the sense that there is no ISI (i.e., $L = 0$), but it is an extension of (3.23), in the sense that the channel coefficient f is a random variable. Being a static channel, the discretization process can be identical to that used in the derivation of (3.23).

For the system model (3.30), the conditional observation is Gaussian and its dependence on the information sequence can be formulated as

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = \frac{1}{\pi\overline{\sigma}_k^2(\boldsymbol{a}_0^k)} \exp\left(-\frac{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2}{\overline{\sigma}_k^2(\boldsymbol{a}_0^k)}\right) \tag{3.31}$$

where the conditional mean and variance are

$$\overline{r}_k(\boldsymbol{a}_0^k) \triangleq \mathrm{E}\{r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\} \tag{3.32}$$

$$\overline{\sigma}_k^2(\boldsymbol{a}_0^k) \triangleq \mathrm{E}\{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2 \, |\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\} \, . \tag{3.33}$$

Unfortunately, the dependence of the conditional mean and variance, hence the observation, on the information sequence is not strictly finite memory in the sense of (2.17). This FMC can be *imposed* as an approximation in order to force the applicability of the proposed finite memory sequence detection algorithms. In particular, one can consider the following approximate conditional mean and variance (conditioned on a finite sequence of previous observations):

$$\overline{r}_k^{(N)}(S_k, a_k) \triangleq \mathrm{E}\{r_k|\boldsymbol{r}_{k-N}^{k-1}, S_k, a_k\} \tag{3.34}$$

$$\overline{\sigma}_k^{2|(N)}(S_k, a_k) \triangleq \mathrm{E}\{|r_k - \overline{r}_k^{(N)}(S_k, a_k)|^2 \, |\boldsymbol{r}_{k-N}^{k-1}, S_k, a_k\} \, . \tag{3.35}$$

The approximation quality improves with increasing values of the *assumed* memory N, which is now a design parameter. Selecting a sufficiently large value for N may entail minor performance degradation. It is possible to show that for the considered Gaussian observation model in (3.30), the approach proposed in this subsection is equivalent to linear predictive detection [101] (see Problem 3.8).

3.6 Estimation-Detection Decomposition

In the previous section, we derived a general VA for combined detection and decoding of an information bearing signal transmitted through a channel, possibly modeled as stochastic and time-varying. We showed that a starting point is the availability of a statistical characterization of the overall mapping from information sequence $\boldsymbol{a}$ to discrete observable $\boldsymbol{r}$. If this mapping is causal and the FMC holds, sequence detection algorithms,[3] based on the branch metric defined in (3.15), can be derived.

The FMC holds in important situations. As shown in Section 3.5.1, transmission on known static dispersive AWGN channels is an important case where this condition is strictly verified. However, there are other important applications in which the FMC does not hold, such as those where the channel inflicts some stochastic alteration upon the transmitted signal, besides thermal noise. At the end of Section 3.5, we considered explicitly the special case of flat slow fading, which is the simplest

[3]Note that in Chapter 4 we will show that the same approach is also the basis for the derivation of FB algorithms to implement MAP symbol detection.

stochastic channel model because it depends on just one complex Gaussian random parameter. Even in this simple case, we observed that the channel-induced memory is not strictly finite and (2.17) does not hold. In fact, the conditional pdf of the observation (3.31) depends on the entire previous information sequence $\boldsymbol{a}_0^k$ through the conditional mean and variance. This unlimited memory feature is present in other simple stochastic models, such as the phase noncoherent channel, characterized by a single random parameter, i.e., the channel phase. Nonetheless, the statistics of a conditional observation depend on all previous observations.

In general, we may conclude that any channel model described in terms of stochastic time-invariant parameters yields a nonfinite memory in the sense of (2.17). As a consequence, optimal sequence detection algorithms can be exactly implemented only by resorting to some type of exhaustive search accounting for all possible transmission acts. The relevant implementation complexity increases exponentially with the length of transmission, i.e., the number of transmitted information symbols K. Hence, optimal detection is implementable only for very limited transmission lengths. Practical transmission lengths are, however, much larger with respect to values compatible with a tractable implementation, even for block or packet transmissions, and the problem of designing suitable *suboptimal* approximate detection algorithms arises.

A general method for devising detection algorithms for transmission systems with stochastic channels, such that the FMC (2.17) does not hold, can be based on the idea of "decomposing" the data detection and parameter estimation functions. This design approach allows one to derive detection algorithms under the assumption of knowledge, to a certain degree of accuracy, of the (channel) parameters and to devise an estimation algorithm for extracting information about the parameters, which is necessary in the detection process. If statistical knowledge about the parameters is available, it can be exploited in this estimation process (but not in the detection process). The detection function can be designed under the assumption of perfect knowledge of the parameters, or it can incorporate a statistical characterization of the estimated parameters (instead of the true ones). We remark, however, that the latter option is often too complex to be pursued.[4]

This general decomposed estimation-detection design approach has the inherent feature of not requiring statistical knowledge about the parameters. In fact, by a subtle modification of the modeling assumptions, stochastic static parameters can be viewed as just unknown quantities and modeled as *deterministic*; hence, estimation algorithms for unknown deterministic parameters can be employed. If a statistical characterization of the channel parameters is not available, this approach is a viable

[4]If a detection algorithm accounting for a statistical parameter characterization is implementable, one would devise it taking directly into account the statistics of the true parameters.

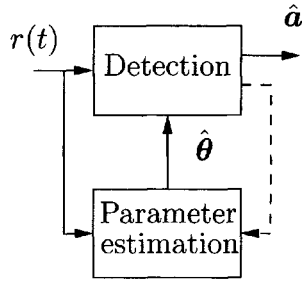


Figure 3.2: Receiver based on estimation-detection decomposition.

alternative.

This estimation-detection approach has the great advantage of conceptually decoupling the detection and estimation problems and physically simplifying the receiver implementation. A general block diagram depicting a receiver based on this decomposed estimation-detection approach is shown in Figure 3.2, where $\widehat{\boldsymbol{\theta}}$ denotes the estimate of an unknown parameter vector $\boldsymbol{\theta}$. The figure suggests that the estimation process observes explicitly the received signal $r(t)$ and possibly the detected data sequence $\widehat{\boldsymbol{a}}$—the latter being an option to be investigated in the following sections. We remark that this approach must be viewed as a logical ad-hoc solution and that no claim of optimality can be made in general. The optimal solution, with minimal error probability, can only be attained if the statistical information about the parameter is known and exploited directly in the detection process.

In general terms, we may refer to the presence of stochastic or unknown deterministic parameters as *parametric uncertainty* affecting the detection problem. Parametric uncertainty can be of a static nature, as in the previous examples, or it can be time-varying, as in several channel models described in Chapter 1. Time-varying stochastic parameters are in fact random processes, but if their correlation properties are such that their time variation is slow enough, they can be considered static in the detection process. As a consequence, one can view these parameters as static in the detection algorithm and account for their slow time variation in the estimation process, which must be capable of *tracking* the parameter values. A critical issue affecting the performance of decoupled estimation-detection schemes in the presence of time-varying (channel) parameters is the ability of the estimation function to track their time variation—the faster this time variation, the more critical the estimation and tracking function.

The reader may wonder whether the previously described unlimited memory caused by a time-invariant stochastic parameter is alleviated when this parameter

is a time-varying random process. The answer in general is negative as can be concluded by considering the case of flat fading, which is one of the simplest examples of a stochastic time-varying channel model. Let us assume that, after discretization, the observation can be modeled as

$$r_k = f_k c_k + n_k \tag{3.36}$$

where $\{f_k\}$ is a circular complex Gaussian random sequence, c_k is the code symbol, and n_k is an AWGN noise sample with variance σ^2 per component. The conditional statistics of the observation (3.36) can be again described by the pdf (3.31). The difference, with respect to the case of slow fading considered in Section 3.5, is implicit in the conditional mean and variance, which now take into account the fading correlation properties. Nonetheless, these conditional moments are still dependent on the entire information sequence and the FMC (2.17) does not hold.

In Section 2.2, it was noted that a discretization process capable of providing a sufficient statistic in the presence of time-invariant random channel parameters exists. If the parameters are modeled as random processes, the discretization of the received signal must account for their time variation. As an example, if the discretization of the received signal is achieved by time sampling, a larger number of samples per signaling interval might be necessary with respect to the number sufficient in a static random channel. The model (3.36) characterized by one sample per code symbol, i.e., per signaling interval, is in fact only an approximation to be considered valid if the bandwidth of the fading process is significantly smaller than that of the transmitted signal. In the time domain, this condition means that the fading sequence is highly correlated, so that it changes very slowly from one signaling interval to the following one.

In order to introduce a formal framework for estimation-detection decomposition, one may recall the results of Section 3.5: MAP sequence detection algorithms can be solely based on the branch metrics (3.15), provided that the FMC is verified. Let us assume that this condition is verified provided some *undesired* parameters, collected into a vector $\boldsymbol{\theta}_k$, are known. For generality, we model this parameter vector as time-varying, as denoted by the time index k. Viewing these undesired parameters as stochastic, we modify the notation to incorporate them and denote the conditional pdf of the observation as $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \boldsymbol{\theta}_k)$. With this notation, a *parameter-conditional* analog of the FMC can be formulated as

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \boldsymbol{\theta}_k) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}, \boldsymbol{\theta}_k) \ . \tag{3.37}$$

To exemplify this condition, let us consider the simple flat fading channel model (3.36). As observed, the FMC is not verified for this model, basically because of

the random fading sequence $\{f_k\}$. In fact, considering this sequence as an undesired parameter, i.e., letting $\boldsymbol{\theta}_k = f_k$ in (3.37), the conditional observation becomes

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, f_k) = p(r_k|c_k, f_k) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|r_k - f_k c_k|^2}{2\sigma^2}\right). \tag{3.38}$$

As the code symbols are the output of the FSM relative to the encoder/modulator, the conditional FMC (3.37) is verified for some value of C. In the special case of an uncoded system, $c_k = a_k$ and $C = 0$; in other words, the observation becomes conditionally memoryless.

This simple example shows that by a clever choice of the parameters to be included in the set of undesired ones, it is possible to transform the transmission system into a conditionally finite memory system. Of course, this property holds conditional on the undesired parameters, i.e., only if they are known. This remark is the key to a decomposed estimation-detection design: one can assume that some undesired parameters are known in devising the detection algorithms, thus avoiding intractable complexity, and can devote some implementation complexity to the estimation of these undesired parameters. These detection algorithms can be simply derived by using a parameter-conditional branch metric obtained by replacing the unlimited memory pdf $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k)$ with its conditionally finite memory counterpart $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \boldsymbol{\theta}_k)$ and computing the corresponding branch metric for an estimated parameter value, in an attempt to approximate the true parameter. Sequence detection algorithms for the estimation of the undesired parameters are the subject of the following Section 3.7.

We conclude this section with a terminological remark. In the technical literature, the word *synchronization* is often used as a synonym of estimation when the channel parameters of interest are the timing epoch, the carrier phase or the carrier frequency [46, 102]. These specific channel parameters must be estimated in virtually any passband communication system,[5] so that a general design methodology can be based on the synchronization-detection decomposition depicted in Figure 3.2, in which the estimation block can be equivalently termed "synchronization block." Although in the following we continue to refer to parameter estimation in general terms, this terminological remark is appropriate.

3.7 Data-Aided Parameter Estimation

In this section, we address the problem of effectively estimating the undesired parameter vector $\boldsymbol{\theta}_k$ in order to provide the detection subsystem with an estimate $\widehat{\boldsymbol{\theta}}_k$

[5] A base-band system requires the estimation of the timing epoch only.

in accordance with the estimation-detection decomposed design shown in Figure 3.2. This estimation problem can be viewed as the "dual" of the detection problem considered in Section 3.6. In fact, the undesired parameters are now parameters of interest which we would like to estimate, whereas the "parameters" of interest in the detection process, namely the data symbols, are now just nuisance (or undesired) parameters.[6] As the knowledge of the undesired parameters simplifies significantly the detection problem, possible knowledge of the data sequence would facilitate the estimation of these undesired parameters. This is quite intuitive, as the overall "randomness" of the received signal is due to all possible sources, including the data symbols; hence, having exact knowledge of these symbols may reduce the degree of randomness and facilitate the estimation of the parameters of interest. Since in coded systems the transmitted signal is directly modulated by the code sequence, knowledge of this sequence is particularly helpful. In this section, we assume that the code sequence plays the role of a data sequence aiding the parameter estimation process.

A general formal description of a data-aided parameter estimator can be obtained by considering the functional dependence of the estimator on the time-continuous received signal $r(t)$ and the code sequence $\boldsymbol{c}$ according to

$$\widehat{\boldsymbol{\theta}}_k \triangleq \overline{\boldsymbol{g}}_k \left[r(t), \boldsymbol{c}\right] \tag{3.39}$$

where $\overline{\boldsymbol{g}}_k[\cdot, \cdot]$ denotes this (vector) functional dependence. If the discretization process is such that the observed sequence $\boldsymbol{r}$ is also a sufficient statistic for the estimation of the undesired parameter $\boldsymbol{\theta}_k$, expression (3.39) can be equivalently formulated in discrete time as follows:

$$\widehat{\boldsymbol{\theta}}_k = \boldsymbol{g}_k \left[\boldsymbol{r}, \boldsymbol{c}\right] \tag{3.40}$$

where $\boldsymbol{g}_k[\cdot, \cdot]$ denotes the functional dependence on the time-discrete sequences $\boldsymbol{r}$ and $\boldsymbol{c}$.

Conditions for the statistical sufficiency of the discrete observable $\boldsymbol{r}$ are considered in Appendix A. If this discrete observation is a sufficient statistic for the original detection problem, it is intuitive that it is so also for the estimation-detection decomposed design. On the other hand, it is intuitive that the simplified detection problem based on the knowledge of the undesired parameters would allow a simplified discretization of the received time-continuous signal. Should this be the case, one should pay attention to the fact that the simplified discretization, sufficient for detection, could not be sufficient for parameter estimation. With this cautionary remark, we assume that the discrete observable is indeed a sufficient statistic for both detection and estimation and we proceed with a discrete-time model.

[6] We recall that the word "detection" is used to exactly mean estimation of a discrete-valued quantity.

The general data-aided estimator (3.40) bases its operation on the sequence of observations and information symbols. This means that an estimate of the "present" parameter, at the k-th epoch, is based on "past" as well as "future" signal observations and data symbols. This assumption is not compatible with the causality property of any physical system; we must therefore restrict the observation sequence up to the current epoch. Regarding the data symbols, should they be exactly known at the receiver, they could be entirely exploited in parameter estimation. However, it is typical that in tracking the time variation of the undesired parameters the estimator uses decisions as approximation of the true data: in this *decision-directed* realization, a causal dependence of the estimator on the data sequence must be assumed as well. As a consequence, we need to modify the data-aided parameter estimator (3.40) in order to incorporate this causality property. As the observation of a few successive signal samples and the knowledge of further data symbols with respect to the current instant may improve the estimation quality, in order to meet the causality condition we could consider the possibility of using current observations for estimating a previous value of the parameter, or we could introduce an *estimation lag* [2,103]. Nonetheless, there might be cases in which a future parameter estimate is useful in the detection process. This case corresponds to a negative estimation lag and is referred to as *prediction*. Finally, we could consider the possibility that in decision-directed operation the data symbols are available after some decision delay. The resulting formalization of a generic causal data-aided estimator is

$$\widehat{\boldsymbol{\theta}}_{k-l} = \boldsymbol{g}_{k-l}\left[\boldsymbol{r}_0^k, \boldsymbol{c}_0^{k-d}\right] \tag{3.41}$$

where l and d are integers denoting the *estimation delay* and the delay affecting the *available data sequence*, respectively.

Expression (3.41) emphasizes that an estimate of the value of the parameter vector at epoch $k-l$ is obtained at epoch k observing the received sequence up to the current epoch k and knowing the data sequence up to epoch $k-d$. In particular, for $d=1$ the most recent data symbol available for this estimation function, i.e., c_{k-1}, is used. The condition $d=1$ is descriptive of the fact that in decision-directed operation the current estimation of the parameter is used in the detection of the current data symbol; hence, only the immediately previous symbol is available for parameter estimation, at most. We sometimes refer to this minimal value of d, which verifies the causality condition, as *zero delay*. In other words, an estimator which is causal with respect to the data sequence requires $d \geq 1$. Negative or zero values of delay d are only compatible with strictly known data symbols (including future ones). For $l \geq d$ an estimate of the parameter at the current or previous time instants with respect to the available data sequence is performed; in particular, for $l = d$ there is no estimation lag. On the contrary, for $l < d$ a *prediction* is performed. We remark

that a delay of l time instants is inherent in the estimate $\widehat{\boldsymbol{\theta}}_k$ with respect to the true parameter vector. Hence, a good estimator would provide $\widehat{\boldsymbol{\theta}}_k \simeq \boldsymbol{\theta}_{k-l}$.

3.8 Joint Detection and Estimation

In the previous section, we introduced a decomposed estimation-detection design method. We showed that a significant reduction of the system memory, hence the implementation complexity, can be achieved by a clever choice of some undesired parameters $\boldsymbol{\theta}_k$. If these parameters were perfectly known, the branch metrics (3.15) could be computed on the basis of the parameter-conditional finite memory pdf $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_k, \boldsymbol{\theta}_k)$. In Section 3.7, the problem of estimating these undesired parameters by means of data-aided techniques was considered under the assumption of perfectly known data, i.e., code sequence. In this section, we consider the combination of these detection and estimation functions.

The detection algorithm is obtained by computing the branch metrics on the basis of the parameter-conditional finite memory pdf (3.37) with the true parameter vector $\boldsymbol{\theta}_k$ replaced by its estimate $\widehat{\boldsymbol{\theta}}_k$. Hence, the branch metric used in the desired sequence detection algorithm can be expressed as

$$\lambda_k(a_k, S_k) = \ln p(r_k|\boldsymbol{r}_0^{k-1}, a_k, S_k, \widehat{\boldsymbol{\theta}}_k) + \ln P\{a_k\} \tag{3.42}$$

with proper definition of trellis state S_k. The parameter estimate used in (3.42) is obtained using a data-aided estimator according to (3.41). The central point is then the choice of the code sequence $\boldsymbol{c}$ to be used in this parameter estimation.

Usually, transmission systems employ a known *training* data sequence to aid the parameter estimator. This aiding data sequence is transmitted during a training period of short duration in order to allow the estimator to provide the detector with a sufficiently reliable parameter estimate. Once this estimate has been obtained it can be used for data detection. If, however, the parameter is time-varying, it could be *tracked* in its time variation by simply using previous data decisions to aid the parameter estimator, i.e., in a decision-directed mode.

A general scheme, depicting the operation of a receiver designed according to the estimation-detection decomposition in the training and tracking modes, is shown in Figure 3.3. In this figure, the received signal $r(t)$ is first discretized, possibly by filtering and time sampling, to yield the sequence $\{r_k\}$. These discrete observables are used in both data detection and parameter estimation. The detection block outputs the detected information sequence $\{a_{k-D}\}$ with a delay D accounting for the overall detection delay of the considered detection scheme. Specifically, this delay can be the typical detection delay of the VA in sequence detection (or the processing delay of

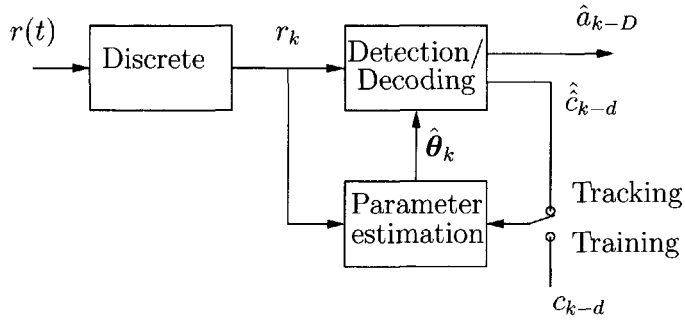


Figure 3.3: Training and tracking operational mode.

the FB algorithm in symbol detection, as shown in Chapter 4), which might include the latency due to the sequence duration. The detection block also outputs a detected code sequence $\{\widehat{\widehat{c}}_{k-d}\}$ with a delay d, usually different than D, which is used in the decision-directed tracking mode for parameter estimation.[7] This emphasizes that the information and code sequences may not verify the coding rule (3.3). The reason why a different value of detection delay is used in parameter estimation, with respect to the delay in the output information sequence, is that the former directly carries over to a delay in the parameter estimate. For this reason, it may be useful to have $d \leq D$. We refer to d as *preliminary* or *tentative* decision delay, to emphasize the fact that the decisions used in parameter estimation are made earlier with respect to the final decisions delayed by D epochs. In the training mode the detected data are replaced by the true data as an aiding sequence. For temporal consistency, in Figure 3.3 the true data are delayed by the same delay d.

3.8.1 Phase-Uncertain Channel

As an example of joint data detection and parameter estimation, let us consider a phase-uncertain channel, as depicted in Figure 1.1 (d). The discrete observable can be modeled as

$$r_k = e^{j\theta_k} c_k + n_k \tag{3.43}$$

where: $\{\theta_k\}$ is the phase rotation process introduced by the channel; $\{c_k\}$ are the code symbols; and $\{n_k\}$ are AWGN samples with variance σ^2 per component. Similarly to the case of flat fading, modeling $\{\theta_k\}$ as a discrete-time random process

[7] Given the different detection delay of the detected information sequence with respect to the code sequence used for parameter estimation, we denote the latter by a double hat.

causes unlimited memory, with the additional drawback that, given the data sequence, the observation is not even conditionally Gaussian. However, if θ_k is known, the parameter-conditional FMC (3.37) holds and one can write:

$$\begin{aligned} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \theta_k) &= p(r_k|a_k, \mu_k, \theta_k) \\ &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|r_k - e^{j\theta_k} \mathrm{o}(a_k, \mu_k)|^2}{2\sigma^2}\right) \end{aligned} \tag{3.44}$$

where $\mathrm{o}(a_k, \mu_k) = c_k$ is the output of the FSM characterized by (3.3). In the special case of an uncoded system, $c_k = a_k$ and the observation becomes conditionally memoryless ($C = 0$).

In order to exploit this conditional FMC, we can decompose the phase synchronization and detection functions. A data-aided phase estimate $\widehat{\theta}_k$ can be obtained through a first order phase-locked loop (PLL) according to the recursion [46]

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k + \eta\, \Im\left\{r_{k-d}\, e^{-j\widehat{\theta}_k}\, c_{k-d}^*\right\} \tag{3.45}$$

where: (i) the parameter η controls the loop bandwidth, (ii) $\Im(\cdot)$ denotes the imaginary part of a complex number and (iii) operation in the training mode is assumed. The estimated phase is inherently delayed by d discrete time instants. However, as (3.45) is relative to the training mode, parameter d can be chosen arbitrarily, except for the causality condition upon the observation which imposes $d \geq 0$. Clearly, the value $d = 0$ is convenient because it reduces the loop delay to the minimum possible value.

The estimated phase (3.45) can be used in place of the true unknown phase in the computation of the branch metric. From (3.44), the following branch metric is obtained:

$$\begin{aligned} \lambda_k(a_k, \mu_k) &= \ln p(r_k|a_k, \mu_k, \widehat{\theta}_k) + \ln P\{a_k\} \\ &\sim -|r_k - e^{j\widehat{\theta}_k}\, c_k|^2 + 2\sigma^2 \ln P\{a_k\} \\ &= -|r_k\, e^{-j\widehat{\theta}_k} - c_k|^2 + 2\sigma^2 \ln P\{a_k\}\,. \end{aligned} \tag{3.46}$$

Equations (3.45) and (3.46) show that the receiver detection and synchronization functions can be based on the phase synchronized observation

$$\overline{r}_k \triangleq r_k\, e^{-j\widehat{\theta}_k}\,.$$

In the decision-directed tracking mode, the data sequence aiding the phase synchronizer is replaced by the sequence of tentative decisions $\{\widehat{c}_{k-d}\}$ yielding the modified recursion

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k + \eta\, \Im\left\{\overline{r}_{k-d}\, \widehat{c}_{k-d}^*\right\}\,. \tag{3.47}$$

Unlike (3.45), the tentative decision delay must now comply with the causality condition upon the detected data, which implies $d \geq 1$. The estimated phase (3.47) can be used in the computation of the branch metric (3.46).

3.8.2 Dispersive Slow Fading Channel

Another example of joint detection and estimation can be considered for a dispersive fading channel as in Figure 1.1 (c). Assuming a slow time variation, a discretization based on sampling with one sample per signaling interval may be sufficient. We relax, to some extent, this statistical sufficiency assumption by modeling the thermal noise as an iid sequence.[8] Hence, the overall mapping from information sequence to observation can be formulated as

$$r_k = \sum_{l=0}^{L} f_{l,k} c_{k-l} + n_k = \boldsymbol{f}_k^{\mathrm{T}} \boldsymbol{c}_k + n_k \tag{3.48}$$

where: $\{f_{l,k}\}_{l=0}^{L}$ denotes the overall time-varying discrete equivalent impulse response at the k-th instant and is collected as an $L+1$ column vector $\boldsymbol{f}_k = (f_{0,k}, f_{1,k}, \ldots, f_{L,k})^{\mathrm{T}}$; $\{c_k\}$ is a code sequence at the output of a channel encoder and is collected as an $L+1$ column vector $\boldsymbol{c}_k = (c_k, c_{k-1}, \ldots, c_{k-L})^{\mathrm{T}}$; and $\{n_k\}$ is an iid Gaussian noise sequence. In fading channels, $\boldsymbol{f}_k$ is a circularly symmetric complex Gaussian random vector, as a result of the Gaussian nature of the fading weights of the physical channel and the linear filtering and sampling operation performed for signal discretization by the receiver front-end [19].

The observation model (3.48) is just a generalization of the flat fading model (3.36), in the sense that it accounts for the ISI generated by the channel time dispersion. The statistics of the observation sequence $\{r_k\}$ are again, given the data sequence, Gaussian, and can be characterized by the conditional pdf (3.31), repeated here for convenience:

$$p(r_k | \boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = \frac{1}{\pi \overline{\sigma}_k^2(\boldsymbol{a}_0^k)} \exp\left(- \frac{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2}{\overline{\sigma}_k^2(\boldsymbol{a}_0^k)} \right) \tag{3.49}$$

with conditional mean

$$\overline{r}_k(\boldsymbol{a}_0^k) = \mathrm{E}\{r_k | \boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\} \tag{3.50}$$

and conditional variance

$$\overline{\sigma}_k^2(\boldsymbol{a}_0^k) = \mathrm{E}\{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2 \, | \boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\} \, . \tag{3.51}$$

[8] We are assuming a suboptimal receiver front-end based on a fixed receiver filter instead of a filter matched to the channel response.

Clearly, the dependence of the conditional mean and variance, hence, the observation, on the information sequence is not strictly finite memory in the sense of (2.17). The differences between the cases of (i) dispersive fading considered here, (ii) flat fading considered in Section 3.6 and (iii) slow flat fading addressed in Section 3.5 lie solely in the conditional mean and variance, which incorporate the specific fading statistics.

As for the previously considered stochastic channel models, the unlimited memory property exhibited by (3.49) suggests an estimation-detection decomposition in which the equivalent fading channel vector $\boldsymbol{f}_k$ is viewed as an undesired parameter. In fact, knowing exactly the channel realization $\boldsymbol{f}_k$ eliminates the stochastically induced unlimited memory, leaving the system with the ISI and the possible coding memory. As in previous similar cases, the resulting parameter-conditional branch metric is

$$\begin{aligned}
\lambda_k(a_k, S_k) &= \ln\left[p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k, \boldsymbol{f}_k)P\{a_k\}\right] \\
&= \ln\left[p(r_k|a_k, S_k, \boldsymbol{f}_k)P\{a_k\}\right] \\
&= \ln\left[\frac{1}{2\pi\sigma^2}\exp\left(-\frac{|r_k - \boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{c}_k|^2}{2\sigma^2}\right)P\{a_k\}\right] \\
&\sim -|r_k - \boldsymbol{f}_k^{\mathrm{T}}\boldsymbol{c}_k|^2 + 2\sigma^2 \ln P\{a_k\}
\end{aligned} \tag{3.52}$$

where the system state S_k is defined as in (3.25), σ^2 denotes the variance of each component (real and imaginary) of the complex noise, and $\boldsymbol{c}_k = (c_k, c_{k-1}, \ldots, c_{k-L})^{\mathrm{T}}$ denotes the code symbol vector uniquely associated with the considered trellis branch (a_k, S_k), in accordance with the considered coding rule.

In the estimation-detection decomposition, the branch metric (3.52) must be computed replacing the true undesired parameter vector $\boldsymbol{f}_k$ with a data-aided estimate. A simple method for estimating this overall channel impulse response may be based on adaptive identification techniques [104]. The least mean squares (LMS) algorithm minimizes the mean square error between the observed signal and the output of a channel identification filter with adaptive coefficients[9] $\widehat{\boldsymbol{f}}_k$. The use of a standard gradient algorithm for channel identification [105] leads to the following recursion:

$$\widehat{\boldsymbol{f}}_{k+1} = \widehat{\boldsymbol{f}}_k + \beta\left(r_{k-d} - \widehat{\boldsymbol{f}}_k^{\mathrm{T}}\boldsymbol{c}_{k-d}\right)\boldsymbol{c}_{k-d}^* \tag{3.53}$$

where the parameter β must be selected as a compromise between adaptation speed and algorithm stability.

Clearly, the LMS identification algorithm (3.53) is a *data-aided* identification technique. In fact, the code vector $\boldsymbol{c}_{k-d}$ depends on the data sequence. During the

[9]We are assuming that the overall ISI length is known or can be estimated.

training mode, this aiding sequence is directly available at the receiver, even with the minimal delay $d = 0$ compatible with the causality condition upon the observation. During the decision-directed tracking mode, the aiding data vector $\boldsymbol{c}_{k-d}$ must be replaced by a corresponding vector defined in terms of tentative decisions according to

$$\widehat{\widehat{\boldsymbol{c}}}_{k-d} = (\widehat{\widehat{c}}_{k-d}, \widehat{\widehat{c}}_{k-d-1}, \ldots, \widehat{\widehat{c}}_{k-d-L})^{\mathrm{T}} . \tag{3.54}$$

The recursion (3.53) must be accordingly modified:

$$\widehat{\boldsymbol{f}}_{k+1} = \widehat{\boldsymbol{f}}_k + \beta \left(r_{k-d} - \widehat{\boldsymbol{f}}_k^{\mathrm{T}} \widehat{\widehat{\boldsymbol{c}}}_{k-d}\right) \widehat{\widehat{\boldsymbol{c}}}^{*}_{k-d} . \tag{3.55}$$

In (3.55), the tentative decision delay must verify $d \geq 1$ in order to comply with the causality condition upon the data.

Depending on the use of either (3.53) or (3.55) in the training or tracking mode, respectively, the branch metric becomes

$$\lambda_k(a_k, S_k) \sim -|r_k - \widehat{\boldsymbol{f}}_k^{\mathrm{T}} \boldsymbol{c}_k|^2 + 2\sigma^2 \ln P\{a_k\} . \tag{3.56}$$

3.9 Per-Survivor Processing

In the previous sections, we considered the detection and estimation functions of a receiver designed according to the decomposition principle. The resulting branch metric (3.42) is computed using the data-aided parameter estimator (3.41). In the training operational mode, the aiding data sequence $\boldsymbol{c}_0^{k-d}$ is known. In the decision-directed tracking mode, this sequence is replaced by a sequence of preliminary decisions $\widehat{\widehat{\boldsymbol{c}}}_0^{k-d}$ yielding the following parameter estimate:

$$\widehat{\boldsymbol{\theta}}_k = \boldsymbol{g}_{k-l}\left[\boldsymbol{r}_0^k, \widehat{\widehat{\boldsymbol{c}}}_0^{k-d}\right] .$$

As we noted, the tentative decision delay must comply with the causality property of the data-aided estimator, namely $d \geq 0$ in the training phase and $d \geq 1$ in the tracking phase. We also mentioned that during training the minimal delay $d = 0$ can be chosen because it allows minimal estimation delay.

Let us now consider the choice of the tentative decision delay in the decision-directed tracking phase. The selected detection algorithm releases final decisions with a delay D which is inherent to the structure of the algorithm itself—good quality, or optimal, decisions cannot be made with a reduced delay, i.e., earlier. Preliminary decisions, with delay $d < D$, can be obtained by these algorithms only with degraded

quality. As an example, considering sequence detection implemented by a VA, the detection delay D must be such that the probability of converging survivors within the time span of the survivor memory is one or several orders of magnitude lower than the desired error probability. In order to release decisions with reduced delay $d < D$, one must accept a significantly larger probability of unmerged survivors. As a consequence, the quality of the tentative decision sequence may be considerably worse than that of the final decisions. From this viewpoint, one would try to choose a large value of d. On the other hand, we have seen that the parameter estimation algorithm is affected by the delay d of the aiding data sequence—the larger this delay, the larger the delay affecting the parameter estimate.[10] Clearly, delayed estimation may affect the detection quality when the true parameter is time-varying. For this reason, during training we would select the minimal value of delay $d = 0$. From this viewpoint, one should select a small value of tentative decision delay d, possibly the minimal value $d = 1$ compatible with the causality condition.

The above discussion indicates that, during the decision-directed tracking mode, *good* values of tentative decision delay d must be the result of a trade-off between two conflicting requirements:

- *large* values, to aid the estimation with a data sequence of good quality;
- *small* values, to limit the parameter estimation delay.

In practice, one would have to experiment with several values of d, measuring the overall error probability each time, in order to select a good compromise value.

Note that a decision-feedback mechanism characterizes the tracking phase, as clearly shown in Figure 3.3, with decisions used for parameter estimation and, therefore, to detect the successive data. In this decision-feedback mechanism, temporary *error propagation* may take place. Namely, wrong data decisions may negatively affect the parameter estimate and cause further decision errors. Although this effect is usually not catastrophic, it may affect the overall performance.

As an alternative to the use of tentative decisions during the tracking mode, a *per-survivor* estimation of the unknown parameters can be implemented [34]. In this technique, the code sequence associated with each survivor is used as the aiding data sequence for per-survivor estimation of the unknown parameters. In order to provide a formal description, let S_k be a trellis state descriptive of the overall FSM modeling the transmission system. Denoting the code sequence associated with the survivor of state S_k as $\breve{\mathbf{c}}_0^{k-1}(S_k)$, per-survivor estimates of the parameter vector $\boldsymbol{\theta}_k$, based on the

[10] Using prediction, i.e., letting $l < d$, would allow one to recover this delay at the cost of reduced estimation quality.

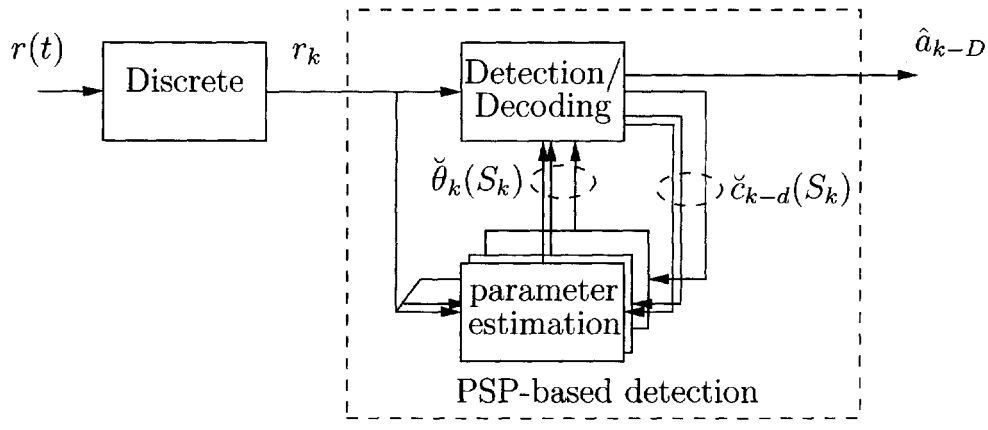


Figure 3.4: PSP-based detection.

data-aided estimator (3.41), can be expressed as

$$\breve{\boldsymbol{\theta}}_k(S_k) = \boldsymbol{g}_{k-l}\left[\boldsymbol{r}_0^k, \breve{\boldsymbol{c}}_0^{k-d}(S_k)\right] . \tag{3.57}$$

These per-survivor estimates can be used in the computation of the branch metric (3.42) according to

$$\lambda_k(a_k, S_k) = \ln p(r_k|\boldsymbol{r}_0^{k-1}, a_k, S_k, \breve{\boldsymbol{\theta}}_k(S_k)) + \ln P\{a_k\} . \tag{3.58}$$

The structure of the branch metric (3.58) is inherently different with respect to the previous cases in the sense that it also depends on the state S_k through the parameter estimate, which is now based on the survivor sequences. Note that there is now a data-aided parameter estimator per trellis state. The estimator uses, as aiding data sequence, the one associated with the survivor of this state. The resulting parameter estimates, one per state, are inherently associated with the survivor sequences—hence, the terminology "per-survivor processing." A block diagram showing a receiver based on this design approach is more difficult to draw because the detection and estimation functions are not clearly separated as in the previous cases based on the estimation-detection decomposition. An attempt to draw a block diagram is shown in Figure 3.4, where a set of parameter estimators observe the received sequence $\boldsymbol{r}_0^k$ and are aided by the survivor sequences $\breve{\boldsymbol{c}}_0^{k-d}(\sigma_k)$. A corresponding set of per-survivor parameter estimates $\breve{\boldsymbol{\theta}}_k(\sigma_k)$ is passed to the detection block.

The intuitive rationale for this type of approximate detection algorithm is the following: whenever the incomplete knowledge of some quantities prevents one from calculating a particular branch metric in a precise and predictable form, it is advisable

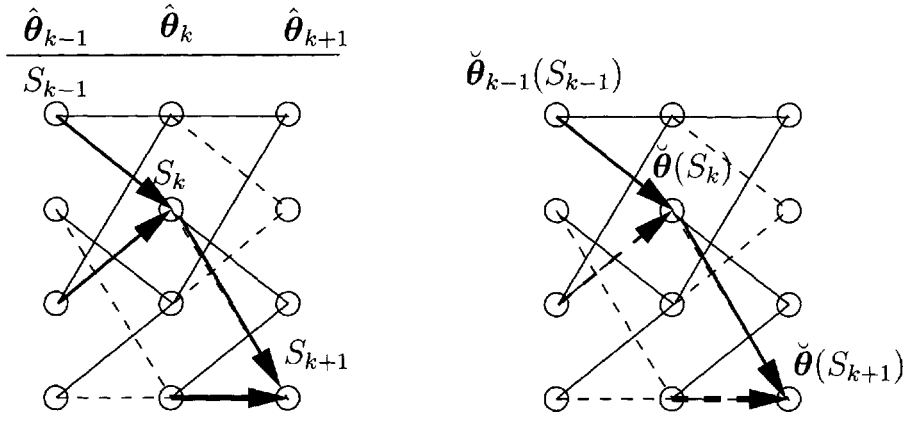


Figure 3.5: Trellis evolution: universal and PSP-based estimation.

to estimate those quantities based on the data sequence associated with the survivor leading to that branch. If any particular survivor is correct (an event of high probability under normal operating conditions), the corresponding estimates are evaluated using the correct data sequence. Since we do not know which survivor is correct (or the best) at each stage of decoding, we extend each survivor based on estimates obtained using its associated data sequence. Roughly speaking, the best survivor is extended using the best data sequence available (which is the sequence associated with it), regardless of our temporary ignorance as to which survivor is the best.

Pictorial descriptions of trellis evolution according to a "universal" estimation scheme, such as that based on tentative decision (or training), and PSP are shown in Figure 3.5. In the universal estimation scheme, only one estimate of the undesired parameter is used for computing all branch metrics. In PSP-based estimation, branch metrics of candidate survivors are computed using the different parameter estimates associated with the survivors of each initial state. In the latter case, the per-survivor parameter estimates are updated along the evolving survivors.

Before concluding this section, let us review, in a PSP-based perspective, the two examples considered in Section 3.8 in the context of universal (decision-directed) parameter estimation, namely phase synchronization and LMS tracking of the overall discrete channel impulse response.

3.9.1 Phase-Uncertain Channel

For the first example considered in Section 3.8.1, one can implement a corresponding PSP-based phase estimation and detection scheme by replacing the branch metric (3.46) with the following one:

$$\lambda_k(a_k, \mu_k) = -|r_k\, e^{-j\breve{\theta}_k(\mu_k)} - c_k|^2 + 2\sigma^2 \ln P\{a_k\} \tag{3.59}$$

and the phase estimate update recursion (3.47) with the following one:

$$\breve{\theta}_{k+1}(\mu_{k+1}) = \breve{\theta}_k(\mu_k) + \eta\, \Im\left\{r_{k-d}\, e^{-j\breve{\theta}_k(\mu_k)}\, \breve{c}^*_{k-d}(\mu_k)\right\} \tag{3.60}$$

in which $\breve{c}_{k-d}(\mu_k)$ is the code symbol at epoch $k-d$ in the survivor associated with state μ_k. The update recursions (3.60) must take place along the branches which actually extend the survivor of state μ_k, i.e., after the "usual" ACS step has been performed.

3.9.2 Dispersive Slow Fading Channel

Considering now LMS tracking of a slowly time-varying fading channel, in order to implement a PSP-based scheme the branch metric (3.52) can be replaced by the following:

$$\lambda_k(a_k, S_k) \sim -|r_k - \breve{\boldsymbol{f}}_k(S_k)^{\mathrm{T}} \boldsymbol{c}\,(a_k, S_k)|^2 + 2\sigma^2 \ln P\{a_k\} \tag{3.61}$$

and the channel estimate update recursion (3.55) with the following:

$$\breve{\boldsymbol{f}}_{k+1}(S_{k+1}) = \breve{\boldsymbol{f}}_k(S_k) + \beta \left[r_{k-d} - \breve{\boldsymbol{f}}_k(S_k)^{\mathrm{T}} \breve{\boldsymbol{c}}_{k-d}(S_k)\right] \breve{\boldsymbol{c}}^*_{k-d}(S_k) \tag{3.62}$$

where $\breve{\boldsymbol{c}}_{k-d}(S_k)$ denotes the following vector:

$$\breve{\boldsymbol{c}}_{k-d}(S_k) \triangleq [\breve{c}_{k-d}(S_k), \breve{c}_{k-d-1}(S_k), \ldots, \breve{c}_{k-d-L}(S_k)]^{\mathrm{T}} \tag{3.63}$$

whose elements are code symbols associated with the survivor of state S_k. The recursion (3.62) must take place over those branches which comply with the VA ACS step at the ending state S_{k+1}.

3.9.3 Remarks

A feature of PSP emerges by examining the parameter update equations (3.60) and (3.62) in the two examples considered above. Unlike decision-directed schemes

based on a tentative decision sequence, where good values of the delay d are a result of a trade-off between decision quality and estimation delay, in PSP-based estimation this delay is not necessary. This means that, since the parameter estimates are updated along each survivor, the best survivor is extended according to its associated data sequence, despite the fact that we do not know which survivor is the best at the current epoch.[11] In this mechanism, there is no reason for delaying the aiding data sequence of the best survivor beyond the minimal delay $d = 1$ complying with the causality condition. Considering now the other survivors, their associated data sequences do not approximate the correct one, regardless of the value of d. We may conclude that in PSP-based sequence detection schemes the minimal value $d = 1$ offers the best overall performance. This remark suggests that the PSP approach allows one to design receivers that are particularly robust when the undesired parameters are time-varying.

3.10 Complexity Reduction Techniques for VA-based Detection Algorithms

As considered in the previous section, we assume an FSM system model based on some FMC (in the strict sense or conditional). A generic VA-based detection algorithm will make use of a suitable branch metric $\lambda(a_k, S_k)$, where S_k is a proper system state, defined as

$$S_k \triangleq (a_{k-1}, \ldots, a_{k-C}, \mu_{k-C}) \tag{3.64}$$

where μ_k is the encoder/modulator state. We refer to the characterization of the encoder/modulator FSM in terms of "output" and "next-state" functions, as given in (3.3). Assuming that the number of states of the encoder/modulator is S_c and that the cardinality of the information symbols is M, we can immediately conclude that the number of expanded states is $\zeta = S_c M^C$. For example, if $S_c = M = C = 4$, then $\zeta = 4 \times 4^4 = 1024$, indicating that state complexity may be significantly large.

We previously observed that in some cases the next-state function might not be invertible, i.e., given S_{k+1} and a_k it is not possible to identify a state S_k in a unique way—this is the case, for instance, for a nonrecursive convolutional code, such that branches terminating in the same trellis state are associated with the same information symbol. In particular cases (for example, for particular recursive convolutional

[11] Eventually, the best survivor will be identified after D further steps, when a survivor merge has likely occurred.

codes), the next-state function is invertible. If this is the case, then an equivalent definition of the system state is the following:[12]

$$S_k = (\mu_k, a_{k-1}, \ldots, a_{k-C}). \tag{3.65}$$

In the following, we outline possible strategies to perform reduced-state sequence detection (RSSD).

3.10.1 State Reduction by Memory Truncation

Assuming that the coding rule is invertible, i.e., assuming that the equivalent definition of state given by (3.65) can be used, we consider a *genie-aided* trellis folding. Suppose that at each epoch k a genie passes a group of (previous) correct symbols $(\tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C})$ to the branch metric computer, with $Q \leq C$—if $Q = C$ then the genie's help is not needed. The genie-aided branch metric could be defined as

$$\tilde{\lambda}_k(a_k, S_k) = \lambda_k(a_k, \tilde{S}_k) \tag{3.66}$$

for each state S_k whose first $Q + 1$ entries coincide with those in

$$\tilde{S}_k = (\mu_k, a_{k-1}, \ldots, a_{k-Q}, \tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C}). \tag{3.67}$$

It is possible to realize that the group of states

$$S_k = (\underbrace{\mu_k, a_{k-1}, \ldots, a_{k-Q}}_{\text{fixed}}, \underbrace{a_{k-Q-1}, \ldots, a_{k-C}}_{\text{variable}}) \tag{3.68}$$

in the trellis would have *identical* path metrics $\forall(a_{k-Q-1}, \ldots, a_{k-C})$ and could then be folded into a reduced (partial, folded or super) state

$$w_k \triangleq (\mu_k, a_{k-1}, \ldots, a_{k-Q}). \tag{3.69}$$

The MAP sequence search based on a VA could be equivalently performed over a folded trellis diagram with this reduced state w_k. The genie-aided branch metric in the reduced-state (RS) trellis can therefore be written as follows:

$$\tilde{\lambda}(a_k, w_k) = \lambda_k(a_k, \tilde{S}_k(w_k)) \tag{3.70}$$

[12]One can immediately notice that the only difference between the state defi nition given by (3.65) and that given by (3.64) is the fact that the encoder/modulator FSM state embedded in (3.65) is at epoch k rather than at epoch $k - C$. If defi nitions (3.64) and (3.65) are equivalent, the coding rule will be referred to as *invertible*.

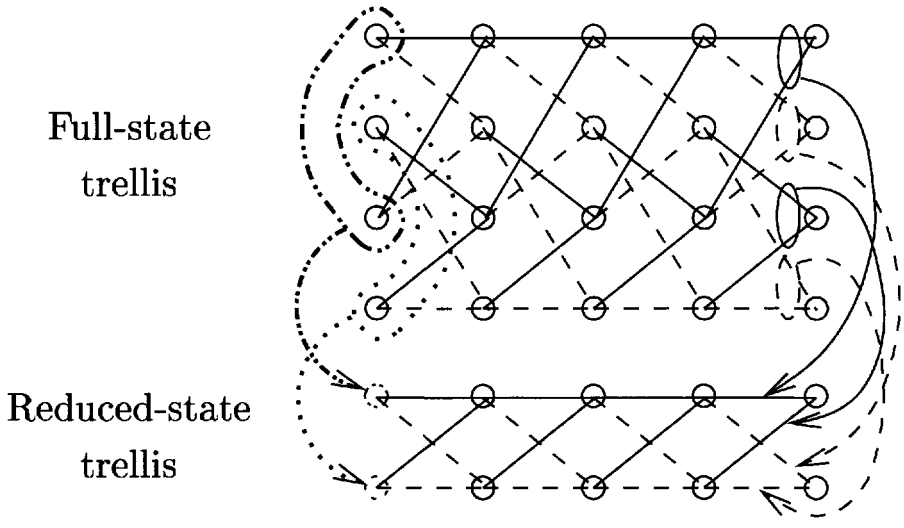


Figure 3.6: Pictorial description of trellis folding.

where $\tilde{S}_k(w_k)$ is a pseudo-state defined as

$$\tilde{S}_k(w_k) = (\underbrace{\mu_k, a_{k-1}, \ldots, a_{k-Q}}_{w_k}, \tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C}) \tag{3.71}$$

where $(\tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C})$ is the genie information. Thus there is a genie-aided *truncation* of the system memory, with RS parameter $Q \leq C$. The reduced number of states is then $\zeta' = S_c M^Q \leq \zeta = S_c M^C$. Finally, the full-state trellis folds into an RS trellis. A pictorial description of trellis folding is shown in Figure 3.6.

In the above derivation we assumed that the coding rule is invertible. If this is not the case, then, in order to perform RS sequence detection, the genie should provide $(\tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C}, \tilde{\mu}_{k-C})$. In other words, the main difference with respect to the previous case consists of the fact that the starting state of the encoder/modulator in the considered window, i.e., $\tilde{\mu}_{k-C}$, has also to be assigned by the genie. Reasoning as before, one can conclude that in this case the number of reduced states would simply be $\zeta' = M^Q$. For ease of derivation, in the following we will refer to the case of an invertible coding rule.

At this point, assuming that the genie disappears, the information symbols $(\tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C})$ can be viewed as an *undesired set of parameters*. Hence, a parameter-conditional *reduced* memory property holds. The estimation-detection decomposition can (again) be the route to the approximation of the branch metric in the presence of this special parametric uncertainty. In other words, the genie information

$(\tilde{a}_{k-Q-1}, \ldots, \tilde{a}_{k-C})$ must be estimated in order to implement detection schemes with reduced state-complexity. Note that we do not need a data-aided parameter estimator, but only the aiding code sequence. In particular, two strategies are possible: one based on *tentative decisions* and the other based on PSP.

1. As shown before, indicating by $\lambda_k(a_k, S_k)$ the branch metric in the original full-state trellis, the branch metric in the RS trellis is $\tilde{\lambda}_k(a_k, S_k) = \lambda_k(a_k, \tilde{S}_k(w_k))$, in which, in the case of tentative-decision feedback, the pseudo-state is defined as

$$\tilde{S}_k(w_k) = (\underbrace{\mu_k, a_{k-1}, \ldots, a_{k-Q}}_{w_k}, \hat{\hat{a}}_{k-Q-1}, \ldots, \hat{\hat{a}}_{k-C}) \tag{3.72}$$

where $(\hat{\hat{a}}_{k-Q-1}, \ldots, \hat{\hat{a}}_{k-C})$ are preliminary decisions on the code sequence.

2. In this case, the pseudo state $\tilde{S}_k$ is

$$\tilde{S}_k(w_k) = (\underbrace{\mu_k, a_{k-1}, \ldots, a_{k-Q}}_{w_k}, \breve{a}_{k-Q-1}, \ldots, \breve{a}_{k-C}) \tag{3.73}$$

where $(\breve{a}_{k-Q-1}, \ldots, \breve{a}_{k-C})$ are the code symbols in the survivor associated with reduced state w_k. As a result, the pseudo state depends on w_k through the feedback of survivor symbols as well. As an example, consider the case of linear modulation on a static dispersive channel, whose output is

$$r_k = \sum_{i=0}^{L} f_i c_{k-i} + n_k \tag{3.74}$$

where $\{f_i\}$ are the channel coefficients and n_k is a complex AWGN sample with variance per component equal to σ^2. The branch metric in the original full-state trellis ($C = L$, $\zeta = S_c M^L$ states) can be written as:

$$\lambda_k(a_k, S_k) \sim -\left| r_k - f_k c_k - \sum_{i=1}^{L} f_i c_{k-i}(S_k) \right|^2 + 2\sigma^2 \ln P\{a_k\} \tag{3.75}$$

where the sequence of code symbols $\boldsymbol{c}_{k-L}^{k}$ is uniquely identified by the transition $T_k = (a_k, S_k)$. In the case of an RS trellis, the branch metric can be written as

$$\begin{aligned} \tilde{\lambda}_k(a_k, w_k) &= \lambda_k(a_k, \tilde{S}_k(w_k)) \\ &\sim -\left| r_k - f_k c_k - \sum_{i=1}^{Q} f_i c_{k-i}(S_k) - \sum_{i=Q+1}^{L} f_l \breve{c}_{k-i}(w_k) \right|^2 \\ &\quad + 2\sigma^2 \ln P\{a_k\} \end{aligned} \tag{3.76}$$

where $(c_{k-1}(w_k), \ldots, c_{k-Q}(w_k))$ are code symbols uniquely identified by reduced state w_k, whereas $(\check{c}_{k-Q-1}(w_k), \ldots, \check{c}_{k-L}(w_k))$ are the code symbols in the survivor associated with reduced state w_k.

3.10.2 State Reduction by Set Partitioning

Another possible approach to performing state reduction is based on folding by *set partitioning*. In the following derivation, we will assume, for ease of notational simplicity, that the output function, besides the next-state function, is invertible. Given this, the "full" state S_k and transition T_k can be equivalently defined in terms of coded symbols, i.e.:

$$S_k = (\mu_k, c_{k-1}, \ldots, c_{k-C}) \tag{3.77}$$

$$T_k = (c_k, S_k). \tag{3.78}$$

The basic idea of set partitioning consists of replacing the code symbols $\{c_{k-i}\}$ in the full state with subsets of the code symbol alphabet (or *constellation*), which will be indicated by $\mathcal{A}$. A reduced state is then defined as

$$w_k \triangleq (\mu_k, I_{k-1}(1), I_{k-2}(2), \ldots, I_{k-L}(L)) \tag{3.79}$$

where $I_{k-i}(i) \in \Omega(i)$, $i = 1, \ldots, L$, are subsets of the constellation $\mathcal{A}$, and $\Omega(i)$, $i = 1, \ldots, L$ constitute a partition of the code constellation $\mathcal{A}$. A given reduced state in this case specifies only the constellation subsets $\{I_{k-i}(i)\}$, and $c_{k-i} \in I_{k-i}(i)$ is a code symbol compatible with the given state. Let $J_i = \text{card}\{\Omega(i)\}$ and $M' = \text{card}\{\mathcal{A}\}$ $(1 \leq J_i \leq M')$, and denote J_i as the partition depth at level i. The reduced state is well-defined if current state w_k and subset $I_k(1)$ (to which the current symbol c_k belongs) uniquely specify the next state

$$w_{k+1} = (\mu_{k+1}, I_k(1), I_{k-1}(2), \ldots, I_{k-L+1}(L)). \tag{3.80}$$

It follows that $\Omega(i)$ must be a further partition of $\Omega(i+1)$.

The partition depths $\{J_i\}$ must satisfy the condition:

$$J_1 \geq J_2 \geq \ldots \geq J_L. \tag{3.81}$$

If Q is such that $J_Q > 1$ and $J_{Q+1} = \ldots = J_L = 1$, the definition of the reduced state can be simplified as follows:

$$w_k = (\mu_k, I_{k-1}(1), I_{k-2}(2), \ldots, I_{k-Q}(Q)). \tag{3.82}$$

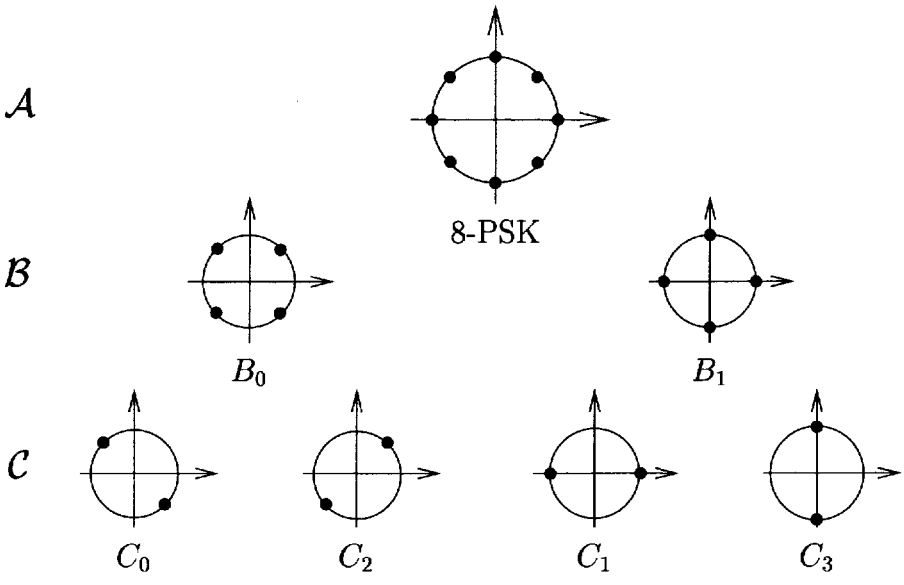


Figure 3.7: Set partitioning for 8-PSK constellation.

If $J_1 = \ldots = J_Q = M'$, memory truncation, i.e., the state reduction technique considered in Section 3.10.1, arises as a special case.

As an example of state complexity reduction, in Figure 3.7 a possible partition of an 8-ary phase shift keying (8-PSK) constellation is shown, where the starting constellation ($\mathcal{A}$) and two successive constellations ($\mathcal{B}$ and $\mathcal{C}$) are shown. Assuming uncoded transmission[13] and a channel length $L = 2$, the full-complexity state is

$$S_k = (a_{k-1}, a_{k-2}) \tag{3.83}$$

and the number of states is $\zeta = M^L = 8^2 = 64$. Various set partitionings are possible, as indicated in the following.

1. A reduced state could be obtained by memory truncation considering $Q = 1$. This corresponds to considering a set partitioning state reduction with $J_1 = 8$ and $J_2 = 1$. In this case, the reduced state can be written as

$$w'_k = a_{k-1} \tag{3.84}$$

and the number of reduced states is $\zeta' = M^Q = 8^1 = 8$.

[13]In this case, there is no encoder at the transmitter side, so that, as the modulation is memoryless, the possible encoder/modulator state μ_k disappears from the defi nition of state S_k.

2. A reduced state can be obtained by considering set partition with

$$\begin{array}{ll} \Omega(1) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} & (J_1 = 4) \\ \Omega(2) = \mathcal{B} = \{B_0, B_1\} & (J_2 = 2). \end{array} \tag{3.85}$$

The reduced state can be written as follows:

$$w_k'' = (I_{k-1}(1), I_{k-2}(2)) \tag{3.86}$$

and the number of reduced states is given by $\zeta' = J_1 J_2 = 4 \times 2 = 8$.

3. Finally, we consider the set partitioning characterized by

$$\Omega(1) = \Omega(2) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} \quad (J_1 = J_2 = 4). \tag{3.87}$$

In this case the reduced state can be written as

$$w_k''' = (I_{k-1}(1), I_{k-2}(2)) \tag{3.88}$$

and the number of reduced states is $\zeta' = J_1 J_2 = 4 \times 4 = 16$.

It is worth observing that set partitioning should follow the partition rules used in trellis coded modulation (TCM) [9]. Note also that if $J_1 < M'$ *parallel transitions* may occur (they do, for sure, in the uncoded case). In particular, if $J_1 < M'$ a state transition and the corresponding branch metric are defined as $(I_k(1), w_k)$ and $\lambda_k(I_k(1), w_k)$, respectively. The branch metric can then be written as follows:

$$\lambda_k(I_k(1), w_k) = \max_{c_k \in I_k(1)} \lambda_k(c_k, \tilde{S}_k(w_k)) \tag{3.89}$$

where the pseudo state $\tilde{S}_k(w_k)$ must be compatible with w_k, in the sense that $c_{k-i} \in I_{k-i}(i)$. The missing information can be based on tentative decisions or PSP. In particular, the PSP-based pseudo state can be expressed as

$$\tilde{S}_k(w_k) = (\mu_k, \tilde{c}_{k-1}(w_k), \ldots, \tilde{c}_{k-Q}(w_k), \breve{c}_{k-Q-1}(w_k), \ldots, \breve{c}_{k-L}(w_k)) \tag{3.90}$$

where $(\tilde{c}_{k-1}(w_k), \ldots, \tilde{c}_{k-Q}(w_k))$ are code symbols compatible with state w_k to be found in the survivor history of state w_k, and $(\breve{c}_{k-Q-1}(w_k), \ldots, \breve{c}_{k-L}(w_k))$ are code symbols in the survivor of state w_k.

Considering linear modulation on a static dispersive channel, the branch metric in the full-state trellis (with $\zeta = S_c M^L$ states) has the expression (3.76). In the case of

state reduction by set partitioning, the branch metric in the corresponding RS trellis can be written as follows:

$$\begin{aligned}\tilde{\lambda}_k(I_k(1), w_k) &= \max_{c_k \in I_k(1)} \lambda_k(c_k, \tilde{S}_k(w_k)) \\ &= \max_{c_k \in I_k(1)} \left\{ - \left| r_k - f_k c_k - \sum_{i=1}^{Q} f_i c_{k-i}(w_k) - \sum_{i=Q+1}^{L} f_l \breve{c}_{k-i}(w_k) \right|^2 \right. \\ &\qquad \left. + 2\sigma^2 \ln P\{a_k(c_k, w_k)\} \right\} \end{aligned} \tag{3.91}$$

where $a_k(c_k, w_k)$ is the information symbol associated with code symbol c_k and reduced state w_k—recall that we are considering an invertible coding rule. In particular, $(c_{k-1}(w_k), \ldots, c_{k-Q}(w_k))$ are code symbols associated with the reduced state w_k, while $(\breve{c}_{k-Q-1}(w_k), \ldots, \breve{c}_{k-L}(w_k))$ are code symbols in the survivor associated with state w_k.

3.10.3 A Case Study: TCM on an ISI Channel

As an example of an application of the state reduction techniques considered above, namely memory truncation and set partitioning, we consider the case of TCM signaling on an ISI channel. In particular, we consider TCM characterized by a 3/4-rate convolutional code followed by mapping over a 16-ary quadrature amplitude modulation (QAM) constellation. The convolutional code has $S_c = 4$ states. The structure of the convolutional encoder is shown in Figure 3.8. The information bits are $\{a_k^{(i)}\}_{i=0}^{2}$ and the code bits are indicated as $\{c_k^{(i)}\}_{i=0}^{3}$. Note that the *gross* spectral efficiency of the TCM code is 3 bits/s/Hz. Using raised cosine pulses with roll-off $\epsilon = 0.3$ [17], the spectral efficiency has to be reduced by the expansion factor $1 + \epsilon = 1.3$, so that the *net* spectral efficiency is 2.3 bits/s/Hz. In Figure 3.8, the mapping over the branches relative to a trellis transition is also indicated. In particular, note that each branch indicated in the section of the trellis diagram corresponds to 2^2 parallel branches—as one can see, two of the three bits at the input of the convolutional encoder do not influence the evolution of the encoder and are systematic bits. In Figure 3.9, the set partition for the considered 16-QAM modulation and the mapping rule are shown. In particular, note that each branch in the trellis section shown in Figure 3.9 is associated with a subset of $\mathcal{C}$.

The model of the discrete observable is the following:

$$r_k = \sum_{l=0}^{L} f_l c_{k-l} + n_k \tag{3.92}$$

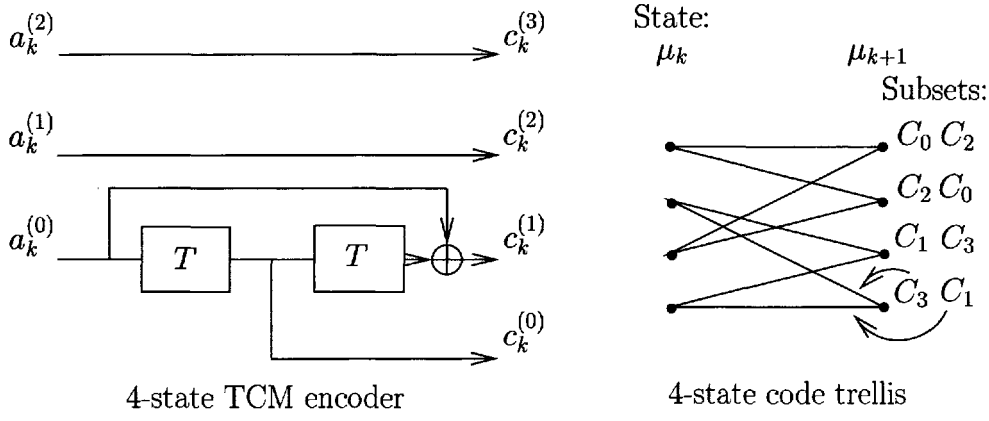


Figure 3.8: TCM encoder and mapping for 16-QAM (the subsets are specified in Figure 3.9).

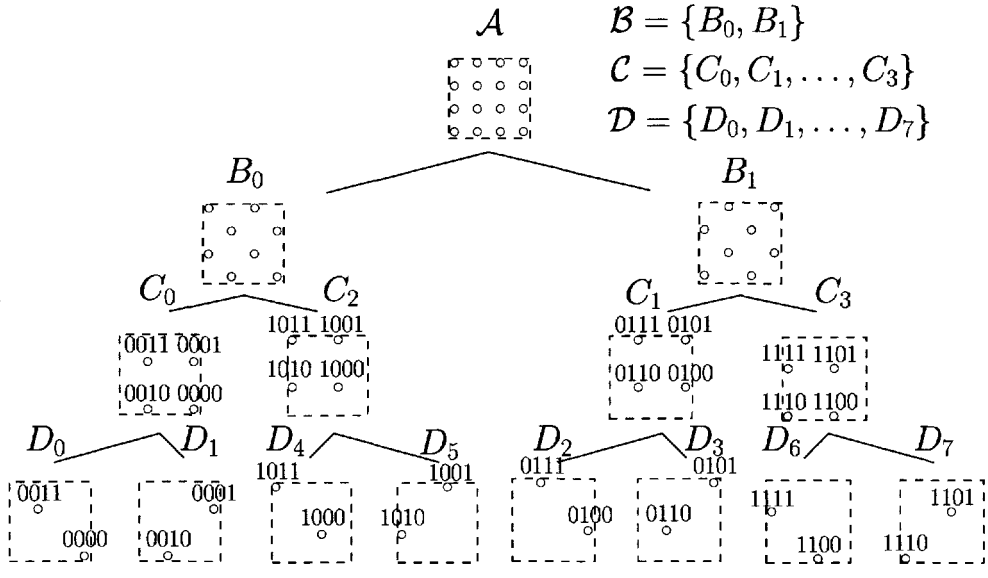


Figure 3.9: Set partition and mapping rule for 16-QAM constellation.

where $\{f_l\}_{l=0}^{L}$ is the *white noise discrete equivalent* impulse response of the ISI channel, $\{n_k\}$ is an iid Gaussian noise sequence with variance σ^2 per component, and $\{c_k\}$ is the code sequence. The parameter L, representing the discrete-time equivalent channel length, must be large enough to accommodate the significant pulses. In particular, we consider $L = 3$, and the distribution of the channel coefficients $\{f_l\}$ is

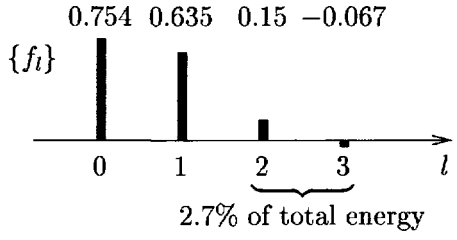


Figure 3.10: Equivalent discrete-time channel response of an ISI channel.

shown in Figure 3.10. The full-complexity state, in this case, is

$$S_k = (\mu_k, c_{k-1}, c_{k-2}, c_{k-3}) \tag{3.93}$$

and the number of states is $\zeta = S_c M^L = 4 \times 8^3 = 2048$. Various state reductions are possible and in the following we enumerate a few interesting options.

1. The first possibility is to consider memory truncation with $Q = 1$, in which case the reduced state can be written as follows:

$$w_k = (\mu_k, c_{k-1}) \tag{3.94}$$

and the number of reduced states is $\zeta' = S_c M^Q = 4 \times 8^1 = 32$. Note that memory truncation corresponds to set partitioning with $J_1 = 16$, $J_2 = J_3 = 1$.

2. It is possible to consider set partition with

$$\begin{array}{ll} \Omega(1) = \mathcal{D} = \{D_0, D_1, \ldots, D_7\} & (J_1 = 8) \\ \Omega(2) = \Omega(3) = \mathcal{A} & (J_2 = J_3 = 1). \end{array} \tag{3.95}$$

In this case, the partial state can be written as

$$w_k = (\mu_k, I_{k-1}(1)) \tag{3.96}$$

and the number of reduced states is $\zeta' = S_c \frac{J_1}{2} = 4 \times 4 = 16$.

3. Another possible set partition is characterized by

$$\begin{array}{ll} \Omega(1) = \Omega(2) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} & (J_1 = J_2 = 4) \\ \Omega(3) = \mathcal{A} & (J_3 = 1). \end{array} \tag{3.97}$$

In this case, the partial state is

$$w_k = (\mu_k, I_{k-1}(1), I_{k-2}(2)) \tag{3.98}$$

and the number of reduced states is $\zeta' = S_c \frac{J_1}{2} \frac{J_2}{2} = 4 \times 2 \times 2 = 16$.

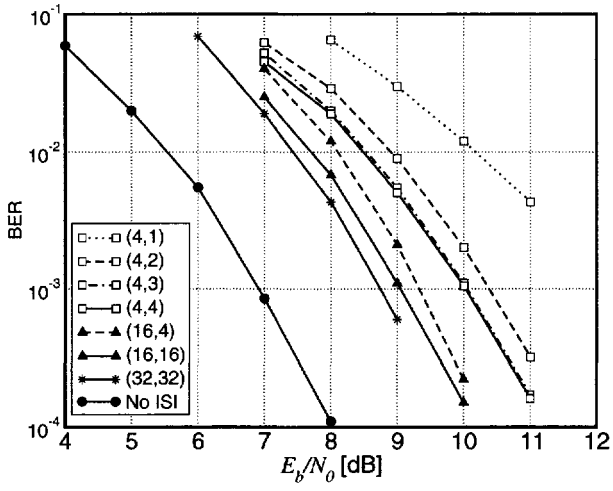


Figure 3.11: Performance of TCM with 16-QAM for transmission over the 4-tap ISI channel considered in Figure 3.10. Reproduced from [106], ©1996 IEEE, by permission of the IEEE.

4. Another possible set partition is characterized by

$$\begin{aligned} &\Omega(1) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} \qquad (J_1 = 4) \\ &\Omega(2) = \Omega(3) = \mathcal{A} \qquad\qquad\quad (J_2 = J_3 = 1). \end{aligned} \tag{3.99}$$

In this case, the partial state becomes

$$w_k = (\mu_k, I_{k-1}(1)) \tag{3.100}$$

and the number of reduced states is $\zeta' = S_c \frac{J_1}{2} = 4 \times 2 = 8$.

5. Finally, one can consider the case of memory truncation with $Q = 0$, i.e., considering only the code trellis. This results in $w_k = \mu_k$ and the number of states is $\zeta' = S_c = 4$.

We now analyze the performance of a receiver using VA-based algorithms for combined detection/decoding and the state complexity reduction techniques enumerated above. The results are shown in Figure 3.11, in terms of bit error rate (BER) versus E_b/N_0, E_b being the received energy per information bit and N_0 the one-sided noise power spectral density. In particular, two numbers are indicated in correspondence to each curve: the first one corresponds to the number of reduced states—recall

that the number of full states in this case is $\zeta = 2048$—and the second one corresponds to the number of estimators considered in the PSP-based VA (more details can be found in [106]). In particular:

- the case with $\zeta' = 32$ corresponds to the reduced combined code/ISI trellis (case 1);
- the case with $\zeta' = 16$ corresponds to the reduced combined code/ISI trellis (case 2);
- the case with $\zeta' = 4$ corresponds to the code trellis only (case 5).

For comparison, the performance curve relative to the case of absence of ISI is also shown.

3.10.4 Reduced-Search Algorithms

If optimal processing is infeasible, any type of suboptimal processing may deserve our attention. Nevertheless, ranking of suboptimal solutions is difficult because of lacking reference criteria. Note that RSSD must be considered but an alternative among many others. More precisely, while the state reduction techniques proposed in the previous subsection allow for *systematic* state reduction over successive trellis transitions, another approach to performing complexity reduction is possible. In particular, *reduced-search* (or *sequential*) algorithms may be used to search a small part of a large FSM trellis diagram or a non-FSM tree diagram. As opposed to state-complexity reduction, the original full-complexity trellis (or tree) diagram is partially searched. These algorithms date back to the pre-VA era [107]. They were first proposed to decode convolutional codes. The denomination "sequential" emphasizes the "novelty" compared to the then-established algebraic decoding of block codes [4]. These algorithms can be applied to any system characterized by large memory or state complexity (if an FSM model holds).

In the following, we assume that an FSM model exists, and we refer to its corresponding trellis diagram. A general formulation of breadth-first algorithms is presented in [108,109]. Assuming that an FSM model holds, let ζ be the number of states in the full-size trellis. It is possible to partition the ζ states into n_{cl} disjoint classes—that is, these classes constitute a partition of the ensemble of ζ states. We assume that n_{pa} paths per class are maintained, by selecting those which maximize the *a posteriori* probabilities (APPs) under the constraint imposed by the partition rule and the class structure. The resulting search algorithm is denoted as $\mathrm{SA}(n_{\mathrm{pa}}, n_{\mathrm{cl}})$ [108]. In the following, M corresponds to the cardinality of the symbols at the input of the FSM. Special interesting cases can be summarized as follows.

- $n_{\mathrm{pa}} > 1$ and $n_{\mathrm{cl}} = \zeta$: the SA$(n_{\mathrm{pa}}, n_{\mathrm{cl}})$ reduces to the list Viterbi algorithm (LVA) with B survivors per state.

- $n_{\mathrm{pa}} = 1$ and $n_{\mathrm{cl}} = \zeta$: the classical VA is obtained.

- $n_{\mathrm{pa}} = 1$ and $n_{\mathrm{cl}} < \zeta$: this situation corresponds to RSSD with n_{cl} states.

- $n_{\mathrm{pa}} > 1$ and $n_{\mathrm{cl}} < \zeta$: this corresponds to list RSSD with n_{pa} survivors per state and n_{cl} states.

- $n_{\mathrm{pa}} = M$ and $n_{\mathrm{cl}} = 1$: the M-algorithm is obtained.

Whenever $n_{\mathrm{cl}} < \zeta$, the branch metric can be computed by applying the principle of PSP. Moreover, PSP allows also the above formalization to be applied when an FSM model does not hold ($\zeta \to \infty$). For this class of algorithms the complexity level is related to the total number of paths being traced, i.e., the product $n_{\mathrm{pa}} n_{\mathrm{cl}}$. Imposing a constraint on the complexity, i.e., $n_{\mathrm{pa}} n_{\mathrm{cl}} \leq \eta$, where η is a suitable complexity threshold, constrained optimality can be defined. According to this constrained optimality criterion, it is possible to show that the M-algorithm is the constrained optimal search algorithm [108].

Finally, we mention another reduced-search technique, namely the T-algorithm, where T stands for *threshold*. This algorithm is applicable to a VA running over a full-complexity (or possibly reduced-complexity) trellis. The basic idea of this algorithm consists of maintaining only the paths with partial metric above a predefined threshold. If a path has a partial path metric above the threshold, the path is discarded. Note that while the SA$(n_{\mathrm{pa}}, n_{\mathrm{cl}})$ class of algorithms has a predetermined complexity (in terms of total number of paths being traced), the T-algorithm has a time-varying complexity, since the number of paths maintained can vary across consecutive trellis transitions.

3.11 Applications to Wireless Communications

In this section, we show how the general approach proposed in Section 3.5 allows one to derive, in a unified manner, novel, as well as known, sequence detection algorithms. In particular, we focus on the case of sequence detection for phase-uncertain channels and fading channels. Various algorithms, depending on the considered channel model and particular estimation/detection approach, will be derived.

3.11.1 Adaptive Sequence Detection: Preliminaries and Least Mean Squares Estimation

Broadly speaking, the channel model parameters can be time-varying (e.g., carrier phase, timing epoch and channel impulse response). A receiver based on the estimation-detection decomposition must be able to *track* these time variations, *provided they are not too fast.* The receiver must adapt itself to the time-varying channel conditions: the concept of PSP can then be successfully applied to the design of *adaptive receivers.*

- The per-survivor estimator associated with the best survivor is derived from data information which can be perceived as high-quality zero-delay decisions. PSP is thus useful in *fast* time-varying channels.
- Many hypothetical data sequences are simultaneously considered in the parameter estimation process. In this sense, acquisition without training (*blind*) may be facilitated.

Assume a parameter-conditional FSM model with state S_k (i.e., we are implicitly assuming that the FMC holds). In this case, it is possible to consider two types of PSP-based parameter estimation strategy: *feed-forward* and *feedback.*

- A PSP-based *feed-forward* data-aided parameter estimator at time k can be written as follows:

$$\breve{\boldsymbol{\theta}}_k^{\mathrm{ff}}(S_k) = \mathbf{p}[\boldsymbol{r}_{k-N}^{k-1}, \breve{\boldsymbol{c}}_0^{k-1}(S_k)] \tag{3.101}$$

  where $\mathbf{p}[\cdot,\cdot]$ is a suitable function of the observations and coded symbols. The estimator is a function of the N most recent signal observations[14] $\boldsymbol{r}_{k-N}^{k-1}$ and the per-survivor aiding data sequence $\breve{\boldsymbol{c}}_0^{k-1}(S_k)$ is associated with state S_k. Based on the computation of a parameter estimate, the branch metric associated with a transition at epoch k starting from state S_k can be written as follows:

$$\lambda_k(a_k, S_k) = \ln p\left(r_k | \boldsymbol{r}_{k-N}^{k-1}, a_k, S_k, \breve{\boldsymbol{\theta}}_k^{\mathrm{ff}}(S_k)\right) + \ln P\{a_k\} \tag{3.102}$$

  and the update rule of the parameter estimate at epoch $k+1$ becomes

$$\breve{\boldsymbol{\theta}}_{k+1}^{\mathrm{ff}}(S_{k+1}) = \mathbf{p}[\boldsymbol{r}_{k-N+1}^{k}, \breve{\boldsymbol{c}}_0^{k}(S_{k+1})]. \tag{3.103}$$

  Note that these estimates are simply recomputed for the new observation vector $\boldsymbol{r}_{k-N+1}^{k}$ and each new survivor sequence $\breve{\boldsymbol{c}}_0^{k}(S_{k+1})$. In this sense, they are

[14] A CMP, implying an FMC, is implicitly assumed.

formally independent from the estimates at the previous epoch. Examples of applications, considered in the following, are the designs of linear predictive and noncoherent detection algorithms.

- We indicate a PSP-based *feedback* data-aided parameter estimate at time k as $\breve{\boldsymbol{\theta}}_k^{\mathrm{fb}}(S_k)$. Based on this estimate, the branch metric associated with a branch starting from state S_k is

$$\lambda_k(a_k, S_k) = \ln p(r_k | \boldsymbol{r}_{k-N}^{k-1}, a_k, S_k, \breve{\boldsymbol{\theta}}_k^{\mathrm{fb}}(S_k)) + \ln P\{a_k\} \tag{3.104}$$

and the update rule of the parameter estimate at epoch $k+1$ can be written as

$$\breve{\boldsymbol{\theta}}_{k+1}^{\mathrm{fb}}(S_{k+1}) = \mathbf{q}[\breve{\boldsymbol{\theta}}_{k-n_{\mathrm{fb}}+1}^{k,\mathrm{fb}}(S_k)] + \mathbf{p}[\boldsymbol{r}_{k-N+1}^{k}, \breve{\boldsymbol{c}}_0^k(S_{k+1})] \tag{3.105}$$

where $\mathbf{p}[\cdot,\cdot]$ has the same meaning as in the feed-forward parameter estimate in (3.101), whereas $\mathbf{q}[\cdot,\cdot]$ is a function of the n_{fb} previous feedback estimates. These estimates are computed for the N most recent observations $\boldsymbol{r}_{k-N+1}^{k}$ and each new survivor sequence $\breve{\boldsymbol{c}}_0^k(S_{k+1})$. The previous n_{fb} parameter values $\breve{\boldsymbol{\theta}}_{k-n_{\mathrm{fb}}+1}^{k,\mathrm{fb}}(S_k)$ are those associated with the survivors of states S_k in the transitions $(S_k \rightarrow S_{k+1})$ selected during the ACS step. Feedback parameter estimation is usually used in adaptive receivers [105].

Note that in feed-forward and feedback parameter estimation, *tentative decisions* $\widehat{\breve{\boldsymbol{c}}}_0^k$ can be used in place of the survivor data sequences $\breve{\boldsymbol{c}}_0^k(S_{k+1})$ for updating the parameter estimate. If this is the case, then the parameter estimate becomes *universal*, i.e., identical for all survivors. Formally, the updating recursions yield identical estimates for all survivors. The parameter estimator becomes *external* to the detection block. During *training* the correct data sequence would be used.

In the following, we consider a few significant examples of sequence detection algorithms using feedback parameter estimation.

Tracking of a Dispersive Time-Varying Channel

Assume the following model for a linearly modulated discrete observable (slow variation):

$$r_k = \sum_{l=0}^{L} f_{l,k} c_{k-l} + n_k = \boldsymbol{f}_k^{\mathrm{T}} \boldsymbol{c}_k + n_k \tag{3.106}$$

where: $\boldsymbol{f}_k = (f_{0,k}, f_{1,k}, \ldots, f_{L,k})^{\mathrm{T}}$ is the overall time-varying discrete equivalent impulse response at the k-th epoch, $\boldsymbol{c}_k = (c_k, c_{k-1}, \ldots, c_{k-L})^{\mathrm{T}}$ is a code sequence of

length L relative to the encoder/modulator FSM with state μ_k, and n_k is an AWGN sample with variance per component equal to σ^2. We define as $S_k = (a_{k-1}, a_{k-2}, \ldots, a_{k-L}, \mu_{k-L})$ the system state including the encoder/modulator and the channel. The code symbol vector uniquely associated with the considered trellis branch (a_k, S_k), in accordance with the coding rule, is indicated as follows:

$$\boldsymbol{c}_k(a_k, S_k) = [c_k(a_k, \mu_k), c_{k-1}(a_{k-1}, \mu_{k-1}), \ldots, c_{k-L}(a_{k-L}, \mu_{k-L})]^{\mathrm{T}}. \quad (3.107)$$

- Considering LMS adaptive identification, the parameter estimator updating rule can be written as

$$\widehat{\boldsymbol{f}}_{k+1} = \widehat{\boldsymbol{f}}_k + \beta(r_{k+1-d} - \widehat{\boldsymbol{f}}_k^{\mathrm{T}} \boldsymbol{c}_{k+1-d})\boldsymbol{c}_{k+1-d}^* \quad (3.108)$$

  where $d \geq 1$ represents a suitable delay and complies with the causality condition upon the data. The parameter β compromises between adaptation speed and algorithm stability [105]. In this case, the branch metric can be expressed as

$$\lambda_k(a_k, S_k) = -\left| r_k - \widehat{\boldsymbol{f}}_k^{\mathrm{T}} \boldsymbol{c}_k(a_k, S_k) \right|^2 + 2\sigma^2 \ln P\{a_k\} \quad (3.109)$$

  In the (tentative) decision-directed tracking mode, the recursive estimator updating rules are

$$\widehat{\boldsymbol{f}}_{k+1} = \widehat{\boldsymbol{f}}_k + \beta(r_{k+1-d} - \widehat{\boldsymbol{f}}_k^{\mathrm{T}} \widehat{\boldsymbol{c}}_{k+1-d})\widehat{\boldsymbol{c}}_{k+1-d}^* \quad (3.110)$$

$$\widehat{\boldsymbol{c}}_{k+1-d} = (\widehat{c}_{k+1-d}, \widehat{c}_{k-d}, \ldots, \widehat{c}_{k+1-d-L})^{\mathrm{T}}. \quad (3.111)$$

- In the case of PSP-based LMS adaptive identification, the branch metric becomes

$$\lambda_k(a_k, S_k) = -\left| r_k - \breve{\boldsymbol{f}}_k(S_k)^{\mathrm{T}} \boldsymbol{c}_k(a_k, S_k) \right|^2 + 2\sigma^2 \ln P\{a_k\} \quad (3.112)$$

  where the channel estimate $\breve{\boldsymbol{f}}_k(S_k)$ depends in this case on the survivor associated with state S_k. The channel estimate update recursion can be written as follows:

$$\breve{\boldsymbol{f}}_{k+1}(S_{k+1}) = \breve{\boldsymbol{f}}_k(S_k) + \beta(r_k - \breve{\boldsymbol{f}}_k(S_k)^{\mathrm{T}} \breve{\boldsymbol{c}}_k(a_k, S_k))\breve{\boldsymbol{c}}_k^*(a_k, S_k) \quad (3.113)$$

  where

$$\breve{\boldsymbol{c}}_k(a_k, S_k) = [\breve{c}_k(a_k, S_k), \breve{c}_{k-1}(S_k), \ldots, \breve{c}_{k-L}(S_k)]^{\mathrm{T}}. \quad (3.114)$$

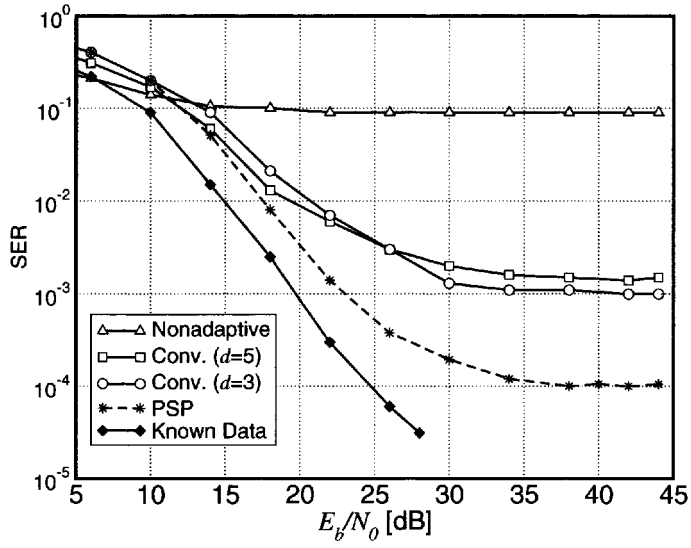


Figure 3.12: SER performance of uncoded QPSK transmission over a 3-tap dispersive fading channel and LMS-based adaptive detection. The normalized Doppler rate is $f_{\mathrm{D}}T = 1.85 \times 10^{-3}$. Reproduced from [34], ©1995 IEEE, by permission of the IEEE.

> The parameter estimate update recursions must take place along the transitions $(S_k \rightarrow S_{k+1})$ which extend the survivors of states $\{S_k\}$, i.e., those selected during the ACS step at time epoch k.

As an example of application, we consider the case of transmission of uncoded quaternary PSK (QPSK) ($M = 4$) over a Rayleigh fading channel with 3 independent tap weights. The channel power delay profile (standard deviations of the tap gains) is $\frac{1}{\sqrt{6}}(1, 2, 1)$. We consider data blocks of $K = 60$ symbols, with training preamble and tail. In Figure 3.12, the performance, in terms of symbol error rate (SER) versus signal-to-noise ratio (SNR), in the case of a normalized Doppler rate $f_{\mathrm{D}}T = 1.85 \times 10^{-3}$ is shown. In particular, in the 1.8 GHz band this Doppler rate corresponds to (i) $1/T = 24.3$ kHz in the case of a speed of 32.5 km/h and to (ii) $1/T = 270.8$ kHz for a speed equal to 300 km/h. Note that the results shown in Figure 3.12 refer to the case of full-state sequence detection with $Q = L = 2$ ($\zeta = 16$ states). In Figure 3.13, the performance in the case of a normalized Doppler rate $f_{\mathrm{D}}T = 3.69 \times 10^{-3}$ is shown. In particular, in the 1.8 GHz band this Doppler rate corresponds to (i) $1/T = 24.3$ kHz in the case of a speed of 65 km/h and to (ii) $1/T = 270.8$ kHz for a speed equal

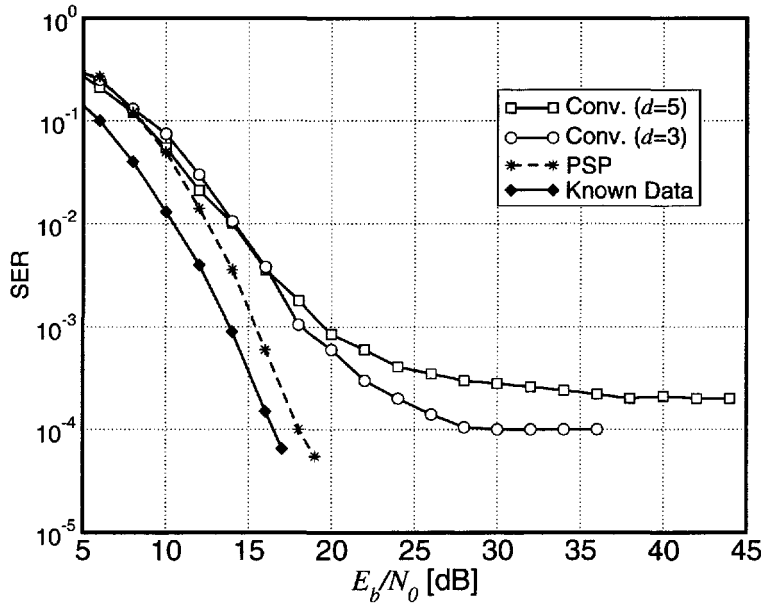


Figure 3.13: SER performance of uncoded QPSK transmission over a 3-tap dispersive fading channel and LMS-based adaptive detection with dual diversity. The normalized Doppler rate is $f_D T = 3.69 \times 10^{-3}$.

to 600 km/h. In this case as well, a full-state receiver is considered and dual diversity is used.

Joint Detection and Phase Synchronization

The problem of optimal detection of a possibly encoded information sequence transmitted over a bandpass AWGN channel is commonly approached by trying to approximately implement coherent detection. In applications in which a coherent phase reference is not available, this approximation is based on the use of a phase synchronization scheme, which extracts a phase reference from the incoming signal, in conjunction with a detection scheme which is optimal under the assumption of perfect synchronization. Since the reconstructed phase reference is only an approximation of the correct one, the overall detection scheme is only an approximation of ideal, i.e., with perfect phase reference, coherent detection. Although widely adopted, this solution should be regarded as just an ad hoc heuristic procedure based on one possible logical approach and will be referred to as *pseudocoherent*. In this subsection, we examine a simple strategy characterized by PLL-based estimation.

The model of the linearly modulated discrete observable is the following (valid in the case of slow variation):

$$r_k = e^{j\theta_k} c_k + n_k \tag{3.115}$$

where: θ_k is a channel-induced phase rotation; $\{c_k\}$ is the coded/modulated sequence generated by an encoder/modulator modeled as an FSM with state μ_k; and $\{n_k\}$ is an iid Gaussian noise sequence with variance per component equal to σ^2. In this case, the feedback parameter estimation is based on the use of a first order data-aided PLL and reads as follows:

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k + \eta \Im \left\{ r_{k+1-d} e^{-j\widehat{\theta}_k} c^*_{k+1-d} \right\} \tag{3.116}$$

where the parameter η controls the loop bandwidth. In the following, we briefly recall two possible phase tracking strategies: decision-directed and PSP-based.

- In the case of *decision-directed* phase tracking, the branch metric can be written as follows:

$$\lambda_k(a_k, S_k) = -\left| r_k e^{-j\widehat{\theta}_k} - c_k(a_k, \mu_k) \right|^2 + 2\sigma^2 \ln P\{a_k\} \tag{3.117}$$

  where $c_k(a_k, S_k)$ is the code symbol associated with transition (a_k, μ_k). The PLL phase-update (feedback) recursion is

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k + \eta \Im \left\{ r_{k+1-d} e^{-j\widehat{\theta}_k} \widehat{c}^*_{k+1-d} \right\}. \tag{3.118}$$

  The tentative decision delay d must comply with the causality condition upon the detected data, which implies $d \geq 1$.

- In the case of *PSP-based* phase tracking, the branch metric can be written as follows:

$$\lambda_k(a_k, S_k) = -\left| r_k e^{-j\breve{\theta}_k(\mu_k)} - c_k(a_k, \mu_k) \right|^2 + 2\sigma^2 \ln P\{a_k\}. \tag{3.119}$$

  The phase estimate update recursion in this case is

$$\breve{\theta}_{k+1}(\mu_{k+1}) = \breve{\theta}_k(\mu_k) + \eta \Im \left\{ r_k e^{-j\breve{\theta}(\mu_k)} \breve{c}^*_k(a_k, \mu_k) \right\} \tag{3.120}$$

  where $\breve{c}_k(a_k, \mu_k)$ is the code symbol associated with transition (a_k, μ_k). The phase estimate update recursions must take place along the transitions $(\mu_k \rightarrow \mu_{k+1})$ which extend the survivors of states $\{\mu_k\}$, i.e., those selected in correspondence to each state μ_{k+1} during the ACS step in correspondence to the trellis section at epoch k.

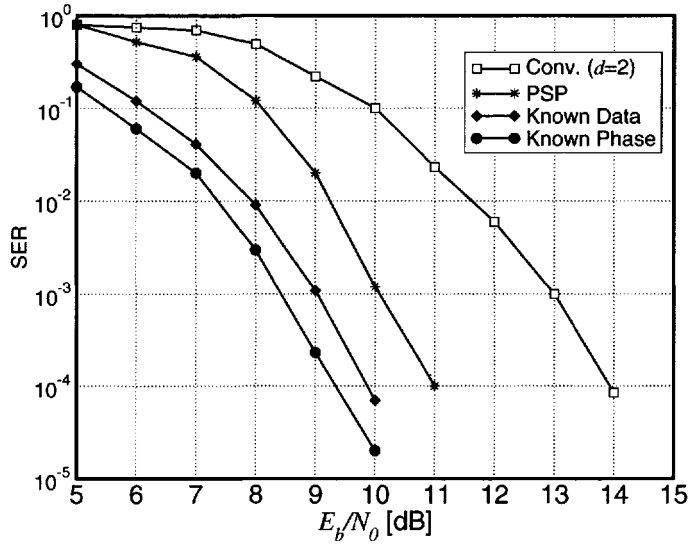


Figure 3.14: SER performance of PSP-based detection of a TCM with 8-PSK. For comparison, the performance of a conventional data-aided receiver (with $d = 2$) is also shown. Reproduced from [34], ©1995 IEEE, by permission of the IEEE.

As an example of application, we consider the transmission of TCM over a channel introducing a phase noise modeled as a Wiener process according to

$$\theta_{k+1} = \theta_k + \Delta_k \tag{3.121}$$

where Δ_k are Gaussian, iid random variables with standard deviation $\sigma_\Delta = 2$ degrees. In particular, the considered TCM scheme has $S_c = 4$ states and the output modulation is 8-PSK. The performance of a PSP-based receiver is shown in Figure 3.14, where it is also compared to that of a conventional data-aided receiver with $d = 2$. As one can immediately see, the improvement brought about by the use of a PSP-based detection strategy, compared with the case where a conventional detection strategy is used, is significant.

3.11.2 Noncoherent Sequence Detection for Phase-Uncertain Channels

Unlike in the approach considered in the previous subsection, where the use of a PLL and explicit recursive phase estimation were presented, noncoherent sequence

detection (NSD) stems from communication over channels affected by a random time-invariant phase rotation [47]—this channel is usually referred to as *noncoherent*. In [47], the starting point for the derivation of VA-based algorithms is the approximation of the overall likelihood function according to the considered channel model. In the following, we cast this class of receivers into the proposed finite memory detection framework introduced in Chapter 2. By first applying a CMP, we simply evaluate the basic branch metric on the basis of the specific noncoherent channel model. This approach is somehow the "specular version"[15] of that proposed in [47] and the final VA-based detection algorithm is obviously the same. We will further comment on this aspect below.

In principle, the time-invariant phase model is a key assumption because the information contained in the received signal may be shown to be unaffected by such a phase uncertainty, provided that a sufficiently long observation is available. On the other hand, the approximations involved in the derivation of the proposed detection schemes have the convenient side-effect of allowing time-varying phase models.

We consider the case of linear coded modulations (the interested reader is referred to [47] for an extension to the case of continuous phase modulation, CPM). The information sequence $\{a_n\}$, composed of independent symbols belonging to an M-ary alphabet, is mapped into a code sequence $\{c_n\}$ by means of some coding rule. This code sequence is further mapped by a modulator in cascade with a channel filter into a time-continuous signal with complex envelope $s(t, \boldsymbol{a})$, which depends on the information sequence denoted by the vector $\boldsymbol{a}$. This signal undergoes a phase rotation θ and is transmitted over an AWGN channel modeled by a complex-valued Gaussian white noise process $n(t)$ with independent components, each with two-sided power spectral density N_0. The complex envelope of the received signal may be expressed as

$$r(t) = s(t, \boldsymbol{a})e^{j\theta} + n(t)\,. \tag{3.122}$$

The phase rotation θ is modeled as a random variable with uniform distribution in the interval $[0, 2\pi)$. Hence, it is assumed to be constant during the entire transmission. In the case of absence of ISI, the discrete-time equivalent of (3.122) leads to the following expression for the observation at time epoch k:

$$r_k = c_k e^{j\theta} + n_k \tag{3.123}$$

where n_k has variance per component equal to $\sigma^2 = N_0$. In this case, since the statistical distribution of the parameter θ is known, it is possible to evaluate the conditional

[15] In [47], first there is *averaging* over the channel phase and *then* truncation, whereas here we first consider truncation and then averaging.

pdf in the branch metric characterizing a VA with branch metric based on the FMC. First of all, observe that

$$\begin{aligned} p(\boldsymbol{r}_{k-N}^{k}|\boldsymbol{a}_{k-N}^{k},\mu_{k-N}) &= \mathrm{E}_{\theta}\{p(\boldsymbol{r}_{k-N}^{k}|\boldsymbol{a}_{k-N}^{k},\mu_{k-N},\theta)\} \\ &= \int_{-\infty}^{\infty} p(\boldsymbol{r}_{k-N}^{k}|\boldsymbol{a}_{k-N}^{k},\mu_{k-N},\theta)p_{\theta}(\theta)\,\mathrm{d}\theta \\ &= \int_{0}^{2\pi} \frac{1}{(2\pi\sigma^2)^N} \exp\left(-\frac{\sum_{i=0}^{N}\left|r_{k-i}-c_{k-i}e^{j\theta}\right|^2}{2\sigma^2}\right)\frac{1}{2\pi}\,\mathrm{d}\theta \\ &\sim \exp\left(-\frac{\sum_{i=0}^{N}|c_{k-i}|^2}{2\sigma^2}\right) \\ &\quad \cdot \int_{0}^{2\pi} \exp\left(\frac{\Re\{e^{-j\theta}\sum_{i=0}^{N} r_{k-i}c_{k-i}^{*}\}}{\sigma^2}\right)\frac{1}{2\pi}\,\mathrm{d}\theta \\ &= \exp\left(-\frac{\sum_{i=0}^{N}|c_{k-i}|^2}{2\sigma^2}\right)\mathrm{I}_0\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^{*}\right|}{\sigma^2}\right) \end{aligned} \tag{3.124}$$

where $\mathrm{I}_0(\cdot)$ is the zero-th order modified Bessel function of the first kind [110]. Hence, one can immediately conclude that in this case the branch metric to be used to perform MAP sequence detection with a VA can be written as follows:

$$\begin{aligned} \lambda_k(a_k,S_k) &= \ln p(r_k|\boldsymbol{r}_{k-N}^{k-1},\boldsymbol{a}_{k-N}^{k},\mu_{k-N}) + \ln P\{a_k\} \\ &= \ln p(\boldsymbol{r}_{k-N}^{k}|\boldsymbol{a}_{k-N}^{k},\mu_{k-N}) - \ln p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{a}_{k-N}^{k-1},\mu_{k-N}) + \ln P\{a_k\} \\ &= -\frac{|c_k|^2}{2\sigma^2} + \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^{*}\right|}{\sigma^2}\right) - \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^{*}\right|}{\sigma^2}\right) \\ &\quad + \ln P\{a_k\}. \end{aligned} \tag{3.125}$$

Moreover, since $\ln \mathrm{I}_0(x) \simeq x$ for large values of the argument [110], (3.125) can be approximately computed as follows:

$$\begin{aligned} \lambda_k(a_k,S_k) &\simeq -\frac{|c_k|^2}{2\sigma^2} + \frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^{*}\right|}{\sigma^2} - \frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^{*}\right|}{\sigma^2} + \ln P\{a_k\} \\ &\sim -\frac{|c_k|^2}{2} + \left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^{*}\right| - \left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^{*}\right| + \sigma^2 \ln P\{a_k\}. \end{aligned} \tag{3.126}$$

Note that the metric (3.125) was obtained in [47]. However, as mentioned above in this subsection, the approach in [47] is specular to the general approach proposed in this book. More precisely, in [47] the sequence detection strategy is first written as follows

$$\hat{\mathbf{a}} = \underset{\tilde{\mathbf{a}}}{\operatorname{argmax}} \left\{ -\frac{1}{2N_0} \int_{T_0} |s(t,\tilde{\mathbf{a}})|^2 \, \mathrm{d}t + \log \mathrm{I}_0 \left(\frac{1}{N_0} \left| \int_{T_0} r(t) s^*(t,\tilde{\mathbf{a}}) \, \mathrm{d}t \right| \right) \right\} \tag{3.127}$$

where N_0 is the one-sided power spectral density of the AWGN process. From the likelihood function (3.127), it is possible to derive an expression for a partial path metric. At this point, by considering a truncation assumption on the observation window (i.e., equivalent to the FMC considered before) on which a partial path metric depends, one can immediately obtain (3.125).

The performance of receivers using this detection strategy is assessed by means of computer simulation in terms of BER versus E_b/N_0. Besides full-state receivers, RS techniques introduced in Section 3.10 will be considered to limit the complexity of the proposed solutions.

As a first example of application, the performance of 16-ary differential QAM (16-DQAM) is analyzed. In particular, we assume that symbols $\{c_n\}$ belong to a 16-QAM alphabet and are derived from information symbols $\{a_n\}$, belonging to the same alphabet, by means of the following *quadrant differential encoding* rule [111]. The generic information symbol is uniquely represented as $a_n = \mu_n p_n$, where μ_n belongs to the first quadrant and $p_n \in \{\pm 1, \pm j\}$. The encoded symbol c_n is given by $c_n = \mu_n q_n$, where $q_n \in \{\pm 1, \pm j\}$ is defined by $q_n = p_n q_{n-1}$, i.e., the usual differential encoding rule for QPSK modulations applied to symbols $\{p_n\}$. The branch metric (3.126) must be expressed in terms of the information symbols in such a way that the VA also implements differential decoding. Multiplying the terms $|\sum_{i=0}^{N-1} r_{n-i} c_{n-i}^*|$ and $|\sum_{i=1}^{N-1} r_{n-i} c_{n-i}^*|$ in (3.126) by $|q_n| = 1$, noting that[16]

$$\tilde{q}_n c_{n-i}^* = \mu_{n-i}^* q_{n-i}^* q_n = \mu_{n-i}^* \prod_{m=0}^{i-1} p_{n-m} \tag{3.128}$$

[16]Symbols μ_n, p_n, q_n are defined by this differential encoding rule applied to the data sequence $\{a_n\}$.

and $|c_n| = |a_n| = |\mu_n|$, from (3.126) one can immediately obtain:

$$\lambda_k(a_k, S_k) = \left| \sum_{i=0}^{N} r_{k-i}\mu_{k-i}^* \prod_{m=0}^{i-1} p_{k-m} \right| - \left| \sum_{i=1}^{N} r_{k-i}\mu_{k-i}^* \prod_{m=0}^{i-1} p_{n-m} \right| - \frac{1}{2}|a_n|^2 \tag{3.129}$$

where the trellis state is evidently defined as

$$\begin{aligned} S_n &\triangleq (\mu_{n-1}, \mu_{n-2}, \ldots, \mu_{n-N}, p_{n-1}, p_{n-2}, \ldots, p_{n-N+1}) \\ &= (a_{n-1}, a_{n-2}, \ldots, a_{n-N+1}, \mu_{n-N}) . \end{aligned} \tag{3.130}$$

The number of states $\zeta = \frac{1}{4}M^N$ depends exponentially on the integer N. Complexity reduction techniques may also be adopted. We only consider schemes with a number of states $\zeta' = 4$ (i.e., the reduced state is defined as $w_n = \tilde{\mu}_n$) and $\zeta' = 1$ (i.e., symbol-by-symbol detection with decision-feedback is performed). Figure 3.15 shows the relevant performance, obtained by computer simulation, for various values of N and compares it with that of optimal coherent detection. It may be observed that the proposed receivers exhibit a loss of only 0.5 dB at a BER of 10^{-5} with respect to coherent detection, with affordable complexity. For larger complexity, the performance approaches that of coherent detection. We also note that the performance tends to that of coherent detection with a rate which is independent of the SNR.

We then consider the application of NSD to TCM. Two 8-state trellis coded (TC) 16-QAM schemes, optimal under coherent detection, are considered: the first is a nonrotationally invariant (NRI) code [9] and the second a 90° rotationally invariant (RI) code [57]. The RI code is used in conjunction with a differential encoder and, in the proposed algorithms, the VA also implements differential decoding. Various noncoherent receivers with different complexities have been analyzed. Figure 3.16 shows the performance of the considered receivers along with that of coherent detection. Receivers based on the code trellis ($\zeta' = 8$) exhibit a performance loss of about 1.5 dB (for $N = 8$), but with an increase of the number of states to $\zeta' = 64$ the performance loss becomes negligible. The state of the receivers with $\zeta' = 16$ is defined by a complete representation of the previous information symbol $\tilde{a}_{n-1}$ and a partial representation of symbol $\tilde{a}_{n-2}$.

The proposed approach can also be extended to the case of ISI channels. In fact, proceeding as in [112], an alternative set $\{z_k\}$ of sufficient statistics an be obtained through a whitened matched filter (WMF) whose output may be expressed as

$$z_k = y_k e^{j\theta} + n_k \tag{3.131}$$

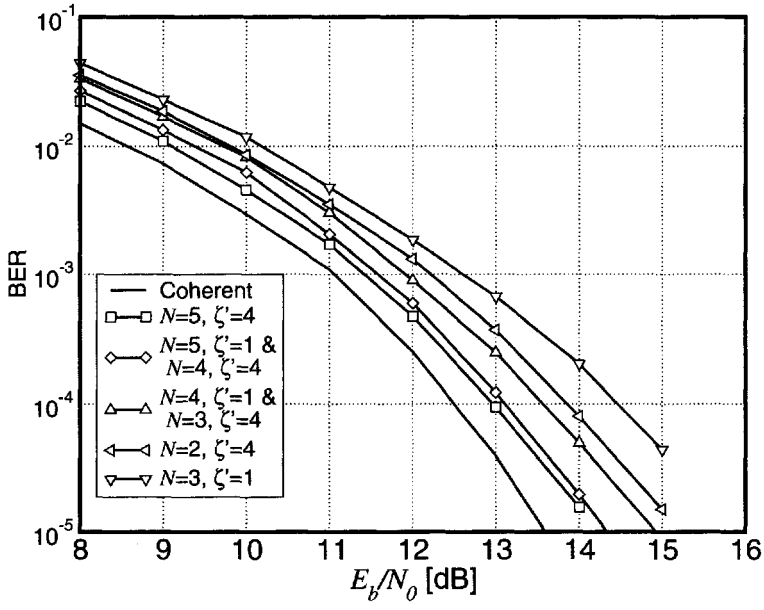


Figure 3.15: BER of NSD detection schemes for 16-DQAM with various degrees of complexity. Reproduced from [47], ©1999 IEEE, by permission of the IEEE.

in which $\{n_k\}$ are Gaussian zero-mean independent complex random variables of variance per component equal to $\sigma^2 = N_0$ and

$$y_k \triangleq \sum_{l=0}^{L} f_l c_{k-l} \tag{3.132}$$

where L is a channel memory parameter. The equivalent discrete-time channel impulse response $\{f_k\}$ can be obtained from sequence $\{g_k\}$ following the method described in [112], where $g_k \triangleq g(kT)$ represents a sample of the overall channel impulse response $g(t)$. Reasoning in the same way as in the absence of ISI, it can be immediately concluded that, in this case, the branch metric can be eventually written as follows:

$$\lambda_k(a_k, S_k) \triangleq \left| \sum_{i=0}^{N} z_{k-i} y_{k-i}^* \right| - \left| \sum_{i=1}^{N} z_{k-i} y_{k-i}^* \right| - \frac{1}{2}|y_k|^2 . \tag{3.133}$$

As an example of application, we consider the case of transmission of 16-DQAM over two ISI channels with $L = 2$. The receiver is composed of a WMF and a VA based on branch metric (3.133). The considered channels correspond to the following overall discrete impulse responses at the output of the WMF: $(f_0, f_1, f_2) =$

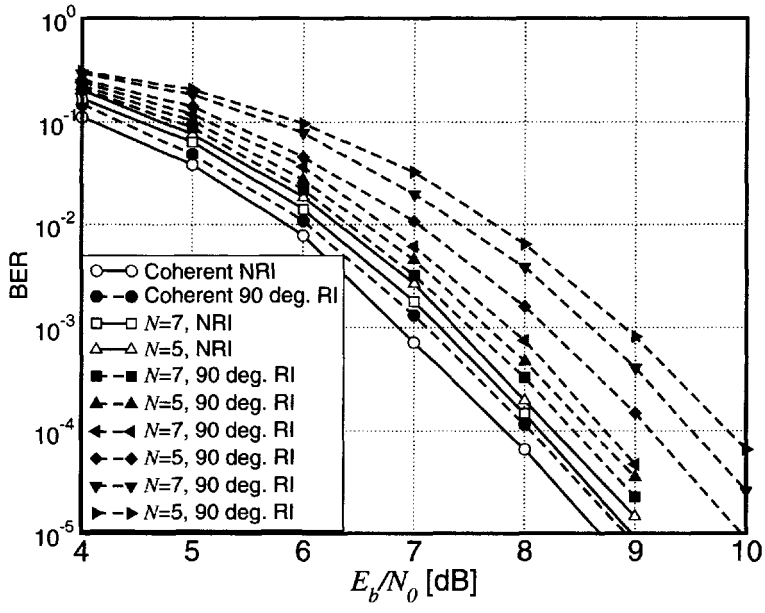


Figure 3.16: BER of NSD detection schemes for 8-state TC-16-QAM. Reproduced from [47], ©1999 IEEE, by permission of the IEEE.

$(0.3, 0.9, 0.3)$ for Channel 1 and $(f_0, f_1, f_2) = (0.408, 0.816, 0.408)$ for Channel 2 [17]. The optimal coherent receiver is characterized by $S_c = 256$ states—in this case the ISI channel is the underlying FSM. In Figure 3.17, the performance of the optimal coherent receiver for both channels is shown and compared with the performance of the proposed NSD schemes with various values of N. For a given value of N, the number of states resulting from (3.133) is $\zeta = \frac{1}{4}M^{N+L} = 4 \times 16^{N+1}$. In the simulations, we used the described state-complexity reduction techniques and considered a number of states $\zeta' = 256$, i.e., the same number of states of the optimal coherent receivers for the two channels. The state of these noncoherent receivers is defined by a complete representation of the symbols a_{n-1} and a_{n-2}. Despite the constraint on state-complexity, for increasing values of N the BER tends to approach that of coherent detection. State-complexity reduction has also been considered, both in noncoherent and in coherent receivers. In the figure, the performance for $\zeta' = 16$ states is shown for the considered channels. The state definition takes into account a complete representation of the single symbol a_{n-1}. We then have a two-fold comparison between a coherent receiver and an NSD-based receiver. With reference to the proposed noncoherent receivers, the figure shows their performance loss, under an equal state-complexity constraint, and their complexity increase, under an approxi-

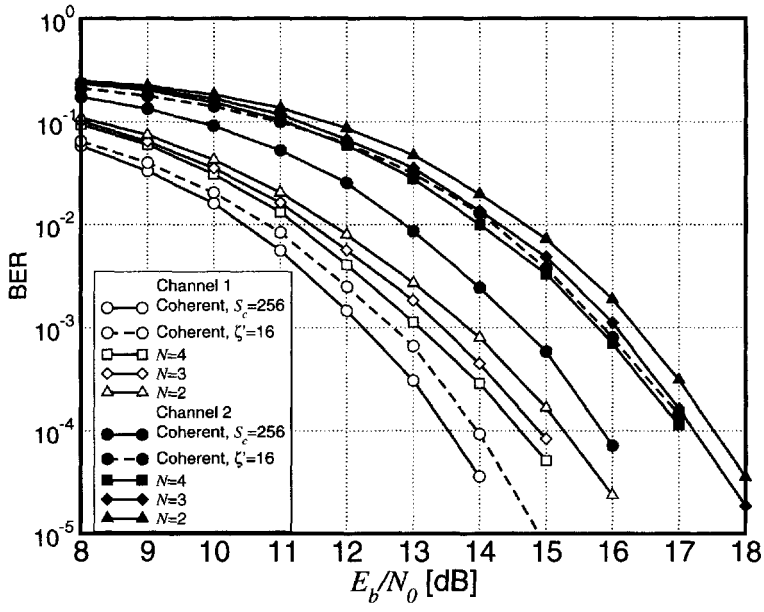


Figure 3.17: BER of the proposed detection schemes for 16-DQAM on the two considered ISI channels and various values of N. The noncoherent detectors search a trellis with $\zeta' = 256$ states. Reproduced from [47], ©1999 IEEE, by permission of the IEEE.

mate equal-performance constraint.

As stated in Chapter 1, it is possible to derive an algorithm considering a particular channel model (which usually simplifies the design) and then apply the same algorithm to other channel models—which must be obviously related to the original channel model and somehow compatible with it. Owing to the FMC applied in the derivation of the proposed detection algorithm for phase noncoherent channels, these detection algorithms can also be applied in the case of dynamic channel conditions. As an example, the performance under dynamic channel conditions has also been investigated, assuming the transmitter and receiver filters have square root raised-cosine frequency response with roll-off 0.5. Two types of time-varying phase model are jointly considered.

- The first is the well-known stochastic model of phase jitter. Accordingly, the phase of the received signal is modeled as a time-continuous Wiener process $\theta(t)$ with incremental variance over a signaling interval equal to σ_Δ^2.

- The second is a deterministic model of a frequency offset Δf.

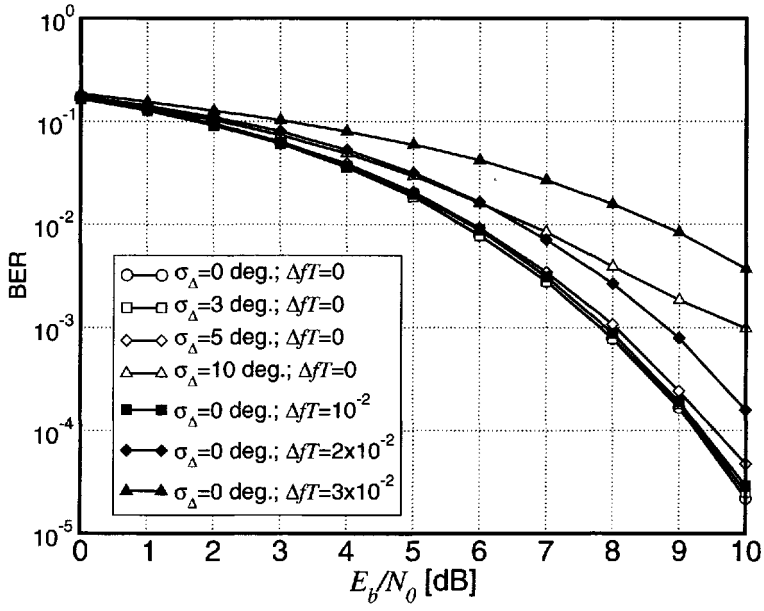


Figure 3.18: BER of the proposed receiver for DQPSK with $N = 5$ and $\zeta' = 1$ for various values of phase jitter standard deviation (white marks) and frequency offset (black marks). Reproduced from [47], ©1999 IEEE, by permission of the IEEE.

The proposed noncoherent receivers are robust to phase jitter and frequency offset, as may be observed in Figure 3.18 for differential QPSK (DQPSK), $N = 5$ and $\zeta' = 1$. A jitter standard deviation up to 5 degrees (per signaling interval) does not degrade significantly the receiver performance. Values of the frequency offset Δf, normalized to the symbol frequency $1/T$, up to $\Delta f T = 10^{-2}$ do not entail appreciable degradation.

The proposed approach can also be extended to the case of transmission over a channel affected by *frequency uncertainty*, besides phase noise. In this case, the assumed model is depicted in Figure 3.19, where θ is modeled as a random variable with uniform distribution in $[0, 2\pi)$, while the frequency offset ν is assumed deterministic. We consider the case of linearly modulated signals, so that the information bearing signal can be written as

$$s(t, \mathbf{a}) = \sum_{k=0}^{K-1} c_k h(t - kT) \tag{3.134}$$

where: K denotes the number of transmitted code symbols; T is the signaling interval; and $h(t)$ is a properly normalized shaping pulse. Assuming perfect symbol

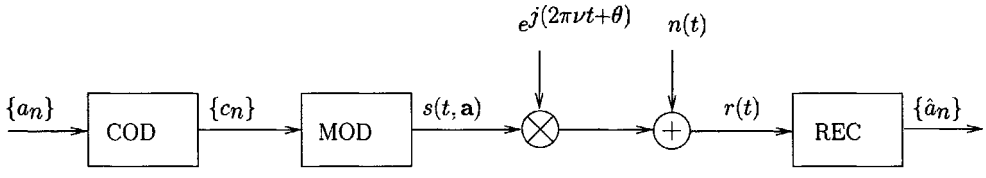


Figure 3.19: System model in the case of a channel with phase and frequency uncertainty.

synchronization and absence of frequency offset, the output, sampled at time $t = kT$, of a filter matched to the shaping pulse $h(t)$ is a sufficient statistic for this detection problem. If a moderate frequency offset is present, this is not true in a strict sense; however, this sampled output may still be considered as an approximate sufficient statistic. This assumption is commonly used in the derivation of frequency estimation algorithms (see [113, 114] and references therein).[17] Assuming that a genie tells us the value of the (unknown but deterministic) frequency offset, one could use the following metric, obtained by directly extending the metric in (3.125):

$$\begin{aligned}\lambda_k(a_k, S_k) &= -\frac{|c_k|^2}{2\sigma^2} + \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^* e^{-j2\pi(k-i)\nu T}\right|}{\sigma^2}\right) \\ &\quad - \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^* e^{-j2\pi(k-i)\nu T}\right|}{\sigma^2}\right) + \ln P\{a_k\} \\ &\simeq -\frac{|c_k|^2}{2\sigma^2} + \frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^* e^{-j2\pi(k-i)\nu T}\right|}{\sigma^2} \\ &\quad - \frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^* e^{-j2\pi(k-i)\nu T}\right|}{\sigma^2} + \ln P\{a_k\} \end{aligned} \tag{3.135}$$

where σ^2 is the variance per component of the AWGN sample. In particular, it is possible to consider an estimate based on the L most recent observations $\boldsymbol{r}_{k-L}^k$ and code symbols $\boldsymbol{c}_{k-L}^k$, so that we can write:

$$\widehat{\nu} = \widehat{\nu}(a_k, \ldots, a_{k-L}, \mu_{k-L}, \boldsymbol{r}_{k-L}^k) \tag{3.136}$$

[17] For larger values of the frequency offset ν, a different front-end, possibly based on oversampling techniques, could be used [94].

and rewrite (3.135) as follows:

$$\begin{aligned}\lambda_k(a_k, S_k) &= -\frac{|c_k|^2}{2\sigma^2} + \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=0}^{N} r_{k-i} c_{k-i}^* e^{-j2\pi(k-i)\widehat{\nu}T}\right|}{\sigma^2}\right) \\ &\quad - \ln \mathrm{I}_0\left(\frac{\left|\sum_{i=1}^{N} r_{k-i} c_{k-i}^* e^{-j2\pi(k-i)\widehat{\nu}T}\right|}{\sigma^2}\right) + \ln P\{a_k\} \\ &\simeq -\frac{|c_k|^2}{2\sigma^2} + \frac{\left|\sum_{i=0}^{N} r_{k-i} c_{k-i}^* e^{-j2\pi(k-i)\widehat{\nu}T}\right|}{\sigma^2} \\ &\quad - \frac{\left|\sum_{i=1}^{N} r_{k-i} c_{k-i}^* e^{-j2\pi(k-i)\widehat{\nu}T}\right|}{\sigma^2} + \ln P\{a_k\}. \end{aligned} \tag{3.137}$$

The frequency estimate $\widehat{\nu}$ may be obtained, for instance, from ML data-aided estimation based on the algorithm proposed in [115]. However, in general, a simpler data-aided algorithm may be used. We consider some of the algorithms described in [114], namely those in [116] and in [113]. In the technical literature, these algorithms have been used for M-ary PSK (M-PSK) signals [113, 114]. They may be easily extended to the general case of non-equal-energy signals such as M-ary QAM (M-QAM) [117]. In particular, in [117] the starting point for the derivation of this class of algorithms is the likelihood function for joint data detection and frequency estimation, which can be written as

$$\Lambda_K(\boldsymbol{a}, \nu) = \sigma^2 \ln \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{k=0}^{K-1} r_k c_k^* e^{-j2\pi k\nu T}\right|\right) - \frac{1}{2}\sum_{k=0}^{K-1} |c_k|^2 \tag{3.138}$$

where; $\boldsymbol{c} = \boldsymbol{c}_0^{K-1}$ is the code sequence uniquely associated with the information sequence $\boldsymbol{a}$ by the given coding rule; ν denotes a trial value of the frequency offset; and $r_k \stackrel{\triangle}{=} r(t) \otimes h(-t)|_{t=kT}$ is the matched filter output, sampled at time $t = kT$. The likelihood function (3.138) is similar to that obtained in [47] in the absence of frequency offsets—the likelihood function considered in [47] may be obtained by letting $\tilde{\nu} = 0$ in (3.138). A property of interest of the likelihood function (3.138) is now described. Let us assume that two code sequences $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$ corresponding to distinct information sequences $\boldsymbol{a}^{(1)}$ and $\boldsymbol{a}^{(2)}$, respectively, are such that $c_k^{(2)} = c_k^{(1)} e^{j(2\pi\nu_0 kT+\theta)}$, where ν_0 is some frequency value and θ is some phase rotation. The structure of the likelihood function (3.138) implies that

$$\Lambda_K(\boldsymbol{a}^{(1)}, \tilde{\nu}) = \Lambda_K(\boldsymbol{a}^{(2)}, \tilde{\nu} + \nu_0)\,. \tag{3.139}$$

A consequence of this property is that if the coding rule admits such sequences, a decoding strategy based on (3.138), directly or by means of some approximations, will not be able to distinguish between the information sequences $\boldsymbol{a}^{(1)}$ and $\boldsymbol{a}^{(2)}$.

For generality, the concept of *phase rotation of order* l may be introduced. A time-varying phase rotation ϕ_k is said to be of order l if it may be expressed as

$$\phi_k = \phi^{(l)}(kT)^l + \phi^{(l-1)}(kT)^{l-1} + \cdots + \phi^{(1)}kT + \phi^{(0)} \tag{3.140}$$

where $\phi^{(l)}, \ldots, \phi^{(0)}$ are suitable constants. The above property (3.139) may be reformulated in the following terms: two code sequences which differ for a phase rotation of the *first* order are indistinguishable. For this reason, a code which admits code sequences that differ for a first order phase shift is *catastrophic* when decoded by means of the strategy entailed by (3.138). Note that this problem resembles, in a higher order sense, the characteristic of noncoherent receivers of being unable to distinguish code sequences which differ for a phase rotation of zero-th order [47]. Although the usual differential encoding (DE) rule solves this zero-th order ambiguity, it is catastrophic in a first order sense. In fact, DE has the property that two code sequences which differ for a zero-th order phase rotation are associated with the same information sequence, but code sequences which differ for a first order phase rotation may derive from different information sequences.

The proposed detector with branch metric (3.135) may be affected by the catastrophic property (3.139). In order to overcome this problem, two possible solutions can be derived by extending simple countermeasures for the zero-th order ambiguity problems of noncoherent receivers to the first order case. An example of indistinguishable sequences in a noncoherent receiver for uncoded QPSK modulation is shown in Figure 3.20 (a). Sequence $\boldsymbol{a}^{(2)}$ is a rotation of sequence $\boldsymbol{a}^{(1)}$ by $\pi/2$ and cannot be distinguished. However, if the phase rotation θ is limited as $|\theta| < \pi/4$, as in the case of the sequence $\{z_k\} = \{a_k^{(1)} e^{j\theta}\}$, the sequence $\boldsymbol{a}^{(1)}$ can be correctly recovered. Alternatively, differential encoding/decoding associates code sequences which differ for a constant phase shift with the same information sequence, eliminating this ambiguity. In Figure 3.20(b), a first order analogy is shown. Sequence $\boldsymbol{a}^{(2)}$ is a first order rotation of sequence $\boldsymbol{a}^{(1)}$ with $\nu T = 1/4$ and cannot be distinguished.[18] However, if the frequency offset is limited to $|\nu T| < 1/8$, as in the case of sequence $\{z_k\} = \{a_k^{(1)} e^{j2\pi\nu kT}\}$, the ambiguity may be resolved. Alternatively, double differential encoding (DDE) with suitable differential decoding may be adopted [45, 118]. These solutions are described in the following.

The first solution (method 1) consists of limiting the frequency offset range and estimation interval. Denote by ψ_0 the angle of invariance of the considered constel-

[18] In (3.140), $\phi^{(1)} = 2\pi\nu$.

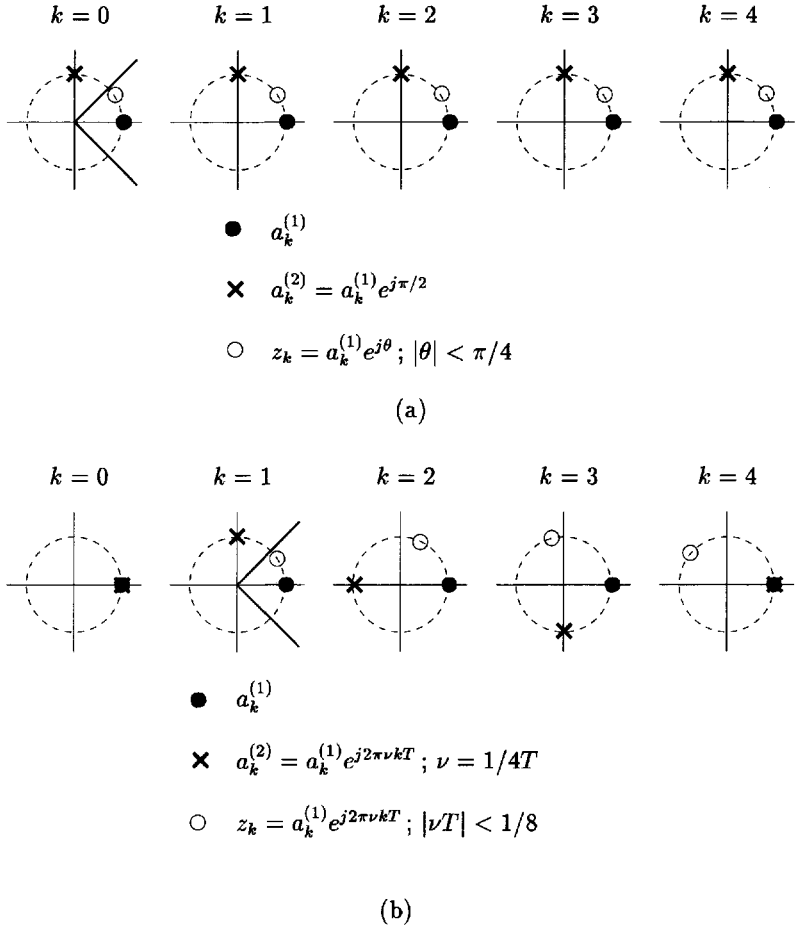


Figure 3.20: Examples of indistinguishable sequences: (a) noncoherent receiver (zero-th order); (b) advanced receiver with frequency estimation (first order). Reproduced from [117], ©2002 IEEE, by permission of the IEEE.

lation. In the case of M-PSK signals, $\psi_0 = 2\pi/M$, whereas, in the case of square M-QAM, $\psi_0 = \pi/2$. As $2\pi\nu T$ is the phase rotation introduced by the channel in one signaling interval, when $-\psi_0/2 \leq 2\pi\nu T \leq \psi_0/2$, this phase rotation does not yield ambiguities. At the same time, this condition should be also satisfied by the estimate $\widehat{\nu}_L(\tilde{\boldsymbol{a}}_n)$. To this purpose, each estimate such that $|2\pi\widehat{\nu}_L(\tilde{\boldsymbol{a}}_n)T| > \psi_0/2$ may be replaced by the closest value in the allowed interval. This method may only be applied for limited values of the frequency offset. That means that for QPSK the condition $|\nu T| \leq 1/8 = 0.125$ must be satisfied, whereas the allowed interval reduces to $|\nu T| \leq 0.03125$ for 16-PSK constellations. To avoid zero-th order ambiguities, DE

can be used.

An alternative solution (method 2) consists of the use of DDE, possibly followed by a code invariant to first order rotations.[19] As a consequence, unlike the previous case in which a limitation of the estimation interval must be imposed at the receiver, this solution does not require any modifications of the receiver operation (except for the fact that the new encoding rule must be taken into account). DDE is described for M-PSK modulations in [45, Chapter 8] and [118]. In this case, symbols $\{c_k\}$ belonging to an M-PSK alphabet are derived from symbols $\{a_k\}$, belonging to the same alphabet, by means of the following DDE rule:

$$c_k = a_k c_{k-1}^2 c_{k-2}^* \,. \tag{3.141}$$

In the case of M-QAM signals, *quadrant* DDE may be used. Expressing the generic information symbol of an M-QAM constellation as $a_k = \mu_k p_k$, where μ_k belongs to the first quadrant and $p_k \in \{\pm 1, \pm j\}$, the encoded symbol c_k is given by $c_k = \mu_k q_k$, where $q_k \in \{\pm 1, \pm j\}$ are defined by the DDE rule for QPSK modulations applied to symbols $\{p_k\}$, that is according to the rule

$$q_k = p_k q_{k-1}^2 q_{k-2}^* \,. \tag{3.142}$$

At this point it is possible to consider a PSP-based estimate of the frequency offset. In other words, the frequency estimate could be based on the survivor associated with a reduced state w_k, so that one can substitute ν by $\widehat{\nu}$, defined as

$$\widehat{\nu} = \widehat{\nu}(a_k, w_k, \mathrm{FS}(w_k)) \tag{3.143}$$

where the state w_k has to be properly defined and $\mathrm{FS}(w_k)$ indicates the survivor[20] associated with the reduced state w_k. Note that in (3.136) the definition of $\mathrm{FS}(w_k)$ is very general, since, depending on the coding/modulation structure, the definition of the survivor could change. Moreover, this technique allows one to choose the parameters N and L independently from the number of states ζ of the VA. The number of reduced states ζ' may therefore be limited retaining sufficiently large values for N and L. In order to compute the branch metric in an RS trellis, the necessary symbols not included or not completely specified in the reduced state definition w_k are found in the survivor history embedded in $\mathrm{FS}(w_k)$.

The performance of the receiver based on branch metric (3.137) for DQPSK, $N = 6$, $\zeta' = 16$, various values of L, and limitation of the estimation interval (method 1), is shown in Figure 3.21. The algorithm in [116] has been used for frequency

[19] These codes may be viewed as an extension of the usual RI codes [119].

[20] The terminology used to indicate the survivor, i.e., FS, recalls the fact that the recursion in the VA proceeds, time-wise, forwards. In Chapter 4 it will be shown that in an FB algorithm the backward recursion might rely on a backward survivor.

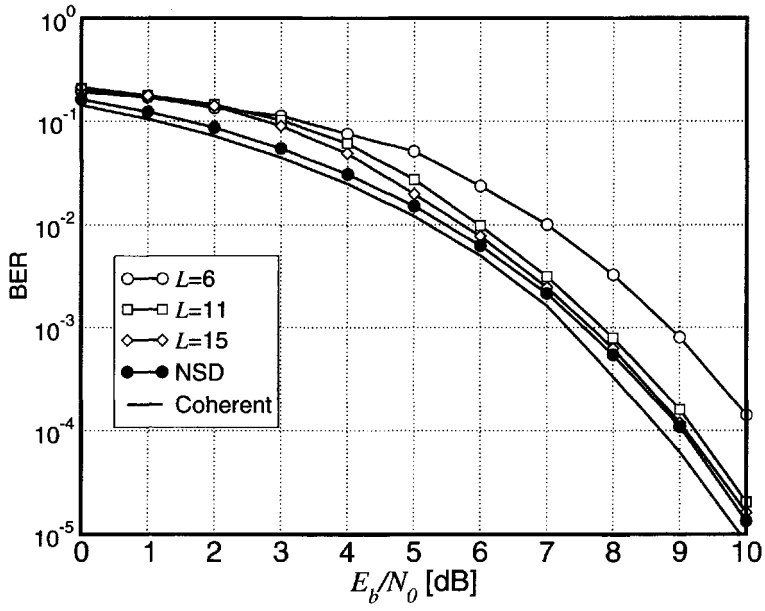


Figure 3.21: BER of the receiver based on (3.137) (white marks) for DQPSK and comparison with NSD (black marks) and coherent receivers. The frequency offset is $\nu = 0$. Reproduced from [117], ©2002 IEEE, by permission of the IEEE.

estimation, but no differences can be observed using more complex algorithms such as those in [113, 115]. The optimal coherent receiver and a NSD receiver with $N = 6$ and $\zeta' = 16$ are also considered for comparison. The frequency offset is $\nu = 0$. It may be observed that, for increasing values of L, the performance of the NSD receiver may be approached at least at high SNR. For $L = 6$, a loss of about 1 dB is exhibited at a BER of 10^{-4}. This power loss reduces to 0.2 dB for $L = 11$ and is negligible for $L = 15$.

In Figure 3.22, the performance of the proposed receiver based on branch metric (3.137) under dynamic channel conditions is evaluated. It is also compared with that of an NSD receiver in order to assess its robustness. The modulation format and the considered receiver are those in the previous figure with $L = 11$. The transmit and receive filters have square root raised-cosine frequency response with roll-off 0.5.[21] In addition to a frequency offset, a phase noise is considered, modeled as a time-continuous Wiener process with incremental standard deviation over a signaling interval equal to σ_Δ. The performance of the NSD receiver rapidly degrades for values of normalized frequency offset greater than 10^{-2}, whereas the proposed receiver is

[21] Under dynamic channel conditions, this specifi cation is necessary as well.

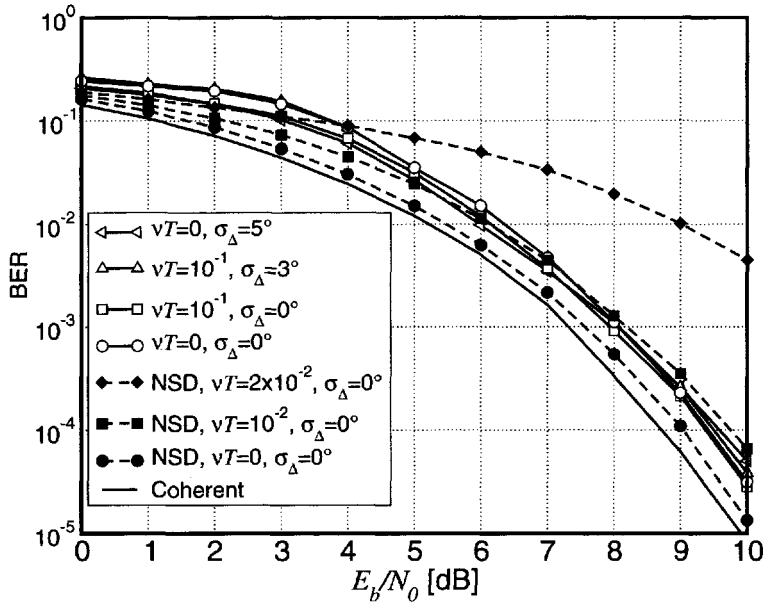


Figure 3.22: BER of the receiver based on (3.137) (white marks) for DQPSK, $N = 6$, $L = 11$, and $\zeta' = 16$. The performance of an NSD receiver (black marks) with $N = 6$ and $\zeta' = 16$ is also shown for comparison. Reproduced from [117], ©2002 IEEE, by permission of the IEEE.

practically unaffected by frequency offset up to 10^{-1} and phase noise standard deviation up to 5 degrees. As already mentioned, in the case of DQPSK, the limiting value of $|\nu T|$ is 0.125 (method 1). On the other hand, for higher values of $|\nu T|$ the receiver performance would be limited, in any case, by the front-end bandwidth and the interference generated by the resulting mismatch. As previously noted, the algorithm used for frequency estimation is irrelevant.

In the previous figures, receivers based on branch metric (3.137) and limitation of the estimation interval (method 1) for DQPSK were considered. Alternatively, method 2 can be used, but in this case DDE has to be employed. The different behavior of the receivers based on the two methods as well as the robustness of the proposed schemes is emphasized in Figure 3.23, in which the BER versus the normalized frequency offset νT, for $E_b/N_0 = 10$ dB, is shown for two receivers. The modulation format, the transmit and the receive filters are those of the previous figure. The curve with white marks corresponds to the performance of the receiver based on branch metric (3.137) with $N = 6$, $L = 6$, and $\zeta' = 16$ for DQPSK with limitation of the estimation interval (method 1). The curve with black marks corresponds to

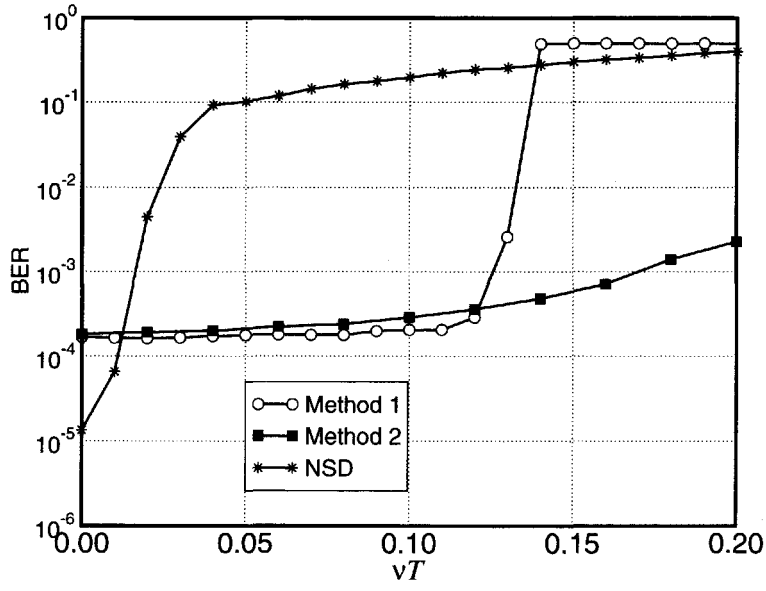


Figure 3.23: BER at an SNR of 10 dB versus the normalized frequency offset of the receiver based on (3.137) for DQPSK and $N = 6$, $L = 6$, and $\zeta' = 16$. Method 1 (white marks) based on a limitation of the estimation interval and method 2 (black marks) based on DDE are considered. The performance of an NSD receiver with $N = 6$ and $\zeta' = 16$ is also shown for comparison. Reproduced from [117], ©2002 IEEE, by permission of the IEEE.

a receiver based on the same branch metric, the same values of N, L, and ζ', and using DDE (method 2). Frequency offset values up to 10% of the symbol rate do not affect the receiver performance. We can also note that the performance of a receiver based on method 1 rapidly degrades when the normalized frequency offset exceeds the limiting value which, for QPSK, is 0.125. On the contrary, the receiver based on method 2 has a performance which slowly degrades with the frequency offset due to the front-end filter. The performance of the NSD receiver already degrades rapidly for ν equal to 1% of the signaling interval.

3.11.3 Noncoherent Sequence Detection for Slowly Varying Frequency Nonselective Fading Channels

The system model is shown in Figure 3.24. An information sequence $\{a_n\}$, composed of iid symbols belonging to an M-ary alphabet, is mapped into a code sequence by means of some coding rule. We assume a burst-mode transmission with K code

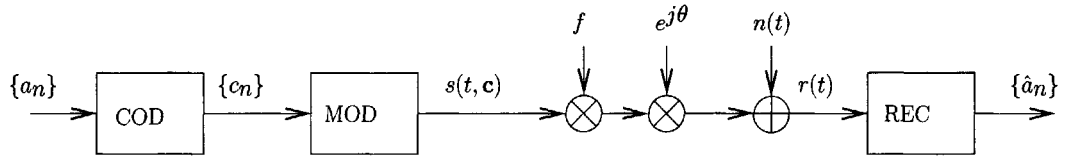


Figure 3.24: System model for transmission over a frequency nonselective fading channel. Reproduced from [120], ©2000 IEEE, by permission of the IEEE.

symbols per burst and denote with $\boldsymbol{c} = (c_0, c_1, \ldots, c_{K-1})^{\mathrm{T}}$ a column vector whose elements are the code symbols transmitted in a burst. The code sequence is further mapped by a modulator into a time-continuous signal $s(t, \boldsymbol{c})$ which is transmitted over a frequency nonselective slowly fading channel, represented by a multiplicative complex random variable f, modeled as Gaussian with mean η_f and variance σ_f^2. The transmitted signal also undergoes a phase rotation θ, modeled as a random variable with uniform distribution in the interval $[0, 2\pi)$ and independent of f. Although both f and θ are assumed to be constant during the transmission of each data burst, the application of the CMP has the convenient feature of allowing one to remove the channel time-invariance assumption and encompass time-varying models.

The choice of the value of η_f allows us to take into account both Rayleigh ($\eta_f = 0$) and Rice fading distributions, with a factor $K_{\mathrm{R}} \triangleq |\eta_f|^2/\sigma_f^2$. The channel also introduces AWGN of complex envelope $n(t)$ with independent real and imaginary components, each with single-sided power spectral density σ^2. Assuming $\eta_f = 1$ and $\sigma_f^2 = 0$, one obtains the limiting case of an AWGN channel.

Assuming that signal $s(t, \boldsymbol{c})$ is linearly modulated, it may be easily shown that sampling the output of a filter matched to the shaping pulse, with one sample per signaling interval, yields a sufficient statistic for optimal detection of the information sequence.[22] In the absence of ISI, this sampled output may be expressed as

$$r_k = f e^{j\theta} c_n + n_k \tag{3.144}$$

where $\{n_k\}$ are iid complex noise samples with independent components, each with variance σ^2. In the following, the samples $\{r_n\}$ will be referred to as *observations*.

Perfect Channel State Information

We now assume that the receiver has perfect channel state information (CSI), represented by the channel gain $|f|$. As an example, this knowledge may be well ap-

[22]This is not true in the case of a time-varying fading channel, where a sufficient statistic may require the use of *oversampling* [121].

proximated by means of a simple data-aided estimate based on a known preamble preceding the data burst. In the case of perfect CSI, an alternative sufficient statistic may be obtained by normalizing the received samples $\{r_n\}$ using the known gain value as

$$r'_n \triangleq \frac{r_n}{|f|}. \tag{3.145}$$

Using polar coordinates to express the fading coefficient as $f = |f|e^{j\alpha}$, we have

$$r'_n = c_n e^{j\theta'} + r'_n \tag{3.146}$$

where

$$\theta' \triangleq \theta + \alpha. \tag{3.147}$$

With respect to a channel which introduces only an unknown phase shift, considered in detail in Section 3.11.2, we now have two differences: (i) the phase shift is θ' and (ii) the variance of the noise samples is $\sigma^2/|f|^2$. The random variable θ' has uniform distribution, regardless of the distribution of α. Under the final channel dependence approximation, the noise variance does not affect the receiver strategy. Therefore, the results in Section 3.11.2 still apply and the receiver, in the case of perfect CSI, is based on a VA with branch metric

$$\lambda_k(a_k, S_k) = \left|\sum_{i=0}^{N-1} r'_{k-i}c^*_{k-i}\right| - \left|\sum_{i=1}^{N-1} r'_{k-i}c^*_{k-i}\right| - \frac{1}{2}|c_k|^2 \tag{3.148}$$

where $T_k \triangleq (a_k, S_k)$ is a transition in a suitably expanded trellis and N is the order of Markovianity of the channel.

Limited Channel State Information

We now turn our attention to the case characterized by the absence of CSI, in the sense that the receiver has statistical information about the fading gain only. Specifically, using polar coordinates to express the mean η_f of the random variable f as $\eta_f = |\eta_f|e^{j\theta_f}$, we assume that the receiver knows the value $|\eta_f|$, the fading variance σ_f^2 and the noise variance σ^2. In principle, an estimation of these parameters could be obtained from the observation of a few preambles of consecutive bursts. However, it is shown in the following that in all significant cases estimation of these parameters is not necessary.

Recalling (3.144), given a code vector $\boldsymbol{c}$ and a hypothetical phase θ the conditional pdf of the observation vector $\boldsymbol{r} \triangleq (r_0, r_1, \ldots, r_{K-1})^{\mathrm{T}}$ is obviously Gaussian:

$$p(\boldsymbol{r}|\boldsymbol{c}, \theta) = \frac{1}{\pi^K \det(\boldsymbol{C_r})} \exp\left[-(\boldsymbol{r} - \boldsymbol{\eta_r})^{\mathrm{H}} \boldsymbol{C_r}^{-1} (\boldsymbol{r} - \boldsymbol{\eta_r})\right] \tag{3.149}$$

where $[\cdot]^{\mathrm{H}}$ is the transpose conjugate operator and

$$\boldsymbol{\eta_r} \triangleq E\{\boldsymbol{r}|\boldsymbol{c},\theta\} = \eta_f e^{j\theta}\boldsymbol{c} \tag{3.150}$$

$$\boldsymbol{C_r} \triangleq E\{(\boldsymbol{r}-\boldsymbol{\eta_r})(\boldsymbol{r}-\boldsymbol{\eta_r})^{\mathrm{H}}|\boldsymbol{c},\theta\} = \sigma_f^2\boldsymbol{c}\boldsymbol{c}^{\mathrm{H}} + 2\sigma^2\mathbf{I} \tag{3.151}$$

are the conditional mean and covariance matrix, respectively. Using equation (3.54) of [122], it is possible to express the inverse covariance matrix as

$$\boldsymbol{C}_{\boldsymbol{r}}^{-1} = \left(\sigma_f^2\boldsymbol{c}\boldsymbol{c}^{\mathrm{H}} + \sigma^2\mathbf{I}\right)^{-1} = \frac{1}{2\sigma^2}\mathbf{I} - \frac{\sigma_f^2}{4\sigma^4}\frac{\boldsymbol{c}\boldsymbol{c}^{\mathrm{H}}}{1+\frac{\sigma_f^2}{2\sigma^2}|\boldsymbol{c}|^2}. \tag{3.152}$$

The following relation may also be easily verified

$$\det(\sigma_f^2\boldsymbol{c}\boldsymbol{c}^{\mathrm{H}} + 2\sigma^2\mathbf{I}) = \left(2\sigma^2\right)^{K-1}\left(\sigma_f^2|\boldsymbol{c}|^2 + 2\sigma^2\right) \tag{3.153}$$

where $\det(\cdot)$ denotes the determinant of a matrix.

Substituting (3.152) and (3.153) into (3.149), after some tedious but straightforward manipulations, one obtains:

$$\begin{aligned} p(\boldsymbol{r}|\boldsymbol{c},\theta) &= \frac{1}{\pi^K\left(2\sigma^2\right)^{K-1}\sigma_f^2}\,\frac{1}{|\boldsymbol{c}|^2+\frac{2\sigma^2}{\sigma_f^2}}\,\exp\left(-\frac{|\boldsymbol{r}|^2}{2\sigma^2}\right) \\ &\quad\cdot\exp\left(-\frac{1}{2\sigma^2}\left[\frac{2\frac{\sigma^2}{\sigma_f^2}|\eta_f|^2|\boldsymbol{c}|^2 - \left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|^2 - 4\frac{\sigma^2}{\sigma_f^2}\mathrm{Re}\left\{\eta_f^* e^{-j\theta}\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right\}}{\frac{2\sigma^2}{\sigma_f^2}+|\boldsymbol{c}|^2}\right]\right) \\ &= \frac{1}{\pi^K\left(2\sigma^2\right)^{K-1}\sigma_f^2}\,\exp\left(-\frac{|\boldsymbol{r}|^2}{2\sigma^2}\right)\,\frac{1}{|\boldsymbol{c}|^2+\frac{1}{\xi}} \\ &\quad\cdot\exp\left(-\frac{1}{2\sigma^2}\left[\frac{\frac{1}{\xi}|\eta_f|^2|\boldsymbol{c}|^2 - \left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|^2 - \frac{2|\eta_f||\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}|}{\xi}\cos\left(\arg\left\{\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right\}-\theta_f-\theta\right)}{\frac{1}{\xi}+|\boldsymbol{c}|^2}\right]\right) \end{aligned} \tag{3.154}$$

where, in the last step, $\xi \triangleq \sigma_f^2/2\sigma^2$ has been introduced. This quantity is useful because the branch metric derived below from (3.154) can be expressed as a function of $1/\xi$ in order to avoid numerical problems for high SNR, when $\xi \to +\infty$. Averaging (3.154) with respect to θ (recall that $p(\theta) = 1/2\pi$ for $\theta \in [0,\pi)$), we

obtain

$$p(\boldsymbol{r}|\boldsymbol{c}) = \frac{1}{2\pi}\int_0^{2\pi} p(\boldsymbol{r}|\boldsymbol{c},\theta)\,\mathrm{d}\theta = \frac{1}{\pi^K\left(2\sigma^2\right)^{K-1}\sigma_f^2}\exp\left(-\frac{|\boldsymbol{r}|^2}{2\sigma^2}\right)$$
$$\cdot\frac{1}{|\boldsymbol{c}|^2+\frac{1}{\xi}}\exp\left(\frac{1}{2\sigma^2}\left[\frac{\left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|^2-\frac{|\eta_f|^2|\boldsymbol{c}|^2}{\xi}}{\frac{1}{\xi}+|\boldsymbol{c}|^2}\right]\right)\mathrm{I}_0\left(\frac{1}{\sigma^2}\left[\frac{\frac{|\eta_f|\left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|}{\xi}}{\frac{1}{\xi}+|\boldsymbol{c}|^2}\right]\right). \tag{3.155}$$

As intuitively expected, due to the noncoherent nature of the strategy, the result in (3.155) is independent of θ_f. Knowledge of this phase is therefore irrelevant for the detection process. In order to perform ML detection, we should maximize the following likelihood function:

$$\begin{aligned}\Lambda_K(\boldsymbol{c}) &\triangleq \ln p(\boldsymbol{r}|\boldsymbol{c})\\ &\sim -\sigma^2\ln\left(|\boldsymbol{c}|^2+\frac{1}{\xi}\right)+\frac{1}{2}\left[\frac{\left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|^2-\frac{|\eta_f|^2|\boldsymbol{c}|^2}{\xi}}{\frac{1}{\xi}+|\boldsymbol{c}|^2}\right]\\ &\quad+\sigma^2\ln\mathrm{I}_0\left(\frac{1}{\sigma^2}\left[\frac{\frac{|\eta_f|\left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|}{\xi}}{\frac{1}{\xi}+|\boldsymbol{c}|^2}\right]\right).\end{aligned} \tag{3.156}$$

Equation (3.156) represents the exact expression of the likelihood function. Some approximations are now introduced in order to realize simple suboptimal detection schemes based on a VA. In the following, we pursue the same conceptual approach considered in [120]—this is the direct extension to the case of fading channels of the approach proposed in [47] in the case of phase noncoherent channels. More precisely, the starting point is the approximation of the likelihood function, and consequently the introduction of a suitable branch metric. Moreover, we point out that exactly the same final result would be obtained by starting from a generic metric based on a CMP, and then averaging over the fading coefficient (see Problem 3.9).

We adopt the approximation $\ln \mathrm{I}_0(x) \simeq x$, valid for large values of the argument. Although this approximation is valid for high values of $|\eta_f|$ and $1/\sigma^2$, i.e., for high SNR and a high Rice factor K_{R}, simulation results show that it is very good even in the case of a Rayleigh fading channel. With this approximation, (3.156) may be

expressed as

$$\begin{aligned}\Lambda_K(\boldsymbol{c}) &\simeq -\sigma^2 \ln\left(|\boldsymbol{c}|^2+\frac{1}{\xi}\right)+\frac{1}{2}\frac{\left|\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}\right|^2-\frac{|\eta_f|^2|\boldsymbol{c}|^2}{\xi}+2\frac{|\eta_f||\boldsymbol{c}^{\mathrm{H}}\boldsymbol{r}|}{\xi}}{\frac{1}{\xi}+|\boldsymbol{c}|^2} \\ &= -\sigma^2 \ln\left(\sum_{k=0}^{K-1}|c_k|^2+\frac{1}{\xi}\right)+\frac{1}{2}\frac{\left|\sum_{k=0}^{K-1}r_k c_k^*\right|^2-\frac{|\eta_f|^2}{\xi}\sum_{k=0}^{K-1}|c_k|^2+2\frac{|\eta_f|}{\xi}\left|\sum_{k=0}^{K-1}r_k c_k^*\right|}{\frac{1}{\xi}+\sum_{k=0}^{K-1}|c_k|^2}.\end{aligned} \tag{3.157}$$

According to a CMP, we may define a partial sequence metric at the k-th signaling interval

$$\Lambda_k(\boldsymbol{c}) \triangleq -\sigma^2 \ln\left(\sum_{i=0}^{k-1}|c_i|^2+\frac{1}{\xi}\right)+\frac{1}{2}\frac{\left|\sum_{i=0}^{k-1}r_i c_i^*\right|^2-\frac{|\eta_f|^2}{\xi}\sum_{i=0}^{k-1}|c_i|^2+2\frac{|\eta_f|}{\xi}\left|\sum_{i=0}^{k-1}r_i c_i^*\right|}{\frac{1}{\xi}+\sum_{i=0}^{k-1}|c_i|^2} \tag{3.158}$$

and an incremental metric

$$\begin{aligned}\Delta_k(\boldsymbol{c}) &\triangleq \Lambda_{k+1}(\boldsymbol{c}) - \Lambda_k(\boldsymbol{c}) \\ &= -\sigma^2 \ln \left(\frac{\sum_{i=0}^{k} |c_i|^2 + \frac{1}{\xi}}{\sum_{i=0}^{k-1} |c_i|^2 + \frac{1}{\xi}} \right) \\ &\quad + \frac{1}{2} \frac{\left| \sum_{i=0}^{k} r_i c_i^* \right|^2 - \frac{|\eta_f|^2}{\xi} \sum_{i=0}^{k} |c_i|^2 + 2\frac{|\eta_f|}{\xi} \left| \sum_{i=0}^{k} r_i c_i^* \right|}{\frac{1}{\xi} + \sum_{i=0}^{k} |c_i|^2} \\ &\quad - \frac{1}{2} \frac{\left| \sum_{i=0}^{k-1} r_i c_i^* \right|^2 - \frac{|\eta_f|^2}{\xi} \sum_{i=0}^{k-1} |c_i|^2 + 2\frac{|\eta_f|}{\xi} \left| \sum_{i=0}^{k-1} r_i c_i^* \right|}{\frac{1}{\xi} + \sum_{i=0}^{k-1} |c_i|^2}. \end{aligned} \tag{3.159}$$

The difficulty inherent in the incremental metric (3.159) is its unlimited memory. In fact, this metric depends on the entire previous code sequence. This implies that the maximization of the sequence metric may, in principle, be realized by a search of the path in a properly defined tree diagram. From an implementation viewpoint, approximate tree search algorithms must be used, unless a very short transmission length is assumed. In order to limit the memory of the incremental metric (3.159) we apply a CMP, which allows one to search a trellis diagram by means of a VA. To this end, in (3.159) we may consider $(N+1) \ll K$ most recent observations $\boldsymbol{r}_{k-N}^{k}$ and code symbols $\boldsymbol{c}_{k-N}^{k}$ only. After an initial transient period, i.e., for $k \geq N$, the

resulting approximate finite memory incremental, or branch, metric is

$$\lambda_k(a_k, S_k) \triangleq -\sigma^2 \ln \left(\frac{\sum_{i=0}^{N} |c_{k-i}|^2 + \frac{1}{\xi}}{\sum_{i=1}^{N} |c_{k-i}|^2 + \frac{1}{\xi}} \right) + \frac{1}{2} \frac{\left| \sum_{i=0}^{N} r_{k-i} c_{k-i}^* \right|^2 - \frac{|\eta_f|^2}{\xi} \sum_{i=0}^{N} |c_{k-i}|^2 + 2\frac{|\eta_f|}{\xi} \left| \sum_{i=0}^{N} r_{k-i} c_{k-i}^* \right|}{\frac{1}{\xi} + \sum_{i=0}^{N} |c_{k-i}|^2} - \frac{1}{2} \frac{\left| \sum_{i=1}^{N} r_{k-i} c_{k-i}^* \right|^2 - \frac{|\eta_f|^2}{\xi} \sum_{i=1}^{N} |c_{k-i}|^2 + 2\frac{|\eta_f|}{\xi} \left| \sum_{i=1}^{N} r_{k-i} c_{k-i}^* \right|}{\frac{1}{\xi} + \sum_{i=1}^{N} |c_{k-i}|^2} \quad (3.160)$$

where $S_k \triangleq (a_{k-1}, \ldots, a_{k-N}, \mu_{k-N})$ and the transition $T_k = (a_k, S_k)$ uniquely identify the code symbols $\boldsymbol{c}_{k-N}^k$ used in (3.160). As already mentioned, the corresponding receiver is based on a VA and requires knowledge of the parameters $|\eta_f|$, σ^2 and σ_f^2. The branch metric (3.160) is valid in the case of any linear modulation and any (Rayleigh, Rice or AWGN) channel. However, they may be significantly simplified in the following special cases.

1. **Equal-energy Signals**
   We now concentrate on the special case of equal-energy signals, such as PSK modulations in the absence of ISI. Eliminating from (3.160) the terms which do not depend on the hypothetical code vector $\boldsymbol{c}$, because the code symbols have now constant modulus, one obtains the following expression of the sequence metric:

$$\Lambda_K(\boldsymbol{c}) = \left| \sum_{i=0}^{K-1} r_i c_i^* \right|^2 + 2\frac{|\eta_f|}{\xi} \left| \sum_{i=0}^{K-1} r_i c_i^* \right| . \quad (3.161)$$

   Adding the term $\frac{|\eta_f|^2}{\xi^2}$, independent of $\boldsymbol{c}$, the following equivalent sequence

metric is obtained:

$$\begin{aligned}\Lambda_K(\boldsymbol{c}) &\sim \left|\sum_{i=0}^{K-1} r_i c_i^*\right|^2 + 2\frac{|\eta_f|}{\xi}\left|\sum_{i=0}^{K-1} r_i c_i^*\right| + \frac{|\eta_f|^2}{\xi^2} \\ &= \left(\left|\sum_{i=0}^{K-1} r_i c_i^*\right| + \frac{|\eta_f|}{\xi}\right)^2 \\ &\sim \left|\sum_{i=0}^{K-1} r_i c_i^*\right| . \end{aligned} \tag{3.162}$$

In the last step, the fact that the square of a positive quantity is monotonically increasing on the quantity has been exploited.

Introducing partial sequence metrics and then finite memory branch metrics, we finally obtain:

$$\lambda_k(a_k, S_k) = \left|\sum_{i=0}^{N} r_{k-i} c_{k-i}^*\right| - \left|\sum_{i=1}^{N} r_{k-i} c_{k-i}^*\right| \tag{3.163}$$

which evidently does not require any statistical information about the channel, because it does not depend on $|\eta_f|$, σ_f^2 and σ^2. This result coincides with that found in Section 3.11.2 for a noncoherent channel, in agreement with the intuition that the channel gain is irrelevant in the detection of equal-energy signals.

2. **Rayleigh Fading Channels**
The branch metric in the case of a Rayleigh fading channel may be obtained letting $|\eta_f| = 0$ in (3.160). Accordingly, one has

$$\lambda_k(a_k, S_k) = -\sigma^2 \ln\left(\frac{\displaystyle\sum_{i=0}^{N} |c_{k-i}|^2 + \frac{1}{\xi}}{\displaystyle\sum_{i=1}^{N} |c_{k-i}|^2 + \frac{1}{\xi}}\right) + \frac{1}{2}\frac{\left|\displaystyle\sum_{i=0}^{N} r_{k-i} c_{k-i}^*\right|^2}{\frac{1}{\xi} + \displaystyle\sum_{i=0}^{N} |c_{k-i}|^2} - \frac{1}{2}\frac{\left|\displaystyle\sum_{i=1}^{N} r_{k-i} c_{k-i}^*\right|^2}{\frac{1}{\xi} + \displaystyle\sum_{i=1}^{N} |c_{k-i}|^2} . \tag{3.164}$$

The evaluation of this branch metric obviously requires knowledge of σ^2 and ξ (or, equivalently, σ^2 and σ_f^2). Since high values of SNR are typical of communications over Rayleigh fading channels, we may use the approximations

$\sigma^2 \simeq 0$ and $1/\xi \simeq 0$, obtaining the following expression:

$$\lambda_k(a_k, S_k) = \frac{\left|\sum_{i=0}^{N} r_{k-i} c_{k-i}^*\right|^2}{\sum_{i=0}^{N} |c_{k-i}|^2} - \frac{\left|\sum_{i=1}^{N} r_{k-i} c_{k-i}^*\right|^2}{\sum_{i=1}^{N} |c_{k-i}|^2} \tag{3.165}$$

which is clearly independent of the channel parameters. Simulation results show that the performance of a receiver based on (3.165) is equivalent to that of the receiver based on (3.164), in which σ^2 and σ_f^2 are assumed perfectly known. Therefore, the approximation used in the derivation of (3.165) may be considered valid.

An interesting result obtained by means of computer simulations is that, in the case of a Rice fading channel, a receiver based on branch metric (3.165) is practically equivalent, in terms of performance, to a receiver based on branch metric (3.160) in which $|\eta_f|$, σ_f^2 and σ^2 are assumed perfectly known (see [120] for more details). We note that a receiver based on branch metric (3.165) would deliver reliable decisions independently of the actual value of the fading gain, given that this gain is slowly varying. This provides some intuitive understanding of the fact that this receiver would work regardless of the distribution of the fading gain, although it is difficult to predict that its performance is good. For these reasons, the branch metric (3.165) will be used in the numerical results for non-equal-energy signals and both Rice and Rayleigh fading channels.

3. **AWGN Channels**
For an AWGN channel, the results in [47] may be obtained letting $\sigma_f^2 = 0$, $\xi = 0$ and $|\eta_f| = 1$. Multiplying the numerator and denominator of each fraction in (3.160) by ξ, one can immediately obtain

$$\lambda_k(a_k, S_k) = \left|\sum_{i=0}^{N} r_{k-i} c_{k-i}^*\right| - \left|\sum_{i=1}^{N} r_{k-i} c_{n-i}^*\right| - \frac{1}{2}|c_k|^2 \tag{3.166}$$

which coincides with the result in [47].

Interestingly, simulation results show that the branch metric (3.165) may also be used in this case as well, with a minor energy loss (of a fraction of dB only), with respect to the use of the metric (3.166). This fact confirms the correctness of the interpretation given at the end of the previous paragraph.

4. **Summary of NSD without CSI**
   In Table 3.11.3, the expressions of the proposed branch metrics in the considered cases are summarized. In all cases, knowledge of channel statistics is not required.

Table 3.1: Summary of the derived branch metrics in the case of the absence of CSI.

	Equal-energy signals	Non-equal-energy signals
Rayleigh channels	eq. (3.163)	eq. (3.165)
Rice channels		
AWGN channels		eq. (3.166) or (3.165)

The performance of the proposed detection algorithms is assessed by means of computer simulations in terms of BER versus E_b/N_0, E_b being the average received signal energy per information bit.

The state-complexity of the proposed detection schemes may be limited by means of RSSD techniques introduced in Section 3.10. These techniques allow one to choose independently the two parameters: phase memory N and number of states ζ' of the VA. Hence, the number of states may be limited without excessively reducing the value of N. In order to compute the branch metrics (3.148), (3.163), (3.165) or (3.166) in an RS trellis, the necessary symbols not included or not completely specified in the state definition are found in the survivor history according to PSP. Receivers with $\zeta' = 1$ and $\zeta' = 4$ are considered. We note that, in the limiting case of $\zeta' = 1$, the trellis diagram degenerates and symbol-by-symbol detection with decision feedback is performed.

The transmitter and receiver filters are assumed to have equal square root raised-cosine frequency response with roll-off 0.5. Each burst is composed of a preamble and an information field of $K = 100$ symbols. CSI-based receivers use a preamble of 10 symbols. These known symbols are used to estimate the fading gain $|f|$ using a simple method based on the sample mean. A more refined estimation may be implemented but, in the case of a slow fading channel, the proposed method guarantees a performance equal to that achievable with a perfect knowledge of $|f|$. For receivers without CSI, a preamble of N symbols is used. These symbols are necessary to initialize state and branch metrics of the VA. In both cases of presence or absence of CSI, a postamble of 1 symbol for the receivers with $S = 4$ is present in order to correctly terminate the VA.

In the case of 16-QAM, quadrant differential encoding is used. The performance of the proposed receivers with $S = 4$ and $N = 2$ for Rice fading with $K_R = 10$ dB

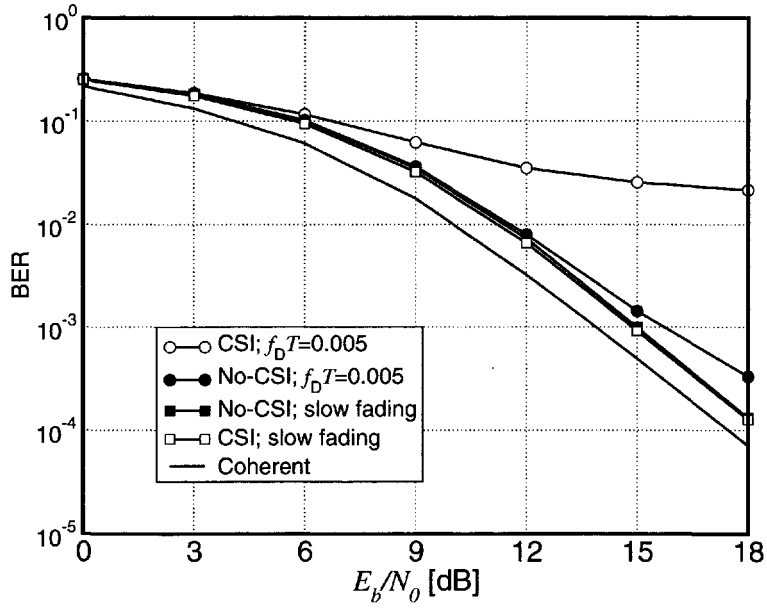


Figure 3.25: BER of the proposed receivers with and without CSI, $S = 4$ and $N = 2$, for differentially encoded 16-QAM and Rice fading with $K_{\mathrm{R}} = 10$ dB. The performance of an ideal coherent receiver is also shown for comparison. Reproduced from [120], ©2000 IEEE, by permission of the IEEE.

and Rayleigh fading is shown in Figure 3.25 and Figure 3.26, respectively. The performance of an ideal coherent receiver, i.e., a receiver which perfectly knows the fading coefficient f and the channel phase θ, is also shown for comparison. It may be observed that the proposed receivers exhibit a moderate performance loss with respect to that of the ideal coherent detector. Furthermore, the receivers with and without CSI are practically equivalent.

The robustness of these receivers in the case of time-varying fading channels is also investigated. The fading autocorrelation function is assumed equal to $J_0(2\pi f_{\mathrm{D}}\tau)$, where $J_0(\cdot)$ is the Bessel function of zero-th order, according to an isotropic scattering model [20] considered in Chapter 1. The fading speed depends on the normalized Doppler rate $f_{\mathrm{D}}T$, which is set to the values 0.005 and 0.002 for Rice and Rayleigh fading, respectively. From Figure 3.25 and Figure 3.26, we may conclude that, in the presence of time-varying fading, the receivers which operate without CSI are clearly more robust than those with CSI, whose performance rapidly degrades. The high error floor exhibited by the CSI-based receiver is mainly due to the significant time variation of the fading amplitude within the data burst and to the fact that the consid-

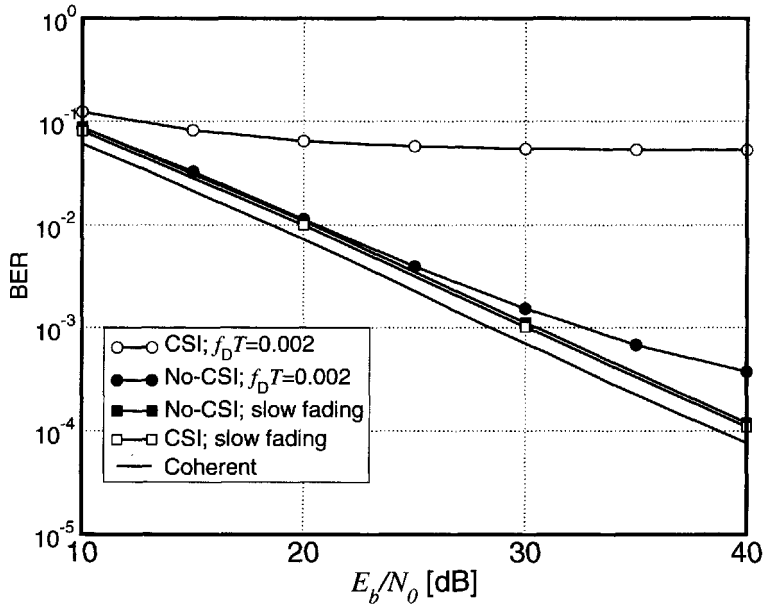


Figure 3.26: BER of the proposed receivers with and without CSI, $\zeta' = 4$ and $N = 2$, for differentially encoded 16-QAM and Rayleigh fading. The performance of an ideal coherent receiver is also shown for comparison. Reproduced from [120], ©2000 IEEE, by permission of the IEEE.

ered receivers do not employ a tracking subsystem in order to adaptively estimate the channel gain value.

Figure 3.27 compares the performance of the proposed noncoherent receiver with $N = 3$ and $\zeta' = 1$ for differentially encoded QPSK and a slow Rayleigh fading channel, with that of a coherent receiver based on a second order maximum-likelihood decision-directed PLL. In this case, the coherent receiver does not need any knowledge about the fading coefficient and is based on a symbol-by-symbol receiver followed by a differential decoder. For the considered burst-mode transmission scheme, the PLL is ideally reinitialized at the beginning of each data burst with the correct phase value; hence, it starts in the lock-in condition. In the figure, the value of the PLL noise equivalent bandwidth has been optimized by simulation for each value of SNR. As we may note, a phase noise standard deviation up to 5 degrees does not degrade significantly the performance of the proposed noncoherent receiver. This may also be noted in Figure 3.28 which shows, for both receivers, the SNR for a BER of 10^{-3}, as a function of the phase noise standard deviation. Unlike the proposed receivers, the PLL-based ones are affected by unavoidable losses of the lock-in con-

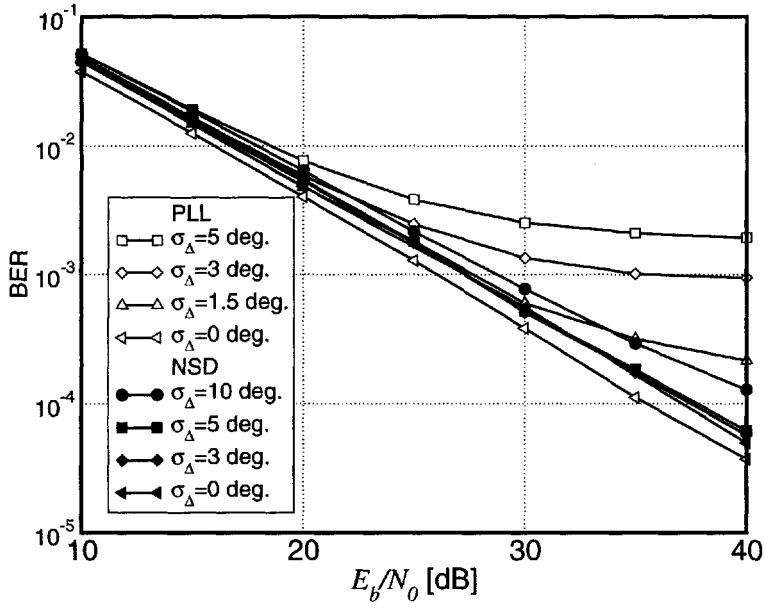


Figure 3.27: BER of the proposed receiver without CSI, $\zeta' = 4$ and $N = 3$, for differentially encoded QPSK, slow Rayleigh fading and various values of phase noise standard deviation (black marks). The performance of a coherent receiver based on a decision-directed PLL (white marks) is also shown for comparison. Reproduced from [120], ©2000 IEEE, by permission of the IEEE.

dition within the data burst for high phase noise.

3.11.4 Linear Predictive Sequence Detection for Phase-Uncertain Channels

Linear prediction has been widely used in estimation problems in various technical areas [123]. In particular, linear predictive detection for fading channels was proposed in [101] for CPM, extended in [124] to PSK and generalized by means of the innovations process in [125]. The starting point of these works is the observation that, conditional on the knowledge of the transmitted signal, linear predictive receivers with increasing prediction order for flat fading channels are asymptotically optimal. Therefore, it is shown that a finite prediction order causes minimal performance degradation.[23]

[23]Note that considering a fi nite prediction order is equivalent to applying the CMP.

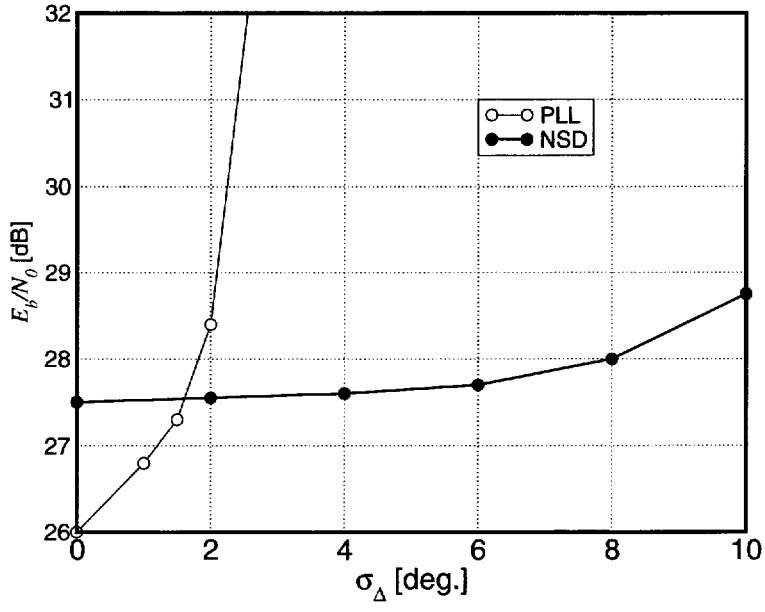


Figure 3.28: E_b/N_0 as a function of the phase noise standard deviation for BER equal to 10^{-3} of the proposed receiver without CSI, $\zeta' = 1$ and $N = 3$, for differentially encoded QPSK, slow Rayleigh fading (black marks), and comparison with a coherent receiver based on a decision-directed PLL (white marks). Reproduced from [120], ©2000 IEEE, by permission of the IEEE.

In the following, we consider MAP sequence detection based on linear prediction. Since the concept of detection based on linear prediction can be generalized, linear predictive receivers can be designed for transmission over any channel where the transmitted signal is affected by a multiplicative distortion. In particular, we focus on the case of a channel causing an unknown phase rotation in the transmitted signal. We will show how the proposed linear predictive sequence detection schemes compare with other solutions considered in Section 3.11.2.

We consider the base-band complex equivalent system depicted in Figure 3.29. We assume that a sequence of K independent and uniformly distributed M-ary symbols $\{a_k\}_{k=0}^{K-1}$, denoted by the vector $\boldsymbol{a}$ in the figure, feeds an encoder/modulator, which can be modeled as an FSM with state s_k and the output of which is the sequence $\boldsymbol{c}$ of coded symbols. We also define the relevant state transition as $t_k \triangleq (a_k, s_k)$. The linearly modulated continuous-time signal $s(t, \boldsymbol{a})$ is obtained by letting the code symbol c_k be carried by a suitable shaping pulse $p(t)$. Although suboptimal in the presence of a time-varying channel, a matched-filter front-end with sampling

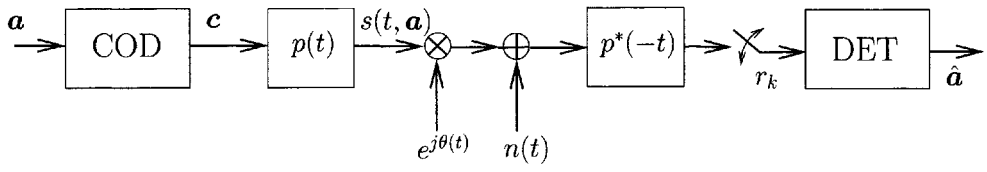


Figure 3.29: System model for linear prediction-based receivers.

rate of one sample per symbol can be practically used, provided that the phase process is not affected by very strong variations [114]. The resulting observation model is

$$r_k = c_k e^{j\theta_k} + n_k \tag{3.167}$$

where $\{n_k\}$ is an iid complex AWGN sequence with variance per component $\sigma^2 = N_0$. The channel phase process θ_k is assumed stationary and zero-mean, and the autocorrelation sequence of the phasor process $e^{j\theta_k}$ is denoted by $R_\theta(n) \triangleq \mathrm{E}\{e^{j\theta_{k+n}} e^{-j\theta_k}\}$.

Should the phase process $\{\theta_k\}$ be known exactly, a coherent sequence detector based on the VA (run over a trellis diagram relative to the encoder/modulator FSM) could be derived, with branch metric

$$\lambda_k^{\text{coher}}(t_k) \triangleq -\frac{\left|r_k - e^{j\theta_k} c_k\right|^2}{2\sigma^2} + \ln P\{a_k\} = -\frac{|c_k|^2 \left|r'_k - e^{j\theta_k}\right|^2}{2\sigma^2} + \ln P\{a_k\} \tag{3.168}$$

where $r'_k \triangleq r_k / c_k$ is a normalized observation—note that by using the metric in (3.168) the VA has to perform recursive maximization. By assuming that an information symbol is *a priori* equally likely to assume any of the M possible values, i.e., assuming that $P\{a_k\} = 1/M$, a metric equivalent to (3.168) is the following:

$$\lambda_k^{\text{coher}}(t_k) \triangleq -\left|r_k - e^{j\theta_k} c_k\right|^2 = -|c_k|^2 \left|r'_k - e^{j\theta_k}\right|^2. \tag{3.169}$$

If only a statistical characterization of the phase process is available, a practical sequence detector can be obtained by using (3.168) with the exact value θ_k replaced by a suitable estimate $\widehat{\theta}_k$. In the general case of a time-varying phase process, the data-aided MMSE phase estimate based on N previous observations is given by the following conditional mean:

$$\widehat{\theta}_k = \mathrm{E}\left\{\theta_k | \boldsymbol{c}_{k-N}^{k-1}, \boldsymbol{r}_{k-N}^{k-1}\right\} \tag{3.170}$$

where an indexed vector notation is used to denote code symbols and observations from time epoch $k - N$ to $k - 1$. The expectation in (3.170) leads to a nonlinear

estimate, which is usually rather difficult to compute. Instead of directly estimating the phase, we use the following indirect estimation strategy. We note that for large SNR it holds that $r'_k \simeq e^{j\theta_k}$, so that $\theta_k \simeq \arg\{r'_k\}$. In order to exploit the phase correlation characteristics in the estimation process, we consider a data-aided linear prediction $\widehat{r'_k}$ of r'_k based on the previous N normalized observations $\boldsymbol{r'}_{k-N}^{k-1}$, defined as

$$\widehat{r'_k} = \sum_{i=1}^{N} p_i r'_{k-i} \tag{3.171}$$

where $\{p_i\}_{i=1}^N$ are the *prediction coefficients* and N is the *prediction order*. The prediction coefficients in (3.171) can be computed by solving the MMSE problem [101, 124–126]

$$\min_{\{p_i\}_{i=1}^N} \mathrm{E}\left\{ \left| r'_k - \sum_{i=1}^{N} p_i r'_{k-i} \right|^2 \middle| \boldsymbol{c}_{k-N}^{k-1} \right\} \tag{3.172}$$

which leads to a Wiener Hopf linear system $\boldsymbol{R}\boldsymbol{p} = \boldsymbol{b}$, where $\boldsymbol{R}$ is a square $N \times N$ matrix whose elements have the following expression

$$[\boldsymbol{R}]_{l,m} = \begin{cases} R_\theta(|l-m|) & \text{if} \quad l \neq m \\ R_\theta(0) + \frac{2\sigma^2}{|c_{k-l}|^2} & \text{if} \quad l = m \end{cases} \tag{3.173}$$

$\boldsymbol{p} \triangleq [p_1 \cdots p_N]^{\mathrm{T}}$ is the unknown vector and $\boldsymbol{b} = [R_\theta(1), R_\theta(2), \cdots, R_\theta(N)]^{\mathrm{T}}$. Considering the set of prediction coefficients solving the problem given by (3.172), it is easy to express the corresponding MMSE, indicated as ϵ_k^2, as follows:

$$\epsilon_k^2 = R_\theta(0) + \frac{2\sigma^2}{|c_k|^2} - \sum_{i=1}^{N} p_i R_\theta(i). \tag{3.174}$$

The solution of this Wiener Hopf system entitles one to consider the following indirect data-aided phase estimate:

$$\widehat{\theta}_k = \arg\left\{\widehat{r'_k}\right\} = \arg\left\{\sum_{i=1}^{N} p_i r'_{k-i}\right\} = \arg\left\{\sum_{i=1}^{N} p_i \frac{r_{k-i}}{c_{k-i}}\right\}. \tag{3.175}$$

Upon the definition of an extended state $S_k \triangleq (a_{k-1}, \ldots, a_{k-N}, s_{k-N})$, one can immediately conclude that the optimal prediction coefficients depend in general on S_k,

and the MMSE depends on the corresponding transition $T_k \triangleq (a_k, S_k)$. Observe that the prediction order N corresponds to the order of Markovianity (and finite memory parameter) introduced in the general framework in Chapter 2. Defining an extended trellis diagram (with respect to the trellis diagram relative to the encoder/modulator FSM) in terms of the state S_k, a VA with the following branch metric can be adopted:

$$\begin{aligned}\lambda_k(T_k) &= -\frac{|c_k|^2 \left| r'_k - e^{j\widehat{\theta}_k} \right|^2}{\epsilon_k^2} + \ln P\{a_k\} \\ &= -\frac{|c_k|^2 \left| r'_k - \frac{\sum_{i=1}^{N} p_i r'_{k-i}}{\left| \sum_{i=1}^{N} p_i r'_{k-i} \right|} \right|^2}{\epsilon_k^2} + \ln P\{a_k\}. \end{aligned} \tag{3.176}$$

In this case, the VA based on metric (3.176) performs combined detection and decoding with a per-survivor linear predictive estimate, based on N consecutive observations, of the channel phase.

In the case of equal-energy signaling, since the system matrix $\boldsymbol{R}$ in (3.173) no longer depends on $\{c_{k-l}\}_{l=1}^{N}$, the prediction coefficients and the MMSE do not depend on T_k, but only on the SNR. In this case, they can be precomputed at once off-line and then used in the VA. Moreover, if the *a priori* symbol probabilities are the same, the metric in (3.176) can be simplified as in the coherent case, obtaining:

$$\lambda_k(T_k) = -|c_k|^2 \left| r'_k - e^{j\widehat{\theta}_k} \right|^2. \tag{3.177}$$

With respect to classical linear predictive receivers for fading channels [101, 124–126], the proposed solution features a few differences.

- First, the denominator in (3.176) normalizes the phasor estimate to unit modulus. This normalization is essential for non-equal-energy signaling such as QAM.
- Second, the correlation matrix $\boldsymbol{R}$ models the statistics of the phase process, whereas in classical receivers it models the amplitude process as well.

Since in this case the channel does not introduce any amplitude variations, it is rather intuitive to expect that the linear predictive detection strategy for transmission over phase-uncertain channels will offer good performance. The complexity of the proposed receivers can be limited by applying RSSD techniques introduced in Section 3.10.

In the case of non-equal-energy signaling, a simple modification which leads to the calculation of the prediction coefficients associated with any transition, regardless of the sequence of coded symbols, consists of considering a modified mean square error (MSE), where the expectation is carried out with respect to both the channel phase and the coded symbols. In other words, the prediction coefficients are obtained by solving the following minimization problem:

$$\min_{\{p_i\}_{i=1}^N} \mathrm{E}\left\{\left|r'_k - \sum_{i=1}^{N} p_i r'_{k-i}\right|^2\right\}$$
$$= \min_{\{p_i\}_{i=1}^N} \mathrm{E}_{\boldsymbol{c}_{k-N}^{k-1}}\left\{\mathrm{E}\left\{\left|r'_k - \sum_{i=1}^{N} p_i r'_{k-i}\right|^2 \,\middle|\, \boldsymbol{c}_{k-N}^{k-1}\right\}\right\}. \quad (3.178)$$

The minimization problem in (3.178) leads to a Wiener Hopf system $\boldsymbol{R}_{\mathrm{simp}}\boldsymbol{p}' = \boldsymbol{b}$, where the generic element $[\boldsymbol{R}_{\mathrm{simp}}]_{i,j}$ of the system matrix $\boldsymbol{R}_{\mathrm{simp}}$ has the following expression:

$$[\boldsymbol{R}_{\mathrm{simp}}]_{i,j} = \begin{cases} R_\theta(|i-j|) & \text{if} \quad i \neq j \\ R_\theta(0) + 2\sigma^2 \mathrm{E}\left\{\frac{1}{|c_{k-i}|^2}\right\} & \text{if} \quad i = j. \end{cases} \quad (3.179)$$

Assuming that a code symbol c_k can coincide with any point of the modulation constellation with equal probability, it follows that the quantity $\mathrm{E}\left\{\frac{1}{|c_{k-i}|^2}\right\}$ depends only on the specific linear modulation format. For example, in the case of 16-QAM, it is easy to show that

$$\mathrm{E}\left\{\frac{1}{|c_{k-i}|^2}\right\} \simeq \frac{1.89}{\mathrm{E}_S} \quad (3.180)$$

where E_S is the average symbol energy.

As an example of application, we consider approximate MAP sequence detection, based on linear prediction, of an eight-state rate-3/4 TCM with 16-QAM output modulation and 90 degree rotational invariance [57]. The performance of the system, shown in Figure 3.30, was investigated for a phase jitter standard deviation σ_Δ equal to 0 degrees and to 5 degrees. For comparison the performance of the simplified algorithm proposed above was also evaluated. The number of states was kept fixed to 64 ($Q = 2$) in all linear predictive receivers. While for $\sigma_\Delta = 0$ degrees the receiver with prediction order $N = 7$ asymptotically approaches the performance of the coherent system, for $\sigma_\Delta = 5$ degrees the best performance is obtained with a small prediction order, namely $N = 2$. A possible explanation is that a large prediction order combined with heavy state reduction is likely to degrade the performance for significant

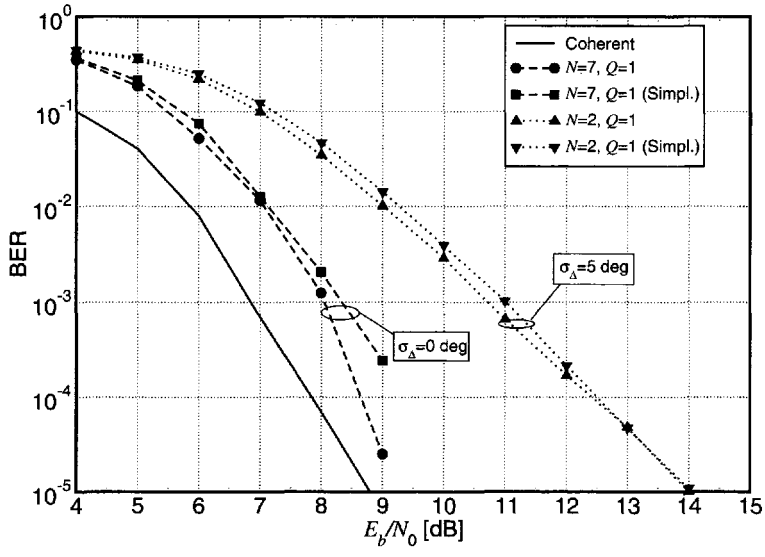


Figure 3.30: BER of a TCM scheme with 16-QAM. Linear predictive receivers with various complexity levels are considered. For comparison, the performance of the equivalent coherent receiver is also shown.

phase jitters. It is also worth observing that the simplified predictive receiver shows a small performance loss with respect to the original one.

Consider a phase process $\{\psi_k\}$ modeled as a discrete-time Wiener process [19] described by the following recursion:

$$\psi_k = \psi_{k-1} + \Delta_k \tag{3.181}$$

where $\{\Delta_k\}$ are iid Gaussian increments with zero mean and variance σ_Δ^2, descriptive of the phase noise intensity. In this case, the autocorrelation of the corresponding phasor process $\{e^{j\psi_k}\}$ becomes

$$R_\psi(n) = \mathrm{E}\left\{e^{j(\psi_k - \psi_{k-n})}\right\} = \mathrm{E}\left\{e^{j(\sum_{i=0}^{n-1}\Delta_{k-i})}\right\} = \prod_{i=0}^{n-1}\mathrm{E}\left\{e^{j\Delta_{k-i}}\right\}. \tag{3.182}$$

It is possible to interpret $\mathrm{E}\{e^{j\Delta_{k-i}}\}$ as the characteristic function, evaluated in one, of the Gaussian random variables Δ_{k-i}. Since $\{\Delta_{k-i}\}$ are equally distributed, it follows that $\mathrm{E}\{e^{j\Delta_{k-i}}\} = \exp(-\sigma_\Delta^2/2)$, and consequently that

$$R_\psi(n) = \exp\left(-|n|\sigma_\Delta^2/2\right). \tag{3.183}$$

The presence of a time-invariant random frequency offset can also be incorporated by indicating the channel phase as $\theta_k = \psi_k + 2\pi\nu kT$, where ν is a random variable uniformly distributed in $(-\alpha/T, \alpha/T)$, α is the normalized frequency offset intensity, and T denotes the signaling interval. For this phase model, assuming that the phase jitter and the frequency offset are independent, it follows that

$$\begin{aligned} R_\theta(n) &= \mathrm{E}\left\{e^{j(\theta_k-\theta_{k-n})}\right\} = \mathrm{E}\left\{e^{j(\psi_k-\psi_{k-n})}\right\}\mathrm{E}\left\{e^{j2\pi\nu nT}\right\} \\ &= \exp\left(-|n|\sigma_\Delta^2/2\right)\,\mathrm{sinc}(2\alpha n) \end{aligned} \tag{3.184}$$

where $\mathrm{sinc}(x) \triangleq \sin(\pi x)/\pi x$. In the absence of phase noise ($\sigma_\Delta = 0$) and frequency offset ($\alpha = 0$), the metric in (3.176) reduces to one of the NSD solutions proposed in [47] for equal-energy signaling. Other phase models may be considered, possibly incorporating a time varying frequency offset, such as that caused by a Doppler rate [127].

Figure 3.31 shows the prediction coefficients as a function of the phase noise standard deviation σ_Δ for an equal-energy modulation, either PSK or CPM, a prediction order $N = 4$ and $E_b/N_0 = 4$ dB, where E_b denotes the received energy per information bit. Three values of frequency offset intensity α are considered, namely 0, 0.01 and 0.02. In the absence of phase instabilities ($\sigma_\Delta = 0$ and $\alpha = 0$) all the prediction coefficients are equal, as expected due to the equivalence with NSD. For increasing phase noise or frequency offset, the prediction coefficients take on different values—the stronger the phase variations, the larger the difference.

DQPSK is considered in Figure 3.32. The performance of the communication system is assessed by computer simulations in terms of BER as a function of the phase noise standard deviation σ_Δ for various values of the frequency offset intensity α and $E_b/N_0 = 10$ dB. We consider symbol-by-symbol detection with decision feedback (i.e., $Q = 0$). The performance of the proposed linear predictive receiver for $N = 5$ is compared with that of the NSD algorithm proposed in Section 3.11.2. For $\alpha = 0$, the curve corresponding to the proposed receiver is the "envelope" of the performance curves of the NSD algorithm for $N = 1, 2, \ldots, 5$. For any given σ_Δ and α, there exists an optimum value of N within the considered range such that the BER obtained with the NSD algorithm is minimized—this optimal N reduces for increasing values of σ_Δ and α. The proposed linear predictive receiver with $N = 5$ automatically minimizes the BER, provided the prediction coefficients are adaptively updated. In the presence of frequency offset, the advantage of the proposed receiver over NSD is even more pronounced, as it appears from the curves relative to $\alpha = 0.02$ and $\alpha = 0.05$. In the latter case, the performance of NSD schemes with $N > 1$ is appreciably worse.

The case of a random constant channel phase can be derived from the Wiener

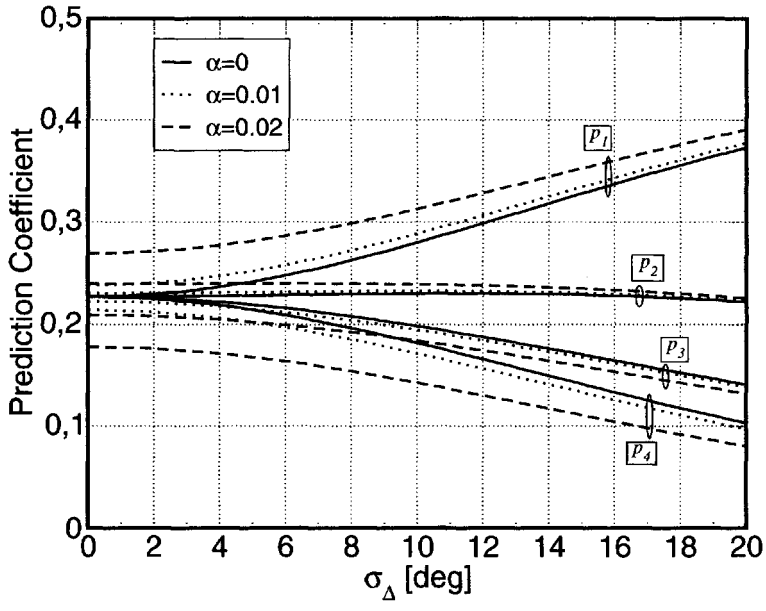


Figure 3.31: Prediction coefficients as a function of the phase noise standard deviation σ_Δ, for an equal-energy modulation, prediction order $N = 4$ and $E_b/N_0 = 4$ dB. Various values of the frequency offset intensity α are considered. Reproduced from [128], ©2003 IEEE, by permission of the IEEE.

phase process model previously considered by imposing $\sigma_\Delta = 0$. Assuming equal-energy modulation, the Wiener Hopf system can be written as $\boldsymbol{R}^{\mathrm{TI}}\boldsymbol{p}^{\mathrm{TI}} = \boldsymbol{b}^{\mathrm{TI}}$ where the system matrix $\boldsymbol{R}^{\mathrm{TI}}$ is

$$\boldsymbol{R}^{\mathrm{TI}} = 2\sigma^2\mathbf{I} + \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{3.185}$$

$\mathbf{I}$ being the identity matrix and $\boldsymbol{b}^{\mathrm{TI}} = [1 \cdots 1]^{\mathrm{T}}$. After a few simple manipulations, it follows that the solution of the Yule Walker system consists of a sequence of identical prediction coefficients, i.e., $p_i^{\mathrm{TI}} = p$, $i \in \{1, \ldots, N\}$, where

$$p \triangleq \frac{1}{N + 2\sigma^2}. \tag{3.186}$$

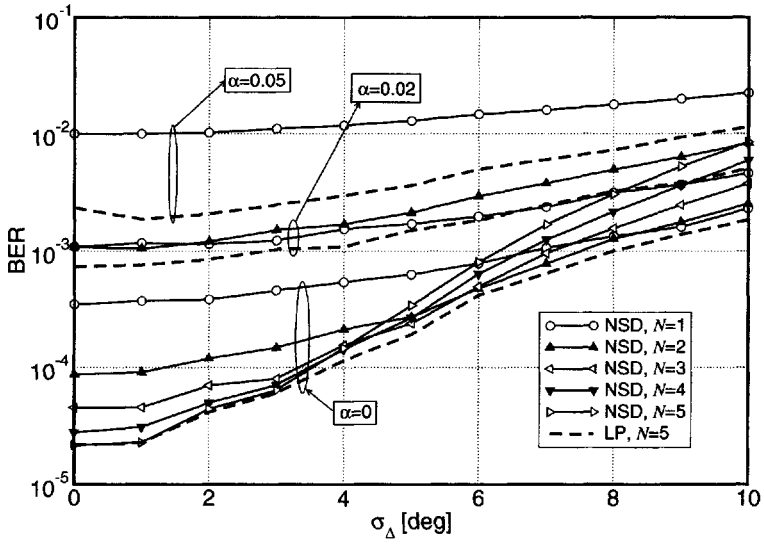


Figure 3.32: BER as a function of the phase noise standard deviation σ_Δ for DQPSK, symbol by symbol decision, and various values of the frequency offset intensity α. Reproduced from [128], ©2003 IEEE, by permission of the IEEE.

Hence, the estimate $\widehat{e^{j\theta_k}}$ becomes

$$\widehat{e^{j\theta_k}} = \frac{\sum_{i=1}^{N} r'_{k-i}}{\left|\sum_{i=1}^{N} r'_{k-i}\right|} = \frac{\sum_{i=1}^{N} \frac{r_{k-i}}{c_{k-i}}}{\left|\sum_{i=1}^{N} \frac{r_{k-i}}{c_{k-i}}\right|} \tag{3.187}$$

and the corresponding branch metric can be written, in the case of equally probable information symbols, as follows:

$$\lambda_k^{\mathrm{pLP-TI}}(T_k) = |c_k|^2 \left| r'_k - \frac{\sum_{i=1}^{N} r'_{k-i}}{\left|\sum_{i=1}^{N} r'_{k-i}\right|} \right|^2 = |c_k|^2 \left| r'_k - \frac{\sum_{i=1}^{N} \frac{r_{k-i}}{c_{k-i}}}{\left|\sum_{i=1}^{N} \frac{r_{k-i}}{c_{k-i}}\right|} \right|^2 . \tag{3.188}$$

In [129], it is shown that the basic metric used in [47] can be approximately written, in the case of equally probable information symbols, as

$$\lambda_k^{\mathrm{NSD}}(T_k) \simeq \Re\{r_k c_k^* e^{-j\widehat{\theta}_k^{(N)}}\} - \frac{|c_k|^2}{2} \tag{3.189}$$

where

$$\widehat{\theta}_k^{(N)} = \arg\left(\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right). \tag{3.190}$$

After simple manipulations, it can be immediately concluded that the branch metric (3.189) coincides with (3.188) in the case of equal-energy modulation. More generally, the metric (3.188) coincides with that proposed in [130].

Considering a generic modulation format (not necessarily with constant amplitude), for large SNR, i.e., $\sigma^2/|c_k|^2 \to 0$, the system matrix $\boldsymbol{R}^{\mathrm{TI}}$ in (3.185) can be approximated as follows:

$$\boldsymbol{R}^{\mathrm{TI}} = \begin{bmatrix} \frac{2\sigma^2}{|c_{k-1}|^2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{2\sigma^2}{|c_{k-N}|^2} \end{bmatrix} + \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \simeq \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \tag{3.191}$$

which implies $p_i \simeq 1/N$, $i \in \{1, \ldots, N\}$. The same phasor estimate (3.187) (exact in the case of equal-energy modulation) can therefore be approximately used in this case (i.e., for any linear modulation format) as well.

3.11.5 Linear Predictive Sequence Detection for Frequency Flat Fading Channels

In the case of linear coded modulation, we assume that the discrete observable can be written as follows:

$$r_k = f_k c_k + n_k \tag{3.192}$$

where: $\{f_k\}$ is a circular complex Gaussian random sequence; $\{c_k\}$ is a code sequence; and $\{n_k\}$ is an iid Gaussian noise sequence with variance per component equal to σ^2. We assume that the encoder/modulator can be modeled as an FSM with state μ_k, characterized by "output" and "next-state" functions as given in (3.3).

The statistics of the observable, conditioned on the data, are Gaussian. More precisely, the conditional pdf of the observation at time epoch k can be written as follows:[24]

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = \frac{1}{\pi\overline{\sigma}^2(\boldsymbol{a}_0^k)} \exp\left(-\frac{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2}{\overline{\sigma}^2(\boldsymbol{a}_0^k)}\right) \tag{3.193}$$

[24]We are implicitly assuming that the starting state μ_0 of the encoder/modulator is known.

where the conditional mean and conditional variance are

$$\overline{r}_k(\boldsymbol{a}_0^k) \triangleq \mathrm{E}\{r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\} \tag{3.194}$$

$$\overline{\sigma}^2(\boldsymbol{a}_0^k) \triangleq \mathrm{E}\{|r_k - \overline{r}_k(\boldsymbol{a}_0^k)|^2|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k\}. \tag{3.195}$$

In particular, the conditional mean and variance depend on the entire previous information sequence, so that one can conclude that the memory of the system is unlimited. By considering the CMP, one can conclude that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) \simeq p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-N}^k, \mu_{k-N}) \tag{3.196}$$

where the second equality follows immediately upon the considered observation model in (3.192). The intuitive motivation is that "old" observations do not contribute much information regarding the current observation. If this condition were strictly met, the random sequence r_k would be Markov of order N, conditional[25] upon $\boldsymbol{a}_0^k$. This Markov assumption is never verified in an exact sense for a realistic fading channel. Even assuming a Markov fading model, thermal noise destroys the Markovianity of the observation. The quality of this approximation depends on the autocovariance sequence of the fading process $\{f_k\}$ and the value of N, which is an important design parameter.

Upon the application of the CMP, we can concentrate on

$$p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) = \frac{1}{\pi\widehat{\sigma}^2(\boldsymbol{a}_0^k)} \exp\left(-\frac{|r_k - \widehat{r}_k(\boldsymbol{a}_0^k)|^2}{\overline{\sigma}^2(\boldsymbol{a}_0^k)}\right). \tag{3.197}$$

The conditional mean and variance in (3.197), i.e.,

$$\widehat{r}_k(\boldsymbol{a}_0^k) = \mathrm{E}\{r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k\} \tag{3.198}$$

$$\widehat{\sigma}^2(\boldsymbol{a}_0^k) = \mathrm{E}\{|r_k - \widehat{r}_k(\boldsymbol{a}_0^k)|^2|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k\} \tag{3.199}$$

are, respectively, (i) the N-th order mean square prediction of current observation r_k, given the previous N observations and the information sequence, and (ii) the relevant prediction error. Note the difference with respect to the previously introduced notation $\overline{r}_k(\boldsymbol{a}_0^k)$ and $\overline{\sigma}^2(\boldsymbol{a}_0^k)$, which denote similar quantities given the entire previous observation history $\boldsymbol{r}_0^{k-1}$ (k-th order prediction at time k).

For Gaussian random sequences, the conditional mean (i.e., the mean square estimation) is linear in the observation:

$$\widehat{r}_k(\boldsymbol{a}_0^k) = \mathrm{E}\{r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k\} = \sum_{i=1}^{N} p_{i,k}(\boldsymbol{a}_0^k) r_{k-i}. \tag{3.200}$$

[25]Note that the CMP can be equivalently reinterpreted as a Markovianity condition of order N.

At time epoch k, the linear prediction coefficients $p_{i,k}$ and the mean square prediction error $\widehat{\sigma}_k^2$ are the solution of the following (linear) matrix equation (Wiener Hopf system):

$$\boldsymbol{R}_k(\boldsymbol{a}_0^k)\boldsymbol{p} = \boldsymbol{q} \tag{3.201}$$

where

$$\begin{aligned} \boldsymbol{R}_k(\boldsymbol{a}_0^k) &= \mathrm{E}\{\boldsymbol{r}_{k-N}(\boldsymbol{r}_{k-N})^{\mathrm{H}}|\boldsymbol{a}_0^k\} && (3.202)\\ \boldsymbol{p} &= (1, -p_1, -p_2, \ldots, -p_N)^{\mathrm{T}} && (3.203)\\ \boldsymbol{q} &= (\widehat{\sigma}^2, \underbrace{0, \ldots, 0}_{N \text{ zeros}})^{\mathrm{T}}. && (3.204) \end{aligned}$$

The observation correlation matrix $\boldsymbol{R}_k(\boldsymbol{a}_0^k)$ incorporates the dependence on the data sequence $\boldsymbol{a}_0^k$ and vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ include the unknowns.

Given the flat fading model, the observation vector can be expressed as

$$\boldsymbol{r}_{k-N}^k = \boldsymbol{C}_k^{(N)}\boldsymbol{f}_{k-N}^k + \boldsymbol{n}_{k-N}^k \tag{3.205}$$

where $\boldsymbol{C}_k^{(N)} \triangleq \mathrm{diag}(\boldsymbol{c}_{k-N}^k)$ and $\boldsymbol{f}_{k-N}^k \triangleq (f_k, f_{k-1}, \ldots, f_{k-N})$ and $\boldsymbol{n}_{k-N}^k \triangleq (n_k, n_{k-1}, \ldots, n_{k-N})$. Hence, the observation correlation matrix is

$$\begin{aligned} \boldsymbol{R}_k(\boldsymbol{a}_0^k) &= \mathrm{E}\{\boldsymbol{r}_{k-N}(\boldsymbol{r}_{k-N})^{\mathrm{H}}|\boldsymbol{a}_0^k|\boldsymbol{a}_0^k\}\\ &= \mathrm{E}\{[\boldsymbol{C}_k^{(N)}\boldsymbol{f}_{k-N}^k + \boldsymbol{n}_{k-N}^k][(\boldsymbol{f}_{k-N}^k)^{\mathrm{H}}(\boldsymbol{C}_k^{(N)})^{\mathrm{H}} + (\boldsymbol{n}_{k-N}^k)^{\mathrm{H}}]|\boldsymbol{a}_0^k\}\\ &= \boldsymbol{C}_k^{(N)}\boldsymbol{F}(\boldsymbol{C}_k^{(N)})^{\mathrm{H}} + 2\sigma^2\boldsymbol{I} \end{aligned} \tag{3.206}$$

where $\boldsymbol{F} \triangleq \mathrm{E}\{\boldsymbol{f}_{k-N}^k(\boldsymbol{f}_{k-N}^k)^{\mathrm{H}}\}$ is the fading correlation matrix, which does not depend on k under the assumption of *stationary fading*. This expression shows that the dependence of $\boldsymbol{R}_k(\boldsymbol{a}_0^k)$ on the information sequence can be compacted[26] into the dependence on the code sequence $\boldsymbol{c}_{k-N}^k$. Hence, considering the "usual" expanded state $S_k = (a_{k-1}, \ldots, a_{k-N}, \mu_{k-N})$, we can further write:

$$\boldsymbol{R}_k(\boldsymbol{a}_0^k) = \boldsymbol{R}_k(a_k, S_k). \tag{3.207}$$

It can be immediately concluded that a similar dependence on the couple (a_k, S_k) characterizes the prediction coefficients, the conditional mean and variance, and the

[26]It is possible to arrive at the same conclusion immediately after the application of the CMP, since a limited dependence on the observations implies a limited dependence on the information sequence as well, as shown in Chapter 2.

entire conditional statistics of the observation. More precisely, carefully analyzing the previously introduced Wiener Hopf system, it is possible to write:

$$p_i(\boldsymbol{a}_0^k) = p_i(S_k) \tag{3.208}$$

$$\widehat{r}_k(\boldsymbol{a}_0^k) = \widehat{r}_k(S_k) = \sum_{i=1}^{N} p_i(S_k) r_{k-i} \tag{3.209}$$

$$\widehat{\sigma}_k^2(\boldsymbol{a}_0^k) = \widehat{\sigma}^2(a_k, S_k) \tag{3.210}$$

$$p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, a_k, S_k) \tag{3.211}$$

where the unnecessary time indexes can be dropped assuming a stationary fading regime. The resulting branch metric to be used in a VA is

$$\begin{aligned}
\lambda_k(a_k, S_k) &= \ln p(r_k|\boldsymbol{r}_{k-N}^{k-1}, S_k, a_k) + \ln P\{a_k\} \\
&\sim -\frac{|r_k - \widehat{r}_k(S_k)|^2}{\widehat{\sigma}^2(a_k, S_k)} - \ln \widehat{\sigma}^2(a_k, S_k) + \ln P\{a_k\} \\
&= -\frac{\left| r_k - \sum_{i=1}^{N} p_i(S_k) r_{k-i} \right|^2}{\widehat{\sigma}^2(a_k, S_k)} - \ln \widehat{\sigma}^2(a_k, S_k) \ln P\{a_k\}.
\end{aligned} \tag{3.212}$$

The branch metric (3.212) makes use of the linear prediction $\widehat{r}_k(S_k)$ of the current observation r_k, based on the previous observations and *path-dependent* prediction coefficients.

The derived linear predictive detection algorithm can be given the following interpretation. Based on the conditional Gaussian nature of the observation and the CMP, we can concentrate on the Gaussian pdf $p(r_k|\boldsymbol{r}_{k-N}^{k-1}, a_k, S_k)$. The conditional mean $\widehat{r}_k(a_k, S_k)$ and variance $\widehat{\sigma}^2(a_k, S_k)$ can be viewed as *system parameters* to be estimated, and the following steps should be considered.

1. Adopt a linear feed-forward data-aided parameter estimator of order N (see Section 3.11.1).

2. Use a set of estimators by associating one estimator to each trellis path.

3. Compute the estimation coefficients in order to minimize the mean square estimation error with respect to the random variable r_k, conditional on the path data sequence.

The resulting estimator is exactly the described *path-dependent* linear predictor. Linear prediction of r_k based on the previous observations is a form of PSP-based *feed-forward* parameter estimation. We obtained it *naturally* in the derivation of the detection algorithm.

It is possible to obtain an alternative formulation of the branch metric. In fact, the observation prediction can be expressed as follows:

$$\begin{aligned}\widehat{r}_k(a_k, S_k) &= \mathrm{E}\{r_k|\boldsymbol{r}_{k-N}, a_k, S_k\} = \mathrm{E}\{c_k f_k + n_k|\boldsymbol{r}_{k-N}, a_k, S_k\} \\ &= c_k \mathrm{E}\{f_k|\boldsymbol{r}_{k-N}, a_k, S_k\} = c_k \mathrm{E}\{f_k|\boldsymbol{r}_{k-N}, S_k\} \\ &= c_k \widehat{f}_k(S_k) \qquad (3.213)\end{aligned}$$

$$\widehat{f}_k(S_k) = \sum_{i=1}^{N} p_i''(S_k) \frac{r_{k-i}}{c_{k-i}(S_k)} \qquad (3.214)$$

where the notation $\{c_{k-i}(S_k)\}$ indicates the coded symbols associated with state S_k. In particular, $\widehat{f}_k(S_k)$ denotes a *state-dependent* linear prediction of the fading coefficient at time k, based on previous observations.[27] The coefficients $\{p_i''(S_k)\}$ are *state-dependent* prediction coefficients of the fading process based on previous observations of noisy *fading-like state-dependent* sequences $\{r_{k-i}/c_{k-i}(S_k)\}_{i=1}^{N}$. The mean square prediction error of observation and fading coefficient are similarly related as follows:

$$\begin{aligned}\widehat{\sigma}^2(a_k, S_k) &= \mathrm{E}\{|r_k - \widehat{r}_k(a_k, S_k)|^2|\boldsymbol{r}_{k-N}^{k-1}, a_k, S_k\} \\ &= \mathrm{E}\{|c_k f_k + n_k - c_k \widehat{f}_k(a_k, S_k)|^2|\boldsymbol{r}_{k-N}^{k-1}, a_k, S_k\} \\ &= |c_k|^2 \widehat{\sigma}_f^2(a_k, S_k) + 2\sigma^2 \qquad (3.215)\end{aligned}$$

$$\widehat{\sigma}_f^2(a_k, S_k) = \mathrm{E}\{|f_k - \widehat{f}_k(a_k, S_k)|^2|\boldsymbol{r}_{k-N}^{k-1}, a_k, S_k\}. \qquad (3.216)$$

The branch metric can then be expressed as

$$\lambda_k(a_k, S_k) = -\frac{|r_k - c_k \widehat{f}_k(a_k, S_k)|^2}{|c_k|^2 \widehat{\sigma}_f^2(a_k, S_k) + 2\sigma^2} - \ln\left[|c_k|^2 \widehat{\sigma}_f^2(a_k, S_k) + 2\sigma^2\right] + \ln P\{a_k\}. \qquad (3.217)$$

In particular, the fading prediction coefficients $p_i''(S_k)$ and mean square prediction error $\widehat{\sigma}_f^2(a_k, S_k)$ are the solution of the following Wiener Hopf equation:

$$\left(\boldsymbol{F}_k^{(N)} + 2\sigma^2 \boldsymbol{C}_k^{(N)}\right) \boldsymbol{p}'' = \boldsymbol{q}'' \qquad (3.218)$$

[27] Note that in the case of application of state reduction techniques, some of the symbols c_{k-i} have to be retrieved on the survivor associated with the reduced state replacing S_k. In this case, the linear predictions of the fading coeffi cients can be interpreted as *path-dependent*. The same comment holds also for the prediction coeffi cients $\{p_i''(S_k)\}$.

where

$$\boldsymbol{F}_k^{(N)} = \mathrm{E}\{\boldsymbol{f}_{k-N}^k(\boldsymbol{f}_{k-N}^k)^{\mathrm{H}}\} \tag{3.219}$$

$$\boldsymbol{C}_k^{(N)} = \mathrm{diag}\left(\frac{1}{|c_k|^2}, \frac{1}{|c_{k-1}|^2}, \ldots, \frac{1}{|c_{k-N}|^2}\right) \tag{3.220}$$

$$\boldsymbol{p}'' = (1, -p_1'', -p_2'', \ldots, -p_N'')^{\mathrm{T}} \tag{3.221}$$

$$\boldsymbol{q}'' = (\widehat{\sigma}_f^2 + \frac{2\sigma^2}{|c_k|^2}, \underbrace{0, \ldots, 0}_{N \text{ zeros}})^{\mathrm{T}}. \tag{3.222}$$

The dependence of the solution on the hypothetical sequence is only through the moduli of the code symbols.

For code symbols with constant modulus, the prediction coefficients do not depend on the state S_k and symbol a_k. For instance, this is the case for PSK, i.e., $|c_k| = 1$. In this case, the branch metric simplifies as follows:

$$\begin{aligned}\lambda_k(a_k, S_k) &= -|r_k - c_k\widehat{f}_k(S_k)|^2 + (\widehat{\sigma}_f^2 + 2\sigma^2)\ln P\{a_k\} \\ &= -\left|r_k - c_k\sum_{i=1}^{N} p_i''\frac{r_{k-i}}{c_{k-i}}\right|^2 + (\widehat{\sigma}_f^2 + 2\sigma^2)\ln P\{a_k\}. \end{aligned} \tag{3.223}$$

This solution is remarkably similar to what we would obtain in a decomposed estimation-detection strategy by estimating the "undesired" parameter f_k according to PSP. Note that the (Gaussian) prediction error variance $\widehat{\sigma}_f^2$ affects the "overall" thermal noise power.

It is possible to interpret the designed algorithm as embedding linear predictive estimation of the fading coefficient f_k. In particular, the observation model $r_k = c_k f_k + n_k$ satisfies a parameter-conditional FMC by viewing f_k as un undesired parameter. In order to estimate this parameter, the following steps should be considered.

1. Adopt a linear feed-forward data-aided parameter estimator of order N (see Section 3.11.1).

2. Use a set of estimators by associating one estimator to each trellis path.

3. Compute the estimation coefficients in order to minimize the mean square estimation error with respect to the random variable r_k/c_k, conditional on the path data sequence.

The resulting estimator is exactly the described *state-dependent* linear predictor. Hence, linear prediction of f_k based on previous observations is a form of PSP-based *feed-forward* parameter estimation.

The state-complexity of a linear predictive receiver can be naturally decoupled from the prediction order N by means of state reduction techniques. For the sake of simplicity, we consider folding by memory truncation, but set partitioning could be used as well. Let $Q < N$ denote the memory parameter to be taken into account in the definition of a reduced trellis state, so that a reduced state w_k can be written as follows:

$$w_k = (a_{k-1}, \ldots, a_{k-Q}, \mu_{k-Q}). \tag{3.224}$$

The branch metric can be obtained by defining a pseudo-state[28] as

$$\tilde{S}_k(w_k) = (\underbrace{a_{k-1}, \ldots, a_{k-Q}, \mu_{k-Q}}_{w_k}, \breve{a}_{k-Q+1}(w_k), \ldots, \breve{a}_{k-N}(w_k), \breve{\mu}_{k-N}(w_k)) \tag{3.225}$$

where $(\breve{a}_{k-Q+1}(w_k), \ldots, \breve{a}_{k-N}(w_k))$ and $\breve{\mu}_{k-N}(w_k)$ are information symbols and a state in the survivor of state w_k, respectively. The branch metric in the RS trellis can be defined as usual according to

$$\tilde{\lambda}_k(a_k, w_k) = \lambda(a_k, \tilde{S}_k). \tag{3.226}$$

For coded PSK, the branch metric is

$$\begin{aligned} \lambda_k(a_k, S_k) &= -|r_k - c_k \widehat{f}_k(\tilde{S}_k)|^2 + (\widehat{\sigma}_f^2 + 2\sigma^2) \ln P\{a_k\} \\ &= -\left| r_k - c_k \sum_{i=1}^{Q} p_i'' \frac{r_{k-i}}{c_{k-i}(w_k)} - c_k \sum_{i=Q+1}^{N} p_i'' \frac{r_{k-i}}{\breve{c}_{k-i}(w_k)} \right|^2 \\ &\quad + (\widehat{\sigma}_f^2 + 2\sigma^2) \ln P\{a_k\}. \end{aligned} \tag{3.227}$$

The prediction order N and RS parameter Q are design parameters to be jointly optimized by experiment to yield a good compromise between performance and complexity. As an example, we consider the case of uncoded QPSK signaling (the cardinality of the information symbols is $M = 4$) over a time-varying flat fading channel. The BER performance, as a function of the SNR, is shown in Figure 3.33. The normalized maximum Doppler rate is indicated as $f_D T$. In particular, we consider a prediction order $N = 10$ and a reduced state parameter $Q = 2$ (16 possible reduced

[28]Note that the formulation in (3.225) is very general and accounts for the case of a recursive state as well. In the case of a nonrecursive state, a simplifi ed formulation is possible.

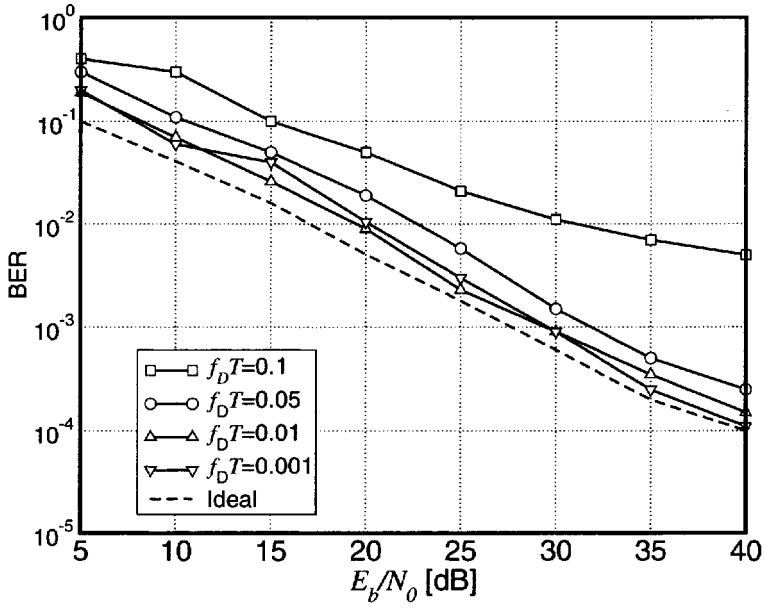


Figure 3.33: BER of a linear predictive receiver for transmission of QPSK over a time-varying flat Rayleigh fading channel. Various values of the normalized maximum Doppler rate are considered. Reproduced from [124], ©1995 IEEE, by permission of the IEEE.

states). Pilot symbols are periodically inserted in the transmitted sequence (the insertion rate is 1 every 9 data symbols). For comparison, the curve relative to the case of perfect channel state information (i.e., when the fading process is perfectly known at the receiver side) is shown.

3.11.6 Linear Predictive Sequence Detection for Frequency Selective Fading Channels

A frequency selective fading channel can be visualized as a time-varying ISI channel. In particular, if the channel response is time-varying according to an arbitrary, but known, stochastic model, we assume that the detector has knowledge of the underlying statistics. In the following, we apply the CMP to derive sequence detection algorithms for signals transmitted over frequency selective fading channels with arbitrary statistics (i.e., amplitude distribution and correlation). No assumption of Gaussian fading or additive noise is therefore made *a priori*, although specific solutions

are proposed for the case of Rayleigh fading channels.

The final discrete-time expression (A.32) for the observable derived in Appendix A can be rewritten by making explicit the indexing of the overall symbol period and of the sampling instants inside each period:

$$r_{k,i} = \sum_{j=0}^{L} f_{i,j}^{(k)} a_{k-j} + n_{k,i} \qquad i = 1, \ldots, \beta. \tag{3.228}$$

Defining the following quantities[29]

$$\boldsymbol{r}_k \triangleq [r_{k,1}, \ldots, r_{k,\beta}]^{\mathrm{T}} \tag{3.229}$$

$$\boldsymbol{a}_{k-L}^{k} \triangleq [a_k, \ldots, a_{k-L}]^{\mathrm{T}} \tag{3.230}$$

$$\boldsymbol{F}^{(k)} \triangleq \begin{bmatrix} f_{1,1}^{(k)} & \cdots & f_{1,L}^{(k)} \\ & \cdots & \\ f_{\beta,1}^{(k)} & \cdots & f_{\beta,L}^{(k)} \end{bmatrix} \tag{3.231}$$

$$\boldsymbol{n}_k \triangleq [n_{k,1}, \ldots, n_{k,\beta}]^{\mathrm{T}} \tag{3.232}$$

one can compactly indicate the oversamples relative to the k-th signaling interval as follows:

$$\boldsymbol{r}_k = \boldsymbol{F}^{(k)} \boldsymbol{a}_{k-L}^{k} + \boldsymbol{n}_k. \tag{3.233}$$

The CMP extends naturally to the case of oversampling, provided that a vector notation is used. In other words, one can write:

$$p(\boldsymbol{r}_k | \boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(\boldsymbol{r}_k | \boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k}) \tag{3.234}$$

where N is the order of Markovianity, C is the finite memory parameter given by $C = N + L$, and $\boldsymbol{r}_{k-N}^{k-1}$ is a $\beta(N+1) \times 1$ column vector defined as

$$\begin{aligned} \boldsymbol{r}_{k-N}^{k-1} &\triangleq [\boldsymbol{r}_{k-N}^T, \boldsymbol{r}_{k-N+1}^T, \ldots, \boldsymbol{r}_k^T]^T \\ &= [r_{k-N,1}, \ldots, r_{k-N,\beta}, r_{k-N+1,1}, \ldots, r_{k-N+1,\beta}, \ldots, r_{k,1}, \ldots, r_{k,\beta}]^T. \end{aligned} \tag{3.235}$$

In other words, in the case of oversampling we assume that the set of β conditional observations relative to a signaling interval depends only on the N most recent previous sets. Hence, the branch metric has formally the general expression given by (3.17). By defining $S_k \triangleq (\boldsymbol{a}_{k-C}^{k-1})$, one can write:

$$\lambda_k(a_k, S_k) = \ln p(\boldsymbol{r}_k | \boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k}) + \ln P\{a_k\}. \tag{3.236}$$

[29]Note that while the observation $\boldsymbol{r}_k$ is a vector with elements the oversamples considered in the k-th signaling interval, the information sequence $\{a_k\}$ is such that there is only one symbol per signaling interval.

Note that (3.236) holds for any fading or noise statistics, as long as the CMP is satisfied. This metric could thus be used for non-Gaussian fading (e.g., Nakagami fading) and non-Gaussian additive noise (e.g., impulsive noise or noise generated by interfering signals).

In the important case of Rayleigh fading, the conditional observation and the relevant pdfs are multivariate zero-mean Gaussian and further simplifications are possible. Defining the following covariance matrices:

$$\boldsymbol{R}_{k-N}^{k}(\boldsymbol{a}_{k-C}^{k}) \triangleq \mathrm{E}\{\boldsymbol{r}_{k-N}^{k}(\boldsymbol{r}_{k-N}^{k})^{\mathrm{H}}|\boldsymbol{a}_{k-C}^{k}\} \tag{3.237}$$

$$\boldsymbol{R}_{k-N}^{k-1}(\boldsymbol{a}_{k-C}^{k-1}) \triangleq \mathrm{E}\{\boldsymbol{r}_{k-N}^{k-1}(\boldsymbol{r}_{k-N}^{k-1})^{\mathrm{H}}|\boldsymbol{a}_{k-C}^{k-1}\} \tag{3.238}$$

simple algebra leads one to the following expression for the branch metric (3.236):

$$\begin{aligned}\lambda_k(a_k, S_k) \sim & \ln\left[\frac{\det \boldsymbol{R}_{k-N}^{k}(\boldsymbol{a}_{k-C}^{k})}{\det \boldsymbol{R}_{k-N}^{k-1}(\boldsymbol{a}_{k-C}^{k-1})}\right] \\ & -(\boldsymbol{r}_{k-N}^{k})^{\mathrm{H}}\left[\boldsymbol{R}_{k-N}^{k}(\boldsymbol{a}_{k-C}^{k})\right]^{-1}\boldsymbol{r}_{k-N}^{k} \\ & +(\boldsymbol{r}_{k-N}^{k-1})^{\mathrm{H}}\left[\boldsymbol{R}_{k-N}^{k-1}(\boldsymbol{a}_{k-C}^{k-1})\right]^{-1}\boldsymbol{r}_{k-N}^{k-1} + \ln P\{a_k\}. \end{aligned} \tag{3.239}$$

We note that the matrices $\boldsymbol{R}_{k-N}^{k}(\boldsymbol{a}_{k-C}^{k})$ and $\boldsymbol{R}_{k-N}^{k-1}(\boldsymbol{a}_{k-C}^{k-1})$ are actually independent of time epoch k if the channel and noise are stationary.

In the considered case of a Rayleigh fading channel, the branch metric (3.239) can also be given a different equivalent formulation. In fact, by applying the chain factorization rule to the conditional pdf (3.234) over the corresponding samples of a vectorial observation $\boldsymbol{r}_k = [r_{k,1}, \ldots, r_{k,\beta}]^{\mathrm{T}}$, one can write:

$$\lambda_k(a_k, S_k) = \sum_{i=1}^{\beta} \ln p(r_{k,i}|r_{k,i-1}, \ldots, r_{k,1}, \boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k}) + \ln P\{a_k\}. \tag{3.240}$$

Each conditional observation sample has Gaussian distribution, so that one can further rewrite (3.240) as follows:

$$\begin{aligned}\lambda_k(a_k, S_k) \sim & \sum_{i=1}^{\beta}\left\{-\frac{\left|r_{k,i} - \overline{r_{k,i|k,i-1}}(r_{k,i-1}, \ldots, r_{k,1}, \boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k})\right|^2}{\sigma_{k,i|k,i-1}^{2}(\boldsymbol{a}_{k-C}^{k})}\right. \\ & \left. - \ln \sigma_{k,i|k,i-1}^{2}(\boldsymbol{a}_{k-C}^{k})\right\} + \ln P\{a_k\} \end{aligned} \tag{3.241}$$

where $\overline{r_{k,i|k,i-1}}(r_{k,i-1}, \ldots, r_{k,1}, \boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^{k})$ and $\sigma_{k,i|k,i-1}^{2}(\boldsymbol{a}_{k-C}^{k})$ indicate the conditional mean and variance, respectively, and can be evaluated by solving a suitable

MMSE problem.[30] We remark that only the conditional mean depends on both observations and information symbols, whereas the conditional variance is indeed independent of the observations [131].

The complexity of the proposed algorithms can be straightforwardly reduced by applying the state reduction techniques introduced in Section 3.10.1. In particular, considering the application of memory truncation techniques, it is possible to define a reduced state as

$$s_k \triangleq (a_{k-1}, \ldots, a_{k-Q}) \tag{3.242}$$

where $Q \in \{1, \ldots, L\}$ represents the RS parameter. In order to compute the branch metric (3.241), the information symbols $(\widehat{a}_{k-L}, \ldots, \widehat{a}_{k-Q+1})$ are recovered from the survivor (based on stored decision feedback) associated with reduced state s_k.

For Rayleigh fading, assuming that the tap coefficients $\{f_{i,n}\}_{i=0}^{L}$ in (A.32) obey an autoregressive moving average (ARMA) model, the conditional mean and conditional variance which appear in (3.241) may be computed recursively using a bank of Kalman filters as shown in [132, 133]. Although these algorithms may be operated with trellis diagrams of various state complexities, this aspect is only implicit in [132, 133], where the ISI trellis diagram is often used. These detection schemes based on per-survivor Kalman channel estimation may be shown to coincide with the proposed detection algorithm in the case with $Q = L$, if the CMP expressed by (3.234) holds. As a matter of fact, a Kalman filtering technique is hidden in the proposed algorithms, as shown in [121].

We now present a performance analysis of the proposed detection algorithms for frequency selective fading channels. The assumed performance measure is the average BER versus E_b/N_0, where E_b denotes the received signal energy per information bit averaged over the data and channel statistics. The assumed receiver filter is a root Nyquist filter regardless of the selected oversampling factor. In the case of an oversampled front-end, a root Nyquist filter frequency response with vestigial symmetry around $1/(2T_s)$ satisfies both (A.19) and (A.20) if

$$B \leq \frac{1-\delta}{2T}\beta \tag{3.243}$$

where δ is the roll-off factor of the receiving filter.

The overall time-discrete channel is modeled according to (A.32), in which the span of the ISI is set to $L = 2$ and the time-varying channel tap coefficients $\{f_{i,n}\}_{i=0}^{L}$

[30]Note that for an optimal filtering approach, the dependence on the entire observation sequence should be considered. However, in (3.241) we are already considering the application of the CMP.

are assumed to be characterized by the following first order ARMA model:

$$f_{i,n} = \rho^{\frac{1}{\beta}} f_{i,n-1} + \sqrt{1-\rho^{\frac{2}{\beta}}}\; z_{i,n} \qquad i = 0, 1, \ldots, L \tag{3.244}$$

where ρ is a forgetting factor and $\{z_{i,n}\}_{i=0}^{L}$ are independent, white, complex, Gaussian random processes with independent real and imaginary components. This simple time-discrete model is suitable to be employed both for symbol-spaced sampled detectors ($\beta = 1$) and for oversampled detectors ($\beta > 1$). However, the received signal is not strictly bandlimited and the observation sequence is only an approximate sufficient statistic, even for $\beta > 1$. The correlation between fading samples whose distance in time is a symbol interval is always ρ, regardless of the selected oversampling factor β. The simulations are performed assuming that (A.31) is met for $\beta = 1$ and neglecting the cross-correlation among taps for $\beta > 1$. The latter assumption is known to be pejorative with respect to the case in which this cross-correlation is taken into account [133].

The standard deviation of the tap gains $(f_{0,n}, f_{1,n}, f_{2,n})$ is set to the values $(0.407, 0.815, 0.407)$. Different values of ρ have been considered (the smaller the parameter ρ, the faster the channel variations associated with it), and two different modulation formats: binary phase shift keying (BPSK, $M = 2$) and QPSK ($M = 4$). An information block length of $K = 60$ symbols is assumed, with a known preamble and postamble of 2 symbols.[31]

In Figure 3.34, we show the BER of the blind recursive receiver employing one sample per symbol interval ($\beta = 1$) as a function of the overall assumed memory C, for BPSK modulation. This figure is intended to validate the CMP, which allows us to factorize the conditional pdf. Since $L = 2$ is fixed because of the assumed channel model and since we consider a full-complexity solution ($Q = L$), the figure allows us to infer empirically the value of the channel dependence parameter N. We can see that for the considered values of ρ, the BER levels out for increasing values of N. In fact from an order of Markovianity equal to $N = 8$ ($C = 10$) onwards, the BER curves exhibit an error floor indicating that no BER improvement is further achievable by increasing the assumed order of Markovianity N.

Figure 3.35 shows the performance curves of the blind recursive detector with $\beta = 1$ for QPSK modulation and $\rho = 0.998$. One can notice that a larger (but not dramatic) performance loss, with respect to BPSK, is exhibited by the reduced-complexity receiver (compare $C = 5, Q = 5, 4, 3$). Furthermore, for an equal number of trellis states ($Q = 5$), the use of state reduction techniques allows one to improve

[31] The presence of the preamble and postamble are expedient for the initialization and the termination of the VA. More details can be found in [121].

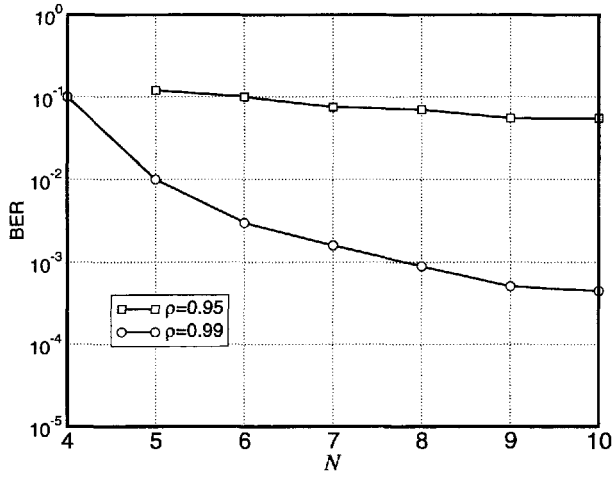


Figure 3.34: BER of the blind recursive detector with $\beta = 1$ for $E_b/N_0 = 40$ dB and BPSK as a function of the assumed memory N. Reproduced from [121] by permission of John Wiley & Sons.

significantly the performance by increasing the assumed channel memory from $C = 5$ to 6 and 7.

In Figure 3.36, we present a performance comparison between symbol-spaced sampled and oversampled blind detectors, for BPSK modulation and $\rho = 0.99$. The oversampled receivers employing two samples per symbol interval ($\beta = 2$) exhibit an error floor which is remarkably lower than that of the corresponding symbol-spaced sampled receivers. Oversampling the received signal may also be a way to compromise in terms of complexity and computational needs. For example, we can compare the curve obtained with $C = Q = 6$ and $\beta = 2$ with that obtained with $C = Q = 7$ and $\beta = 1$: the oversampled receiver performs slightly better. This is due to the fact that even if it has half the number of states, the branch metric makes use of matrices and observation vectors which have double length with respect to that of a symbol-spaced receiver.

3.12 Summary

This chapter has been devoted to MAP sequence detection. After describing this concept, the *Viterbi algorithm*, the most famous and successful detection algorithm

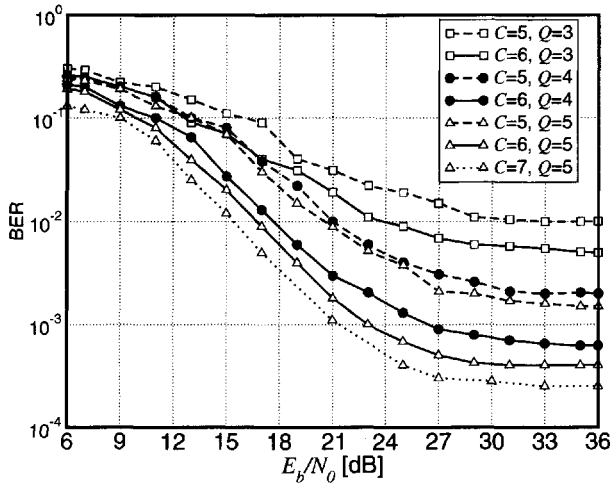


Figure 3.35: BER versus E_b/N_0 of the blind recursive detector with $\beta = 1$ for QPSK modulation and $\rho = 0.998$. Reproduced from [121] by permission of John Wiley & Sons.

to date, has been described and shown to exactly implement MAP sequence detection. *Finite memory sequence detection* has been introduced, as an instance of the general approach introduced in Chapter 2. After a discussion about the relation between *detection* and *estimation*, a few preliminary examples of data-aided parameter estimation and joint detection and estimation have been considered. The concept of *per-survivor processing* has been presented, along with examples of application to phase-uncertain channels and dispersive slow fading channels. This concept turns out to be related, explicitly or implicitly, to several detection algorithms considered in this book. In order to limit the complexity of finite memory sequence detection, state reduction techniques for detection strategies based on the Viterbi algorithm have been introduced. Finally, several applications to wireless communications have been considered: adaptive sequence detection based on least mean squares estimation for tracking a dispersive fading channel and joint detection and phase synchronization; noncoherent sequence detection for phase-uncertain and slowly varying frequency nonselective fading channels; linear predictive sequence detection for phase-uncertain and fading channels.

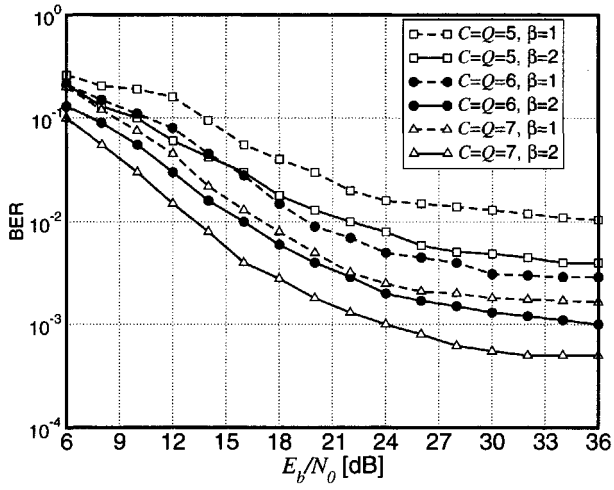


Figure 3.36: BER versus E_b/N_0 of the blind recursive detector with $\beta = 1$ and $\beta = 2$ for BPSK modulation and $\rho = 0.99$. Reproduced from [121] by permission of John Wiley & Sons.

3.13 Problems

Problem 3.1: Consider a base-band binary transmission system which uses equally probable symbols $a_k \in \{-1, +1\}$ and linear modulation. The channel is affected by AWGN. The samples at the output of a matched filter can be expressed as

$$r_k = \sum_{\ell=-1}^{+1} g_\ell c_{k-\ell} + n_k^{(c)}$$

where $\{c_k\}$ are the transmitted symbol and $\{g_k\}$ is the overall discrete impulse response such that

$$g_k = \begin{cases} 1 & k = 0 \\ 0.1 & k = \pm 1. \end{cases}$$

The noise process $\{n_k^{(c)}\}$ is colored, with autocorrelation equal to $R_n^{(c)}(k) = \mathrm{E}\{n_{k+j}^{(c)}(n_j^{(c)})^*\} = 2N_0 g_k$.

A. Design a digital noise-whitening filter to process the observations $\{r_k\}$ and determine the overall discrete-time impulse response.

B. Consider MAP sequence detection based on the samples at the output of the whitening filter: define a suitable state and design the corresponding trellis diagram.

Problem 3.2: Consider a linearly modulated signal with QPSK. The complex envelope of the transmitted signal can be written as follows:

$$s(t) = \sum_{k=0}^{K-1} c_k p(t - kT)$$

where the symbols $\{c_k\}$ are differentially encoded according to the rule $c_k = c_{k-1}a_k$. The information and code symbols belong the alphabet $\{\pm 1, \pm j\}$. Assume that there is AWGN, the information symbols are equally likely and the condition for absence of ISI is satisfied. At the receiver we consider a detector based on the VA.

A. Draw the trellis diagram of the detector.

B. Show that the survivors at epoch $k+1$ can be obtained by directly extending the survivor at epoch k with the best metric (in other words, the depth of convergence of the survivors is one).

C. Show that considering sequence detection based on the VA is, in this case, equivalent to symbol by symbol demodulation.

D. Discuss the effects on the decision process of a phase rotation in the received signal equal to a multiple of $\pi/2$.

Problem 3.3: Independent and equally likely information symbols $\{a_k\} \in \{\pm 1\}$ are coded according to the following rule:

$$c_k = a_k + a_{k-1}.$$

A coded symbol c_k thus belongs to the alphabet $\{-2, 0, +2\}$. The coded symbols are transmitted through a base-band linear modulation with shaping pulse $p(t)$, whose Fourier transform is a raised root cosine with no excess bandwidth (i.e., it is a rectangular spectrum). The transmitted signal has therefore the following expression:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k p(t - kT).$$

Considering transmission over an AWGN channel and assuming that the receiver front-end is a sampled matched filter, derive the VA to implement MAP sequence detection.

Problem 3.4: Consider the general class of reduced-state algorithms, indicated as SA($n_{\rm pa}, n_{\rm cl}$) in Section 3.10.4 and originally described in [108]. Justify why, for a given maximum complexity level given by the product $n_{\rm pa}n_{\rm cl}$, the reduced-state sequence detection algorithm which provides the best performance is the M-algorithm, i.e., the algorithm with $M = n_{\rm pa}$ and $n_{\rm cl} = 1$. Compare your solution with the results in [108, 109].

Problem 3.5: The observation at the output of a phase-uncertain channel can be written as follows:

$$r_k = c_k e^{j\theta_k} + n_k$$

where n_k is an AWGN sample. It is easy to show that

$$\Im\left\{r_k c_k^* e^{-\theta_k}\right\} \sim \frac{\mathrm{d}\ln p(r_k|c_k,\theta_k)}{\mathrm{d}\theta}.$$

A gradient algorithm for estimating the channel phase process would be based on the following recursive estimation:

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k + \eta\Im\left\{r_k c_k^* e^{-\theta_k}\right\}.$$

Comment on the relationship of this algorithm with the joint phase synchronization and MAP sequence detection strategy considered in Section 3.8.1 (data-aided) and Section 3.9.1 (PSP-based) where a first order PLL is used.

Problem 3.6: Suppose transmission of independent and equally distributed symbols belonging to an 8-PSK constellation over an ISI channel with memory $L = 4$. Draw all possible trellis diagrams with four states to perform reduced-state detection. Considering various ISI channel impulse responses, describe the pros and cons of the considered state reduction techniques.

Problem 3.7: In the case of application of NSD to fading channels, derive the basic branch metric (3.160) by considering the approach specular to that proposed in the text. In other words, first apply the CMP, and then average over the (known) channel statistics.

Problem 3.8: Considering the observation model (3.30) for a flat slow fading channel in Section 3.5.2, show that linear predictive sequence detection is equivalent to the strategy proposed in Section 3.5.2 and based on the correlation properties of the observation sequence.

Problem 3.9: Starting from (3.138), derive the metric (3.135). Note that this approach is the "specular" version of that proposed in this book, where, first, an approximate application of the CMP is used, i.e., memory truncation is introduced, and then statistical averaging is performed.

Problem 3.10: Derive (3.160) using the "standard" approach proposed in this book: first apply (approximately) the CMP and then average out the channel statistical parameters.

Problem 3.11: Verify by computer simulations that the use of the approximation $\ln \mathrm{I}_0(x) \simeq x$ does not entail major performance degradation in the performance of a VA-based receiver using (3.160).

Problem 3.12: Verify that the performance of a VA-based receiver using (3.164) and (3.165) is the same.

Problem 3.13: Verify that the performance of a VA-based receiver using (3.165) is equivalent, in the case of Rice fading, to that obtained by (3.160). Devise suitable approximations which allow one to derive (3.165) from (3.160).

Problem 3.14: Consider uncoded transmission of binary symbols $a_k \in \{\pm 1\}$ through a static dispersive channel with white noise discrete equivalent impulse response $\boldsymbol{f} = (1, 2, 1)/\sqrt{6}$.

A. Define a suitable system state μ_k and draw the relevant trellis diagram.

B. Express explicitly the branch metric as a function of the received signal sample r_k for any possible transition.

   Assume that the received sequence is

   $$(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7) = (1.7, 1.2, 1.1, 0.3, -0.2, -1.1, 0.7, 0.4)$$

   and the initial state is $\mu_0 = (+1, +1)$.

C. Use the VA to detect the MAP sequence $\{\hat{a}_k^{(\mathrm{VA})}\}_{k=0}^{7}$.

Problem 3.15: Consider the model of discrete observable

$$r_k = e^{j\theta} c_k + n_k$$

where θ is the overall phase rotation introduced by the channel. Let $\hat{\theta}$ be an estimate of the channel phase and define the "phase-synchronized" observation $\overline{r}_k \triangleq r_k e^{-j\hat{\theta}}$.

A. Derive an explicit expression of the MSE $\mathrm{E}|\overline{r}_k - c_k|^2$ as a function of $\hat{\theta}$.

B. Obtain an estimate of θ minimizing the MSE.

C. Formulate a data-aided iterative stochastic gradient algorithm with the aim to minimize the MSE.

D. Comment on the functional relationship of the obtained synchronization scheme with a first order PLL.

Hint: *Define a stochastic gradient by differentiating the MSE with respect to $\hat{\theta}$ and discarding the expectation.*

Problem 3.16: Consider the model of discrete observable:

$$r_k = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{c}_k + n_k$$

where $\boldsymbol{f}$ is the overall discrete equivalent channel impulse response.

Let $\hat{\boldsymbol{f}}$ be an estimate of the channel response. Assume that the code symbols are zero-mean and uncorrelated.

A. Derive an explicit expression of the MSE $\mathrm{E}\{|r_k - \hat{\boldsymbol{f}}^{\mathrm{T}} \boldsymbol{c}_k|^2\}$ as a function of $\hat{\boldsymbol{f}}$.

B. Formulate a data-aided iterative stochastic gradient algorithm to minimize the MSE.

C. Comment on the functional relationship of the obtained identification scheme and the LMS algorithm.

Hint: *Define a stochastic gradient by differentiating the MSE with respect to $\hat{\boldsymbol{f}}$ and discarding the expectation.*

Problem 3.17: Assume a first order autoregressive fading model

$$f_{k+1} = \sqrt{1-\rho^2} f_k + \rho v_k$$

where $\rho \in [0, 1]$ is a constant and $\{v_k\}$ is an iid zero-mean Gaussian sequence with variance σ_v^2.

A. Show that the fading sequence is Markovian of first order.

Assume f_0 is Gaussian with variance σ_v^2.

B. Show that f_k is a stationary Gaussian sequence.

C. Check if the conditional observation $\{r_k\}$ satisfies a Markov property.

Problem 3.18: Consider the random-phase discrete channel model

$$r_k = c_k e^{j\theta} + n_k.$$

Define a feed-forward data-aided phase estimate $\hat{\theta}$ based on the previous N observations by minimizing the MSE $\mathrm{E}\{||\boldsymbol{r}_{k-N}^{k-1} - \boldsymbol{c}_{k-N}^{k-1} e^{j\hat{\theta}}||^2\}$.

A. Show that this estimate must verify the condition

$$e^{-j\hat{\theta}} = \frac{(\boldsymbol{r}_{k-N}^{k-1})^{\mathrm{H}} \boldsymbol{c}_{k-N}^{k-1}}{\left|(\boldsymbol{r}_{k-N}^{k-1})^{\mathrm{H}} \boldsymbol{c}_{k-N}^{k-1}\right|}.$$

B. Show that the result in part A coincides with the data-aided ML phase estimate based on the previous N observations.

4

Symbol Detection: Algorithms and Applications

4.1 Introduction

This chapter focuses on the application of the general framework proposed in Chapter 2 to the case of maximum *a posteriori* (MAP) *symbol* detection. In particular, performing MAP symbol detection requires the *explicit* computation of a symbol *a posteriori* probability (APP), i.e., the probability that a particular symbol has been transmitted given the observation of the entire received signal, or, considering an optimal discretization of the received signal, the entire sequence of discrete-time channel observations. Since, in various cases, the calculation of the exact APP may be computationally very intensive, one can resort to the computation of suboptimal values which approximate the APPs.

While the bit error rate (BER) performance of a given transmission system based on the MAP symbol detection criterion is substantially identical[1] to that obtained by applying the MAP sequence detection criterion, the discovery of the concept of *iterative decoding/detection* [33] spurred a strong interest and attention for symbol detection algorithms. The most commonly used algorithm to perform MAP symbol detection is the so-called *forward backward* (FB) algorithm. Seminal work in the design of algorithms for soft-output decoding dates back to the late Sixties [134,135]. An instance of the FB algorithm was proposed in [136], but a clear formalization is

Detection Algorithms for Wireless Communications - G. Ferrari, G. Colavolpe and R. Raheli
 ISBN: 0-470-85828-1

[1] This is exactly correct for large signal-to-noise ratios, but it represents a good approximation also at low signal-to-noise ratios (SNRs).

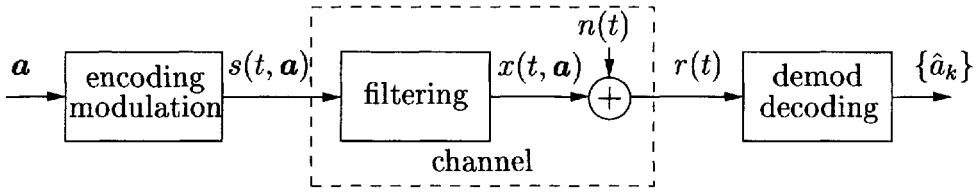


Figure 4.1: Transmission system and MAP symbol detection.

due to Bahl, Cocke, Jelinek and Raviv in 1974 [54]—for this reason, the FB algorithm is also often referred to as BCJR algorithm from the initials of the last names of the authors of [54].

4.2 MAP Symbol Detection Principle

We assume that the transmission system can be represented as shown in Figure 4.1. In particular, a source generates a sequence $\boldsymbol{a} = \boldsymbol{a}_0^{K-1}$ of information symbols, which is transformed by the encoder/modulator into a time-continuous signal $s(t, \boldsymbol{a})$. For the sake of simplicity, we assume that the duration of the signal is KT, where T denotes the symbol interval. The MAP symbol detection criterion leads to the choice of the symbol $\hat{a}_k$ which minimizes the probability of error with respect to the received signal. More precisely, this criterion can be formulated as follows:

$$\hat{a}_k = \text{argmax}_{a_k} P\{a_k | r(t)\} \tag{4.1}$$

where $r(t)$ is the received signal observed over the entire information-bearing interval T_0. Due to the system memory, T_0 may include the interval $(0, KT)$, possible border intervals or, ideally, the entire time axis. Considering a suitable discretization process, through which the received signal $r(t)$ is converted into an equivalent sequence of discrete-time observations $\boldsymbol{r}$, whose dimension depends on the number of samples per symbol interval T, the strategy (4.1) can be reformulated as follows:

$$\hat{a}_k = \text{argmax}_{a_k} P\{a_k | \boldsymbol{r}\}. \tag{4.2}$$

In order to compute the APP $P\{a_k|\boldsymbol{r}\}$ one can write:

$$P\{a_k|\boldsymbol{r}\} = \sum_{\boldsymbol{a}:a_k} P\{\boldsymbol{a}|\boldsymbol{r}\} = \sum_{\boldsymbol{a}:a_k} p(\boldsymbol{r}|\boldsymbol{a}) P\{\boldsymbol{a}\} \tag{4.3}$$

where the notation $\boldsymbol{a} : a_k$ denotes all possible information sequences containing a_k, or compatible with a_k. Note that the computation of the APP, as indicated in (4.3),

requires the computation of the same metric as in the case of MAP sequence detection (i.e., $p(\boldsymbol{r}|\boldsymbol{a})P\{\boldsymbol{a}\}$) and a further *marginalization* based on the sum over all the information sequences compatible with symbol a_k. Assuming that the information symbols are independent, one can further express the APP as follows:

$$P\{a_k|\boldsymbol{r}\} = P\{a_k\} \sum_{\boldsymbol{a}:a_k} p(\boldsymbol{r}|\boldsymbol{a}) \prod_{i=0, i\neq k}^{K-1} P\{a_i\}. \tag{4.4}$$

The first and simplest possible approach for the evaluation of the APP could be based on the computation of expression (4.4). It is immediately concluded that the efficiency of this computation is very low, since one must compute a large number of sequence metrics and then marginalize by adding them together. The complexity would obviously be exponential in the transmission length K. The FB algorithm, introduced in more detail in the following section, represents an efficient way to compute the APP, with a complexity linear with the transmission length K, as in the case of a Viterbi algorithm (VA) for MAP sequence detection.

4.3 Forward Backward Algorithm

As already mentioned, the first clear formulation of the FB algorithm can be found in [54]. In the following, we propose a simple derivation of the FB algorithm for transmission over a *memoryless channel*. We assume that the encoder/modulator can be represented as a finite state machine (FSM), with state μ_k and output symbol c_k. As in Chapter 2, we assume that the next-state and the output functions are known:[2]

$$\mu_{k+1} = \mathrm{ns}(a_k, \mu_k) \tag{4.5}$$

$$c_k = \mathrm{o}(a_k, \mu_k). \tag{4.6}$$

The couple (μ_k, a_k) uniquely identifies a transition in the trellis diagram of the encoder/modulator FSM. We denote this transition as t_k. After a suitable discretization process with one sample per information symbol, we assume that the observable at epoch k can be written as

$$r_k = c_k + n_k. \tag{4.7}$$

[2]Note that the derivation shown in the following holds also in the case of a channel with strictly finite memory, the only difference being the interpretation of μ_k as the state of the FSM obtained by concatenating the encoder/modulator and the channel. Moreover, we assume generation of a single output symbol c_k in correspondence to an information symbol a_k, but the derivation can be straightforwardly extended to the case of multiple output symbols by using a vector notation.

In this case, the APP can be expressed as follows:

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} &\sim p(\boldsymbol{r}|a_k)P\{a_k\} \\ &= \sum_{\mu_k} p(\boldsymbol{r}|a_k,\mu_k)P\{\mu_k|a_k\}P\{a_k\} \\ &= \sum_{\mu_k} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k,a_k,\mu_k)p(r_k|\boldsymbol{r}_0^{k-1},a_k,\mu_k)p(\boldsymbol{r}_0^{k-1}|a_k,\mu_k) \\ &\quad \cdot P\{\mu_k|a_k\}P\{a_k\}. \end{aligned} \tag{4.8}$$

Owing to the independence between the information symbols, since μ_k may depend on $\boldsymbol{a}_0^{k-1}$, it follows that

$$P\{\mu_k|a_k\} = P\{\mu_k\}. \tag{4.9}$$

Upon the assumption of transmission over a memoryless channel, the remaining conditional probability density functions (pdfs) in (4.8) can be simplified as

$$p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k,a_k,\mu_k) = p(\boldsymbol{r}_{k+1}^{K-1}|\mu_{k+1} = \mathrm{ns}(a_k,\mu_k)) \tag{4.10}$$

$$p(r_k|\boldsymbol{r}_0^{k-1},a_k,\mu_k) = p(r_k|a_k,\mu_k) \tag{4.11}$$

$$p(\boldsymbol{r}_0^{k-1}|a_k,\mu_k) = p(\boldsymbol{r}_0^{k-1}|\mu_k). \tag{4.12}$$

Hence, the APP in (4.8) can be rewritten as follows:

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} \sim \sum_{\mu_k} & p(\boldsymbol{r}_{k+1}^{K-1}|\mu_{k+1} = \mathrm{ns}(a_k,\mu_k)) \\ & \cdot p(r_k|a_k,\mu_k)p(\boldsymbol{r}_0^{k-1}|\mu_k)P\{\mu_k\}P\{a_k\}. \end{aligned} \tag{4.13}$$

By defining

$$\alpha_k(\mu_k) \triangleq p(\boldsymbol{r}_0^{k-1}|\mu_k)P\{\mu_k\} \tag{4.14}$$

$$\gamma_k(a_k,\mu_k) \triangleq p(r_k|a_k,\mu_k)P\{a_k\} \tag{4.15}$$

$$\beta_{k+1}(\mu_{k+1}) \triangleq p(\boldsymbol{r}_{k+1}^{K-1}|\mu_{k+1}) \tag{4.16}$$

the desired symbol APP finally becomes

$$P\{a_k|\boldsymbol{r}\} \sim \sum_{\mu_k} \alpha_k(\mu_k)\gamma_k(a_k,\mu_k)\beta_{k+1}(\mu_{k+1}) \tag{4.17}$$

where, for the sake of notational simplicity, the dependence of μ_{k+1} on μ_k and a_k is not explicitly indicated.[3] In the following, since the generated soft-output value is a

[3]This simplifying notational assumption (and other similar assumptions) will also be considered later in the book. The context should eliminate any ambiguity.

quantity monotonically related to (or approximating) the APP, we will indicate this value with the general notation $\mathrm{S}[a_k]$. In other words, one can write:

$$\mathrm{S}[a_k] \triangleq \sum_{\mu_k} \alpha_k(\mu_k)\gamma_k(a_k, \mu_k)\beta_{k+1}(\mu_{k+1}). \tag{4.18}$$

Note that the operation (4.18), where the quantities $\{\alpha_k(\mu_k)\}$ and $\{\beta_{k+1}(\mu_{k+1})\}$ are combined to generate the extrinsic information, is usually referred to as *completion*.

The quantities $\alpha_k(\mu_k)$ and $\beta_{k+1}(\mu_{k+1})$ can be computed by means of forward and backward recursions, respectively. More precisely, one can write:

$$\begin{aligned}
\alpha_k(\mu_k) &= p(\boldsymbol{r}_0^{k-1}|\mu_k)P\{\mu_k\} \\
&= \sum_{\substack{(\mu_{k-1}, a_{k-1}):\\ \mathrm{ns}(a_{k-1}, \mu_{k-1}) = \mu_k}} p(\boldsymbol{r}_0^{k-1}|a_{k-1}, \mu_{k-1}, \mu_k)P\{a_{k-1}, \mu_{k-1}|\mu_k\}P\{\mu_k\} \\
&= \sum_{\substack{(\mu_{k-1}, a_{k-1}):\\ \mathrm{ns}(a_{k-1}, \mu_{k-1}) = \mu_k}} p(r_{k-1}|\boldsymbol{r}_0^{k-2}, a_{k-1}, \mu_{k-1}, \mu_k) \\
&\quad \cdot p(\boldsymbol{r}_0^{k-2}|a_{k-1}, \mu_{k-1}, \mu_k)P\{a_{k-1}, \mu_{k-1}, \mu_k\} \\
&= \sum_{\substack{(\mu_{k-1}, a_{k-1}):\\ \mathrm{ns}(a_{k-1}, \mu_{k-1}) = \mu_k}} p(r_{k-1}|\boldsymbol{r}_0^{k-2}, a_{k-1}, \mu_{k-1}, \mu_k) \\
&\quad \cdot p(\boldsymbol{r}_0^{k-2}|a_{k-1}, \mu_{k-1}, \mu_k)P\{\mu_k|a_{k-1}, \mu_{k-1}\}P\{a_{k-1}|\mu_{k-1}\}P\{\mu_{k-1}\}
\end{aligned} \tag{4.19}$$

where the index of the summation indicates all possible transitions $\{(\mu_{k-1}, a_{k-1})\}$ compatible, through the next-state function, with μ_k—this notation is general and accounts also for the case of underlying recursive and nonrecursive FSM models. The summation in (4.19) can be re-interpreted as being carried over all possible trellis branches $\{t_{k-1}\}$ compatible with the final state μ_k. Since we are considering possible combinations of μ_{k-1} and a_{k-1} compatible with μ_k, it follows that

$$P\{\mu_k|a_{k-1}, \mu_{k-1}\} = 1. \tag{4.20}$$

Based on the independence between the information symbols and on the absence of channel memory, one can write:

$$p(r_{k-1}|\boldsymbol{r}_0^{k-2}, a_{k-1}, \mu_{k-1}, \mu_k) = p(r_{k-1}|\mu_{k-1}, a_{k-1}) \tag{4.21}$$

$$p(\boldsymbol{r}_0^{k-2}|a_{k-1}, \mu_{k-1}, \mu_k) = p(\boldsymbol{r}_0^{k-2}|\mu_{k-1}) \tag{4.22}$$

$$P\{a_{k-1}|\mu_{k-1}\} = P\{a_{k-1}\}. \tag{4.23}$$

Finally, a step in the forward recursion in (4.19) can be concisely expressed as follows:

$$\begin{aligned}\alpha_k(\mu_k) &= \sum_{t_{k-1}:\mu_k} p(\boldsymbol{r}_0^{k-2}|\mu_{k-1})P\{\mu_{k-1}\}p(r_{k-1}|\mu_{k-1},a_{k-1})P\{a_{k-1}\} \\ &= \sum_{t_{k-1}:\mu_k} \alpha_{k-1}(\mu_{k-1})\gamma_{k-1}(t_{k-1}). \end{aligned} \tag{4.24}$$

A similar derivation holds also for the backward recursion. More precisely, one can write:

$$\begin{aligned}\beta_k(\mu_k) &= p(\boldsymbol{r}_k^{K-1}|\mu_k) \\ &= \sum_{a_k} p(\boldsymbol{r}_k^{K-1}|a_k,\mu_k)P\{a_k|\mu_k\} \\ &= \sum_{a_k} p(r_k|\boldsymbol{r}_{k+1}^{K-1},a_k,\mu_k)p(\boldsymbol{r}_{k-1}^{K-1}|a_k,\mu_k)P\{a_k|\mu_k\}. \end{aligned} \tag{4.25}$$

Based on the independence between information symbols and the absence of memory of the considered transmission channel, the following simplifications hold:

$$p(r_k|\boldsymbol{r}_{k+1}^{K-1},a_k,\mu_k) = p(r_k|a_k,\mu_k) \tag{4.26}$$

$$p(\boldsymbol{r}_{k-1}^{K-1}|a_k,\mu_k) = p(\boldsymbol{r}_{k-1}^{K-1}|\mu_{k+1}=\mathrm{ns}(a_k,\mu_k)) \tag{4.27}$$

$$P\{a_k|\mu_k\} = P\{a_k\}. \tag{4.28}$$

A step in the backward recursion, as indicated in (4.25), can be rewritten as follows:

$$\begin{aligned}\beta_k(\mu_k) &= \sum_{a_k} p(\boldsymbol{r}_{k-1}^{K-1}|\mu_{k+1})p(r_k|a_k,\mu_k)P\{a_k\} \\ &= \sum_{a_k} \beta_{k+1}(\mu_{k+1})\gamma_k(t_k). \end{aligned} \tag{4.29}$$

Note that, unlike the forward recursion, in the backward recursion there is no ambiguity in the summation index. In fact, the trellis diagram corresponding to any possible coding/modulation scheme, either recursive or nonrecursive, is such that branches exiting from a state are associated with different information symbols.

Given the inherent complexity of the FB algorithm, a number of suboptimal approximate versions have been proposed in the literature. Among the various approximations, we consider one which arises naturally once the algorithm is formulated in the metric (or logarithmic) domain. Unlike sequence detection, taking the natural

logarithm of (4.18), (4.24), and (4.29) would not seem very helpful because the logarithm does not commute with the summations. However, given two values x and y, and noting that for large $|x - y|$

$$\ln(e^x + e^y) \simeq \max(x, y) \tag{4.30}$$

and, consequently, that

$$\ln(e^{x_1} + e^{x_2} + \cdots + e^{x_n}) \simeq \max(x_1, x_2, \ldots, x_n) \tag{4.31}$$

it is possible to formulate a computationally efficient approximation of the FB algorithm. This formulation of the FB algorithm is usually referred to either as *max-log* [137] or *min-sum* [98]. The following definitions are expedient:[4]

$$\Lambda_k^{\mathrm{F}}(\mu_k) \triangleq \ln \alpha_k(\mu_k) \tag{4.32}$$

$$\lambda_k(a_k, \mu_k) \triangleq \ln \gamma_k(a_k, \mu_k) \tag{4.33}$$

$$\Lambda_k^{\mathrm{B}}(\mu_k) \triangleq \ln \beta_k(\mu_k). \tag{4.34}$$

In the metric domain, the completion operation in (4.18) therefore becomes:

$$\begin{aligned} \ln P\{a_k|\boldsymbol{r}\} &\sim \ln \sum_{\mu_k} e^{\ln \alpha_k(\mu_k) + \ln \gamma_k(a_k, \mu_k) + \ln \beta_k(\mu_k)} \\ &= \ln \sum_{\mu_k} e^{\Lambda_k^{\mathrm{F}}(\mu_k) + \lambda_k(a_k, \mu_k) + \Lambda_k^{\mathrm{B}}(\mu_k)} \\ &\simeq \max_{\mu_k} \left[\Lambda_k^{\mathrm{F}}(\mu_k) + \lambda_k(a_k, \mu_k) + \Lambda_k^{\mathrm{B}}(\mu_k)\right] \end{aligned} \tag{4.35}$$

where, in the last step, the max-log approximation has been used. Similarly, the two

[4]Note that the second definition derives naturally from the definitions of the branch metric $\lambda_k(a_k, \mu_k)$ in Chapter 2 and $\gamma_k(a_k, \mu_k)$. In particular, γ_k can be interpreted as an "exponential metric."

recursions can be written as follows:

$$\begin{aligned}
\Lambda_k^{\mathrm{F}}(\mu_k) = \ln \alpha_k(\mu_k) &= \ln \sum_{t_{k-1}:\mu_k} e^{\ln \alpha_{k-1}(\mu_{k-1}) + \ln \gamma_k(t_{k-1})} \\
&= \ln \sum_{t_{k-1}:\mu_k} e^{\Lambda_{k-1}^{\mathrm{F}}(\mu_{k-1}) + \lambda_{k-1}(t_{k-1})} \\
&\simeq \max_{t_{k-1}:\mu_k} \left[\Lambda_{k-1}^{\mathrm{F}}(\mu_{k-1}) + \lambda_{k-1}(t_{k-1})\right] \qquad (4.36) \\
\Lambda_k^{\mathrm{B}}(\mu_k) = \ln \beta_k(\mu_k) &= \sum_{a_k} e^{\ln \beta_{k+1}(\mu_{k+1}) + \ln \gamma_k(a_k, \mu_k)} \\
&= \ln \sum_{a_k} e^{\Lambda_{k+1}^{\mathrm{B}}(\mu_{k+1}) + \lambda_k(a_k, \mu_k)} \\
&\simeq \max_{a_k} \left[\Lambda_{k+1}^{\mathrm{B}}(\mu_{k+1}) + \lambda_k(a_k, \mu_k)\right]. \qquad (4.37)
\end{aligned}$$

The "beauty" of the max-log approximation is that forward and backward recursions in the metric domain, given by (4.36) and (4.37), respectively, can be implemented by two VAs and the two quantities $\Lambda_k^{\mathrm{F}}(\mu_k)$ and $\Lambda_k^{\mathrm{B}}(\mu_k)$ can be interpreted directly as forward and backward path metrics associated with state μ_k, respectively. Specifically, the first VA is run forward, with usual branch metric $\lambda_k(t_k)$, to compute the forward path metrics $\{\Lambda_k^{\mathrm{F}}(\mu_k)\}$, whereas the second VA is run backward, still with branch metric $\lambda_k(t_k)$, to compute the backward path metrics $\{\Lambda_k^{\mathrm{B}}(\mu_k)\}$. Finally, the forward and backward path metrics are combined with the branch metrics and a comparison is performed to determine the APPs, or the symbol decisions, according to (4.35).

The computational efficiency of the max-log FB algorithm obviously comes at the price of a slight performance degradation; hence, a number of different approximations have been studied as intermediate solutions between the full-complexity optimal FB algorithm and its max-log approximation. For conciseness, we do not pursue this topic further and refer the interested reader to the existing literature [98, 137]. The max-log approximation of the FB algorithm is sufficient for our purposes because of its direct interpretation in terms of VAs running in direct and inverse time directions.

4.4 Iterative Decoding and Detection

The concept of *iterative detection* is a natural extension of the concept of *iterative decoding*. The latter, originally introduced by Gallager in his Ph.D. thesis [31], was crystallized by Berrou and Glavieux in 1993 with the introduction of *turbo codes*

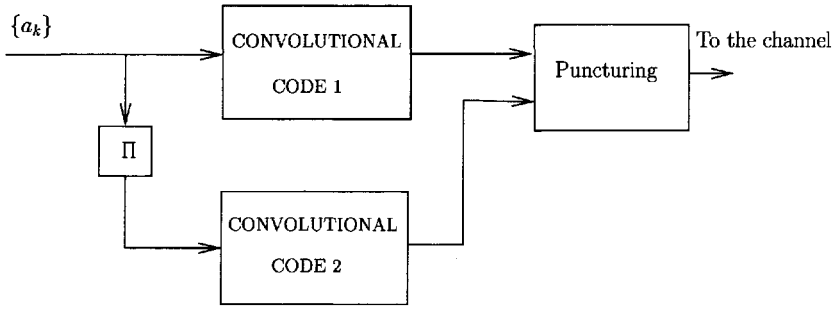


Figure 4.2: Parallel concatenated convolutional code, or *turbo code*.

and the concept of iterative decoding [33, 66]. In this revolutionary work, the authors showed that a complicated code, with a particular structure, can be decoded efficiently with limited complexity. In particular, they considered a parallel concatenated convolutional code (PCCC), constituted by the parallel concatenation, through interleaving, of two convolutional codes.

The structure of a parallel concatenated convolutional code is shown in Figure 4.2, where the interleaver is indicated with the symbol π. As shown in Figure 4.2, the output sequences from the two component convolutional codes can be punctured in order for the overall code to have the desired code rate.

The receiver is based on two component decoders corresponding to the two constituent convolutional encoders. In Figure 4.3, the basic structure of a turbo decoder, corresponding to the encoder in Figure 4.2, is shown. The two component decoders exchange *soft information* between each other, and it can thus be obtained by using soft-output algorithms at each component decoder. More precisely, the decoders exchange *extrinsic information*, which represents the component of the generated soft output on a symbol not depending on the soft-input information on the same symbol [33]. Referring to the FB algorithm formulation proposed in Section 4.3, one can immediately recognize that the final soft-output quantity (4.18), can be always written as follows:

$$\mathrm{S}[a_k] = P\{a_k\}\mathrm{S}^{(E)}[a_k] \tag{4.38}$$

where

$$\mathrm{S}^{(E)}[a_k] \triangleq \sum_{\mu_k} \alpha_k(\mu_k)\gamma'_k(a_k, \mu_k)\beta_{k+1}(\mu_{k+1}) \tag{4.39}$$

and

$$\gamma'_k(a_k, \mu_k) \triangleq p(r_k|a_k, \mu_k) = \frac{\gamma_k(a_k, \mu_k)}{P\{a_k\}}. \tag{4.40}$$

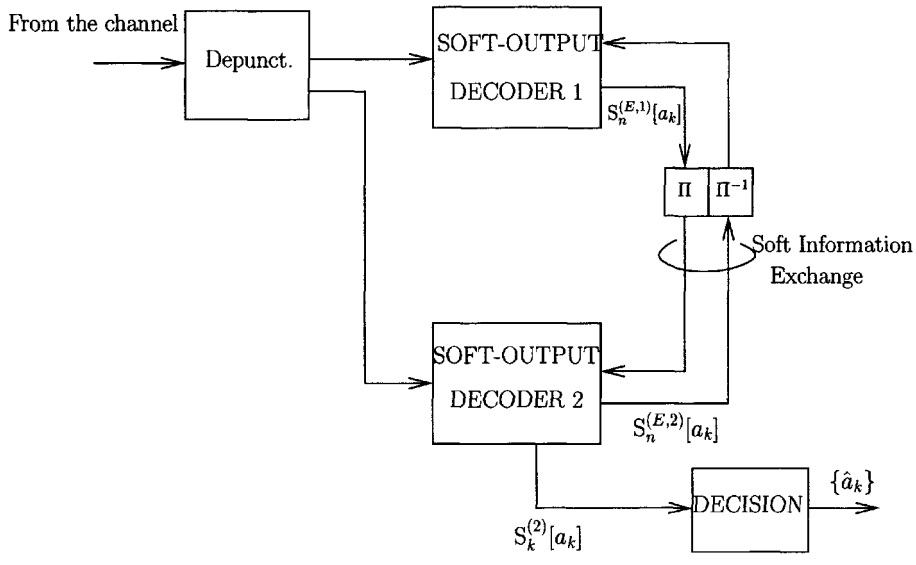


Figure 4.3: Turbo decoder for a PCCC.

In Figure 4.3, the extrinsic information values on information symbol a_k generated by the first and second decoders in the n-th iteration are denoted by $S_n^{(E,1)}[a_k]$ and $S_n^{(E,2)}[a_k]$, respectively. Note that the *a priori* probability $P\{a_k\}$ of an information symbol used by each component decoder is given, in the iterative decoding process, by the extrinsic information generated by the other decoder. Assuming, as a convention, that an iteration is constituted by the sequence of decoding acts of the first and second component decoders, the soft-output values generated by each component decoder can be rewritten as follows:

$$S_n^{(1)}[a_k] = \underbrace{S_{n-1}^{(E,2)}[a_k]}_{\text{input}} \underbrace{S_n^{(E,1)}[a_k]}_{\text{output}} \tag{4.41}$$

$$S_n^{(2)}[a_k] = \underbrace{S_n^{(E,1)}[a_k]}_{\text{input}} \underbrace{S_n^{(E,2)}[a_k]}_{\text{output}}. \tag{4.42}$$

In other words, the soft-output value generated by the first decoder at the n-th iteration is the product of the soft value at its input, corresponding to the extrinsic information generated by the second decoder at the $(n-1)$-th iteration, and the generated extrinsic information. The soft-output value generated by the second decoder at the n-th iteration is the product of the soft value at its input, corresponding to the extrinsic information generated by the first decoder at the same iteration, and the generated extrinsic information. The two soft-output values in (4.41) and (4.42) indicate

clearly that the soft-output decoding processes (based on the FB algorithm) in the two component decoders are *coupled*. A *decoding iteration* is basically constituted by four steps:

1. The first decoder generates the extrinsic information on an information symbol a_k by using the extrinsic information generated by the second decoder—note that at the very first iteration the first decoder cannot rely on the extrinsic information generated by the second decoder.

2. The extrinsic information sequence generated by the first decoder is interleaved and passed to the second decoder.

3. The second decoder, upon receiving the sequence of soft reliability values generated by the first decoder, generates a sequence of extrinsic information values.

4. The extrinsic information sequence generated by the second decoder is deinterleaved and passed to the first decoder. At this point, a single iteration ends, and a new iteration can start.

The concept of extrinsic information is very subtle. This information can be interpreted as the "surplus" of information, on the reliability of a symbol, generated by a component decoder. The extrinsic information value of a symbol at epoch k is unbiased with respect to the corresponding soft-input information at the same epoch.[5] In this way, considering the turbo decoder scheme shown in Figure 4.3, each decoder receives from the other decoder a "new" suggestion regarding the reliability of a symbol. This is fundamental for the iterative decoding process to *converge* to a correct decision. In fact, suppose that the first decoder does not decode correctly symbol a_k, i.e., it attributes to this symbol a wrong reliability value—this means that the extrinsic information value corresponding to a_k suggests that a specific wrong realization is more likely than the others. Upon receiving this wrong reliability value, if the second decoder passed back to the first decoder a "complete" soft-output information $\mathrm{S}^{(2)}[a_k]$, then the latter decoder would be heavily influenced by previous decisions of the former decoder. In other words, the second decoder would "push" the first decoder to decide in the same way. Instead, if the second decoder manages to generate a correct surplus of information (i.e., the extrinsic information is a reliability value consistent with the correct information symbol a_k), then this surplus of information can positively influence the first decoder, which can decide in the right direction in the following iterations.

[5]Note that the extrinsic information at epoch k depends, however, on all soft-input information values at epochs different from k.

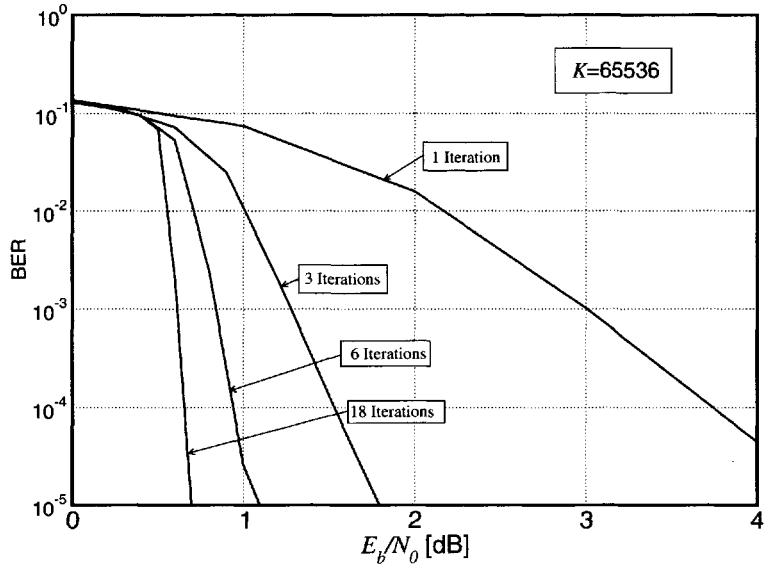


Figure 4.4: Typical BER performance curves of a turbo code for an increasing number of iterations. Reprinted from [33], ©IEEE, by permission of the IEEE.

Note that performing MAP sequence or symbol decoding of the entire turbo code, seen as a *single* FSM, is practically impossible, since the presence of the interleaver makes the overall FSM possess a huge number of states. Hence, nobody knows exactly the ultimate BER performance of a turbo code, but various analyses suggest that it is reasonable to assume that the performance obtained with iterative decoding basically approaches that obtained performing sequence decoding of the overall code. Typical BER curves for iterative decoding of a parallel concatenated code are shown in Figure 4.4. In particular, the BER curves shown in Figure 4.4 refer to the turbo code originally introduced in [66], which is a rate-1/2 turbo code, with constituent 16-state recursive systematic convolutional (RSC) codes with generators (in octal notation) $G_1 = 37$, $G_2 = 21$, and interleaving 256×256 (i.e., the transmitted information sequence has length $K = 65\,536$) as described in [33]. The considered number of iterations increases from 1 to 18. One can immediately recognize a characteristic *waterfall* effect of the BER curves, as the number of decoding iterations increases above a specific *threshold* value of the signal-to-noise ratio (SNR). In the particular case considered in Figure 4.4, the threshold SNR value is around 0.7 dB. In other words, if the SNR value at the channel output (i.e., at the input of the turbo decoder) is larger than this value, then the iterative decoding process converges, in the sense that the number of errors (on the eventually decided symbol) rapidly re-

duces to a very small value. If, on the other hand, the SNR is below this value, then the iterative decoding process does not converge and the BER remains unacceptable.

In the light of the previous interpretation of the iterative decoding process, the existence of a threshold in terms of SNR can be further interpreted as follows. If the SNR is too low, then the first decoder, at the very first iteration, passes completely wrong reliability values to the second decoder. The latter decoder is "pushed too hard" in the wrong direction by the first decoder, and it passes back soft-output values which aim in the same wrong decision direction. The successive iterations worsen the situation, so that the final decisions made by the iterative decoder are completely wrong.

In the last few years, in order to avoid long computer simulations with numerical problems, new techniques have been developed to evaluate the performance of concatenated codes decoded by iterative decoding techniques. In particular, as indicated in the previous paragraph, a figure of interest may be the threshold SNR value above which there is convergence, i.e., above which the BER rapidly decreases to zero for an increasing number of iterations. Among these techniques, we recall two interesting ones.

- A first technique is based on *density evolution* [84]. According to this approach, the generated extrinsic information is associated with a pdf, and the evolution of this pdf over successive decoding iterations is tracked—this is related to the approach proposed in Section 4.5.

- A second technique is based on the extrinsic information transfer (EXIT) charts [86, 87], which characterize the input output SNR relation for a considered code. For example, comparing the EXIT curves of both component convolutional encoders of a turbo code allows determination of the value of the SNR threshold and the speed of the iterative decoding process.

In the following Section 4.5, we propose a different (and simpler) approach to analyzing the iterative decoding process on the basis of the mean value and variance of the generated extrinsic information. In particular, it will be shown that simply scaling the generated soft-output information, whenever produced by algorithms introducing suboptimal approximations, may lead to substantial performance improvements. As an example, this will be the case using state reduction techniques.

4.5 Extrinsic Information in Iterative Decoding: a Unified View

As described in the previous sections, the principle of iterative decoding is based on the exchange, between component subdecoders, of *soft information* [33, 138]. More precisely, in [33, 138] the original concept of *extrinsic information* was introduced to identify the component of the reliability value generated by a decoding subsystem, relative to a specific input or output variable, which does not depend on the input soft information on the same variable.[6]

In this section, we analyze the extrinsic information, showing that simply scaling, in the logarithmic domain, the soft information generated by each decoder can improve in some cases the overall performance. For the sake of simplicity, we will refer to the case of transmission over a memoryless channel. The same conclusions hold, however, also in the case of iterative detection over channels with memory, as will be shown in the following sections. We consider, as examples, a PCCC, or turbo code, and a serially concatenated convolutional code (SCCC) transmitted over an additive white Gaussian noise (AWGN) channel. In particular, we consider as component algorithms at the receiver side the FB algorithm, and the soft-output VA (SOVA) [99, 139]. In both cases, we derive the two algorithms in the logarithmic domain. In fact, a natural reliability value, in the binary case,[7] is the *logarithmic likelihood ratio* (LLR), defined as

$$\mathrm{LLR}[a_k] \triangleq \ln \frac{P\{a_k = +1|\mathrm{inputs}\}}{P\{a_k = -1|\mathrm{inputs}\}} \tag{4.43}$$

where the word "inputs" refers to all decoder inputs. We remark that we consider a binary input a_k belonging to the set $\{-1, +1\}$, in a one-to-one correspondence to the set $\{0, 1\}$. We consider this notation because it will make the following derivation simpler. The LLR can be exactly computed employing the FB algorithm, which allows one to calculate the APPs $P\{a_k = i|\mathrm{inputs}\}$, $i \in \{\pm 1\}$ [54]. As we have seen, the FB algorithm is the optimal algorithm to generate the sequence of APPs, but its computational complexity is large with respect to that of the VA. Besides "hard" symbol decisions, SOVA provides a symbol reliability information which can be interpreted as an approximation of the LLR [33, 139, 140].

For both the FB algorithm and SOVA, in the literature there exist essentially two methods to process the extrinsic information at the input of a soft-output decoder.

[6] In [138], the extrinsic information is referred to as "refi nement factor."

[7] In this section, we limit our attention to the case of binary information symbols. The proposed derivation can, however, be extended to the case of information symbols with larger cardinality.

1. In a first method, the extrinsic information at the input of a decoder is modeled as the output of an AWGN *meta-channel* [33, 141, 142].

2. In a second method, the extrinsic information is used to update the *a priori* probabilities used in the next decoding step, in the sense that the APPs computed by a decoder become *a priori* probabilities for the other one [73, 143, 144]—this method was discussed in Section 4.4 when introducing the concept of iterative decoding.

In this section, we present a unified interpretation of these two methods and emphasize their commonalities and differences. More precisely, we show that, either using the FB algorithm or SOVA, the two methods only differ for a multiplicative factor used in the metric computation. When the input is modeled as a Gaussian random variable, this multiplicative factor depends on the variance and mean of the sequence of LLRs, whereas, in the case of extraction of the *a priori* probabilities, it is a constant equal to 1/2. We finally consider the use of an heuristic multiplicative parameter for both algorithms and evaluate the performance of the considered decoding schemes for various values of this parameter.

4.5.1 A Review of the Use of the Extrinsic Information

As mentioned in Section 4.4, decoding of parallel and serially concatenated codes is based on a suboptimal iterative processing in which each component decoder takes advantage of the extrinsic information produced by the other decoder at the previous time [33]. This iterative decoding process is made possible by employing soft-output component decoders. As an example, for the rate-1/2 turbo code described in [33], the turbo decoder is shown in Figure 4.5—we recall that this turbo code is composed of two rate-1/2 RSC codes, with puncturing on the coded symbols generated by the two encoders. In the figure: blocks Π and Π^{-1} denote the interleaver and deinterleaver, respectively; $\{r_k^{(j)}\}$, $j = 1, 2$, denote the channel output sequences; and $\{z_k^{(j)}\}$, $j = 1, 2$, denote the extrinsic information sequence at the input of the j-th soft-output decoder (i.e., produced by the other one). This sequence is derived, by means of an interleaver or a deinterleaver, from the sequences $\{w_k^{(j)}\}$, $j = 1, 2$, produced by the other component decoder. Obviously, a serial concatenated decoder presents a serial concatenation, instead of a parallel concatenation, of two component decoders. In particular, while in a parallel decoder at the input of each component subdecoder there are channel observations, in a serial decoder only the inner subdecoder is directly connected to the channel.

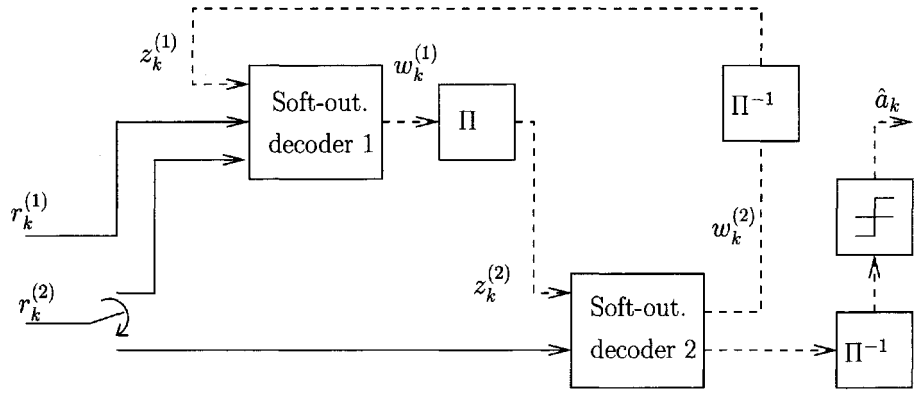


Figure 4.5: Decoder for a turbo code of rate 1/2.

In this subsection, we describe the possible methods for the use of the extrinsic information at the input of each decoder. To this purpose we consider, without loss of generality, a soft-output decoder which receives a sequence $\{r_k\}$ of channel outputs and a sequence $\{z_k\}$ of extrinsic information values generated by another decoder and produces a sequence $\{w_k\}$ of soft-output values—this is the case for decoder 2 in Figure 4.5, but may be easily generalized to the other decoder with an extended vector notation for r_k. Moreover, we assume that the input sequence $\{z_k\}$ and the generated sequence $\{w_k\}$ are both relative to the sequence $\{a_k\}$ of information symbols. The proposed formulation can be generalized computing the extrinsic information relative to the code symbols $\{c_k\}$ if the relevant soft outputs are needed, besides those for information symbols, as in the case of nonsystematic codes. In [83, 138], this generalization is carried out considering the FB algorithm; however, an extension to SOVA is straightforward. For simplicity, we consider an RSC code (usually the component code of a turbo code and the inner code of a serially concatenated code), in which case the extrinsic information on an information symbol needs to be computed.

We assume that a channel output, after suitable filtering and sampling, can be expressed as

$$r_k = c_k + n_k \tag{4.44}$$

where $\{n_k\}$ is a sequence of independent, zero-mean, real Gaussian random variables, with variance σ^2. In (4.44) we assume that $c_k \in \{\pm 1\}$, i.e., c_k represents a symbol modulated with binary phase shift keying (BPSK). In the original paper on turbo codes and turbo decoding [33], the input sequence $\{z_k\}$, i.e., the extrinsic information relative to the information sequence $\{a_k\}$, is interpreted as the output of a

Gaussian meta-channel. Specifically, it is assumed that

$$z_k = \eta_z a_k + n'_k \tag{4.45}$$

where the information symbols $\{a_k\}$ belong to the binary alphabet $\{\pm 1\}$, $\{n'_k\}$ are independent, zero-mean, real Gaussian random variables, with variance σ_z^2 and $\eta_z \triangleq |\mathrm{E}\{z_k|a_k\}|$. In [33], it is observed that the Gaussian assumption, even if not satisfied in the initial iterations, is a good approximation when the number of iterations increases. The values of η_z and σ_z^2 are estimated for each data block. The Gaussian assumption has acquired a new and more powerful interpretation in recent works appearing in the literature on the analysis of the convergence of iterative decoding based on the concept of *density evolution* [84], where the term density refers to the pdf of the extrinsic information, interpreted as a Gaussian random variable.

An alternative method, which does not require an estimation of η_z and σ_z^2, is proposed in [73, 143]. In this case, the extrinsic information z_k at the input of the considered decoder is used to extract a new estimate of the *a priori* probabilities to be employed in the next decoding step. In fact, each decoder interprets z_k as an approximation of the LLR of the *a priori* probabilities according to

$$z_k \simeq \ln \frac{P\{a_k = +1\}}{P\{a_k = -1\}} \tag{4.46}$$

which allows one to derive (taking into account that $P\{a_k = +1\} + P\{a_k = -1\} = 1$)

$$\begin{aligned} P\{a_k = +1\} &\simeq \frac{e^{z_k}}{1 + e^{z_k}} \\ P\{a_k = -1\} &= 1 - P\{a_k = +1\} \simeq \frac{1}{1 + e^{z_k}} . \end{aligned} \tag{4.47}$$

Therefore, the APPs generated by a decoder are used as *a priori* probabilities by the other one. This approach can be generalized. In fact, it is possible to consider a *normalization* of a generic soft-input information $\mathrm{SI}[a_k]$ [98]. For example, defining $\mathrm{SI}[a_k] \triangleq P\{a_k\}$, i.e., defining the soft-input quantity as the "ideal" *a priori* probability, and $\mathrm{MI}[a_k] \triangleq \ln \mathrm{SI}[a_k]$, one can rewrite z_k as

$$z_k = \ln P\{a_k = +1\} - \ln P\{a_k = -1\} = \mathrm{MI}[a_k = 1] - \mathrm{MI}[a_k = -1]. \tag{4.48}$$

One can consider a normalization such that $\mathrm{MI}[a_k = -1] = 0\,, \forall k$, and therefore write

$$\mathrm{MI}[a_k] = \begin{cases} z_k & \text{if } a_k = 1 \\ 0 & \text{if } a_k = 0. \end{cases} \tag{4.49}$$

In the following, we will consider the alternative normalization expressed as

$$\mathrm{MI}[a_k] = \begin{cases} z_k/2 & \text{if } a_k = 1 \\ -z_k/2 & \text{if } a_k = 0. \end{cases} \tag{4.50}$$

It is easy to show that any *shift* in the logarithmic domain, i.e., adding a constant value to all soft inputs epoch by epoch, creates an equivalent set of soft measures [98]. In the binary case, what really matters is the difference, in the logarithmic domain, between the soft values associated with $+1$ and -1. This is why the LLR "captures," as a single quantity, all the soft information relative to a binary symbol a_k. Generalizing, if M-ary symbols are considered, instead of considering M different soft values it is sufficient to consider $(M-1)$ LLRs.

4.5.2 Forward Backward Algorithm

In Section 4.3, we proposed a simple derivation of the FB algorithm for the computation of soft-output reliability values. In order to simplify the comparison of the FB algorithm with the SOVA, we limit our attention to the case of binary information symbols. Considering the formulation proposed in [33]—slightly different from, but equivalent to, the one we considered in Section 4.3—one can immediately show that the LLR can be written as follows:

$$\mathrm{LLR}[a_k] = \ln \frac{\sum_m \sum_{m'} \gamma_k(+1, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_k(-1, m', m)\alpha_{k-1}(m')\beta_k(m)}. \tag{4.51}$$

The pdfs $\gamma_k(i, m', m)$, $i \in \{\pm 1\}$, are defined as

$$\begin{aligned} \gamma_k(i, m', m) &\triangleq p(R_k | a_k = i, \mu_k = m, \mu_{k-1} = m') \\ &\cdot P\{a_k = i | \mu_k = m, \mu_{k-1} = m'\} P\{\mu_k = m | \mu_{k-1} = m'\} \end{aligned} \tag{4.52}$$

where: (i) $R_k = (r_k, z_k)$ if z_k is interpreted as the output of a Gaussian meta-channel, or (ii) $R_k = r_k$ if z_k is used to update the *a priori* probabilities; $P\{a_k = i | \mu_k = m, \mu_{k-1} = m'\}$ is either one or zero depending on whether bit i is or is not associated with the transition from state m' to state m, respectively; and $P\{\mu_k = m | \mu_{k-1} = m'\}$ is the transition probability.

As we have previously seen, the pdfs $\{\alpha_k(m)\}$ and $\{\beta_k(m)\}$ can be calculated using forward and backward recursions, respectively. As an example, using the notation introduced in this subsection, a generic step in the forward recursion for the

computation of $\alpha_k(m)$ can be written as follows:

$$\alpha_k(m) = \frac{\sum_{m'}\sum_{i=0}^{1}\gamma_k(i,m',m)\alpha_{k-1}(m')}{\sum_{m''}\sum_{m'}\sum_{i=0}^{1}\gamma_k(i,m',m'')\alpha_{k-1}(m')} . \tag{4.53}$$

Note that, unlike (4.24), the ensemble $\{\alpha_k(m)\}$ is normalized (with their sum) over all possible states at each epoch. In this case, the quantity $\alpha_k(m)$ can be interpreted as the "state probability." Such normalization has no influence on the value of the LLR, but it is often useful for numerical stability. Moreover, from (4.51) and (4.53), it is obvious that $\gamma_k(i,m',m)$ may be arbitrarily multiplied by any constant independent of i, m' and m.

In the following two subsections, we present the mentioned methods for using the extrinsic information within a unified interpretation. In the third subsection, we propose an heuristic method to treat the extrinsic information. Finally, in the fourth subsection, we present some numerical results relative to iterative decoding of a PCCC and an SCCC using the FB algorithm with the proposed methods for using the extrinsic information.

Extrinsic Information as Gaussian-distributed Input

In this case, the information symbols are assumed independent and identically distributed (iid), i.e., $P\{a_k = +1\} = P\{a_k = -1\} = 1/2$. Hence, $P\{\mu_k = m|\mu_{k-1} = m'\} = 1/2$ for each possible transition. Since $R_k = (r_k, z_k)$ and due to the assumed independence of r_k and z_k, we may write:

$$\begin{aligned} p(R_k|a_k = i, \mu_k = m, \mu_{k-1} = m') &= p(r_k|a_k = i, \mu_k = m, \mu_{k-1} = m') \\ &\quad \cdot p(z_k|a_k = i, \mu_k = m, \mu_{k-1} = m') . \end{aligned} \tag{4.54}$$

Recalling (4.45) and the Gaussian assumption for z_k, we have:

$$\begin{aligned} p(z_k|a_k = i, \mu_k = m, \mu_{k-1} = m') &= p(z_k|a_k = i) \\ &= \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left\{-\frac{(z_k - i\eta_z)^2}{2\sigma_z^2}\right\} \\ &= \frac{1}{\sqrt{2\pi\sigma_z^2}} \exp\left\{-\frac{z_k^2}{2\sigma_z^2} - \frac{\eta_z^2}{2\sigma_z^2}\right\} \\ &\quad \exp\left\{\frac{iz_k\eta_z}{\sigma_z^2}\right\} . \end{aligned} \tag{4.55}$$

Thus, from (4.52) we may express the pdf $\gamma_k(i, m', m)$ used in the forward and backward recursions as

$$\begin{aligned}\gamma_k(i, m', m) &= \delta_k p(r_k|a_k = i, \mu_k = m, \mu_{k-1} = m') \\ &\quad \cdot P\{a_k = i|\mu_k = m, \mu_{k-1} = m'\} \exp\left\{\frac{i z_k \eta_z}{\sigma_z^2}\right\} \end{aligned} \tag{4.56}$$

where δ_k is a suitable constant, independent of i, m' and m.

In this case, the LLR (4.51) can be expressed as

$$\text{LLR}[a_k] = 2\frac{z_k \eta_z}{\sigma_z^2} + w_k \tag{4.57}$$

where w_k is the extrinsic component of the generated LLR defined as

$$w_k \triangleq \ln \frac{\sum_m \sum_{m'} \gamma'_k(+1, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma'_k(-1, m', m)\alpha_{k-1}(m')\beta_k(m)} \tag{4.58}$$

where

$$\gamma'_k(i, m', m) \triangleq \delta_k p(r_k|a_k = i, \mu_k = m, \mu_{k-1} = m') P\{a_k = i|\mu_k = m, \mu_{k-1} = m'\}. \tag{4.59}$$

Extrinsic Information Used to Update the *a Priori* Probabilities

In this case, $R_k = r_k$ and it is easy to show (see Section 4.5.1) that

$$P\{\mu_k = m|\mu_{k-1} = m'\} = \begin{cases} \dfrac{e^{z_k}}{1 + e^{z_k}} & \text{if } P\{a_k = 1|\mu_k = m, \mu_{k-1} = m'\} = 1 \\ \dfrac{1}{1 + e^{z_k}} & \text{if } P\{a_k = -1|\mu_k = m, \mu_{k-1} = m'\} = 1. \end{cases} \tag{4.60}$$

Defining $\rho_k \triangleq e^{\frac{z_k}{2}}(1 + e^{z_k})^{-1}$, one may write:

$$\begin{aligned} P\{\mu_k = m|\mu_{k-1} = m'\} &= \begin{cases} \rho_k \exp\left\{\dfrac{z_k}{2}\right\} \\ \quad \text{if } P\{a_k = 1|\mu_k = m, \mu_{k-1} = m'\} = 1 \\ \rho_k \exp\left\{-\dfrac{z_k}{2}\right\} \\ \quad \text{if } P\{a_k = -1|\mu_k = m, \mu_{k-1} = m'\} = 1 \end{cases} \\ &= \rho_k \exp\left\{\frac{a_k z_k}{2}\right\}. \end{aligned} \tag{4.61}$$

Substituting into (4.52), one obtains:

$$\begin{aligned}\gamma_k(i,m',m) &= \rho_k p(r_k|a_k=i,\mu_k=m,\mu_{k-1}=m') \\ &\quad \cdot P\{a_k=i|\mu_k=m,\mu_{k-1}=m'\}\exp\left\{\frac{iz_k}{2}\right\}. \end{aligned} \tag{4.62}$$

In this case, we can express the LLR (4.51) as

$$\mathrm{LLR}[a_k] = z_k + w_k \tag{4.63}$$

where w_k, the extrinsic component of the generated LLR, is defined as in (4.58), the only difference being the expression for γ'_k, which in this case is

$$\gamma'_k(i,m',m) \triangleq \rho_k p(r_k|a_k=i,\mu_k=m,\mu_{k-1}=m')P\{a_k=i|\mu_k=m,\mu_{k-1}=m'\}. \tag{4.64}$$

This corresponds to the normalization (4.50) for the soft inputs relative to a symbol a_k.

Discussion and Heuristic Method

Based on the above results we may observe that, with the exception of the irrelevant constants δ_k and ρ_k, the two methods presented above differ in the sense that the extrinsic information is weighted by different coefficients. This may be noted by comparing the expressions of the pdfs $\gamma_k(i,m',m)$ (4.52) and (4.62) (the coefficient is η_z/σ_z^2 in the first method and $1/2$ in the second one), and the relations which implicitly define the extrinsic information w_k, i.e., (4.57) and (4.63) (the coefficient is $2\eta_z/\sigma_z^2$ in the first method and 1 in the second one).

Based on this interpretation, an heuristic method can be conceived with the aim of determining an "optimal" weight for the extrinsic information. In this case, the extrinsic information z_k is weighted by a parameter θ to be optimized by trial and error. The performance of the receiver for various values of the parameter θ leads to useful remarks about the way the extrinsic information should be processed when the FB algorithm is used in the component decoders. In particular, in the following, the parameter θ substitutes η_z/σ_z^2 (first method) or 1/2 (second method). It is obvious that multiplication by a constant θ in the logarithmic domain is equivalent to exponentiation in the probability domain.[8] In fact

$$\theta\mathrm{MI}[a_k] = \theta \ln \mathrm{SI}[a_k] = \ln(\mathrm{SI}[a_k])^\theta. \tag{4.65}$$

[8]This aspect will be considered again in Section 4.8.5, where generalized reduced-state FB algorithms will be introduced.

We now make another comment, regarding *density evolution* analysis of the convergence of iterative decoding techniques [84, 145, 146]. In this case, a fictitious SNR for the extrinsic information, interpreted as a random variable, is defined as $\eta_z^2/\sigma_z^2 = \eta_z \frac{\eta_z}{\sigma_z^2}$. In particular, if the extrinsic information has a pdf which is Gaussian and *symmetric*,[9] then $\eta_z/\sigma_z^2 = 1/2$. As shown in the following numerical results, the approach where the extrinsic information is interpreted as a Gaussian random variable will prove asymptotically optimal if $\eta_z/\sigma_z^2 \to 1/2$. A possible interpretation of the obtained result is that the Gaussian approach, when asymptotically optimal, satisfies the symmetry condition.

Numerical Results

The performance achieved by the FB algorithm in the proposed iterative decoding schemes is assessed for the classical turbo code of rate-1/2, 16-state RSC constituent codes with generators $G_1 = (37)_8$, $G_2 = (21)_8$ (octal notation), and 256×256 nonuniform interleaver described in [33], and for an SCCC of rate 1/4, outer 4-state nonrecursive nonsystematic convolutional (NRNSC) code with generators $G_1 = (7)_8$, $G_2 = (5)_8$ and inner 4-state RSC code with generators $G_1 = (5)_8$, $G_2 = (7)_8$ and 64×64 nonuniform interleaver [72].[10] We refer to the method in which the extrinsic information is assumed Gaussian, as the *first method*; similarly, the method in which the extrinsic information is used to update the *a priori* probabilities, and the heuristic method described in Section 4.5.2 are referred to as the *second method* and *third method*, respectively. The same formalism will be adopted when considering SOVA as the component decoding algorithm. We first consider the performance of the turbo code and then the performance of the serially concatenated code. In the following simulation results, the performance is expressed in terms of BER versus the bit SNR E_b/N_0, E_b being the received signal energy per information bit and N_0 the one-sided noise power spectral density.

In Figure 4.6, the BER performance relative to the FB algorithm is shown for various numbers of iterations. It may be observed that the second method, in which the extrinsic information is used to update the *a priori* probabilities, gives better performance (i.e., the extrinsic information is better used) than the first method, which models the extrinsic information as a Gaussian-distributed random variable. Specifically, the second method exhibits a BER of 10^{-5} for an SNR value equal to approximately 0.7 dB. Moreover, the third (heuristic) method does not give any improvement. In

[9] Indicating by $f(\lambda)$ the pdf of the extrinsic information, the symmetry condition is satisfied if $\lambda = \log[f(\lambda)/f(-\lambda)]$ [146].

[10] In the case of a recursive code, the generator G_1 refers to the feedback line, whereas in the case of a nonrecursive code the same generator is relative to the first generated code symbol.

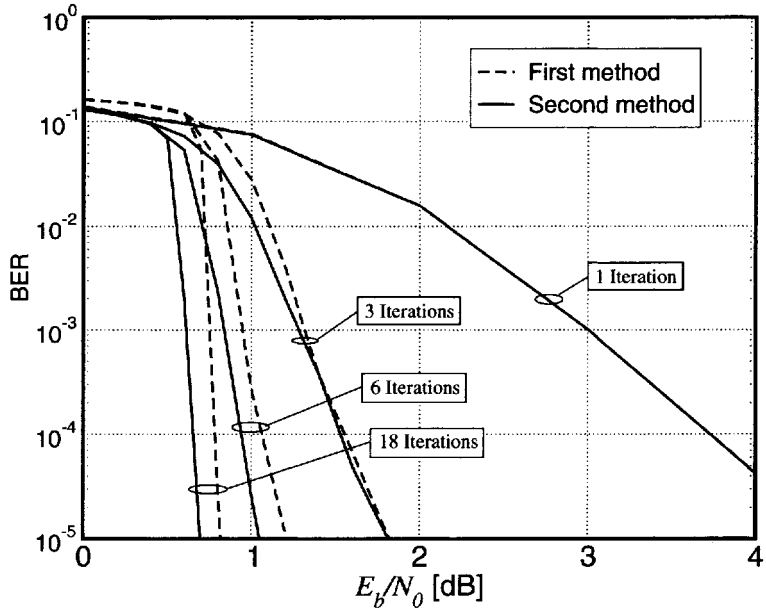


Figure 4.6: BER of a turbo code and the FB algorithm. The extrinsic information generated by each decoder is either modeled as a Gaussian-distributed random variable (first method) or used to update the *a priori* probabilities (second method). The considered numbers of iterations are 1, 3, 6 and 18. Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

fact, for each iteration and each SNR, the best value of θ is 1/2, which corresponds to the second method.

For the first method, Figure 4.7 shows the behavior of the average value of the ratio η_z/σ_z^2 for the extrinsic information at the input of the first decoder as a function of the number of iterations and for various values of SNR. It may be observed that, almost regardless of the considered number of iterations, for values of SNR below a convergence threshold (about 0.7 dB), this ratio takes on values greater than 1/2. Therefore, in the case of the first method, the extrinsic information is overestimated. This has been previously observed in [33], where an heuristic normalization of the extrinsic information has been proposed with the aim of improving the performance at low SNR. We may conclude that, in the FB algorithm, the second method corresponds to a better use of the extrinsic information, whereas the first method is asymptotically optimal for an SNR above 0.7 dB and a sufficiently large number of iterations. In addition, the second method does not require the estimation of the ratio η_z/σ_z^2.

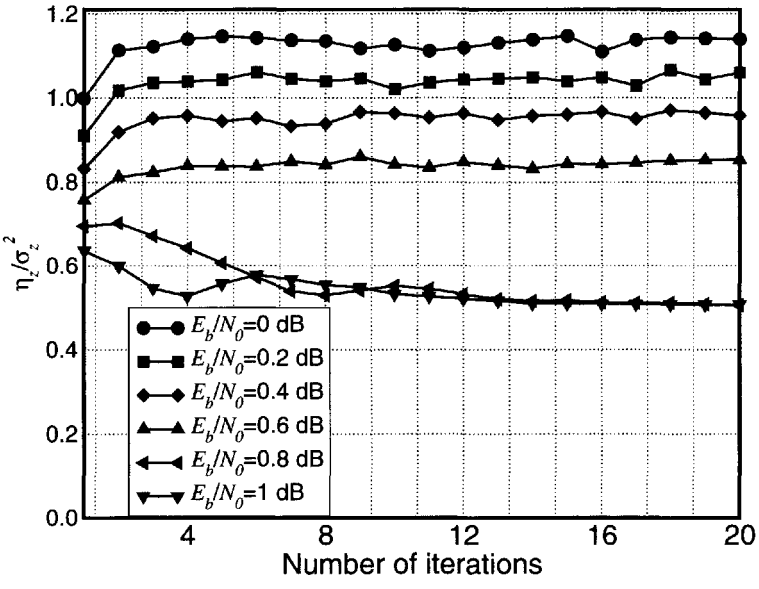


Figure 4.7: Average value of ratio η_z/σ_z^2 versus the number of iterations, for various values of SNR and a turbo code. The component decoders use the FB algorithm. The extrinsic information generated by each decoder is modeled as a Gaussian-distributed random variable (first method). Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

As for a PCCC, when considering an SCCC in conjunction with the FB algorithm, the optimal method proves to be the second one. In Figure 4.8, only the BER performance of the first two methods is shown, since for values of θ different from 0.5 (third method) the performance degrades. In this case, the first method is not asymptotically optimal. In fact, Figure 4.9 shows that the ratio η_z/σ_z^2 at the input of the first encoder tends to a value approximately equal to 0.35, whereas the optimal performance was obtained with the second method, i.e., with $\theta = 0.5$.

4.5.3 Soft-Output Viterbi Algorithm

An alternative to the FB algorithm is represented by SOVA [33, 139, 140], whose soft output is an approximation of the LLR. In the numerical results, we use the soft-output Viterbi decoder architecture proposed in [33] (with the updating rule proposed in [139]) in order to obtain a real-time scheme, whose complexity is roughly doubled with respect to that of a classical Viterbi decoder. The conclusions drawn when using this algorithm also hold for a suboptimal version of the FB algorithm,

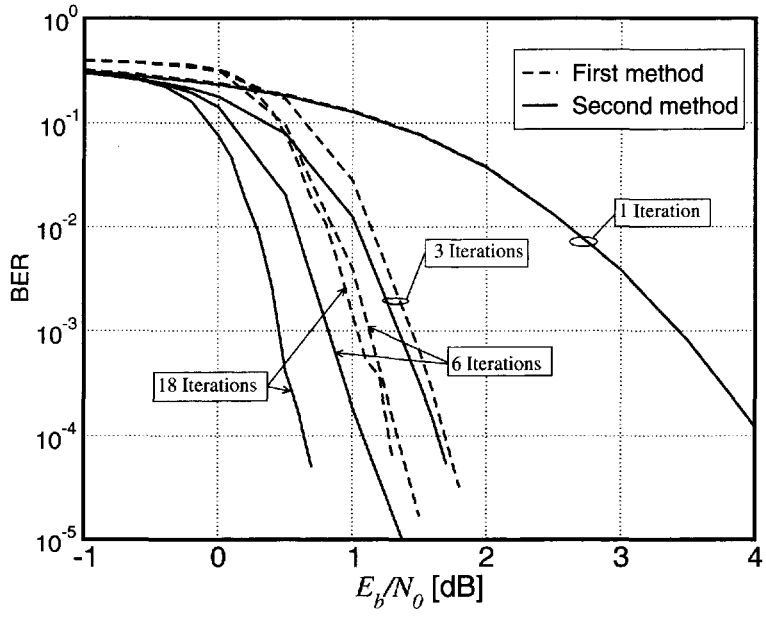


Figure 4.8: BER of a serially concatenated code and FB algorithm. The considered numbers of iterations are 1, 3, 6 and 18. Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

obtained by applying the max-log approximation [137], since these two algorithms are equivalent [148].

Recalling that K indicates the number of samples of each data block, we define by $\boldsymbol{r} \triangleq \{r_k\}_{k=0}^{K-1}$ and $\boldsymbol{z} \triangleq \{z_k\}_{k=0}^{K-1}$ the sequences of observations and input extrinsic information, respectively. We also denote by $\boldsymbol{R} \triangleq \{R_k\}_{k=0}^{K-1}$ the sequence of inputs of the considered decoder and by $\boldsymbol{a} \triangleq \{a_k\}_{k=0}^{K-1}$ the sequence of information symbols. As in the case of the FB algorithm, (i) $R_k = (r_k, z_k)$ if z_k is interpreted as the output of a Gaussian meta-channel, or (ii) $R_k = r_k$ if z_k is used to update the *a priori* probabilities. The MAP sequence detection strategy corresponds to the maximization of the following metric:

$$\Lambda(\boldsymbol{a}) = \ln\left[p(\boldsymbol{R}|\boldsymbol{a})P\{\boldsymbol{a}\}\right] = \ln p(\boldsymbol{R}|\boldsymbol{a}) + \ln P\{\boldsymbol{a}\}\,. \tag{4.66}$$

This metric may be arbitrarily multiplied by any constant independent of the information sequence.

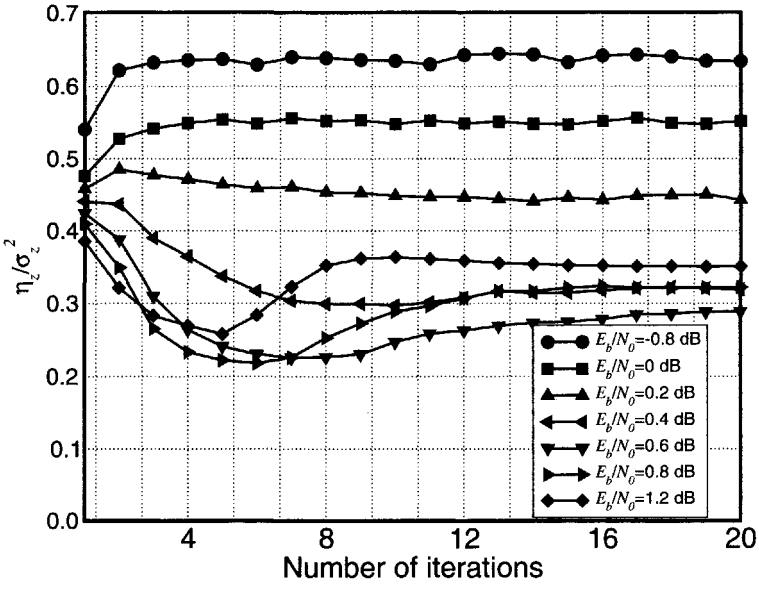


Figure 4.9: Average value of ratio η_z/σ_z^2 versus number of iterations, for various values of SNR and a serially concatenated code. The component decoders use the FB algorithm. The extrinsic information generated by each decoder is modeled as a Gaussian-distributed random variable (first method). Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

Extrinsic Information as Gaussian-Distributed Input

Since $R_k = (r_k, z_k)$ and due to the assumed independence of r_k from z_k, one can write:

$$\Lambda(\boldsymbol{a}) = \ln p(\boldsymbol{r}|\boldsymbol{a}) + \ln p(\mathbf{z}|\boldsymbol{a}) + \ln P\{\boldsymbol{a}\}\,. \tag{4.67}$$

The pdfs in (4.67) may be expressed as

$$p(\boldsymbol{r}|\boldsymbol{a}) = \frac{1}{(2\pi\sigma^2)^{\frac{K}{2}}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=0}^{K-1}(r_k - c_k)^2\right\} \tag{4.68}$$

$$p(\boldsymbol{z}|\boldsymbol{a}) = \frac{1}{(2\pi\sigma_z^2)^{\frac{K}{2}}} \exp\left\{-\frac{1}{2\sigma_z^2}\sum_{k=0}^{K-1}(z_k - a_k\eta_z)^2\right\}. \tag{4.69}$$

In this case, the information symbols are assumed iid, i.e., $P\{a_k = +1\} = P\{a_k = -1\} = 1/2$. Therefore,

$$P\{\boldsymbol{a}\} = \frac{1}{2^K}\,. \tag{4.70}$$

Substituting (4.68), (4.69), and (4.70) into (4.67) and discarding irrelevant terms, we obtain the equivalent metric

$$\Lambda(\boldsymbol{a}) = \sum_{k=0}^{K-1} \left(\frac{r_k c_k}{\sigma^2} + \frac{z_k a_k \eta_z}{\sigma_z^2} \right) \tag{4.71}$$

which may be recursively computed using the following branch metric:

$$\lambda_k(\boldsymbol{a}) = \frac{r_k c_k}{\sigma^2} + \frac{z_k a_k \eta_z}{\sigma_z^2} . \tag{4.72}$$

SOVA does not produce soft outputs by considering all paths in the trellis diagram as in the case of the FB algorithm, but only two paths—the maximum likelihood path and its strongest competitor. Considering a binary RSC code such that paths terminating in a specific state μ_k are relative to different information symbols,[11] i.e., $a_k = +1$ and $a_k = -1$. We denote the corresponding code symbols by c_k^{+1} and c_k^{-1} and the corresponding cumulative metrics by Λ_k^{+1} and Λ_k^{-1}, respectively. Assuming that the winning path includes state μ_k at time k, an initial reliability value for symbol a_k is obtained by considering the absolute value of the difference between the cumulative metrics of the two paths terminating in state μ_k.

Let us consider the case $\Lambda_k^{+1} > \Lambda_k^{-1}$. An initial reliability value is [33, 139, 140]

$$\begin{aligned} \Lambda_k^{+1} - \Lambda_k^{-1} &= \lambda_k^{+1} - \lambda_k^{-1} + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1} \\ &= \frac{2\eta_z z_k}{\sigma_z^2} + \frac{r_k}{\sigma^2}(c_k^{+1} - c_k^{-1}) + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1} = \frac{2\eta_z z_k}{\sigma_z^2} + \Xi \end{aligned} \tag{4.73}$$

where

$$\Xi \triangleq \frac{r_k}{\sigma^2}(c_k^{+1} - c_k^{-1}) + \Lambda_{k-1}^{+1} - \Lambda_{k-1}^{-1}. \tag{4.74}$$

Similarly, in the case $\Lambda_k^{-1} > \Lambda_k^{+1}$, one can write $\Lambda_k^{-1} - \Lambda_k^{+1} = -\frac{2\eta_z z_k}{\sigma_z^2} - \Xi$. In general, the initial reliability value may be expressed as

$$\left| \Lambda_k^{+1} - \Lambda_k^{-1} \right| = a_k \frac{2\eta_z z_k}{\sigma_z^2} + a_k \Xi . \tag{4.75}$$

This value is then updated at successive time instants $(k+1, k+2, \ldots)$, according to a suitable rule [99, 139, 140]. Denoting by v_k the final reliability value derived from (4.75), a reasonable definition of the extrinsic information component w_k of the LLR relative to symbol a_k is

$$w_k \triangleq a_k v_k - \frac{2\eta_z z_k}{\sigma_z^2} . \tag{4.76}$$

[11] This is not the case if a nonrecursive code is considered, since all paths entering into a state are associated with the same information symbol.

Extrinsic Information Used to Update the *a Priori* Probabilities

In this case, $R_k = r_k$ and the decoder assumes that $P\{a_k = +1\}$ and $P\{a_k = -1\}$ can be obtained from the expression in (4.61). Since

$$P\{\boldsymbol{a}\} = \prod_{k=0}^{K-1} P\{a_k\} \tag{4.77}$$

and $p(\boldsymbol{r}|\boldsymbol{a})$ is given by (4.68), substituting (4.68) and (4.77) into (4.66) and discarding terms independent of the information sequence, we obtain

$$\Lambda(\boldsymbol{a}) = \sum_{k=0}^{K-1} \left[\frac{r_k c_k}{\sigma^2} + \ln P\{a_k\}\right] \tag{4.78}$$

Adding the constant $\sum_{k=0}^{K-1}[\frac{z_k}{2} + \ln(1+e^{z_k})]$, independent of $\boldsymbol{a}$, we have the equivalent metric

$$\Lambda(\boldsymbol{a}) = \sum_{k=0}^{K-1} \left[\frac{r_k c_k}{\sigma^2} + \frac{a_k z_k}{2}\right] \tag{4.79}$$

and the corresponding branch metric

$$\lambda_k(\boldsymbol{a}) = \frac{r_k c_k}{\sigma^2} + \frac{a_k z_k}{2}\,. \tag{4.80}$$

In this case, the extrinsic information at the decoder output can be written as

$$w_k \triangleq a_k v_k - z_k\,. \tag{4.81}$$

Discussion and Heuristic Method

As in the case of the FB algorithm, the two methods differ for the constant which multiplies the received extrinsic information z_k, both in the expression of the branch metrics (4.72) and (4.80) (η_z/σ_z^2, which appears in the first method, is substituted by 1/2 in the second one), and in the definition of the soft output w_k in (4.76) and (4.81) (in this case, the constant is $2\eta_z/\sigma_z^2$ in the first method and 1 in the second one).[12]

[12] In [142], the reliability value z_k at the input of each component decoder is normalized by multiplying it by the factor $2\eta_z/\sigma_z^2$, and is used to update the *a priori* probabilities. Although [142] claims to use the second method, because of this normalization the method actually used is the first one.

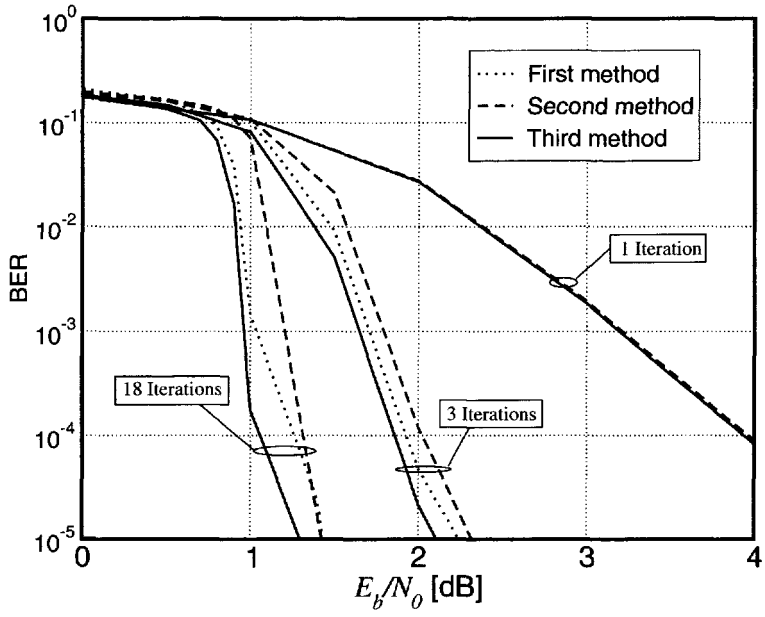


Figure 4.10: BER of the proposed detection schemes for a turbo code and SOVA. The extrinsic information generated by each decoder is either modeled as a Gaussian-distributed random variable (first method) or used to update the *a priori* probabilities (second method) or heuristically weighted (third method). The considered numbers of iterations are 1, 3 and 18. Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

Even in this case, an heuristic method may be implemented by introducing a weighting parameter θ to be optimized by trial and error.

Numerical Results

Similar simulations as for the FB algorithm, but considering SOVA as the component decoding algorithm, are considered. The performance of the three proposed methods for decoding the considered PCCC is shown in Figure 4.10. As expected, for any of the three methods, the performance degrades with respect to that of the corresponding scheme which uses the FB algorithm, due to the suboptimality of SOVA (compare the results in Figure 4.10 with those in Figure 4.6). Using SOVA, we may note that, unlike the FB algorithm, the best method is the heuristic method, with a value $\theta \simeq 0.4$ (optimal for any number of iterations). Moreover, in this case the second method is even worse than the first one. As observed in [142, 144], SOVA overestimates the

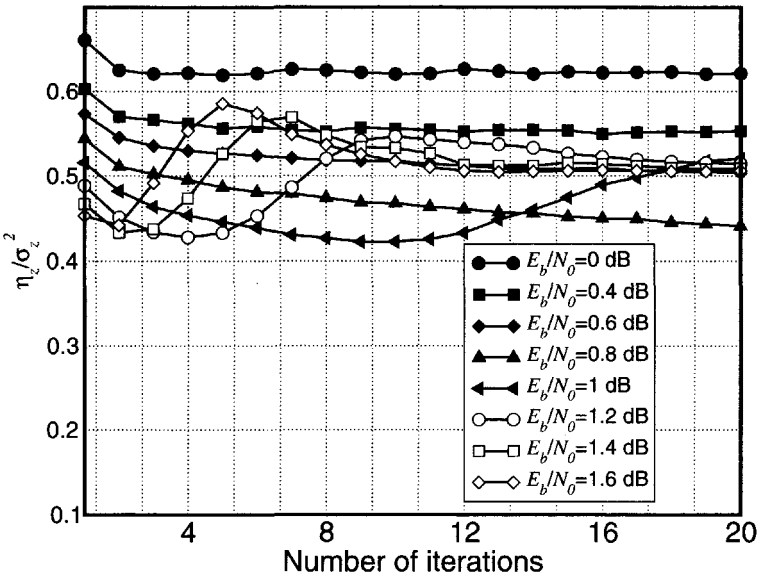


Figure 4.11: Average value of the ratio η_z/σ_z^2 versus the number of iterations, for various values of SNR by considering a turbo code. The component decoders use SOVA. The extrinsic information generated by each decoder is considered as a Gaussian-distributed random variable (first method).

reliability values—the obtained results are consistent with these observations. In fact, the coefficient θ multiplies the extrinsic information z_k generated by the other decoder; hence, a reduced value of θ "compresses" the sequence $\{z_k\}$, correcting the overestimation. Figure 4.11 shows the behavior of the average value of the ratio η_z/σ_z^2 in this case. As one can see, this ratio does not tend to the optimal value $\theta = 0.4$ when the number of iterations increases and the SNR is sufficiently high. A simple conclusion is that in this case the first method is not asymptotically optimal.

Similarly to the case of turbo codes, when using SOVA to iteratively decode the considered serially concatenated code, the best method is the third one, and the optimal value of θ is approximately 0.3. Figure 4.12 shows the performance of the first and third method. This is consistent with the analysis of the limit of the ratio η_z/σ_z^2, shown in Figure 4.13. We can conclude that the first method is asymptotically optimal in this case. However, in Figure 4.13 there is no evidence of the convergence, in terms of BER, of the first and third methods. According to the proposed analysis there should be convergence, which could actually occur at a lower BER.

We now comment further on the numerical results obtained with SOVA. In fact, both in the PCCC case and in the SCCC case, the best performance was obtained

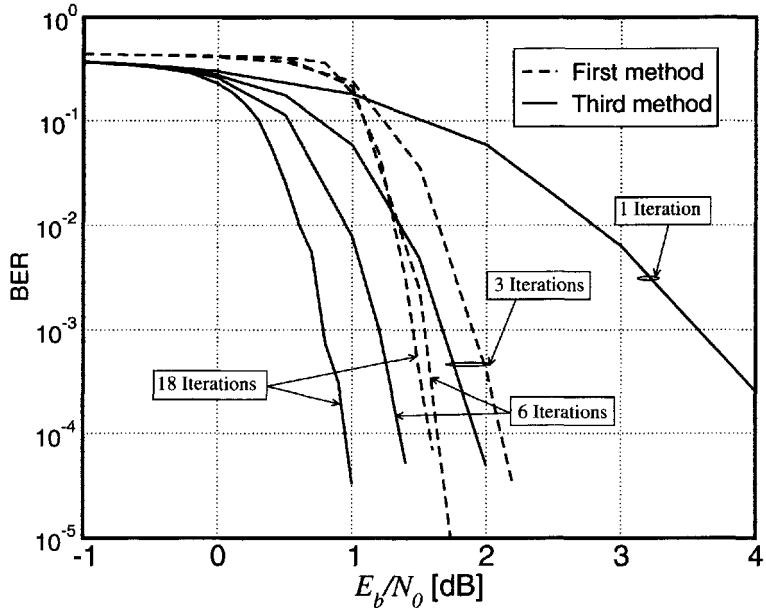


Figure 4.12: BER of a serially concatenated code and SOVA. The considered numbers of iterations are 1, 3, 6 and 18. Reproduced from [147], ©2001 IEEE, by permission of the IEEE.

considering the third approach with an heuristic parameter $\theta < 0.5$ (0.4 in the PCCC case and 0.3 in the SCCC case). This is because of the overestimation in the generated reliability values, and this phenomenon is most likely due to the suboptimality of the decoding algorithm. A general intuitive conclusion is that *any* suboptimal soft-output algorithm might suffer from the same overestimation problem. A practical example is given by reduced-state FB algorithms developed in Section 4.8. In that case, simply scaling the generated soft output helps significantly the iterative decoding convergence. Scaling the soft outputs in the logarithmic domain consists indeed of smoothing strong variations, and this is intuitively useful for the stability of the iterative decoding process.

4.6 Finite Memory Symbol Detection

We now show how to apply the general framework proposed in Chapter 2 to derive FB algorithms for channels with memory. The considered communication system model is shown in Figure 4.14. We recall briefly a few basic assumptions introduced

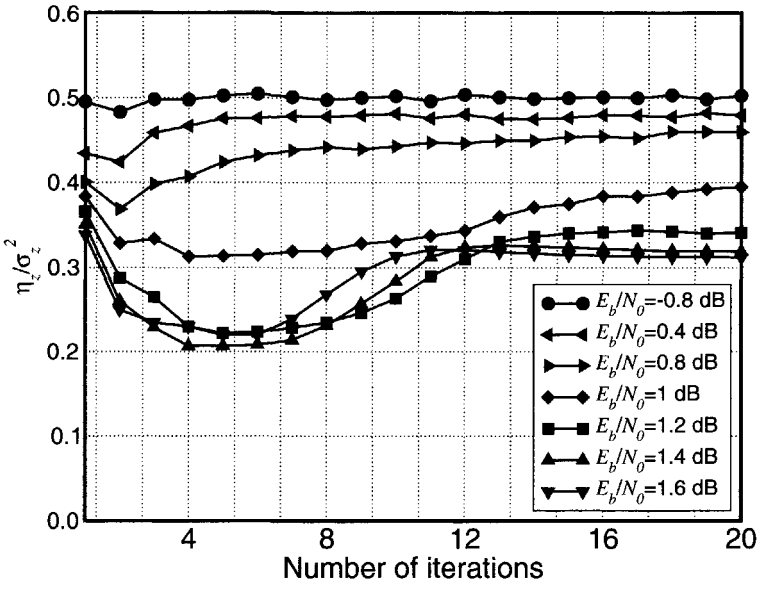


Figure 4.13: Average value of the ratio η_z/σ_z^2 versus the number of iterations, for various values of SNR and a serially concatenated code. The component decoders use SOVA. The extrinsic information generated by each decoder is considered as a Gaussian-distributed random variable (first method).

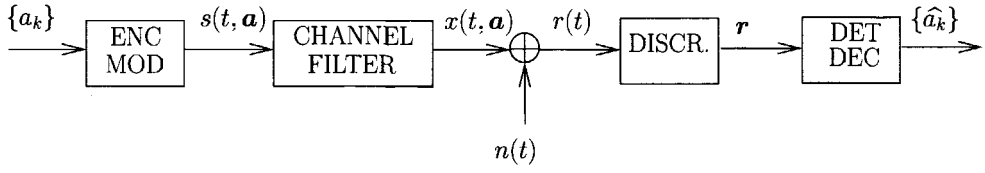


Figure 4.14: Communication system.

in Chapter 2. The encoder/modulator block can be described as a time-invariant FSM with state μ_k, which evolves according to a suitable "next-state" function $\mathrm{ns}(\cdot,\cdot)$ such that:

$$\mu_k = \mathrm{ns}(\mu_{k-1}, a_{k-1}). \tag{4.82}$$

The received signal, written as $r(t) = x(t, \boldsymbol{a}) + n(t)$, is converted, through a suitable discretization process (see Appendix A), into a time-discrete sequence $\boldsymbol{r}$. The following two statistical conditions are the keys to deriving detection algorithms for channels with memory:

- *causality condition*:

$$p(\boldsymbol{r}_0^k|\boldsymbol{a}) = p(\boldsymbol{r}_0^k|\boldsymbol{a}_0^k); \tag{4.83}$$

- *finite memory condition* (FMC):

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \tag{4.84}$$

where C is a suitable *finite memory parameter* and μ_{k-C} represents the state, at epoch $k-C$, of the encoder/modulator.

Based on causality and FMC, the following probabilistic derivation of an FB algorithm is obtained:

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} &= \sum_{\boldsymbol{a}_{k-C}^{k-1}} \sum_{\mu_{k-C}} P\{\boldsymbol{a}_{k-C}^k, \mu_{k-C}|\boldsymbol{r}\} \\ &\sim \sum_{\boldsymbol{a}_{k-C}^{k-1}} \sum_{\mu_{k-C}} p(\boldsymbol{r}|\boldsymbol{a}_{k-C}^k, \mu_{k-C}) P\{\boldsymbol{a}_{k-C}^k, \mu_{k-C}\} \\ &= \sum_{\boldsymbol{a}_{k-C}^{k-1}} \sum_{\mu_{k-C}} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \\ &\quad \cdot p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^k, \mu_{k-C}) P\{\boldsymbol{a}_{k-C}^k, \mu_{k-C}\} \end{aligned} \tag{4.85}$$

where Bayes', marginalization and chain rules have been used. Based on (2.24) and causality, it follows that

$$p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) = p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^k, \mu_{k-C+1}). \tag{4.86}$$

Based on causality, one can also write:

$$p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^k, \mu_{k-C}) = p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}). \tag{4.87}$$

Recalling the independence between information symbols, (4.85) can be rewritten as follows:

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} \sim & \sum_{\boldsymbol{a}_{k-C}^{k-1}} \sum_{\mu_{k-C}} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^k, \boldsymbol{a}_{k-C+1}^k, \mu_{k-C+1}) p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \\ & \cdot p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}) P\{\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}\} P\{a_k\}. \end{aligned} \tag{4.88}$$

Considering the augmented state[13] $S_k = (\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C})$ and the transition $T_k = (S_k, a_k)$, and defining

$$\gamma_k(T_k) \triangleq p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^{k}, \mu_{k-C})P\{a_k\} \tag{4.89}$$

$$\beta_{k+1}(S_{k+1}) \triangleq p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^{k}, \boldsymbol{a}_{k-C+1}^{k}, \mu_{k-C+1}) \tag{4.90}$$

$$\alpha_k(S_k) \triangleq p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C})P\{\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}\} \tag{4.91}$$

the APP in (4.85) can be finally formulated as follows:

$$P\{a_k|\boldsymbol{r}\} \sim \sum_{S_k} \beta_{k+1}(S_{k+1})\gamma_k(T_k)\alpha_k(S_k). \tag{4.92}$$

Based on the causality condition and FMC, the quantities $\alpha_k(S_k)$ and $\beta_{k+1}(S_{k+1})$ can be computed by means of forward and backward recursions, respectively. We first consider the forward recursion. Applying Bayes' and marginalization rules, one can write:

$$\begin{aligned}\alpha_k(S_k) &= p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C})P\{\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}\} \\ &= P\{\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}|\boldsymbol{r}_0^{k-1}\}p(\boldsymbol{r}_0^{k-1}) \\ &= \sum_{\substack{\mu_{k-C-1}, a_{k-C-1}: \\ \mathrm{ns}(\mu_{k-C-1}, a_{k-C-1}) = \mu_{k-C}}} P\{\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}|\boldsymbol{r}_0^{k-1}\}p(\boldsymbol{r}_0^{k-1}).\end{aligned} \tag{4.93}$$

Indicating concisely by $T_{k-1} : S_k$ the index of the summation in (4.93) (i.e., all transitions T_{k-1} compatible with state S_k) and applying Bayes' and chain factorization rules, (4.93) can be expressed as follows:

$$\begin{aligned}\alpha_k(S_k) &= \sum_{T_{k-1}:S_k} p(\boldsymbol{r}_0^{k-1}|\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1})P\{\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}\} \\ &= \sum_{T_{k-1}:S_k} p(r_{k-1}|\boldsymbol{r}_0^{k-2}, \boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1})p(\boldsymbol{r}_0^{k-2}|\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}) \\ &\qquad \cdot P\{\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}\}.\end{aligned} \tag{4.94}$$

Owing to causality and independence of the information symbols, respectively, the following identities hold:

$$p(\boldsymbol{r}_0^{k-2}|\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}) = p(\boldsymbol{r}_0^{k-2}|\boldsymbol{a}_{k-C-1}^{k-2}, \mu_{k-C-1}) \tag{4.95}$$

$$P\{\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1}\} = P\{a_{k-1}\}P\{\boldsymbol{a}_{k-C-1}^{k-2}, \mu_{k-C-1}\} \tag{4.96}$$

[13] As explained in Chapter 2, one can immediately introduce an augmented trellis diagram, with state S_k, characterized by a "next-state" function $\mathrm{NS}(S_k, a_k) = (\mathrm{ns}(\mu_{k-C}, a_{k-C}), \boldsymbol{a}_{k-C+1}^{k})$.

and, finally, (4.94) can be expressed as follows:

$$\begin{aligned}\alpha_k(S_k) &= \sum_{T_{k-1}:S_k} \underbrace{p(r_{k-1}|\boldsymbol{r}_0^{k-2}, \boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1})P\{a_{k-1}\}}_{\gamma_{k-1}(T_{k-1})} \\ &\quad \cdot \underbrace{p(\boldsymbol{r}_0^{k-2}|\boldsymbol{a}_{k-C-1}^{k-1}, \mu_{k-C-1})P\{\boldsymbol{a}_{k-C-1}^{k-2}, \mu_{k-C-1}\}}_{\alpha_{k-1}(S_{k-1})} \\ &= \sum_{T_{k-1}:S_k} \gamma_{k-1}(T_{k-1})\alpha_{k-1}(S_{k-1}). \end{aligned} \tag{4.97}$$

The backward recursion can be similarly obtained. In fact, by applying Bayes' and marginalization rules and recalling the independence between information symbols, one obtains:

$$\begin{aligned}\beta_k(S_k) &= p(\boldsymbol{r}_k^{K-1}|\boldsymbol{r}_0^{k-1}, S_k) = p(\boldsymbol{r}_k^{K-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}) \\ &= \sum_{a_k} p(\boldsymbol{r}_k^{K-1}|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^{k}, \mu_{k-C})P\{a_k|\boldsymbol{a}_{k-C}^{k-1}, \mu_{k-C}\} \\ &= \sum_{a_k} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^{k}, \boldsymbol{a}_{k-C}^{k}, \mu_{k-C})p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_{k-C}^{k}, \mu_{k-C})P\{a_k\}. \end{aligned} \tag{4.98}$$

Using (4.86), it follows that the backward recursion can be finally rewritten as

$$\beta_k(S_k) = \sum_{a_k} \beta_{k+1}(S_{k+1})\gamma_k(T_k). \tag{4.99}$$

Proper boundary conditions for the initial forward metrics $\{\alpha_0(S_0)\}$ and backward metrics $\{\beta_{K-1}(S_{K-1})\}$ must be specified. Assuming that the starting state $\overline{S}_0$ is known and that the final state is unknown (i.e., trellis termination is not considered), the correct boundary conditions can be written as follows:

$$\alpha_0(S_0) = \begin{cases} 1 & S_0 = \overline{S}_0 \\ 0 & S_0 \neq \overline{S}_0 \end{cases} \qquad \beta_K(S_K) = \frac{1}{S_c M^C} \quad \forall S_K \tag{4.100}$$

where S_c is the number of states of the encoder/modulator FSM and M is the cardinality of the information symbols.

The obtained FB algorithm can also be formulated in the logarithmic domain. It can also be approximated applying the max-log approximation introduced in Section 4.3. More precisely, defining the following metrics (as direct extensions of those introduced in the case of detection over channels without memory):

$$\Lambda_k^{\mathrm{F}}(S_k) \triangleq \ln \alpha_k(S_k) \tag{4.101}$$

$$\lambda_k(a_k, S_k) \triangleq \ln \gamma_k(a_k, S_k) \tag{4.102}$$

$$\Lambda_k^{\mathrm{B}}(S_k) \triangleq \ln \beta_k(S_k) \tag{4.103}$$

an approximation of the APP, i.e., a reliability value close to the APP, can be written as follows:

$$\mathrm{S}[a_k] = \max_{S_k} \left\{ \Lambda_k^{\mathrm{F}}(S_k) + \lambda_k(a_k, S_k) + \Lambda_{k+1}^{\mathrm{B}}(S_{k+1}) \right\}. \tag{4.104}$$

The quantities $\Lambda_k^{\mathrm{F}}(S_k)$ and $\Lambda_k^{\mathrm{B}}(S_k)$ (corresponding to forward and backward path metrics), can be recursively computed as follows:

$$\Lambda_k^{\mathrm{F}}(S_k) = \max_{T_{k-1}:S_k} \left\{ \Lambda_k^{\mathrm{F}}(S_k) + \lambda_{k-1}(T_{k-1}) \right\} \tag{4.105}$$

$$\Lambda_k^{\mathrm{B}}(S_k) = \max_{a_k} \left\{ \Lambda_{k+1}^{\mathrm{B}}(S_{k+1}) + \lambda_k(T_k) \right\}. \tag{4.106}$$

Obviously, the two recursions given by (4.105) and (4.106) can be implemented by two VAs. It can therefore immediately be concluded that any detection strategy designed for sequence detection can be systematically extended to the case of symbol detection.

In the case of a channel characterized by parameters affected by stochastic uncertainty, the observations $\{r_k\}$ are *dependent*, so that the channel memory may not be finite. A very general parametric model for the observation r_k is the following:

$$r_k = g(\boldsymbol{a}_{k-L}^k, \mu_{k-L}, \boldsymbol{\theta}_0^k) + n_k \tag{4.107}$$

where L is an integer, $\boldsymbol{\theta}_0^k$ is a sequence of stochastic parameters independent of $\boldsymbol{a}$, and n_k is an additive noise sample.[14] Under this channel model, the following *conditional Markov property* (CMP) is sufficient to guarantee an FMC:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_0^k) \tag{4.108}$$

where N is the order of Markovianity. In fact, (4.108) implies the following:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{a}_{k-C}^k, \mu_{k-C}) \tag{4.109}$$

where the finite memory parameter is $C = N + L$. It is immediately recognized that (4.109) represents a special case of (2.17). As a consequence, all the derivations in the previous section hold.

A statistical description of the stochastic parameter allows one to compute $\gamma_k(T_k)$ as

$$\gamma_k(T_k) = \frac{p(\boldsymbol{r}_{k-N}^k|T_k)}{p(\boldsymbol{r}_{k-N}^{k-1}|S_k)} P\{a_k\} = \frac{\mathrm{E}_{\boldsymbol{\theta}_0^k}\{p(\boldsymbol{r}_{k-N}^k|T_k, \boldsymbol{\theta}_0^k)\}}{\mathrm{E}_{\boldsymbol{\theta}_0^{k-1}}\{p(\boldsymbol{r}_{k-N}^{k-1}|S_k, \boldsymbol{\theta}_0^{k-1})\}} P\{a_k\}. \tag{4.110}$$

[14] The additive noise is not required to be Gaussian for the validity of the following result.

Unfortunately, the above exact result is limited by the fact that *in realistic scenarios* the CMP (4.108) is seldom met exactly [47,95]. However, this result suggests a reasonable approach to devising approximate detection algorithms whenever the conditional observations are asymptotically independent for increasing index difference.

4.7 An Alternative Approach to Finite Memory Symbol Detection

In this section, we present an alternative formulation to derive an FB algorithm for channels with memory.[15] For the sake of simplicity, we refer directly to the case where a CMP holds and $L = 0$—a more general formulation, where an FMC (not involving any truncation of the number of conditioning observables) applies, can be straightforwardly obtained. In this case, we slightly modify the CMP, referring to it as an *extended CMP* (ECMP). More precisely, we refer to the present property as extended in the sense that we consider the dependence of an observation from $N+1$ previous observations, and not from N as before. In other words, we assume that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{a}_0^k) = p(r_k|\boldsymbol{r}_{k-N+1}^{k-1}, \boldsymbol{a}_{k-N-1}^k, \mu_{k-N-1}). \tag{4.111}$$

We now focus on the computation of the APP $P\{a_k|\boldsymbol{r}\}$, which can be given the following expression (alternative to (4.85)):

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} &= p(\boldsymbol{r}_0^{k-N-1}, \boldsymbol{r}_{k-N}^k, \boldsymbol{r}_{k+1}^{K-1}|a_k)P\{a_k\} \\ &= \sum_{T_k:a_k} p(\boldsymbol{r}_0^{k-N-1}, \boldsymbol{r}_{k-N}^k, \boldsymbol{r}_{k+1}^{K-1}|T_k)P\{T_k|a_k\}P\{a_k\} \\ &= \sum_{T_k:a_k} \underbrace{p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^{k-N-1}, \boldsymbol{r}_{k-N}^k, T_k)}_{p_1} \underbrace{p(\boldsymbol{r}_0^{k-N-1}|\boldsymbol{r}_{k-N}^k, T_k)}_{p_2} \\ &\qquad \cdot \underbrace{p(\boldsymbol{r}_{k-N}^k|T_k)P\{S_k\}P\{a_k\}}_{p_3} \end{aligned} \tag{4.112}$$

where S_k and T_k are the "usual" expanded state and transition, and independence between information symbols has been used considering $P\{T_k|a_k\} = P\{S_k\}$. We

[15] This is a possible example, nonexhaustive, which shows that in the case of FB algorithms, it is possible to consider detection strategies which are not obtained directly from the VA. However, it is likely that any solution 'borrowed" from a sequence detection algorithm, can be extended to modifi ed FB algorithms as well [149].

define

$$\mathring{\alpha}_k(T_k) \triangleq p_2 = p(\boldsymbol{r}_0^{k-N-1}|\boldsymbol{r}_{k-N}^k, T_k) \tag{4.113}$$

$$\mathring{\gamma}_k(T_k) \triangleq p_3 = p(\boldsymbol{r}_{k-N+1}^k|T_k)P\{S_k\}P\{a_k\} \tag{4.114}$$

and, applying the ECMP, we can simplify the pdf p_1 as follows:

$$p_1 = p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_0^{k-N-1}, \boldsymbol{r}_{k-N}^k, T_k) = p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_{k-N}^k, T_k) \triangleq \mathring{\beta}_k(T_k). \tag{4.115}$$

The introduced quantities $\mathring{\alpha}_k(T_k)$ and $\mathring{\beta}_k(T_k)$ can be recursively computed by means of forward and backward recursions, respectively. We point out that, unlike in the derivation proposed in Section 4.6, in this case the quantities $\mathring{\alpha}_k(T_k)$ and $\mathring{\beta}_k(T_k)$, computed during the two recursions, depend on a transition T_k, rather than on a state S_k.

We first consider the forward recursion. Given a state S_k, there are M possible predecessor states $\{S_{k-1}\}$. For ease of derivation, we assume that each predecessor state is in a one-to-one correspondence to an information symbol[16] a_{k-N-1}. It follows that $T_{k-1} = (S_{k-1}, S_k) = (a_{k-N}, S_k)$. Hence, given $T_k = (S_k, S_{k+1})$ and a_{k-N-1}, the transition T_{k-1} can be uniquely identified. Therefore,

$$\begin{aligned}
\mathring{\alpha}_k(T_k) &= p(\boldsymbol{r}_0^{k-N-1}|\boldsymbol{r}_{k-N}^k, T_k) \\
&= \sum_{a_{k-N-1}} p(\boldsymbol{r}_0^{k-N-1}|\boldsymbol{r}_{k-N}^k, a_{k-N-1}, T_k) \underbrace{P\{a_{k-N-1}|\boldsymbol{r}_{k-N}^k, T_k\}}_{P\{a_{k-N-1}\}\text{ (caus. and indep.)}} \\
&= \sum_{a_{k-N-1}} p(\boldsymbol{r}_0^{k-N-2}|\boldsymbol{r}_{k-N-1}^k, a_{k-N-1}, T_k) \\
&\qquad \cdot p(r_{k-N-1}|\boldsymbol{r}_{k-N}^k, a_{k-N-1}, T_k)P\{a_{k-N-1}\}
\end{aligned} \tag{4.116}$$

where the fact that $P\{a_{k-N-1}|\boldsymbol{r}_{k-N}^k, T_k\} = P\{a_{k-N-1}\}$ has been used. Considering the CMP and observing that $(a_{k-N-1}, T_k) = (T_{k-1}, T_k)$, one can write:

$$p(\boldsymbol{r}_0^{k-N-2}|\boldsymbol{r}_{k-N-1}^k, T_{k-1}, T_k) = p(\boldsymbol{r}_0^{k-N-2}|\boldsymbol{r}_{k-N-1}^{k-1}, T_{k-1}) = \mathring{\alpha}_{k-1}(T_{k-1}). \tag{4.117}$$

Defining

$$\begin{aligned}
\psi_k(T_{k-1}, T_k) &\triangleq p(r_{k-N-1}|\boldsymbol{r}_{k-N}^k, T_{k-1}, T_k)P\{a_{k-N-1}\} \\
&= \frac{p(\boldsymbol{r}_{k-N-1}^k, T_{k-1}, T_k)}{p(\boldsymbol{r}_{k-N}^k, T_k)}P\{a_{k-N-1}\}
\end{aligned} \tag{4.118}$$

[16]Note that this is true for particular recursive codes. However, the final obtained formulation applies also to nonrecursive codes.

the forward recursion can be expressed as follows:

$$\mathring{\alpha}_k(T_k) = \sum_{a_{k-N-1}} \mathring{\alpha}_{k-1}(T_{k-1})\psi_k(T_{k-1}, T_k). \tag{4.119}$$

The backward recursion can be derived in a similar way. In fact:

$$\begin{aligned}\mathring{\beta}_k(T_k) &= p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_{k-N}^{k}, T_k) \\ &= \sum_{a_{k+1}} p(\boldsymbol{r}_{k+1}^{K-1}|\boldsymbol{r}_{k-N}^{k}, T_k, a_{k+1}) \underbrace{P\{a_{k+1}|\boldsymbol{r}_{k-N}^{k}, T_k\}}_{P\{a_{k+1}\}\text{ (caus. and indep.)}} \\ &= \sum_{a_{k+1}} p(\boldsymbol{r}_{k+2}^{K-1}|\boldsymbol{r}_{k-N}^{k+1}, T_k, a_{k+1}) \\ &\qquad \cdot p(r_{k+1}|\boldsymbol{r}_{k-N}^{k}, T_k, a_{k+1}) P\{a_{k+1}\}. \end{aligned} \tag{4.120}$$

Observing that $(T_k, a_{k+1}) \equiv (T_k, T_{k+1})$ and applying the ECMP, it follows that

$$p(\boldsymbol{r}_{k+2}^{K-1}|\boldsymbol{r}_{k-C+1}^{k+1}, T_k, T_{k+1}) = p(\boldsymbol{r}_{k+2}^{K-1}|\boldsymbol{r}_{k-C+2}^{k+1}, T_{k+1}) = \mathring{\beta}_{k+1}(T_{k+1}). \tag{4.121}$$

By defining

$$\begin{aligned}\phi_{k+1}(T_k, T_{k+1}) &\triangleq p(r_{k+1}|\boldsymbol{r}_{k-N}^{k}, T_k, T_{k+1}) P\{a_{k+1}\} \\ &= \frac{p(\boldsymbol{r}_{k-N}^{k+1}|T_k, T_{k+1})}{p(\boldsymbol{r}_{k-N}^{k}|T_k)} P\{a_{k+1}\} \end{aligned} \tag{4.122}$$

the backward recursion can finally be written as

$$\mathring{\beta}_k(T_k) = \sum_{a_{k+1}} \mathring{\beta}_{k+1}(T_{k+1})\phi_{k+1}(T_k, T_{k+1}). \tag{4.123}$$

At this point, indicating the APP in (4.112) with the general notation[17] $\mathrm{S}[a_k]$, one can finally write:

$$\mathrm{S}[a_k] = \sum_{T_k:a_k} \mathring{\alpha}_k(T_k)\mathring{\beta}_k(T_k)\mathring{\gamma}_k(T_k). \tag{4.124}$$

Proper boundary conditions must be introduced in terms of initial and final states of the encoder. A reasonable approximation for the computation of $P\{S_k\}$ in $\mathring{\alpha}_k(T_k)$ is that of assuming[18] $P\{S_k\} \simeq \prod_{j=1}^{N} P\{a_{k-j}\}$. While the final FB algorithm obtained in Section 4.6 is a natural extension of the coherent FB algorithm [54] (with a

[17] If the ECMP applies only approximately, then $\mathrm{S}[a_k]$ is only an approximation of the APP.

[18] This implicitly assumes that the recursive states μ_k are all equiprobable.

properly extended metric $\gamma_k(T_k)$), the FB algorithm summarized in (4.119), (4.123) and (4.124) represents an alternative nonimmediate generalization. We refer to this class of algorithms as *FB-type* algorithms. We will consider again these algorithms in Section 4.8, where suitable state reduction techniques will be introduced.

We now comment on the relationship between the FB algorithm derived in this section and the FB algorithm proposed in Section 4.6. For a given value of the parameter N, it is intuitive that the algorithm derived in this section, due to the ECMP, should perform better than the one proposed in Section 4.6. In fact, one can equivalently describe the two algorithms by saying that the former assumes a "correlation span" of length $(N+2)$ among the observations, while the latter reduces it to $(N+1)$. However, the simpler formulation in Section 4.6 may be more appealing for the following reasons.

- From a complexity viewpoint, given a specific number of trellis states, the FB-type algorithms proposed in this section present an increased complexity, with respect to the FB algorithms of Section 4.6. We now intuitively quantify the increase of complexity. We assume that a unit of complexity is associated with a transition T_k. Indicating by $\zeta = S_c M^N$ the number of states $\{S_k\}$ at the decoder (S_c is the number of states of the encoder/modulator FSM), we observe that, considering an FB algorithm, each step in the two recursions has complexity ζM. The same complexity is associated with the completion at each epoch. Then, the complexity of an FB algorithm is approximately $3\zeta MK$. Considering FB-type algorithms, each step in the two recursions involves two consecutive transitions, with complexity ζM^2. Since the combination has the same complexity as for an FB algorithm, the complexity of FB-type algorithms is $\zeta MK(2M+1)$. Finally, the increase in complexity of FB-type algorithms with respect to FB algorithms is approximately $(2M+1)/3$. If the cardinality of the information symbols is large, as in the case of spectrally efficient trellis coded modulation (TCM) [9], the increase in complexity may become significant, and could be justified only for a proportional performance improvement. As we will show later on in this section, in the case of a noncoherent channel the performance improvement, if any, is not remarkable.

- The complexity can be further reduced by applying state reduction techniques. In particular, the state reduction technique which will be introduced in Section 4.8, with possible generalizations, can be applied. As will become clearer below, one can refer to the FB algorithms of Section 4.6 as *feed-forward*. The feed-forward structure allows simple implementation of state reduction techniques, whereas for FB-type algorithms the application of state reduction techniques is more complicated.

As a specific example of comparison between the proposed FB algorithmic classes, we focus on transmission over a noncoherent channel. We assume that the channel introduces an unknown phase rotation, modeled as a time-invariant random variable with uniform distribution in $[0, 2\pi)$. We consider linear modulation at the transmitter side. In this case, the samples at the output of a matched filter are a sufficient statistic [47] and have the following expression:

$$r_k = c_k e^{j\theta} + n_k \tag{4.125}$$

where c_k represents the modulated, and possibly coded, symbol and n_k is a complex AWGN sample with variance per component equal to σ^2. In this case, the order of Markovianity N is related to the phase memory parameter introduced in [47].

We first consider the feed-forward FB algorithm proposed in Section 4.6. Based on the considered channel model, one can write:

$$\begin{aligned}
\gamma_k(T_k) &= \frac{\mathrm{E}_\theta\left\{p(\boldsymbol{r}_{k-N}^{k}|\theta, T_k)\right\}}{\mathrm{E}_\theta\left\{p(\boldsymbol{r}_{k-N}^{k-1}|\theta, S_k)\right\}} P\{a_k\} \\
&\sim \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|\right)} P\{a_k\} \\
&\simeq \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|}{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|} P\{a_k\}.
\end{aligned} \tag{4.126}$$

The final expression (4.126) is obtained by considering the approximation $\log \mathrm{I}_0(x) \simeq x$ as in [47]. In the following, we will refer to this algorithm as the NCSOa algorithm.

When considering the FB-type algorithm proposed in Section 4.7, the pdfs $\psi_k(T_{k-1}, T$ and $\phi_k(T_{k-1}, T_k)$ can be computed by averaging over the stochastic parameters as for

$\gamma_k(T_k)$, and have the following expressions:

$$\begin{aligned}\psi_k(T_{k-1},T_k) &= p(r_{k-N-1}|\boldsymbol{r}_{k-N}^{k},T_{k-1},T_k)P\{a_{k-N-1}\}\\ &\sim \exp\left(-\frac{|c_{k-N-1}|^2}{2\sigma^2}\right)\frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N+1}r_{k-i}c_{k-i}^{*}\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N}r_{k-i}c_{k-i}^{*}\right|\right)}P\{a_{k-N-1}\}\end{aligned} \tag{4.127}$$

$$\begin{aligned}\phi_k(T_{k-1},T_k) &= p(r_k|\boldsymbol{r}_{k-N-1}^{k-1},T_{k-1},T_k)P\{a_k\}\\ &\sim \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right)\frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N+1}r_{k-i}c_{k-i}^{*}\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N+1}r_{k-i}c_{k-i}^{*}\right|\right)}P\{a_k\}.\end{aligned} \tag{4.128}$$

As one can see from (4.127) and (4.128), the FB-type algorithm is such that:

- in the forward recursion, the phase at epoch $(k-N-1)$ is implicitly estimated based on future observations (the observations $\boldsymbol{r}_{k-N}^{k}$);
- in the backward recursion, the phase at epoch k is estimated based on the past (the observations $\boldsymbol{r}_{k-N-1}^{k-1}$).

This means that in the forward recursion there is a sort of backward phase estimation, and vice versa in the backward recursion. The completion is such that the two quantities are "linked" together by the pdf $\mathring{\gamma}_k(T_k)$, which has the following expression

$$\begin{aligned}\mathring{\gamma}_k(T_k) &= p(\boldsymbol{r}_{k-N}^{k}|T_k)P\{a_k\}\\ &\sim \exp\left(-\frac{1}{2\sigma^2}\sum_{i=0}^{N}|c_{k-i}|^2\right)\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N}r_{k-i}c_{k-i}^{*}\right|\right)P\{a_k\}.\end{aligned} \tag{4.129}$$

We will refer to this algorithm as the NCSOb algorithm.

The structure of the estimation process embedded inside the NCSOa algorithm is shown in Figure 4.15. Figure 4.16 shows the estimation strategy embedded in the NCSOb algorithm. As one can catch from these figures, both algorithms consider an implicit phase estimation[19] (in different ways) and their performance is basically

[19]For more details, see Section 4.9.3.

comparable, as will become apparent in the following. However, there is a substantial difference. In the NCSOa algorithm, the feed-forward phase estimate at epoch k is the same in the recursions and in the completion. On the other hand, in the NCSOb algorithm the phase estimate at epoch k is different in each recursion and in the completion. In particular, we observe that the NCSOb algorithm hints at another perspective in the design of soft-output algorithms with implicit estimation. More precisely, it suggests that during the two recursions suitable estimation strategies could be used, while in the completion a simpler combination of the quantities recursively computed in the two recursions could be considered (see [149] for further extensions in this direction).

We shortly compare the performance obtained by considering the NCSOa algorithm and the NCSOb algorithm as component detection algorithms for iterative detection of concatenated codes. As examples, we consider two concatenated codes, namely a PCCC and an SCCC. The performance of the proposed decoding algorithms is assessed by means of computer simulations in terms of BER versus the bit SNR E_b/N_0, E_b being the received energy per information bit and N_0 the one-sided noise power spectral density. In the simulation programs we considered, for numerical and practical reasons, the max-log versions of the NCSOa and NCSOb algorithms inside each component adaptive decoder at the receiver side.

We first consider a PCCC of rate-1/2 with 16-state binary RSC codes as component codes with generators $G_1 = (37)_8$ and $G_2 = (21)_8$. The inner pseudorandom bit interleaver is 32×32 [33]. The output modulation is BPSK. The RSC code is noncoherently noncatastrophic [150]. In Figure 4.17, the performance when considering both algorithms is shown in the case with $N = 2$. For comparison, the performance in the coherent case, i.e., assuming perfect channel state information (CSI) at the receiver side, is also shown. In all cases, 1, 5, and 10 decoding iterations are considered. As one can see, the performance when using the NCSOb algorithm is slightly better (about 0.3 dB) for any number of decoding iterations, but this brings an increase in complexity: if $M = 2$ is the cardinality of the information sequence at the input of each component encoder, the complexity will increase on the order of $(2M+1)/3 = (2 \cdot 2 + 1)/3 = 5/3$ times. The already small performance gap can be completely eliminated, without additional complexity increase in the NCSOa algorithm, by using state reduction techniques for FB algorithms which will be described in more detail in Section 4.8. In particular, we will make use of this technique in the following numerical example relative to an SCCC.

The considered SCCC consists of an outer 4-state, rate-1/2 convolutional code connected through a 32×32 pseudorandom interleaver to an inner 4-state, rate-2/3

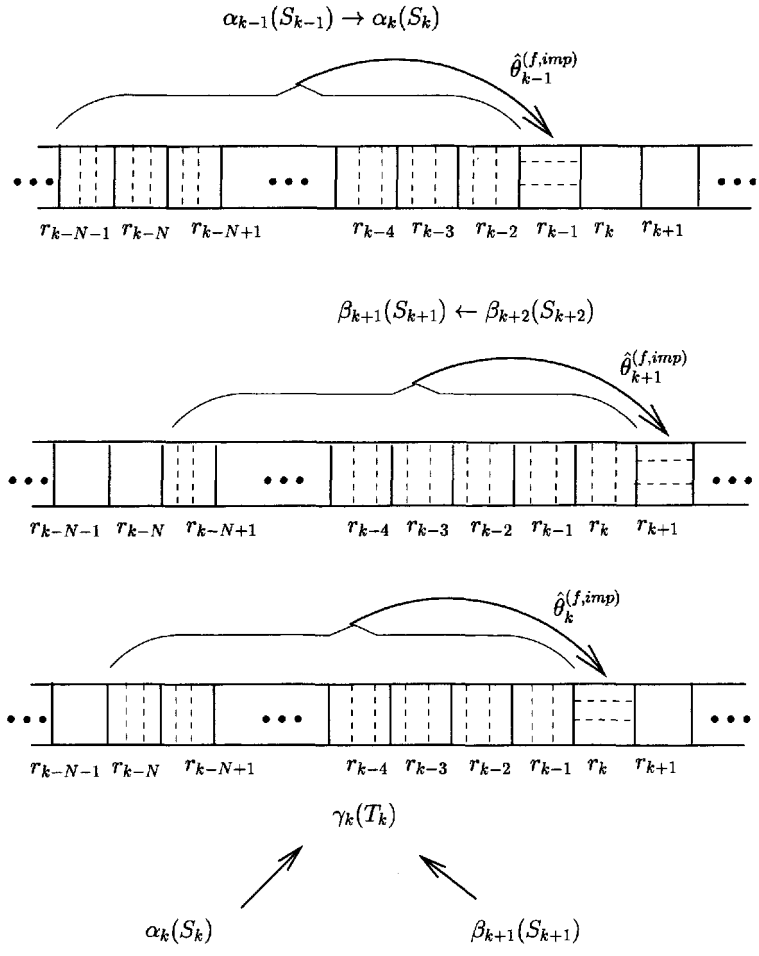


Figure 4.15: Implicit phase estimation in the forward recursion, backward recursion and completion for the NCSOa algorithm.

convolutional code. The corresponding generator matrices are given by

$$\boldsymbol{G}_{\mathrm{o}}(D) = \left[1 \;\; \frac{1+D^2}{1+D+D^2}\right] \qquad \boldsymbol{G}_{\mathrm{i}}(D) = \begin{bmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} \\ 0 & 1 & \frac{1+D}{1+D+D^2} \end{bmatrix} \tag{4.130}$$

where $\boldsymbol{G}_{\mathrm{o}}(D)$ is the generator matrix of the outer code and $\boldsymbol{G}_{\mathrm{i}}(D)$ is the generator matrix of the inner code. The output symbols are mapped to an 8-ary phase shift keying (8-PSK) constellation with natural mapping, resulting in an overall code of spectral efficiency of 1 b/s/Hz. The inner adaptive decoder uses either the NCSOa algorithm or the NCSOb algorithm, while the outer decoder uses the coherent FB

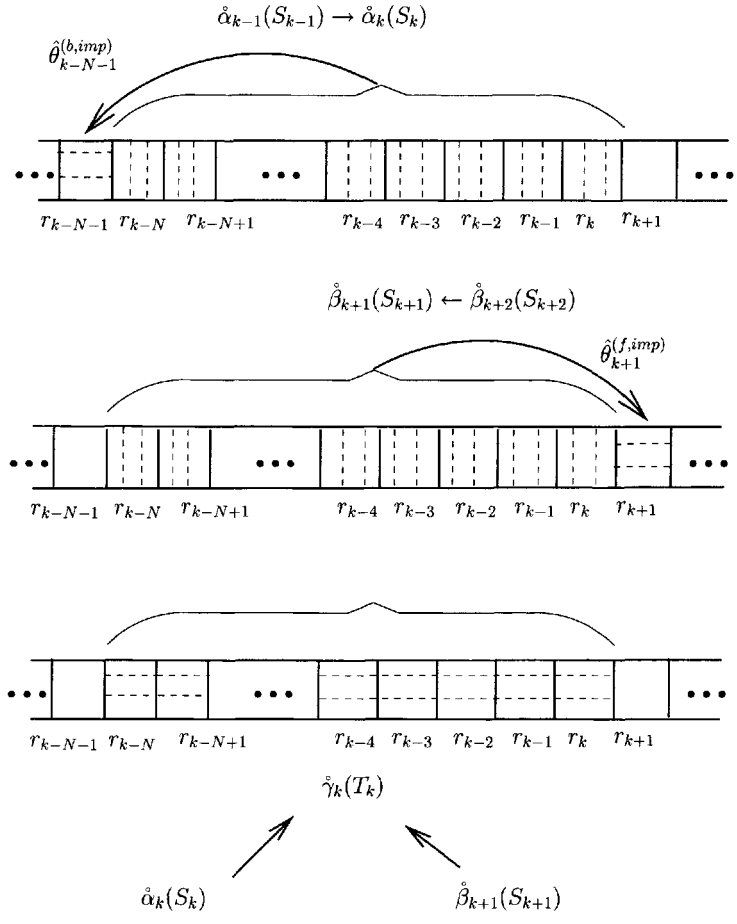


Figure 4.16: Implicit phase estimation in the forward recursion, backward recursion and completion for the NCSOb algorithm.

algorithm. The performance when considering both NCSOa and NCSOb algorithms is shown in Figure 4.18 in various cases. In particular, each setting is identified by a couple of parameters (N, Q). The parameter N has the usual meaning, but in this case its value is decoupled from the complexity of the considered algorithm. In fact, the *reduced-state parameter* Q is related to the actual memory of the decoder, i.e., the number of states in the corresponding trellis. In Figure 4.18, the performance is shown for $(N, Q) = (5, 2)$ and (7,2), considering both algorithms. For comparison, the performance of a coherent receiver is also shown. In all cases, 1 and 5 decoding iterations are considered. As one can see, the performance of the NCSOa algorithm with $(N, Q) = (7, 2)$ is the same as that of the NCSOb algorithm with $(N, Q) =$

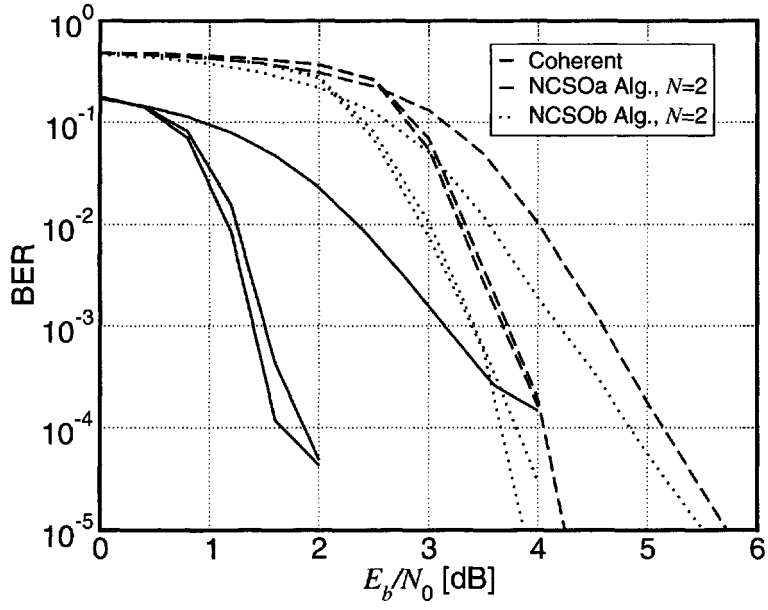


Figure 4.17: Noncoherent iterative decoding of a PCCC with BPSK. For comparison the performance of the corresponding coherent receiver is also shown. In all cases 1, 5, and 10 decoding iterations are considered.

$(5, 2)$. As the inner code rate is equal to 2/3, the information symbols at its input are quaternary. The complexity of the NCSOb algorithm is then $(2M + 1)/3 = (2 \cdot 4 + 1)/3 = 3$ times that of the NCSOa algorithm. For a comparable complexity level, the NCSOa algorithm, besides being simpler, shows a better performance than the NCSOb algorithm.

4.8 State Reduction Techniques for Forward Backward Algorithms

In this section, we propose a general approach to obtaining reduced-complexity FB algorithms by means of state reduction techniques.

In particular, in Section 4.8.1 a very intuitive trellis-based state reduction technique is presented, suggested by the systematic relationship between the FB algorithm and VA—in fact, as shown in Section 4.6, in both cases the same metric (considering the FB algorithm in the logarithmic domain) is used. Considering a reduced-state trellis, the key idea is simple. During the forward recursion a *survivor*, based

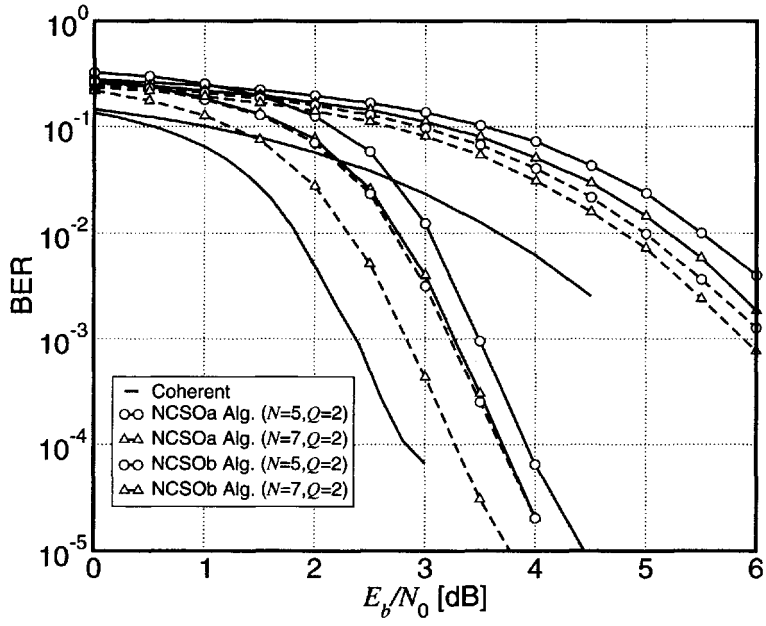


Figure 4.18: Noncoherent iterative decoding of an SCCC with 8-PSK. For comparison the performance of the corresponding coherent receiver is also shown. In all cases 1 and 5 decoding iterations are considered.

on decision feedback, is associated with each state at each epoch. The ensemble of the survivors is then used in the backward recursion. We will refer to the obtained reduced-state (RS) algorithms as *forward-only* (Fwd-only) RS-FB algorithms.

In Section 4.8.5, the proposed state reduction technique is generalized, with a final formulation which includes, as special cases, Fwd-only RS-FB algorithms and *bi-directional* (BiD) RS-FB algorithms, i.e., RS-FB algorithms where the survivors are constructed in both recursions, as originally proposed in [151].

4.8.1 Forward-Only RS-FB Algorithms

For simplicity, we refer to a general FB algorithm for channels with memory as proposed in Section 4.6, and relative to a communication system based on the serial concatenation of an encoder/modulator and a channel with stochastic uncertainty. The total memory of the system is limited on the basis of an FMC (or a suitable CMP). In the following, we will assume that the portion of memory to be reduced is relative to the channel. This assumption can be relaxed, and in Section 4.8.5 we will consider the application of state reduction techniques to more general cases, for

example to decode convolutional codes.

Defining μ_k as the state of the encoder/modulator, at the receiver side we consider an expanded state $S_k \triangleq (\mu_{k-N}, \boldsymbol{a}_{k-N}^{k-1})$, where $\{a_k\}$ is a sequence of independent and identically distributed M-ary information symbols at the input of the encoder and N is the order of Markovianity. The corresponding transition is defined as $T_k = (S_k, a_k)$. We use the same notation introduced in Section 4.6. As an example, we consider the case where the channel introduces an infinite memory and the CMP is applied. In this case, the quantities used in the FB algorithms derived in Section 4.6 have the following expressions:

$$\alpha_k(S_k) \triangleq p(\boldsymbol{r}_0^{k-1}|S_k)P\{S_k\} \tag{4.131}$$

$$\beta_k(S_k) \triangleq p(\boldsymbol{r}_k^{K-1}|\boldsymbol{r}_{k-N}^{k-1}, S_k) \tag{4.132}$$

$$\gamma_k(T_k) \triangleq p(r_k|\boldsymbol{r}_{k-N}^{k-1}, T_k)P\{a_k\} \tag{4.133}$$

where r_k is the channel observation at epoch k after suitable filtering and sampling. As considered in Section 4.6, we assume for simplicity that there is one sample r_k per information symbol a_k. We recall that the quantities $\alpha_k(S_k)$ and $\beta_k(S_k)$ can be computed by means of forward and backward recursions, respectively, as in (4.97) and (4.99). The generated soft output can be written as follows:

$$\mathrm{S}[a_k] = \sum_{T_k:a_k} \alpha_k(S_k)\gamma_k(T_k)\beta_{k+1}(S_{k+1}). \tag{4.134}$$

Given that the number of states $\{\mu_k\}$ of the encoder/modulator is S_c, the total number of states of the receiver is $\zeta = S_c M^N$. We simply define a reduced state as

$$s_k \triangleq (\mu_{k-Q}, \boldsymbol{a}_{k-Q}^{k-1}) \tag{4.135}$$

where Q will be referred to as the *reduced-state parameter*. One can immediately recognize that the reduced state s_k is obtained from S_k by substituting the order of Markovianity N with the reduced-state parameter Q. A transition in the reduced-state trellis is defined as $\epsilon_k \triangleq (s_k, a_k, s_{k+1})$. This redundant definition simplifies the derivation of the following state reduction techniques. The number of states in the reduced-state trellis is $\zeta' = S_c M^Q$. Given a transition ϵ_k, in order to compute $\gamma_k(\epsilon_k)$ it is necessary to somehow recover the information symbols $\boldsymbol{a}_{k-N}^{k-Q-1}$. A simple approach based on reduced-state sequence detection (RSSD) [34, 152], as shown in Chapter 3, is that of associating a *forward* survivor with each reduced state s_k.[20] We

[20]More complex techniques, such as those based on set partitioning introduced in Chapter 3 for sequence detection, may also be employed [153, 154]. For the sake of simplicity, we consider only state reduction techniques based on memory truncation.

define this survivor by the general notation $\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k)$, with the implicit assumption that it includes any "segment" of information symbols $\tilde{\boldsymbol{a}}_{k-Q-m}^{\mathrm{f},k-Q}$, $\forall m \geq 0$, embedded in the survivor associated with state s_k. A more refined formalism will be considered in Section 4.8.3. The ensemble of the survivors for all states and all epochs is denoted by $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ and we call it the (forward) *survivor map*.

Assuming a forward survivor map has been constructed, it is reasonable to compute the soft output relative to a symbol a_k as follows:

$$\mathrm{S}[a_k] = \sum_{\epsilon_k : a_k} \alpha_k(s_k)\gamma_k(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), \epsilon_k)\beta_{k+1}(s_{k+1}). \tag{4.136}$$

In particular, it is worth remarking that the quantities $\alpha_k(s_k)$ and $\beta_{k+1}(s_{k+1})$ are formally equivalent to the corresponding quantities in the full-state case, but, due to state reduction, they are quantitatively different.

We now show how a forward survivor map $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ can be recursively constructed during the forward recursion. We assume that the forward survivors are known up to epoch $k-1$. A generic step at epoch k of the forward recursion can be written as follows:

$$\begin{aligned}
\alpha_k(s_k) &= p(\boldsymbol{r}_0^{k-1}|s_k)P\{s_k\} \\
&= \sum_{\epsilon_{k-1}:s_k} \underbrace{p(r_{k-1}|\boldsymbol{r}_0^{k-2}, \epsilon_{k-1})}_{p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \epsilon_{k-1})\,(\mathrm{CMP})} \; P\{a_k|\boldsymbol{r}_0^{k-2}, s_{k-1}\} \\
&\qquad \cdot p(\boldsymbol{r}_0^{k-2}|s_{k-1})P\{s_{k-1}\} \\
&= \sum_{\epsilon_{k-1}:s_k} p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \epsilon_{k-1}) \underbrace{P\{a_k|\boldsymbol{r}_0^{k-2}, s_{k-1}\}}_{P\{a_k\}\,(\mathrm{caus.\ and\ indep.})} p(\boldsymbol{r}_0^{k-2}|s_{k-1})P\{s_{k-1}\} \\
&= \sum_{\epsilon_{k-1}:s_k} p(\boldsymbol{r}_0^{k-2}|s_{k-1})P\{s_{k-1}\}p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \epsilon_{k-1})P\{a_k\}.
\end{aligned} \tag{4.137}$$

As the survivors are known up to epoch $k-1$, a reasonable way to compute the pdf $p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \epsilon_{k-1})$ is the following:

$$\begin{aligned}
p(r_{k-1}|\boldsymbol{r}_0^{k-2}, \epsilon_{k-1}) &\simeq p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), \epsilon_{k-1}) \\
&= p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \tilde{\boldsymbol{a}}_{k-N-1}^{\mathrm{f},k-Q}, \epsilon_{k-1})
\end{aligned} \tag{4.138}$$

so that one can write:

$$\gamma_{k-1}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), \epsilon_{k-1}) = p(r_{k-1}|\boldsymbol{r}_{k-N-1}^{k-2}, \tilde{\boldsymbol{a}}_{k-N-1}^{\mathrm{f},k-Q}, \epsilon_{k-1})P\{a_{k-1}\}. \tag{4.139}$$

Hence, the forward recursion can be written as

$$\alpha_k(s_k) = \sum_{\epsilon_{k-1}:s_k} \alpha_{k-1}(s_{k-1})\gamma_{k-1}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), \epsilon_{k-1}). \tag{4.140}$$

The forward survivor associated with state s_k is associated with the survivor $\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}^{\max})$ of the state $s_{k-1}^{\max}$ such that the corresponding term in the sum (4.140) is the largest one.

Once the forward survivor map has been constructed, it can be used in the backward recursion, which can be written in the following way:

$$\beta_k(s_k) = \sum_{\epsilon_k:s_k} \beta_{k+1}(s_{k+1})\gamma_k(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), \epsilon_k). \tag{4.141}$$

The computational complexity of the forward and backward recursions is related to the number of branches in a section of the receiver trellis, which is proportional to the number of states. Hence, the complexity of the proposed RS-FB algorithms is approximately $M^{(N-Q)}$ times lower than that of the full-state case.

A direct extension of the Fwd-only RS-FB algorithm proposed in this section can be defined the as *backward-only* (Bwd-only) RS-FB algorithm. In this case, a *backward survivor map* $\tilde{\boldsymbol{a}}^{\mathrm{b}}$ is constructed during the backward recursion, run first, and then used in the forward recursion and in the completion. We remark that a linear time-invariant (LTI) maximum-phase system is a causal and stable system with the maximum energy delay property among all systems with the *same* frequency response magnitude [155]. As we will see in the case of iterative detection for inter-symbol interference (ISI) channels, use of Bwd-only RS-FB algorithms can be greatly beneficial for maximum-phase ISI channels.

The basic structural idea can be further generalized. One could construct two distinct survivor maps during forward and backward recursions, and use an extended metric in the completion. This has been partially proposed in [151], and will be considered in Section 4.8.5.

4.8.2 Examples of Application of Fwd-Only RS-FB Algorithms

As examples of application of the Fwd-only state reduction technique for FB algorithms introduced in Section 4.8.1, we consider the cases of coherent detection for an ISI channel (i.e., assuming perfect knowledge of the ISI channel coefficients), noncoherent detection (using the noncoherent FB algorithm, indicated as NCSOa, introduced in Section 4.6) and linear predictive detection based on linear prediction for a Rayleigh flat fading channel.

ISI Channel

In the case of an ISI channel, we assume uncoded transmission, i.e., there is no state μ_k inside the expanded state S_k (suitably defined as explained in the following). The observation sample at the output of a whitened matched filter (WMF) can be expressed as [100]

$$r_k = \sum_{i=0}^{L} a_{k-i} f_i + n_k. \tag{4.142}$$

The noise samples $\{n_k\}$ are real independent Gaussian random variables, with zero mean and variance σ^2. The distortion on the elementary shaping pulse makes each sample r_k dependent on L information symbols, each weighted by a different coefficient f_i. By defining an expanded state[21] as $S_k \triangleq \boldsymbol{a}_{k-L}^{k-1}$ (consequently, $T_k = \boldsymbol{a}_{k-L}^{k}$), it is easy to show that the FB algorithm can be directly applied [54]. Hence, the metric $\gamma_k(T_k)$ reduces to

$$\begin{aligned} \gamma_k(T_k) &= p(r_k|\boldsymbol{r}_{k-N}^{k-1}, T_k) P\{a_k\} \\ &= p(r_k|T_k) P\{a_k\} \\ &\sim \exp\left\{ -\frac{|r_k - \sum_{i=0}^{L} a_{k-i} f_i|^2}{2\sigma^2} \right\} P\{a_k\} \end{aligned} \tag{4.143}$$

where the symbol $\sim$ denotes proportionality.

We remark that an LTI system is minimum-phase if it is causal and stable and has a causal and stable inverse [155]. A minimum-phase system has the minimum energy delay property among all systems with the *same* frequency response magnitude [155]. If the impulse response of the equivalent time-discrete channel $\{f_i\}_{i=0}^{L}$ is minimum-phase, effective state reduction is obtained by defining a reduced state $s_k = \boldsymbol{a}_{k-Q}^{k-1}$, with $Q < L$, and, consequently, a transition $\epsilon_k = (s_k, s_{k+1}) \equiv \boldsymbol{a}_{k-Q}^{k}$ [100, 152–154]. Hence, searching in the survivor path associated with state s_k, it is possible to recover[22] $\tilde{\boldsymbol{a}}_{k-L}^{\mathrm{f},k-Q-1}$ and then write

$$\gamma_k(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), \epsilon_k) \sim \exp\left\{ -\frac{|r_k - \sum_{i=0}^{Q} a_{k-i} f_i - \sum_{j=Q-1}^{L} \tilde{a}_{k-j}^{\mathrm{f}} f_j|^2}{2\sigma^2} \right\}. \tag{4.144}$$

[21] This would correspond to setting the finite memory parameter C to the length L of the ISI channel, and suppressing the dependence of an observation from previous observations.

[22] For the sake of notational simplicity, in the following we do not explicitly indicate the dependence of the information symbols associated with the survivor path from the state s_k. The correct interpretation should be clear from the context.

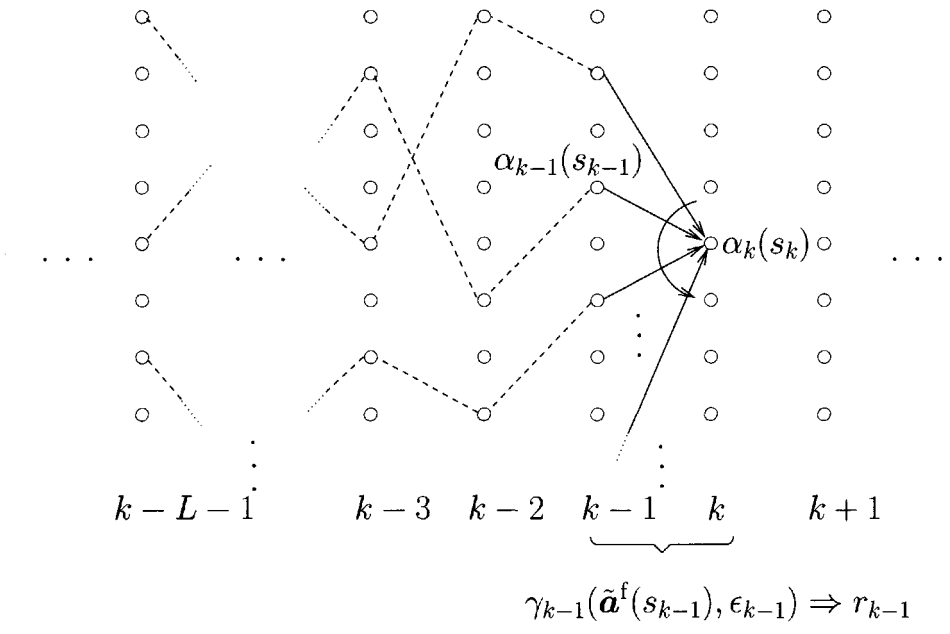


Figure 4.19: Forward recursion for the computation of $\alpha_k(s_k)$ for a Fwd-only RS-FB algorithm in the case of coherent detection for an ISI channel. Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

In Figure 4.19, it is shown how the forward recursion proceeds in the reduced-state trellis in the case of coherent detection for an ISI channel. In this case, we associate with each state s_k a state s_{k-1}, chosen among the beginning states of the M branches ending in s_k, according to the corresponding path metrics—the chosen state s_{k-1} is such that the corresponding path metric is the largest. The backward recursion proceeds similarly, by using the survivors selected during the forward recursion.

If the overall impulse response $\{f_i\}_{i=0}^{L}$ is maximum-phase, e.g., if a causal whitening filter is selected [100], efficient definitions of trellis state and transition are $s'_k = \boldsymbol{a}_{k-L}^{k-L+Q-1}$ and $\epsilon'_{k+1} = \boldsymbol{a}_{k-L}^{k-L+Q}$, respectively. A Bwd-only RS-FB algorithm can be designed, which starts with the backward recursion, constructs a backward survivor map compatible with the new reduced state s'_k, and runs the forward recursion. Related reverse-time processing structures, suited to impulse responses with energy concentrated towards the end, are considered in [157, 158]. In this "specular" version of the previously introduced algorithm, during the backward recursion the information symbol a_{k-L} is relative to the transition from state s'_{k+1} to state s'_k and symbol a_{k-L+Q} is discarded. The formulation is a straightforward extension of that previously introduced, the only modification being a termination of the reduced-state

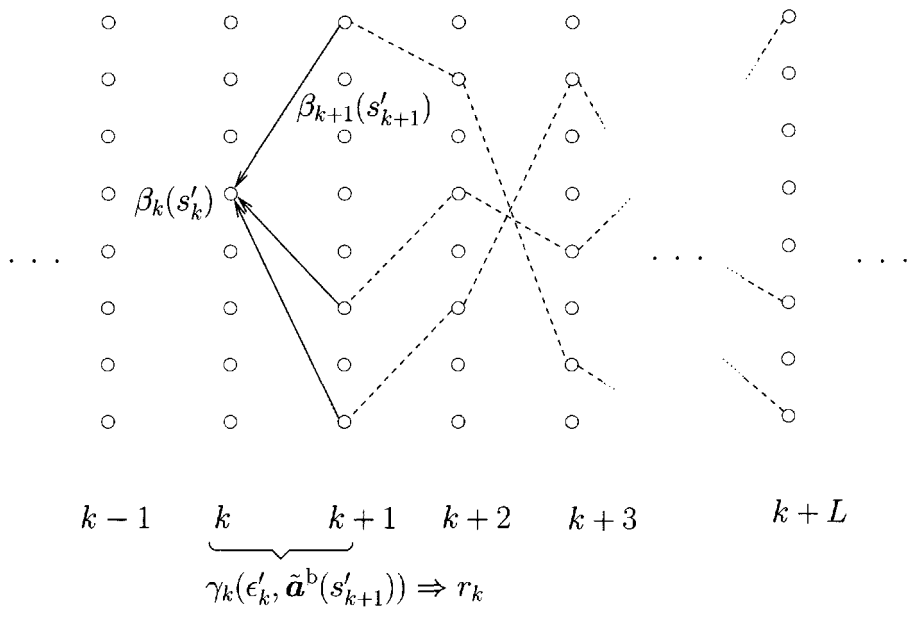


Figure 4.20: Backward recursion for the computation of $\beta_k(s'_k)$ for a Bwd-only RS-FB algorithm in the case of coherent detection for an ISI channel. The survivor map is constructed during this recursion. Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

trellis necessary to better initialize the backward recursion. Figure 4.20 schematically shows how the backward recursion proceeds in a Bwd-only RS-FB algorithm.

This example leads to a more general conclusion on the possible applications of the proposed state reduction techniques. Survivors can be constructed during both recursions, depending on the overall channel impulse response and, consequently, on the structure of the observations. As an example, a proper use of survivor maps constructed during both recursions may prove useful if the overall impulse response is "mixed-phase."[23] This will be considered in detail in Section 4.8.5.

As an example of coherent iterative detection for an ISI channel we consider the scheme of iterative equalization proposed in [73]. The performance is assessed by means of computer simulations in terms of BER versus the SNR E_b/N_0, E_b being the received signal energy per information bit and N_0 being the one-sided noise power

[23] In this context, the term "mixed-phase" channel will be loosely used for channels in which most of the energy is not concentrated around the channel coeffi cients that are associated with the most recent or the earliest input symbols.

spectral density. In any component decoder we consider a simplified logarithmic version of the corresponding FB or RS-FB algorithm [156]. More precisely, we consider a binary ($M = 2$) transmission system characterized by a rate-1/2 16-state RSC encoder with generators $G_1 = (23)_8$, $G_2 = (35)_8$ (octal notation), followed by a 64×64 pseudorandom interleaver [33]. The bits at the output of the interleaver are sent through the channel with BPSK. The minimum-phase discrete-time channel impulse response is characterized by the following coefficients [73]:

$$\boldsymbol{f} = (\sqrt{0.45}, \sqrt{0.25}, \sqrt{0.15}, \sqrt{0.10}, \sqrt{0.05}). \tag{4.145}$$

The receiver is based on a serial concatenation of an inner detector, which uses the RS-FB algorithm, and an outer decoder which is a standard FB module [83]. The extrinsic information is used according to the heuristic method proposed in Section 4.5. By trial and error, we found that a good performance is obtained when the extrinsic information generated by the inner detector is weighted, i.e., multiplied, in the logarithmic domain, by a constant[24] equal to 0.6 and the extrinsic information generated by the outer decoder is weighted by a constant equal to 1 (i.e. it is not modified). The state reduction technique is applied to the inner detector. In Figure 4.21, the performance of the full-state receiver (inner detector with $\zeta = 16$ states) is compared with the performance of the receiver with reduced complexity (inner detector with $\zeta' = 8, 4$ or 2 states). In all cases, we consider 1 and 6 decoding iterations. At 6 decoding iterations, the performance loss for a detector with $\zeta' = 4$ states with respect to that of a receiver without state reduction is only 0.75 dB at a BER of 10^{-4}, and it reduces to 0.25 dB for a reduced-state detector with $\zeta' = 8$ states. For comparison, the performance in the absence of ISI, i.e., for coded transmissions over an AWGN channel, is also shown. In this case, the receiver reduces to the outer decoder of the RSC code considered above.

Noncoherent Channel

We consider the transmission scenario described at the end of Section 4.7, and we assume linear modulation. The channel introduces an unknown phase rotation, which is modeled as a time-invariant random variable with uniform distribution in $[0, 2\pi)$. A sample at the output of a matched filter has the following expression:

$$r_k = c_k e^{j\theta} + n_k \tag{4.146}$$

[24]Note that in Section 4.5, since we were referring to LLRs, the multiplicative constant used in the heuristic method had to be lower than 0.5. However, without considering the LLR and referring directly to the generated soft-output values, the multiplicative constant is between 0 and 1.

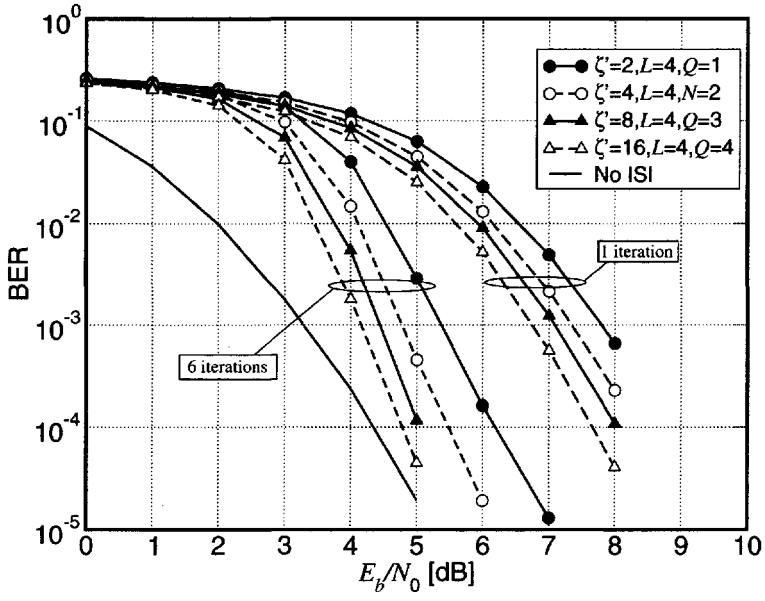


Figure 4.21: Application of the proposed technique to iterative decoding/detection for an ISI channel. Receivers with various levels of complexity are considered and compared with the full-state receiver ($\zeta = 16$). The considered numbers of iterations are 1 and 6 in all cases. The performance in the case of coded transmission over an AWGN channel, without ISI, is also shown (solid lines with circles). Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

where c_k represents the modulated, and possibly coded, symbol and n_k is a complex AWGN sample with variance per component equal to σ^2. Considering the NCSOa algorithm derived and used in Section 4.7, the exponential metric $\gamma_k(T_k)$ can be expressed as follows:

$$\gamma_k(T_k) \sim \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|\right)} P\{a_k\}. \tag{4.147}$$

Introducing a reduced-state parameter Q, which replaces N, a reduced-state trellis is obtained and a forward survivor map $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ can be constructed during the forward recursion. Based on this map, a full-state metric can be associated with each reduced

transition ϵ_k as follows

$$\gamma_k(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k),\epsilon_k) = \gamma_k(\tilde{\boldsymbol{a}}_{k-N}^{\mathrm{f},Q-1},\epsilon_k) = \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \cdot \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{Q} r_{k-i}c_{k-i}^* + \sum_{i=Q-1}^{N} r_{k-i}\left(\tilde{c}_{k-i}^{\mathrm{f}}\right)^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{Q} r_{k-i}c_{k-i}^* + \sum_{i=Q-1}^{N} r_{k-i}\left(\tilde{c}_{k-i}^{\mathrm{f}}\right)^*\right|\right)} P\{a_k\} \quad (4.148)$$

where the notation $\tilde{c}_{k-i}^{\mathrm{f}}$ is relative to a coded and modulated symbol obtained by using the constructed forward survivors. We do not show any numerical result relative to this particular application, since in Section 4.9.1 detailed examples of noncoherent iterative detection schemes will be proposed.

Fading Channel

We consider transmission of differentially encoded M-ary phase shift keying (M-PSK) over a Rayleigh flat fading channel. We refer to the transmission system considered in [159] and denote by $f_{\mathrm{D}}T$ the normalized Doppler rate of the channel. We assume that each information symbol a_k corresponds to a group of $\log_2 M = m$ bits, i.e., $a_k = (a_k^{(1)},\ldots,a_k^{(m)})$. The bits $(a_k^{(1)},\ldots,a_k^{(m)})$ are mapped to an M-PSK symbol a_k through Gray mapping. The differentially encoded M-ary symbols c_k are defined by the rule $c_k = a_k c_{k-1}$, with $c_0 = 1$ (c_0 acts as a reference symbol). The corresponding received signal at the matched filter output can be written as

$$r_k = f_k c_k + n_k \quad (4.149)$$

where the sequence of channel coefficients $\{f_k\}$ is a complex random sequence whose real and imaginary components are independent, Gaussian with zero mean, and $\{n_k\}$ are samples of a zero-mean complex-valued circularly symmetric white Gaussian noise process. The autocorrelation of the fading process is assumed to follow the classical isotropic scattering model [20].

Assuming that the information bits are independent within each symbol, we can consider $P\{a_k\} = P\{a_k^{(1)}\}\cdots P\{a_k^{(m)}\}$. Considering a bit-wise interleaver as in [159], soft outputs on bits $\{a_k^{(i)}\}$ have to be calculated. To obtain these soft outputs, equation (4.92) (or (4.134)) can be modified as follows:

$$\mathrm{S}[a_k^{(i)}] = \sum_{T_k:a_k^{(i)}} \alpha_k(S_k)\gamma_k(T_k)\beta_{k+1}(S_{k+1}) \quad (4.150)$$

where the notation $T_k : a(T_k)^{(i)}$ indicates all the transitions $\{T_k\}$ compatible with the i-th bit component of the symbol a_k associated with T_k. Since the specific realizations of the channel coefficients $\{f_k\}$ are unknown, linear prediction [101, 123, 125, 126] can be used to derive estimates of these coefficients. Denoting by N the prediction order and setting the finite memory parameter as $C = N$, the metric $\gamma_k(T_k)$ can be written as follows [159]:

$$\begin{aligned}
\gamma_k(T_k) &= p(r_k|\boldsymbol{r}_{k-C}^{k-1}, T_k)P\{a_k\} = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, T_k)P\{a_k\} \\
&\sim \exp\left(-\frac{|r_k - \hat{h}_k c_k|^2}{\sigma_{\text{pre}}^2}\right) P\{a_k\} \\
&= \exp\left(-\frac{\left|r_k - c_k \sum_{n=1}^{N} p_n r_{k-n} c_{k-n}^*\right|^2}{\sigma_{\text{pre}}^2}\right) P\{a_k\}
\end{aligned} \tag{4.151}$$

where $\hat{f}_k$ is the fading gain prediction, $\{p_n\}$ are the prediction coefficients and σ_{pre}^2 is the variance of the prediction error, which can be computed as shown in [160]. We may observe that $c_k c_{k-i}^* = \prod_{j=0}^{i-1} a_{k-j}$. By defining $S_k \triangleq \boldsymbol{a}_{k-N+1}^{k-1}$, the corresponding trellis diagram has M^{N-1} states.[25] Assuming that the autocorrelation sequence of the channel fading process is known, the optimal prediction coefficients $p_n, 1 \leq n \leq N$,

[25]Note that this definition of state depending on only $(N-1)$ symbols, instead of $C = N$, is due to the use of differential encoding.

are obtained by solving Wiener Hopf equations [159]. Hence

$$\begin{aligned}\gamma_k(T_k) &\sim \exp\left(-\frac{\left|r_k-\sum_{n=1}^{N}p_n r_{k-n}\prod_{j=0}^{n-1}a_{k-j}\right|^2}{\sigma^2_{\text{pre}}}\right)P\{a_k\}\\ &= \exp\left(-\frac{|r_k|^2+|a_k|^2\left|\sum_{n=1}^{N}p_n r_{k-n}\prod_{j=1}^{n-1}a_{k-j}\right|^2}{\sigma^2_{\text{pre}}}\right)\\ &\quad\cdot\exp\left(\frac{2\Re\left\{r_k^* a_k\sum_{n=1}^{N}p_n r_{k-n}\prod_{j=1}^{n-1}a_{k-j}\right\}}{\sigma^2_{\text{pre}}}\right)P\{a_k\}. \quad (4.152)\end{aligned}$$

Noting that $|a_k|^2$ and $|r_k|^2$ do not depend on T_k and defining

$$g(S_k,\boldsymbol{r}_{k-N}^{k-1}) \triangleq \sum_{n=1}^{N}p_n r_{k-n}\prod_{j=1}^{n-1}a_{k-j} \quad (4.153)$$

one finally obtains

$$\gamma_k(T_k) \sim \exp\left(-\frac{|g(S_k,\boldsymbol{r}_{k-N}^{k-1})|^2}{\sigma^2_{\text{pre}}}+\frac{2\Re\left\{r_k^* a_k g(S_k,\boldsymbol{r}_{k-N}^{k-1})\right\}}{\sigma^2_{\text{pre}}}\right)P\{a_k\}. \quad (4.154)$$

The first term inside the exponential in (4.154) was dropped in [159]. However, it depends on S_k and should not be neglected, unless one assumes perfect CSI.

To reduce the number of trellis states, we proceed as in the case of an ISI channel. We may define a reduced state $s_k = \boldsymbol{a}_{k-Q+1}^{k-1}$, with $Q < N$, obtaining (the formulation

is similar to that used in Section 4.8.2)

$$
\begin{aligned}
g(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), s_k, \boldsymbol{r}_{k-N}^{k-1}) &= \sum_{n=1}^{N} p_n r_{k-n} \prod_{j=1}^{n-1} a_{k-j} + \sum_{n=Q+1}^{N} p_n r_{k-n} \prod_{j=1}^{n-1} \tilde{a}_{k-j}^{\mathrm{f}} \\
\gamma_k(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), \epsilon_k) &\sim \exp\left(-\frac{|g(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s^-(\epsilon_k)), s^-(\epsilon_k), \boldsymbol{r}_{k-N}^{k-1}), s^-(\epsilon_k))|^2}{\sigma_{\mathrm{pre}}^2} \right) \\
&\quad \cdot \exp\left(\frac{2\Re\{r_k^* a_k g(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s^-(\epsilon_k)), s^-(\epsilon_k), \boldsymbol{r}_{k-N}^{k-1}), s^-(\epsilon_k))\}}{\sigma_{\mathrm{pre}}^2} \right) \\
&\quad \cdot P\{a_k\} \qquad (4.155)
\end{aligned}
$$

where the notation $s^-(\epsilon_k)$ indicates the beginning state s_k of transition ϵ_k. Similarly, hereafter the notation $s^+(\epsilon_k)$ will refer to the ending state s_{k+1} of transition ϵ_k. An identical notation will be used in the full-state case, too.

As an example of application, we consider transmission of differentially encoded quaternary PSK (DQPSK) signals over a flat fading channel, as in [159]. The outer code is a 64-state nonrecursive convolutional (NRC) code with generators $G_1 = (133)_8$ and $G_2 = (171)_8$. This code is concatenated, through a 64×64 nonuniform bit interleaver, with an inner differential encoder. In fact, bit interleaving is an appropriate means to combat the effects of fading [65, 161]. The normalized fading rate is $f_{\mathrm{D}}T = 0.01$. The differential inner detector uses linear prediction and state reduction. In Figure 4.22, the performance of the full-state receiver, with prediction orders $N = Q = 3$ and 4, respectively, is compared with the performance of a receiver with various levels of complexity reduction, specified by the couple of parameters (N, Q). In all cases, 1 and 6 decoding iterations are considered. At 6 decoding iterations, a detector with $N = 3$ and $Q = 2$ exhibits a performance loss of only 0.2 dB at a BER of 10^{-4}, with respect to the full-state detector ($N = Q = 3$). For $(N, Q) = (5, 3)$, the performance gain with respect to the full-state receiver with $N = 3$ is about 0.6 dB at a BER of 10^{-4}. For comparison, the performance curve in the case of perfect knowledge of the fading coefficients (coherent detection) is also shown.

4.8.3 Forward-Only RS FB-type Algorithms

We now consider the FB-type algorithms proposed in Section 4.7. We consider the same communication scenario (an encoder/modulator followed by a channel with memory) as in Section 4.6, and we adopt the same formalism. We recall the defini-

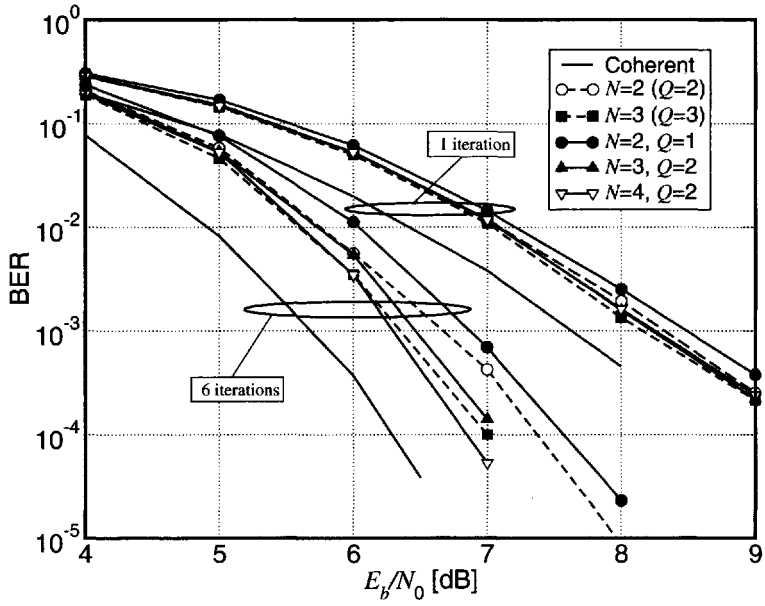


Figure 4.22: Application of the proposed Fwd-only state reduction technique to iterative decoding/detection, through linear prediction, for flat fading channels with $f_D T = 0.01$. Receivers with various levels of complexity (in terms of prediction order N and reduced-state parameter Q) are shown. The considered numbers of iterations are 1 and 6 in all cases. The performance in the case of decoding with perfect knowledge of the fading coefficients is also shown (solid lines). Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

tions of the following quantities:

$$\mathring{\alpha}_k(T_k) = p(\boldsymbol{r}_0^{k-N-1} | \boldsymbol{r}_{k-N}^{k}, T_k) \tag{4.156}$$

$$\mathring{\gamma}_k(T_k) = p(\boldsymbol{r}_{k-N}^{k} | T_k) P\{S_k\} P\{a_k\} \tag{4.157}$$

$$\mathring{\beta}_k(T_k) = p(\boldsymbol{r}_{k+1}^{K-1} | \boldsymbol{r}_{k-N}^{k}, T_k) \tag{4.158}$$

where N is the order of Markovianity. According to the structure of the introduced FB-type algorithms, the soft-output reliability value on an information symbol a_k can be expressed as follows:

$$\mathrm{S}[a_k] = \sum_{T_k : a_k} \mathring{\alpha}_k(T_k) \mathring{\gamma}_k(T_k) \mathring{\beta}_k(T_k). \tag{4.159}$$

where the summation is extended over all transitions of epoch k compatible with the information symbol a_k, and $P\{S_k\}$ represents the *a priori* probability of state S_k,

respectively. As seen in Section 4.7, $\mathring{\alpha}_k(T_k)$ and $\mathring{\beta}_k(T_k)$ can be computed by means of forward and backward recursions, as in (4.119) and (4.123), respectively.

We proceed exactly as in the case of Fwd-only RS-FB algorithms. We consider a trellis diagram with a reduced number of states $\zeta' < \zeta$ and denote its generic state by $s_k \equiv (\mu_{k-Q}, \boldsymbol{a}_{k-Q+1}^k)$, with $Q < N$. The question of how to define a survivor in the case of RS FB-type algorithms arises.

A full-state FB-type algorithm runs first a forward recursion to compute $\mathring{\alpha}_k(T_k)$ for each transition T_k and epochs $k = 0, 1, \ldots, K-1$. Due to the structure of FB-type algorithms, α_k is associated with a transition T_k, instead of a single state S_k as in the case of an FB algorithm. Hence, a survivor associated with a single transition has to be defined. By considering the algorithm in the logarithmic domain, the forward recursion of α_k (equation (4.119)) can be expressed as

$$\ln \mathring{\alpha}_k(T_k) = \ln \sum_{T_{k-1}:S^+(T_{k-1})=S^-(T_k)} \exp\left\{\ln \psi_k(T_{k-1}, T_k) + \ln \mathring{\alpha}_{k-1}(T_{k-1})\right\} \tag{4.160}$$

where $S^+(T_{k-1})$ indicates the final state of transition T_{k-1} and $S^-(T_k)$ indicates the initial state of transition T_k. We can approximate this recursion by using the max-log approximation [137], obtaining

$$\ln \mathring{\alpha}_k(T_k) \simeq \max_{T_{k-1}:S^+(T_{k-1})=S^-(T_k)} \left\{\ln \psi_k(T_{k-1}, T_k) + \ln \alpha_{k-1}(T_{k-1})\right\}. \tag{4.161}$$

In equation (4.161), it is intuitive to interpret the term to be maximized as a "metric" associated with a path ending with transition T_k, as in a classical VA. However, this VA is an "extended" VA, where a single step involves two consecutive trellis transitions.

Considering the reduced-state trellis and assuming that the forward survivor (FS) of each transition ϵ_{k-1} is known, we now show how the survivors can be extended to epoch k by using the forward recursion. We define[26] by $\mathrm{FS}_{k-j}^{(i)}(\epsilon_{k-1})$ the sequence of i transitions reaching epoch $k-j$ along the survivor of transition ϵ_{k-1}, i.e., $\mathrm{FS}_{k-j}^{(i)}(\epsilon_{k-1}) = \tilde{\boldsymbol{\epsilon}}_{k-j-i+1}^{k-j} \equiv \tilde{\boldsymbol{a}}_{k-j-i-Q+1}^{\mathrm{f},k-j-Q}$. Any transition $\tilde{\epsilon}_{k-l}$, for $l \in \{j, \ldots, j+i-1\}$, in $\mathrm{FS}_{k-j}^{(i)}(\epsilon_{k-1})$ and any information symbol $\tilde{a}_{k-j-h}^{\mathrm{f}}$, for $h \in \{Q, \ldots, Q+i-1\}$, depend on the transition ϵ_{k-1}. Therefore, the couple $(\mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1})$ uniquely identifies the sequence of information symbols $(\tilde{\boldsymbol{a}}_{k-N-1}^{\mathrm{f,k-Q}}, \boldsymbol{a}_{k-Q-1}^{k-1})$ where $\tilde{\boldsymbol{a}}_{k-N-1}^{\mathrm{f,k-Q-2}}$

[26] In Section 4.8.1 we considered the generic notation $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ to indicate the forward decision feedback. In the current section we use an explicit notation to indicate a segment of an FS. We also suppress the explicit dependence of the recovered information symbols on the transition ϵ_{k-1} to simplify the notation. Any ambiguity should be clear from the context.

represent the information symbols in the path history of transition ϵ_{k-1}. With these definitions, it is easy to extend the survivors by a step, i.e., to epoch k. In the reduced-state trellis, taking into account the survivor path $\mathrm{FS}_{k-1}^{(i)}(\epsilon_{k-1})$ associated with transition ϵ_{k-1}, equation (4.161) reduces to

$$\ln \mathring{\alpha}_k(\epsilon_k) \simeq \max_{\epsilon_{k-1}:s^+(\epsilon_{k-1})=s^-(\epsilon_k)} \left\{ \ln \psi_k(\mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k) + \ln \mathring{\alpha}_{k-1}(\epsilon_{k-1}) \right\} . \quad (4.162)$$

For each transition ϵ_k, the branch $\epsilon_{k-1}^{\max}$ that maximizes (4.162) has to be stored.

Given the above definitions, we can reformulate equations (4.159)-(4.162), obtaining:

$$\mathrm{S}[a_k] = \sum_{\epsilon_k:a_k} \mathring{\gamma}_k(\mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k) \mathring{\alpha}_k(\epsilon_k) \mathring{\beta}_k(\epsilon_k) P\{s_k\} \quad (4.163)$$

$$\mathring{\alpha}_k(\epsilon_k) = \sum_{\epsilon_{k-1}:s^+(\epsilon_{k-1})=s^-(\epsilon_k)} \psi_k(\mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k) \mathring{\alpha}_{k-1}(\epsilon_{k-1}) \quad (4.164)$$

$$\mathring{\beta}_k(\epsilon_k) = \sum_{\epsilon_{k+1}:s^+(\epsilon_k)=s^-(\epsilon_{k+1})} \phi_{k+1}(\mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1}) \mathring{\beta}_{k+1}(\epsilon_{k+1}). \quad (4.165)$$

When a recursive state μ_k is embedded in s_k, one can use the following approximation for the *a priori* probability of state s_k: $P\{s_k\} \simeq \prod_{i=1}^{Q-1} P\{a_{k-i}\}$ [162]—this implicitly assumes that all recursive states are considered equiprobable. In Figure 4.23, it is shown how the forward recursion proceeds in the reduced-state trellis according to equation (4.164). In order to compute $\alpha_k(\epsilon_k)$, one should consider the M quantities $\{\alpha_{k-1}(\epsilon_{k-1})\}$ such that $s^+(\epsilon_{k-1}) = s^-(\epsilon_k)$ and, for each of them, compute the metric ψ_k by considering the symbols associated with the survivor of transition ϵ_{k-1}. The backward recursion proceeds similarly using the survivor map generated during the forward recursion, as shown in Figure 4.24.

4.8.4 Examples of Application of Fwd-Only RS FB-type Algorithms

Noncoherent Channel

We consider the so-called NCSOb algorithm proposed in Section 4.7. We remark that in this case, the application of the proposed state reduction technique is "tricky." We now show a possible derivation, successfully employed in [163].

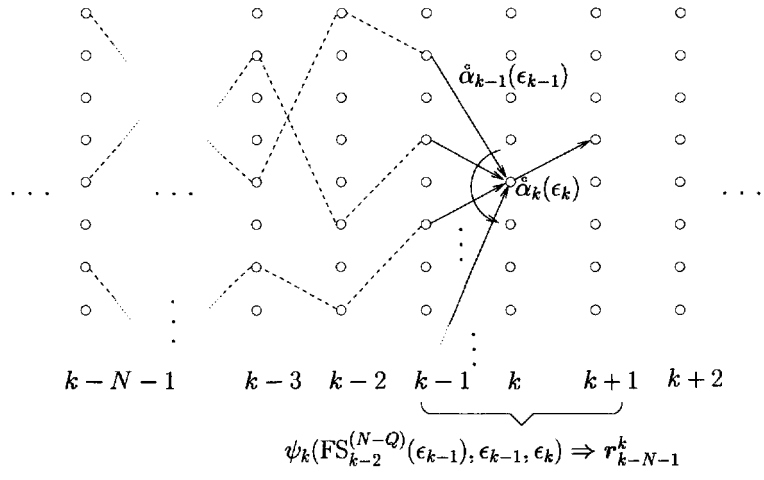


Figure 4.23: Forward recursion of the pdf $\mathring{\alpha}_k(\epsilon_k)$ for a general Fwd-only RS FB-type algorithm. Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

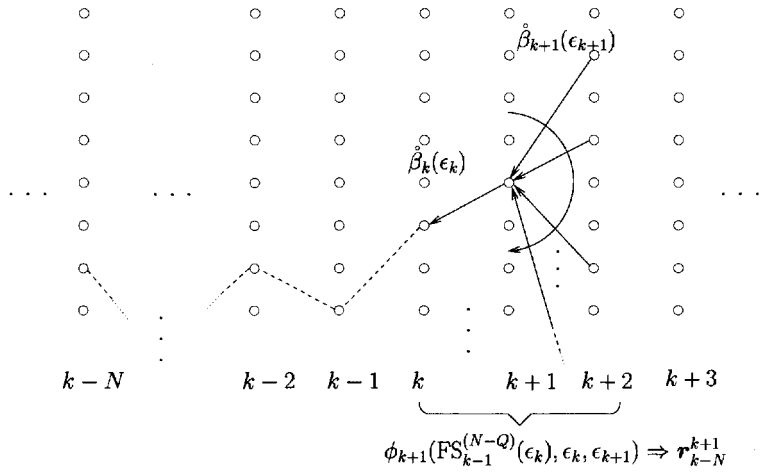


Figure 4.24: Backward recursion of the pdf $\mathring{\beta}_k(\epsilon_k)$ for a general Fwd-only RS FB-type algorithm. The metric ϕ_k is calculated using the survivor map previously constructed in the forward recursion. Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

Assuming a forward survivor map has been constructed, the pdf $\mathring{\gamma}_k$ can be com-

puted as follows:

$$\begin{aligned}\mathring{\gamma}_k(\mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k),\epsilon_k) &= p\left(\boldsymbol{r}_{k-N}^{k}\left|\mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k),\epsilon_k\right.\right)P\{a_k\}\\ &\sim \exp\left\{-\frac{\sum_{i=0}^{Q}|c_{k-i}|^2+\sum_{j=Q-1}^{N}|\tilde{c}_{k-j}^{\mathrm{f}}|^2}{2\sigma^2}\right\}\\ &\quad\cdot \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{Q}r_{k-i}c_{k-i}^{*}+\sum_{j=Q-1}^{N}r_{k-j}\left(\tilde{c}_{k-j}^{\mathrm{f}}\right)^{*}\right|\right)P\{a_k\}\,.\end{aligned} \tag{4.166}$$

In the reduced-state case, we recall that the two quantities $\mathring{\alpha}_k$ and $\mathring{\beta}_k$ can be written, directly extending (4.156) and (4.158), as

$$\mathring{\alpha}_k(\epsilon_k) = p\left(\boldsymbol{r}_1^{k-N-1}\left|\boldsymbol{r}_{k-N}^{k},\epsilon_k\right.\right) \tag{4.167}$$

$$\mathring{\beta}_k(\epsilon_k) = p\left(\boldsymbol{r}_{k+1}^{K}\left|\boldsymbol{r}_{k-N}^{k},\epsilon_k\right.\right). \tag{4.168}$$

If $Q < N$, then $\mathring{\alpha}_k(\epsilon_k)$, as defined in (4.167) for the reduced-state case, is different from $\mathring{\alpha}_k(T_k)$ in (4.156) for the full-state case. Similarly, $\mathring{\beta}_k(\epsilon_k) \neq \mathring{\beta}_k(T_k)$.

We now show the mathematical derivation which leads to the forward recursion in the reduced-state trellis. More precisely, assuming the survivor map is known up to epoch $(k-1)$, we show how to extend it to epoch k. The detailed mathematical derivation of the forward recursion proposed in Section 4.7 for the full-state case cannot be applied. For ease of derivation, we consider the case of an encoder/modulator FSM such that the next-state function is invertible. In other words, we assume that (a_{k-Q-1},ϵ_k) uniquely identifies ϵ_{k-1}. A correct formulation of a step in the reduced-state forward recursion is the following:

$$\begin{aligned}\mathring{\alpha}_k(\epsilon_k) &= p\left(\boldsymbol{r}_1^{k-N-1}\left|\boldsymbol{r}_{k-N}^{k},\epsilon_k\right.\right)\\ &= \sum_{a_{k-Q-1}} p\left(\boldsymbol{r}_1^{k-N-2}\left|\boldsymbol{r}_{k-N-1}^{k},a_{k-Q-1},\epsilon_k\right.\right)p\left(r_{k-N-1}\left|\boldsymbol{r}_{k-N}^{k},a_{k-Q-1},\epsilon_k\right.\right)\\ &\qquad\cdot P\left\{a_{k-Q-1}\left|\epsilon_k,\boldsymbol{r}_{k-N}^{k}\right.\right\}\,.\end{aligned} \tag{4.169}$$

Assuming $Q < N$ (state reduction), a_{k-Q-1} depends on $\boldsymbol{r}_{k-N}^{k}$. Hence $P\left\{a_{k-Q-1}\left|\epsilon_k,\boldsymbol{r}_{k-N}^{k}\right.\right\} \neq P\left\{a_{k-Q-1}\right\}$, making it impossible to evaluate this probability. As a consequence, another approach has to be considered. More precisely, we

may express $\mathring{\alpha}_k$ as follows:

$$\begin{aligned}\mathring{\alpha}_k(\epsilon_k) &= p\left(\boldsymbol{r}_1^{k-N-1}\left|\boldsymbol{r}_{k-N}^{k},\epsilon_k\right.\right)=\frac{p\left(\boldsymbol{r}_1^{k}\left|\epsilon_k\right.\right)}{p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right)}\\ &= \frac{\sum\limits_{a_{k-Q-1}} p\left(\boldsymbol{r}_1^{k}\left|a_{k-Q-1},\epsilon_k\right.\right)P\{a_{k-Q-1}|\epsilon_k\}}{p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right)}\\ &= \left[\sum_{a_{k-Q-1}} p\left(\boldsymbol{r}_1^{k-N-2}\left|\boldsymbol{r}_{k-N-1}^{k-1},r_k,a_{k-Q-1},\epsilon_k\right.\right)p\left(\boldsymbol{r}_{k-N-1}^{k}\left|a_{k-Q-1},\epsilon_k\right.\right)\right.\\ &\qquad \left.\cdot P\{a_{k-Q-1}|\epsilon_k\}\right]/p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right). \end{aligned} \tag{4.170}$$

Since $P\{a_{k-Q-1}|\epsilon_k\}=P\{a_{k-Q-1}\}$, observing that ϵ_{k-1} is uniquely determined by (a_{k-Q-1},ϵ_k) and applying the CMP, i.e.:

$$p\left(\boldsymbol{r}_1^{k-N}\left|\boldsymbol{r}_{k-N-1}^{k-1},r_k,a_{k-Q-1},\epsilon_k\right.\right)=p\left(\boldsymbol{r}_1^{k-N}\left|\boldsymbol{r}_{k-N-1}^{k-1},\epsilon_{k-1}\right.\right) \tag{4.171}$$

the following forward recursion in the reduced-state trellis is obtained:

$$\begin{aligned}\mathring{\alpha}_k(\epsilon_k) &= \frac{\sum\limits_{a_{k-Q-1}} p\left(\boldsymbol{r}_1^{k-N}\left|\boldsymbol{r}_{k-N-1}^{k-1},\epsilon_{k-1}\right.\right)p\left(\boldsymbol{r}_{k-N-1}^{k}\left|a_{k-Q-1},\epsilon_k\right.\right)P\{a_{k-Q-1}|\epsilon_k\}}{p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right)}\\ &= \frac{\sum\limits_{a_{k-Q-1}} \alpha_{k-1}(\epsilon_{k-1})p\left(\boldsymbol{r}_{k-N-1}^{k}\left|a_{k-Q-1},\epsilon_k\right.\right)P\{a_{k-Q-1}\}}{p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right)} \end{aligned} \tag{4.172}$$

where

$$\mathring{\alpha}_{k-1}(\epsilon_{k-1})=p\left(\boldsymbol{r}_1^{k-N}|\boldsymbol{r}_{k-N-1}^{k-1},\epsilon_{k-1}\right) \tag{4.173}$$

in agreement with (4.167).

The problem in the computation of (4.172) is the evaluation of the two pdfs $p\left(\boldsymbol{r}_{k-N-1}^{k}\left|a_{k-Q-1},\epsilon_k\right.\right)$ and $p\left(\boldsymbol{r}_{k-N}^{k}\left|\epsilon_k\right.\right)$. In fact, since $Q<N$, each of the two pdfs should be correctly computed by averaging over previous information symbols. Since at epoch k the survivor of each transition ϵ_{k-1} is known and since $(a_{k-Q-1},\epsilon_k)\equiv(\epsilon_{k-1},\epsilon_k)$, we replace $p(\boldsymbol{r}_{k-N-1}^{k}\left|a_{k-Q-1},\epsilon_k\right.)=p(\boldsymbol{r}_{k-N-1}^{k}\left|\epsilon_{k-1},\epsilon_k\right.)$ with the pdf $p(\boldsymbol{r}_{k-N-1}^{k}\left|\mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}),\epsilon_{k-1},\epsilon_k\right.)$, obtaining the following modified

recursion:

$$\mathring{\alpha}_k(\epsilon_k) = \frac{\sum_{a_{k-Q-1}} \mathring{\alpha}_{k-1}(\epsilon_{k-1}) p\left(\boldsymbol{r}_{k-N-1}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k \right.\right) P\{a_{k-Q-1}\}}{p\left(\boldsymbol{r}_{k-N}^{k} | \epsilon_k\right)}. \tag{4.174}$$

We now express the forward recursion (4.174) in the logarithmic domain as follows:

$$\begin{aligned} \ln \mathring{\alpha}_k(\epsilon_k) \;=\; & \ln \Big\{ \sum_{a_{k-Q-1}} \exp \Big[\ln \mathring{\alpha}_{k-1}(\epsilon_{k-1}) \\ & + \ln p\left(\boldsymbol{r}_{k-N-1}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k \right.\right) \\ & + \ln P\{a_{k-Q-1}\} \Big] \Big\} - \ln p(\boldsymbol{r}_{k-N}^{k} | \epsilon_k) \end{aligned} \tag{4.175}$$

and using the max-log approximation one obtains:

$$\begin{aligned} \ln \mathring{\alpha}_k(\epsilon_k) \simeq \max_{a_{k-Q-1}} \Big\{ & \ln \mathring{\alpha}_{k-1}(\epsilon_{k-1}) + \ln p(\boldsymbol{r}_{k-N-1}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k \right.) \\ & + \ln P\{a_{k-Q-1}\} \Big\} - \ln p(\boldsymbol{r}_{k-N}^{k} | \epsilon_k). \end{aligned} \tag{4.176}$$

The choice of the survivor associated with ϵ_k may be based on the maximization operation in (4.176), which can be correctly carried out since the quantities $\ln \mathring{\alpha}_{k-1}(\epsilon_{k-1})$ and $\ln P\{a_{k-Q-1}\}$ are known and $\ln p(\boldsymbol{r}_{k-N-1}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}), \epsilon_{k-1}, \epsilon_k \right.)$ can be computed. The term $\ln p(\boldsymbol{r}_{k-N}^{k} | \epsilon_k)$ does not affect the maximization (and, consequently, the survivor selection), but affects the numerical value of $\ln \mathring{\alpha}_k(\epsilon_k)$.

We denote by $\epsilon_{k-1}^{\max}$ the transition selected by maximization in (4.176). Equivalently, the symbol $a_{k-Q}^{\max}$ may be considered. Once the transition $\epsilon_{k-1}^{\max}$ has been associated with ϵ_k, we replace $\ln p(\boldsymbol{r}_{k-N}^{k} | \epsilon_k)$ with the following pdf in the logarithmic domain:

$$\begin{aligned} \ln p(\boldsymbol{r}_{k-N}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{\max}), \epsilon_{k-1}^{\max}, \epsilon_k \right.) \;\sim\; & -\frac{\sum_{i=0}^{Q-1} |c_{k-i}|^2 + \sum_{j=Q}^{N} |\tilde{c}_{k-j}^{\mathrm{f}}|^2}{2\sigma^2} \\ & + \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} r_{k-i} c_{k-i}^{*} + \sum_{j=Q}^{N} r_{k-j} \left(\tilde{c}_{k-j}^{\mathrm{f}}\right)^{*} \right| \right). \end{aligned} \tag{4.177}$$

The resulting forward recursion becomes:

$$\begin{aligned}
\ln \mathring{\alpha}_k(\epsilon_k) &= \ln \mathring{\alpha}_{k-1}(\epsilon_{k-1}^{\max}) + \ln p(\boldsymbol{r}_{k-N}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q)}(\epsilon_{k-1}^{\max}), \epsilon_{k-1}^{\max}, \epsilon_k \right.) \\
&\quad - \ln p(\boldsymbol{r}_{k-N}^{k} \left| \mathrm{FS}_{k-2}^{(N-Q-1)}(\epsilon_{k-1}^{\max}), \epsilon_{k-1}^{\max}, \epsilon_k \right.) + \ln P\{a_{k-Q-1}^{\max}\} \\
&\sim \ln \mathring{\alpha}_{k-1}(\epsilon_{k-1}^{\max}) + \frac{|\tilde{c}_{k-N-1}^{\mathrm{f}}|^2}{2\sigma^2} + \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} r_{k-i} c_{k-i}^{*} + \sum_{j=Q}^{N+1} r_{k-j} \left(\tilde{c}_{k-j}^{\mathrm{f}}\right)^{*} \right| \right) \\
&\quad - \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} r_{k-i} c_{k-i}^{*} + \sum_{j=Q}^{N} r_{k-j} \left(\tilde{c}_{k-j}^{\mathrm{f}}\right)^{*} \right| \right) + \ln P\{a_{k-Q-1}^{\max}\}.
\end{aligned} \tag{4.178}$$

The obtained forward recursion in the reduced-state trellis exhibits some analogies with the corresponding forward recursion in the full-state trellis. This indirectly confirms the validity of the proposed intuitive approximations.

The backward recursion can be similarly obtained with the further simplification that the survivor map is now already available because it was previously determined during the forward recursion. More precisely, observing that (ϵ_k, a_k) uniquely identifies ϵ_{k+1}, the backward recursion may be written as follows:

$$\begin{aligned}
\ln \mathring{\beta}_k(\epsilon_k) &= \max_{a_{k+1}} \left\{ \mathring{\beta}_{k+1}(\epsilon_{k+1}) + \ln p \left(\boldsymbol{r}_{k-N}^{k+1} \left| \mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1} \right. \right) \right. \\
&\qquad \left. + \ln P\{a_{k+1}\} \right\} - \ln p \left(\boldsymbol{r}_{k-N}^{k} \left| \mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k \right. \right) \\
&= \ln \mathring{\beta}_{k+1}(\epsilon_{k+1}^{\max}) + \ln p \left(\boldsymbol{r}_{k-N}^{k+1} \left| \mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k, \epsilon_{k+1}^{\max} \right. \right) \\
&\qquad + \ln P\{a_{k+1}^{\max}\} - \ln p(\boldsymbol{r}_{k-N}^{k} \left| \mathrm{FS}_{k-1}^{(N-Q)}(\epsilon_k), \epsilon_k \right.) \\
&\simeq \ln \mathring{\beta}_{k+1}(\epsilon_{k+1}^{\max}) + \frac{|c_{k+1}|^2}{2\sigma^2} \\
&\quad + \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=0}^{Q-1} r_{k+1-i} c_{k+1-i}^{*} + \sum_{j=Q}^{N+1} r_{k+1-j} \left(\tilde{c}_{k+1-j}^{\mathrm{f}}\right)^{*} \right| \right) \\
&\quad - \ln \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_{i=1}^{Q-1} r_{k+1-i} c_{k+1-i}^{*} + \sum_{j=Q}^{N} r_{k+1-j} \left(\tilde{c}_{k+1-j}^{\mathrm{f}}\right)^{*} \right| \right) \\
&\quad + \ln P\{a_{k+1}^{\max}\}.
\end{aligned} \tag{4.179}$$

As an example of application, we consider a single 16-state RSC binary code with generators $G_1 = (37)_8$, $G_2 = (21)_8$ and rate 2/3, obtained by means of puncturing,

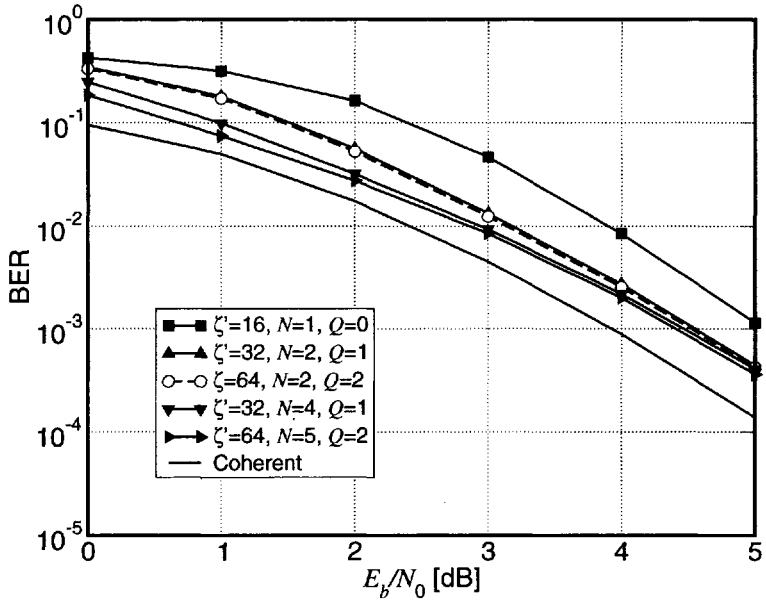


Figure 4.25: Application of the RS-NCSOb algorithm to noncoherent decoding of an RSC code. Receivers with various levels of complexity are considered and compared with a full-state receiver ($N = 2$) and the coherent receiver. Reproduced from [156], ©2001 IEEE, by permission of the IEEE.

used as a component of the turbo code presented in [33]. The modulation format is BPSK. In Figure 4.25, the performance of the noncoherent decoder using the RS-NCSOb algorithm is assessed for various levels of state reduction and compared with that of a coherent receiver. With respect to a decoder with $N = 2$ and $\zeta = 64$ states, for $N = 5$ and $\zeta' = 64$ states the performance is appreciably improved at low SNRs. For $N = 1$ and $\zeta' = 16$, the performance loss with respect to the full-state receiver with $N = 2$ is less than 1 dB for every SNR, and reduces to only 0.5 dB for an SNR value larger than 5 dB. As mentioned previously, we do not consider any example of noncoherent iterative detection, since Section 4.9.1 will be entirely dedicated to this topic.

4.8.5 Generalized RS-FB Algorithms

In this subsection, we present a generalized class of RS-FB algorithms. We first consider a probabilistic derivation, and we then introduce a generalized structure. Various examples of applications of these generalized state reduction techniques for

FB algorithms are proposed. More details about generalized RS-FB algorithms can be found in [164].

Probabilistic Derivation

In this section, a probabilistic approach to deriving generalized RS-FB algorithms is proposed [165]. Let a_k, S_k, and x_k be the input, state, and output of an FSM (it could be the encoder/modulator FSM or the encoder/modulator/channel FSM), respectively, at epoch $k \in \{0, 1, \ldots, K-1\}$, where K represents the length of the information sequence at the input of the FSM. The input symbols $\{a_k\}_{k=0}^{K-1}$ are assumed to be iid. A *valid* transition T_k is defined by (S_k, a_k, S_{k+1}), where $S_{k+1} = \mathrm{ns}_k(S_k, a_k)$. The generated output symbol is denoted by $x_k = \mathrm{o}_k(S_k, a_k)$. The next-state $(\mathrm{ns}_k(\cdot,\cdot))$ and output $(\mathrm{o}_k(\cdot,\cdot))$ functions are determined by the encoder/modulator structure. In order to simplify the notation, we assume, in the remainder of this subsection, that a single output symbol x_k is generated in correspondence to a single input symbol a_k. The extension to a more general case is straightforward by considering a vector notation.

To simplify the derivation, a state of an FSM is assumed to be defined by a sequence of consecutive input symbols $S_k \triangleq \boldsymbol{a}_{k-L}^{k-1}$, where the parameter L is related to the memory of the FSM. We refer to this kind of FSM as *simple* [98]. This is the case, for example, for the FSM corresponding to an ISI channel with $L+1$ taps—as considered, for instance, in Section 4.8.2. Therefore, in the full-state case, a transition T_k is associated with $L+1$ information symbols, i.e., $T_k = (S_k, a_k, S_{k+1}) = \boldsymbol{a}_{k-L}^{k}$. For the reduced-state case, a state is simply defined as $s_k \triangleq \boldsymbol{a}_{k-L_1}^{k-1}$, with $L_1 \leq L$. Similarly to the full-state trellis case, a reduced-state transition t_k is also defined by $(s_k, a_k, s_{k+1}) = \boldsymbol{a}_{k-L_1}^{k}$. Thus, there are $L - L_1 + 1$ possible reduced-state transitions $\{t_{k-(L-L_1)}, \ldots, t_k\}$ embedded in (i.e., uniquely determined by) a full-state transition T_k. In both reduced-state recursions we want to associate with each trellis branch, at any epoch, a metric corresponding to one in the full-state case. Hence, a reduced-state transition has to be associated in a unique way with a particular full-state transition, and this will be done on the basis of decision feedback. In general, a full-state transition T_k can be truncated differently in the two recursions. Specifically, T_k can be reduced to t_{k-n_f} in the forward recursion, and to t_{k-n_b} in the backward recursion, where n_f and n_b are suitable integer parameters such that $0 \leq n_\mathrm{f} \leq n_\mathrm{b} \leq L - L_1$.

Let us suppose that the forward recursion is run first. An intuitive approach to performing reduced-state detection consists of reducing T_k to t_k in the forward recursion (i.e., $n_\mathrm{f} = 0$) and to t_{k-L_2} in the backward recursion (i.e., $n_\mathrm{b} = L_2$), where $L_2 \in \{0, \ldots, L - L_1\}$. In the following, we derive an RS-FB algorithm for this specific case, and generalizations to other cases can be dealt with in the same way. For

the sake of simplicity, we refer to a discrete-time communication system, where the output sequence $\boldsymbol{x}_0^{K-1}$ of a simple FSM is transmitted over an AWGN channel. The observation at the output of the channel at epoch k, indicated as r_k, can be written as

$$r_k = x_k + n_k \tag{4.180}$$

where $\boldsymbol{n}_0^{K-1}$ is a sequence of zero-mean uncorrelated Gaussian random variables. The goal of this derivation is to compute the *a posteriori* likelihood, i.e., the extrinsic soft output, of a_k based on the observations $\boldsymbol{r}_0^{N-1}$. The extrinsic soft output, indicated as SO$[a_k]$, can be written as

$$\mathrm{SO}[a_k] = \frac{\mathrm{S}[a_k]}{\mathrm{SI}[a_k]} = \frac{P\{a_k|\boldsymbol{r}_0^{K-1}\}}{P\{a_k\}} \sim p(\boldsymbol{r}_0^{K-1}|a_k) = \frac{P(\boldsymbol{r}_0^{K-1}, a_k)}{P\{a_k\}} \tag{4.181}$$

where $P(\boldsymbol{r}_0^{K-1}, a_k)$ is the joint probability distribution function[27] of the observation sequence $\boldsymbol{r}_0^{K-1}$ and information symbol a_k, S$[a_k]$ is the "complete" soft output (corresponding, in principle, to the APP) and SI$[a_k]$ is the *a priori* input soft information on a_k (corresponding, in principle, to the *a priori* probability on the symbol). In the following, we will focus on the computation of the joint probability distribution function $P(\boldsymbol{r}_0^{K-1}, a_k)$ over a reduced-state trellis.

Let us assume that forward and backward survivors have been somehow already associated with each reduced state at each epoch (at this point of the derivation we do not know how, yet). The ensembles of forward and backward survivors are indicated with the general notation $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ and $\tilde{\boldsymbol{a}}^{\mathrm{b}}$, respectively. A reasonable approximation, useful for computing the desired soft output (4.181), is the following:

$$P(\boldsymbol{r}_0^{K-1}, a_k) \sim P(\boldsymbol{r}_0^{K-1}, a_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}}). \tag{4.182}$$

[27] We use the term probability distribution function to denote a continuous pdf with some discrete probability masses. For a probability distribution function we will use the symbol $P(.)$.

Using elementary conditional probability relations, one obtains:

$$
\begin{aligned}
P(\boldsymbol{r}_0^{K-1}, a_k) \;\sim\; & P(\boldsymbol{r}_0^{K-1}, a_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}}) = \sum_{t_k:a_k} P(\boldsymbol{r}_0^{K-1}, t_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\
= & \sum_{t_k:a_k} \underbrace{p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\boldsymbol{r}_0^{k-1}, \mathbf{r}_k^{k+L_2}, \tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p(\mathbf{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})} \underbrace{p(\boldsymbol{r}_k^{k+L_2}|\mathbf{r}_0^{k-1}, \tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p(\mathbf{r}_k^{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})} \\
& \cdot \underbrace{P(\mathbf{r}_0^{k-1}, t_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{P\{a_k|\mathbf{r}_0^{k-1}, \tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}\} p(\mathbf{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}})} \\
= & \sum_{t_k:a_k} \underbrace{p(\mathbf{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p_1} \underbrace{p(\mathbf{r}_k^{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p_2} \\
& \underbrace{p(\mathbf{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p_3} P\{a_k\}
\end{aligned}
\tag{4.183}
$$

where the pdfs in (4.183), denoted by p_1, p_2, and p_3, respectively, are further analyzed below.

p_1. Based on the causality of the system (i.e., $t_k = \boldsymbol{a}_{k-L_1}^k$), p_1 can be written as

$$
p_1 = P(\boldsymbol{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, \tilde{\boldsymbol{a}}^{\mathrm{b}}) = P(\boldsymbol{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}) \tag{4.184}
$$

which we define as follows:[28]

$$
\alpha_k(s_k) \triangleq P(\boldsymbol{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}). \tag{4.185}
$$

p_2. As the channel is memoryless and assuming that a forward survivor $\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k)$ and a backward survivor $\tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})$ are associated with states s_k and s_{k+1}, respectively, the *extended metric* p_2 can be computed as

$$
\begin{aligned}
p_2 &= p(\boldsymbol{r}_k^{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\
&\simeq p(\boldsymbol{r}_k^{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})) \\
&= p(\boldsymbol{r}_k^{k+L_2}|\tilde{\boldsymbol{a}}_{k-L}^{\mathrm{f,k-L_1-1}}(s_k) = \boldsymbol{a}_{k-L}^{k-L_1-1}, t_k = \boldsymbol{a}_{k-L_1}^k, \tilde{\boldsymbol{a}}_{k+1}^{\mathrm{b,k+L_2}}(s_{k+1}) = \boldsymbol{a}_{k+1}^{k+L_2}) \\
&= p(\boldsymbol{r}_k^{k+L_2}|\boldsymbol{a}_{k-L}^{k+L_2}) = \prod_{i=0}^{L_2} p(r_{k+i}|x_{k+i}(\boldsymbol{a}_{k-L+i}^{k+i}))
\end{aligned}
\tag{4.186}
$$

[28] In fact, since $P(\boldsymbol{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}) = p(\boldsymbol{r}_0^{k-1}|s_k, \tilde{\boldsymbol{a}}^{\mathrm{f}})P\{s_k\}$, the quantity defined in (4.185) is exactly the quantity $\alpha_k(s_k)$.

and we correspondingly define:

$$\begin{aligned} \mathrm{G}_k^{k+L_2}(t_k) &\triangleq \prod_{i=0}^{L_2} p(r_{k+i}|x_{k+i}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1}))) \\ &= \prod_{i=0}^{L_2} p(r_{k+i}|x_{k+i}(\boldsymbol{a}_{k-L+i}^{k+i})). \end{aligned} \tag{4.187}$$

p_3. The quantity p_3 can be expressed in different ways, depending on L_2. If $L_2 = L - L_1$ (e.g., [151]), we can write:

$$\begin{aligned} p_3 &= p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\ &= p(\boldsymbol{r}_{k+L-L_1+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\ &= p(\boldsymbol{r}_{k+L-L_1+1}^{K-1}|s_{k+1}, \tilde{\boldsymbol{a}}^{\mathrm{b}}). \end{aligned} \tag{4.188}$$

However, if $L_2 < L - L_1$ (e.g., $L_2 = 0$ in [156]), then r_{k+L_2+1} depends on all the information symbols embedded in a reduced transition t_k and, possibly, on a portion of $\tilde{\boldsymbol{a}}^{\mathrm{f}}$. Thus, it is not true that $\boldsymbol{r}_{k+L_2+1}^{K-1}$ depends only on s_{k+1} and $\tilde{\boldsymbol{a}}^{\mathrm{b}}$. However, by assuming that $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ contains the information, relative to t_k, not included in s_{k+1}, one can write:

$$p_3 = p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_{k+1}, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \tag{4.189}$$

and one can correspondingly define:

$$\beta_{k+1}(s_{k+1}) \triangleq p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_{k+1}, \tilde{\boldsymbol{a}}^{\mathrm{b}}). \tag{4.190}$$

The likelihood of (i.e., the extrinsic soft-output value associated with) a_k can be finally obtained through the following completion operation:

$$\begin{aligned} \mathrm{SO}[a_k] &= p(\boldsymbol{r}_0^{K-1}|a_k) = \frac{P(\boldsymbol{r}_0^{K-1}, a_k)}{P\{a_k\}} \\ &= \sum_{t_k:a_k} \alpha_k(s_k)\mathrm{G}_k^{k+L_2}(t_k)\beta_{k+1}(s_{k+1}). \end{aligned} \tag{4.191}$$

As one can see, we have dropped from the definitions of the introduced quantities ($\mathrm{SO}[a_k]$, $\alpha_k(s_k)$, $\mathrm{G}_k^{k+L_2}(t_k)$ and $\beta_{k+1}(s_{k+1})$) the dependence on the survivors. They have been used, however, to justify and motivate the above derivation. The notation $\mathrm{G}_k^{k+L_2}$ is used to indicate the time interval relative to the considered observation

sequence, i.e., from k to $k+L_2$. This represents a difference with respect to a standard FB algorithm. In fact, instead of considering just the observation at epoch k, we consider a *window of observations*. The quantities $\alpha_k(s_k)$ and $\beta_{k+1}(s_{k+1})$ can be computed by means of forward and backward recursions, respectively. During these recursions, the survivors $\tilde{\boldsymbol{a}}^{\mathrm{f}}$ and $\tilde{\boldsymbol{a}}^{\mathrm{b}}$ are recursively constructed. Soft-output values relative to output symbols $\{x_k\}$ can be generated in a similar way.

Let us consider the computation associated with the trellis transition at epoch $k-1$. Assuming that a forward survivor $\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1})$ is associated with each state s_{k-1}, the corresponding step in the forward recursion allows one to recursively compute $\alpha_k(s_k)$ on the reduced-state trellis and, simultaneously, to extend the existing survivors to the states at epoch k. In fact, $\alpha_k(s_k)$ can be expressed as follows:

$$\begin{aligned}
\alpha_k(s_k) &= P(\boldsymbol{r}_0^{k-1}, s_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}) \\
&= \sum_{t_{k-1}:s_k} P(\boldsymbol{r}_0^{k-1}, t_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}) \\
&= \sum_{t_{k-1}:s_k} \underbrace{p(r_{k-1}|\boldsymbol{r}_0^{k-2}, \tilde{\boldsymbol{a}}^{\mathrm{f}}, t_{k-1})}_{p(r_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), t_{k-1})} \cdot \underbrace{P(\boldsymbol{r}_0^{k-2}, t_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}})}_{P\{a_{k-1}|\boldsymbol{r}_0^{k-2}, s_{k-2}, \tilde{\boldsymbol{a}}^{\mathrm{f}}\} p(\boldsymbol{r}_0^{k-2}, s_{k-2}|\tilde{\boldsymbol{a}}^{\mathrm{f}})} \\
&= \sum_{t_{k-1}:s_k} \underbrace{p(r_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), t_{k-1})}_{p(r_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f},k-L_1-2}_{k-L-1}(s_{k-1})=\boldsymbol{a}^{k-L_1-2}_{k-L-1}, t_{k-1}=\boldsymbol{a}^{k-1}_{k-L_1-1})} P(\boldsymbol{r}_0^{k-2}, s_{k-2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}) P\{a_{k-1}\} \\
&= \sum_{t_{k-1}:s_k} P(\boldsymbol{r}_0^{k-2}, s_{k-2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}) p(r_{k-1}|\boldsymbol{a}_{k-L-1}^{k-1}) P\{a_{k-1}\}. \qquad (4.192)
\end{aligned}$$

By defining

$$\begin{aligned}
\mathrm{G}_{k-1}(t_{k-1}) &\triangleq p(r_{k-1}|x_{k-1}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), t_{k-1})) P\{a_{k-1}\} \\
&= p(r_{k-1}|x_{k-1}(\boldsymbol{a}_{k-L-1}^{k-1})) P\{a_{k-1}\} \qquad (4.193)
\end{aligned}$$

where $\boldsymbol{a}_{k-L-1}^{k-L_1-2}$ depend on the forward survivor, the $(k-1)$-th step in the forward recursion becomes:

$$\alpha_k(s_k) = \sum_{t_{k-1}:s_k} \alpha_{k-1}(s_{k-1}) \mathrm{G}_{k-1}(t_{k-1}). \qquad (4.194)$$

A reasonable way to associate a survivor with s_k consists of associating with it the survivor relative to the state s_{k-1} such that the corresponding term in the summation (4.194) is the largest one [151, 156]. Proper boundary conditions, in terms of $\alpha_0(s_0)$, have to be considered, depending on the initial state of the FSM.

Similarly to the forward recursion, the quantity $\beta_k(s_k)$ can be computed through a backward recursion assuming that backward survivors are known down to states

$\{s_{k+1}\}$. Moreover, the backward survivors are simultaneously extended to states $\{s_k\}$ at the k-th epoch step in the backward recursion. Thus,

$$\begin{aligned}
\beta_k(s_k) &= p(\boldsymbol{r}_{k+L_2}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\
&= \sum_{t_k:s_k} P(\boldsymbol{r}_{k+L_2}^{K-1}, t_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\
&= \sum_{t_k:s_k} P(\boldsymbol{r}_{k+L_2}^{K-1}, a_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}) \\
&= \sum_{t_k:s_k} \underbrace{p(\boldsymbol{r}_{k+L_2+1}^{K-1}|r_{k+L_2}, \tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, a_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_{k+1}, \tilde{\boldsymbol{a}}^{\mathrm{b}})} \\
&\qquad \cdot \underbrace{p(r_{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, a_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})}_{p(r_{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}})} P\{a_k|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}\} \\
&\simeq \sum_{t_k:s_k} p(\boldsymbol{r}_{k+L_2+1}^{K-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}, s_{k+1}, \tilde{\boldsymbol{a}}^{\mathrm{b}}) p(r_{k+L_2}|x_{k+L_2}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})))P\{a_k\}.
\end{aligned} \tag{4.195}$$

Defining

$$\begin{aligned}
\mathrm{G}_{k+L_2}(t_k) &\triangleq p(r_{k+L_2}|x_{k+L_2}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})))P\{a_k\} \\
&= p(r_{k+L_2}|x_{k+L_2}(\tilde{\boldsymbol{a}}_{k+L_2-L}^{\mathrm{f},k-L_1-1}(s_k) = \boldsymbol{a}_{k+L_2-L}^{k-L_1-1}, \\
&\qquad\qquad t_k = \boldsymbol{a}_{k-L_1}^{k}, \tilde{\boldsymbol{a}}_{k+1}^{\mathrm{b},k+L_2}(s_{k+1}) = \boldsymbol{a}_{k+1}^{k+L_2}))P\{a_k\} \\
&= p(r_{k+L_2}|x_{k+L_2}(\boldsymbol{a}_{k+L_2-L}^{k+L_2}))P\{a_k\},
\end{aligned} \tag{4.196}$$

a step in the backward recursion at epoch k reads:

$$\beta_k(s_k) = \sum_{t_k:s_k} \beta(s_{k+1})\mathrm{G}_{k+L_2}(t_k). \tag{4.197}$$

Similarly to the forward recursion, it is reasonable to associate with state s_k the backward survivor relative to state s_{k+1} corresponding to the largest term in the summation in (4.197). Note that, unlike the case where forward-only state reduction techniques are used, in this case the step at epoch k in the backward recursion involves the observation at epoch $k + L_2$, rather than that at epoch $k + 1$. This is due to the extended observation window used in the completion operation.

For the sake of clarity, we consider a few interesting cases for the backward recursion by focusing on the metric $\mathrm{G}_{k+L_2}(t_k)$.

- If $L_2 = L - L_1$, then

$$\begin{aligned} \mathrm{G}_{k+L_2}(t_k) = \mathrm{G}_{k+L-L_1}(t_k) &= p(r_{k+L_2}|x_{k+L_2}(\tilde{\mathbf{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1}))) \\ &= p(r_{k+L-L_1}|x_{k+L-L_1}(t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1}))) \\ &= p(r_{k+L-L_1}|x_{k+L-L_1}(t_k = \boldsymbol{a}_{k-L_1}^{k}, \\ &\qquad \tilde{\boldsymbol{a}}_{k+1}^{\mathrm{b},k+L-L_1}(s_{k+1}) = \boldsymbol{a}_{k+1}^{k+L-L_1})) \\ &= p(r_{k+L-L_1}|x_{k+L-L_1}(\boldsymbol{a}_{k-L_1}^{k+L-L_1})). \end{aligned} \quad (4.198)$$

  In this case, the backward recursion relies only on a backward survivor, which means that backward and forward recursions can be run independently [151].

- If $0 < L_2 < L - L_1$, during the backward recursion backward decision feedback survivors are constructed relying on both forward and backward survivors. This means that the forward recursion has to be run first, and must rely only on forward survivors.

- At another extreme case, i.e., $L_2 = 0$ [156], no survivor is built during the backward recursion. In other words, the decision feedback relies completely on the forward survivors, and one obtains:

$$\begin{aligned} \mathrm{G}_{k+L_2}(t_k) &= \mathrm{G}_k(t_k) = p(r_k|x_k(\tilde{\mathbf{a}}^{\mathrm{f}}(s_k), t_k)) \\ &= p(r_k|x_k(\tilde{\boldsymbol{a}}_{k-L}^{\mathrm{f},k-L_1-1}(s_k) = \boldsymbol{a}_{k-L}^{k-L_1-1}, t_k = \boldsymbol{a}_{k-L_1}^{k})) \\ &= p(r_k|x_k(\boldsymbol{a}_{k-L}^{k})). \end{aligned} \quad (4.199)$$

  This corresponds to the approach considered in Section 4.8.1.

Generalized Structures of RS-FB Algorithms

The probabilistic derivation in Section 4.8.5 provides a unified framework for deriving RS-FB algorithms, generalizing previously published solutions [151, 156]. However, further improvements, based on intuition, are possible (for more details, see [164, 166]). In fact, further generalizations can be obtained by modifying various aspects of the derived RS-FB algorithmic structure, including the forward recursion, the backward recursion, and the soft output generation. As in the previous section, for the sake of simplicity we will consider only the generation of soft-output values relative to the inputs of the FSM, i.e., $\{a_k\}$.

First, let us consider forward and backward recursions. In the previous derivation, the forward recursion is run first, by using only forward decision feedback. The backward recursion, where a backward decision feedback might be constructed, can

utilize both forward and backward decision feedbacks. However, both recursions could be designed to use either forward or backward decision feedback, or both, thus providing different RS-FB structures. Moreover, the soft input relative to a_k does not necessarily need to be included in the metric corresponding to the reduced-state trellis section at epoch k. In fact, it could be included at any epoch l, provided that a_k can be extracted from the reduced transition at epoch l. Indicating by $\boldsymbol{a}^{\mathrm{f}}(s_k)$ the forward decision feedback associated with the reduced state s_k and by $\boldsymbol{a}^{\mathrm{b}}(s_{k+1})$ the backward decision feedback associated with s_{k+1}, the soft-input information on x_{k+i}, relative to the reduced-state transition $t_k = (s_k, a_k, s_{k+1})$, can be generally indicated with the following notation:

$$\begin{aligned} \mathrm{SI}[x_{k+i}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})] &= \mathrm{SI}[x_{k+i} = \mathrm{o}_{k+i}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})], \\ &= \mathrm{SI}[x_{k+i}(\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})], \\ &= p(r_{k+i}|x_{k+i}(\boldsymbol{a}_{k-L+i}^{k+i})) \end{aligned} \tag{4.200}$$

in which the sequence of symbols $\boldsymbol{a}_{k-L+i}^{k+i}$ is extracted from $\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k)$, t_k, and $\tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})$. For example, if $t_k = \boldsymbol{a}_{k-L_1}^{k}$ (simple FSM), the forward and backward recursions in (4.194) and (4.197) can then be equivalently rewritten as follows:

$$\alpha_k(s_k) = \sum_{t_{k-1}:s_k} \alpha_{k-1}(s_{k-1})\mathrm{SI}[x_{k-1}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_{k-1}), t_{k-1}]\mathrm{SI}[a_{k-1+m_{\mathrm{f}}}] \tag{4.201}$$

$$\beta_k(s_k) = \sum_{t_k:s_k} \beta_{k+1}(s_{k+1})\mathrm{SI}[x_{k+L_2}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})]\mathrm{SI}[a_{k+m_{\mathrm{b}}}] \tag{4.202}$$

where m_{f} and m_{b} are suitable integers such that $-L_1 \leq m_{\mathrm{f}} \leq 0$ and $-L_1 \leq m_{\mathrm{b}} \leq 0$. These integer values indicate which soft-input information is embedded in the transition at epoch k. Therefore, the forward and backward recursions corresponding to the case where T_k is reduced to $t_{k-n_{\mathrm{f}}}$ in the forward recursion and to $t_{k-n_{\mathrm{b}}}$ in the backward recursion, with $0 \leq n_{\mathrm{f}} \leq n_{\mathrm{b}} \leq L - L_1$, $-L_1 \leq m_{\mathrm{f}} \leq 0$ and $-L_1 \leq m_{\mathrm{b}} \leq 0$, can be generally written as

$$\alpha_{k+1}(s_{k+1}) = \sum_{t_k:s_{k+1}} \alpha_k(s_k)\mathrm{SI}[x_{k+n_{\mathrm{f}}}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})]\mathrm{SI}[a_{k+m_{\mathrm{f}}}] \tag{4.203}$$

$$\beta_k(s_k) = \sum_{t_k:s_k} \beta_{k+1}(s_{k+1})\mathrm{SI}[x_{k+n_{\mathrm{b}}}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})]\mathrm{SI}[a_{k+m_{\mathrm{b}}}]. \tag{4.204}$$

If the parameters n_{f} and n_{b} in the two recursions are set so that both recursions are based only on the forward (backward) decision feedback, the derived algorithm corresponds to a Fwd-only (Bwd-only) RS-FB algorithm. On the other hand, if one

of these two recursions makes use of both forward and backward decision feedbacks, the corresponding RS-FB algorithm will be referred to as a BiD RS-FB algorithm.

At this point, let us consider the completion operation in (4.191). By using the notation introduced in (4.200), one can rewrite (4.191) as follows:

$$\mathrm{SO}_k[a_k] = \sum_{t_k : a_k} \alpha_k(s_k) \left\{ \prod_{i=0}^{L_2} \mathrm{SI}[x_{k+i} | \tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})] \right\} \beta_{k+1}(s_{k+1}) \quad (4.205)$$

where $\mathrm{SO}_k[a_k]$ represents *a* soft output value on a_k, generated through the completion corresponding to the reduced-state transition t_k (the notation will be clear in the following). In this case, the forward and backward recursions are defined as in (4.201) and (4.202), with $0 \leq L_2 \leq L - L_1$ and $m_{\mathrm{f}} = m_{\mathrm{b}} = 0$ (as used in Section 4.8.5).

- The Fwd-only RS-FB algorithm introduced in Section 4.8.1 is obtained by setting $L_2 = 0$:

$$\mathrm{SO}_k[a_k] = \sum_{t_k : a_k} \alpha_k(s_k) \mathrm{SI}[x_k | \tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k] \beta_{k+1}(s_{k+1}). \quad (4.206)$$

  One can immediately notice that this completion is formally identical to that in the full-state case. This structure will be referred to as the *standard* completion operation.

- At the other extreme, if $L_2 = L - L_1$, only backward decision feedback is used during the backward recursion. Therefore, for this BiD RS-FB algorithm, the backward recursion can be run independently from the forward recursion. The completion operation (4.205) for the generation of a soft output value on a_k at the trellis section corresponding to time epoch k can be written as

$$\mathrm{SO}_k[a_k] = \sum_{t_k : a_k} \alpha_k(s_k) \left\{ \prod_{i=0}^{L-L_1} \mathrm{SI}[x_{k+i} | \tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})] \right\} \beta_{k+1}(s_{k+1}). \quad (4.207)$$

  Comparing this BiD RS-FB algorithm with the BiD RS-FB algorithm proposed in [151], one can immediately recognize that both algorithms utilize the same forward and backward recursions. Nevertheless, there is a difference in the completion operation. In fact, the completion operation proposed in [151] can be written as

$$\mathrm{SO}_k[a_k] = \sum_{t_k : a_k} \alpha_k(s_k) \mathrm{SI}[x_k | \tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k] \beta_{k+1}(s_{k+1}). \quad (4.208)$$

The completion in (4.208) has the same structure as the completion of the (full-state) FB algorithm (i.e., a standard completion). Hence, the completions in (4.207) and (4.208) utilize a different amount of soft-input information in order to generate the desired soft-output information. In fact, it is possible to show that the proposed BiD RS-FB algorithm, with the completion (4.207), uses each of the soft inputs $\mathrm{SI}[x_k]$ and $\mathrm{SI}[a_k]$, for $k = 0, \ldots, K-1$, *exactly once*[29] to generate the quantity $\mathrm{SO}_k[a_k]$. On the other hand, for the BiD RS-FB algorithm in [151] (with completion in (4.208)), the soft output $\mathrm{SO}_k[a_k]$ does not depend on the soft-input values $\{\mathrm{SI}[x_{k+1}], \mathrm{SI}[x_{k+2}], \ldots, \mathrm{SI}[x_{k+(L-L_1)}]\}$. The set of missing soft-input values will be referred to as the *gap*. In the RS-FB algorithm in [151], this gap is due to the time misalignment between the soft input $\mathrm{SI}[x_k]$ used in the metric of the forward recursion step at epoch k and the soft input $\mathrm{SI}[x_{k+L_2}]$ used in the metric of the backward recursion step at epoch k. In this sense, the completion (4.207) of the proposed BiD RS-FB algorithm *fully fills* the gap.

In order to further generalize the completion in (4.205), *weight exponents* for the soft-input values can be introduced as follows:

$$\mathrm{SO}_k[a_k] = \sum_{t_k \,:\, a_k} \alpha_k(s_k) \left\{ \prod_{i=0}^{L_2} \mathrm{SI}[x_{k+i}|\tilde{\boldsymbol{a}}^{\mathrm{f}}(s_k), t_k, \tilde{\boldsymbol{a}}^{\mathrm{b}}(s_{k+1})]^{w_x^{\mathrm{c}}(k,i)} \right\} \beta_{k+1}(s_{k+1}). \tag{4.209}$$

The weight exponents $w_x^{\mathrm{c}}(k,i)$ are used to identify whether the soft inputs $\mathrm{SI}[x_{k+i}]$ should be included in the completion. If a particular soft input is included (excluded) in the completion, its associated weight exponent is set to 1 (0). In general, a weight exponent can be any real number $\theta \in (0, 1]$. The completion (4.209) refers to the sum-product version of an RS-FB algorithm—in the min-sum version, a weight exponent becomes a *scaling factor* that multiplies the soft-input information, i.e., it corresponds to the heuristic parameter introduced in Section 4.5 to scale the extrinsic information for optimized iterative detection. Referring to the generalized forward and backward recursions (4.203) and (4.204), the corresponding generalized comple-

[29] A situation where each soft-input $\mathrm{SI}[x_k]$ and $\mathrm{SI}[a_k]$, $k = 0, \ldots, N-1$, is used more than once will be referred to as *double counting* and should be avoided if possible.

tion operation in the reduced-state trellis at epoch k can be written as

$$\mathrm{SO}_k[a_k] = \sum_{t_k : a_k} \frac{\alpha_k(s_k)\beta_{k+1}(s_{k+1})}{\mathrm{SI}[a_k]} \prod_{m=m_\mathrm{f}}^{m_\mathrm{b}} \mathrm{SI}[a_{k+m}]^{w_a^c(k,m)} \cdot \prod_{n=n_\mathrm{f}}^{n_\mathrm{b}} \mathrm{SI}[x_{k+n}|\tilde{\boldsymbol{a}}^\mathrm{f}(s_k), t_k, \tilde{\boldsymbol{a}}^\mathrm{b}(s_{k+1})]^{w_x^c(k,n)} \tag{4.210}$$

where the weight exponents $\{w_a^c(k,m)\}$ have been introduced for the soft-input values on the information symbols and the parameters $n_\mathrm{f}, n_\mathrm{b}, m_\mathrm{f}$ and m_b previously introduced are such that $n_\mathrm{f} \leq n_\mathrm{b}$ and $m_\mathrm{f} \leq m_\mathrm{b}$. In this case, each soft input is utilized at most once. The weight exponents $w_x^c(k,n)$ and $w_a^c(k,m)$ can be *tuned* to use the desired amount of soft-input information. Specifically, $\{\mathrm{SI}[x_i]\}_{i=0}^{k+n_\mathrm{f}-1} \cup \{\mathrm{SI}[a_i]\}_{i=0}^{k+m_\mathrm{f}-1}$ and $\{\mathrm{SI}[x_i]\}_{i=k+n_\mathrm{b}+1}^{N-1} \cup \{\mathrm{SI}[a_i]\}_{i=k+m_\mathrm{b}+1}^{N-1}$ are the soft-input values used during the forward and backward recursions to generate the state metrics $\alpha_k(s_k)$ and $\beta_{k+1}(s_{k+1})$—note that $\mathrm{SI}[x_{k+n_\mathrm{f}}]\mathrm{SI}[a_{k+m_\mathrm{f}}]$ and $\mathrm{SI}[x_{k+n_\mathrm{b}}]\mathrm{SI}[a_{k+m_\mathrm{b}}]$ are the metrics of the forward and backward recursions in (4.203) and (4.204), respectively.

Standard completion is equivalent to setting $w_a^c(k,m) = \delta_\mathrm{K}(m-m_\mathrm{f})$ and $w_x^c(k,n) = \delta_\mathrm{K}(n-n_\mathrm{f})$, where $\delta_\mathrm{K}(\cdot)$ is the Kronecker delta function defined as

$$\delta_\mathrm{K}(i) \triangleq \begin{cases} 1 & i = 0 \\ 0 & i \neq 0. \end{cases} \tag{4.211}$$

The corresponding RS-FB algorithm is identified by *not filling the gap* (Nfg). The gap is in this case constituted by $\{\mathrm{SI}[x_{k+i}]\}_{i=n_\mathrm{f}+1}^{n_\mathrm{b}} \cup \{\mathrm{SI}[a_{k+i}]\}_{i=m_\mathrm{f}+1}^{m_\mathrm{b}}$. For instance, there is no gap if $n_\mathrm{f} = n_\mathrm{b}$ and $m_\mathrm{f} = m_\mathrm{b}$ (e.g., $n_\mathrm{f} = n_\mathrm{b} = m_\mathrm{f} = m_\mathrm{b} = 0$). On the other hand, the gap can be *fully filled* if $w_a^c(k,m) = w_x^c(k,n) = 1$, $\forall\, m,n$. This option will be referred to as *fully filling the gap* (Ffg). The option where only a portion of the gap is filled, by setting $w_a^c(k,m) = \delta_\mathrm{K}(m-m_\mathrm{f}) + \delta_\mathrm{K}(m-m_\mathrm{b})$ and/or $w_x^c(k,n) = \delta_\mathrm{K}(n-n_\mathrm{f}) + \delta_\mathrm{K}(n-n_\mathrm{b})$, will be referred to as *partially filling the gap* (Pfg).

Another possible improvement can be obtained by considering *multiple completion*, as originally suggested in [98, Problem 3.10]. The idea is simple. Instead of simply generating the soft output on a_k when averaging over t_k, a soft output relative to a_k could be generated by averaging over any transition embedding a_k. To be more specific, we could obtain various reliability values on a_k by averaging over t_k, t_{k+1},..., t_{k+L_1}. Due to the suboptimality of RS-FB algorithms, based on the construction and use of survivors, it is likely that the L_1 generated soft outputs are different. Suitably combining them could refine, owing to a *diversity effect*, the final soft-output information on a_k. In the logarithmic domain (min-sum version), an intuitive way to

combine them is considering the arithmetic mean—in fact, this operation returns the same output as the FB algorithm in the absence of state reduction. In the probability domain (sum-product version), this corresponds to the geometric mean. Indicating by $\mathrm{SO}_{k+i}[a_k]$ the soft output on a_k obtained by averaging over t_{k+i}, the final soft output can be expressed as

$$\mathrm{SO}[a_k] = \left(\prod_{i=0}^{L_1} \mathrm{SO}_{k+i}[a_k] \right)^{\frac{1}{L_1+1}} = \prod_{i=0}^{L_1} (\mathrm{SO}_{k+i}[a_k])^{\frac{1}{L_1+1}} . \tag{4.212}$$

In this case as well, weight exponents can be introduced to generalize the multiple completion in (4.212). More precisely, the soft output on a_k could be written as

$$\mathrm{SO}[a_k] = \prod_{i=0}^{L_1} (\mathrm{SO}_{k+i}[a_k])^{w_a^o(k,i)} \tag{4.213}$$

where the weight exponents $w_a^o(k,i)$ in (4.213) have to be properly set—in (4.212), we are implicitly considering $\theta = 1/(L_1+1)$. Typically, a soft-output value on a_k is generated through a single completion at epoch k. This corresponds to considering $w_a^o(k,i) = \theta\, \delta_{\mathrm{K}}(i)$, and will be referred to as a *single completion* (sc). At the other extreme, the final soft-output value on a_k could be obtained by using all the possible quantities in (4.213). An instance of this option is obtained by setting $w_a^o(k,i) = \theta/(1+L_1)$, $\forall\, i \in \{0,\ldots,L_1\}$ and will be referred to as *full multiple completion* (Fmc). Finally, if only a portion of the soft-input values in (4.213) is considered to generate the final value on a_k, such as $w_a^o(k,i) = \frac{\theta}{2}\delta_{\mathrm{K}}(i) + \frac{\theta}{2}\delta_{\mathrm{K}}(i-L_1)$, this option will be called *partial multiple completion* (Pmc).

The factor θ introduced above is used to control the influence of the soft-input information in the generation of the soft output. This might be very useful when considering exchange of soft information in iterative data detectors, as seen in Section 4.5. In fact, RS-FB algorithms usually lead, because of suboptimality, to overestimation of the soft-output reliability information. The beneficial effect of using weight exponents lower than 1 will be confirmed by the considered numerical results.

A probabilistic derivation of generalized RS-FB algorithms has been proposed. However, further useful and effective modifications (e.g., use of weight exponents and multiple completion) are dictated by intuition and do not have an analytical basis. A *graphical approach* represents a systematic way to design RS-FB algorithms, which can take into account all the considered intuitive modifications. In fact, a suitable graph structure should possess the following characteristics:

- it should allow the application of different state reduction techniques;

- it should account for multiple completion;
- in the absence of state reduction, the probabilistic propagation in the graph should provide the same (or exact) soft output as the FB algorithm.

More details about state reduction techniques based on the use of suitable graphical structures, based on the concept of an *over-structured graph*, can be found in [164, 167]. In the following subsection, we further comment on RS-FB algorithms, in order to provide the reader with more intuition regarding state reduction techniques.

Comments on Generalized RS-FB Algorithms for ISI Channels

For any specific communication system, it would be of interest to find the RS-FB algorithm that guarantees the best possible performance for a *given complexity*—this is a legitimate question, since an RS-FB algorithm is an approximation of the FB algorithm. The case of an ISI channel, whose corresponding FSM output x_k can be written as

$$x_k = \sum_{m=0}^{L} f_m a_{k-m}, \tag{4.214}$$

is important and easy to understand. Hence, we will comment on the proposed RS-FB algorithms referring to an ISI channel, according to the values of the channel coefficients $\{f_0, \ldots, f_L\}$. This analysis can be extended to various types of FSM, especially any LTI system. In the following, the discussion will be based on the three general categories of RS-FB algorithm previously introduced: Fwd-only, Bwd-only, and BiD RS-FB algorithms.

First, let us consider Fwd-only RS-FB algorithms, where both forward and backward recursions utilize only forward decision feedback in the metric computation. In other words, this algorithm uses RSSD only during the forward recursion, where a *forward* survivor (corresponding to a sequence of "past" decided symbols) is associated with each state at each epoch. This algorithm can be obtained from the generic recursions (4.203) and (4.204), by setting $n_\mathrm{f} = n_\mathrm{b} = m_\mathrm{f} = m_\mathrm{b} = 0$. Since the reduced state $s_k = \boldsymbol{a}_{k-L_1}^{k-1}$ is the most recent portion of the full state $S_k = \boldsymbol{a}_{k-L}^{k-1}$, the Fwd-only RS-FB algorithm is the best choice if the ISI channel is minimum-phase, because in this case the taps of the ISI channel, where most of the energy is concentrated, are the ones relative to the information symbols embedded in the reduced state. In this way, the past information symbols, recovered by decision feedback, correspond to the smallest taps and have little influence. The error propagation is not significant, and the backward recursion is then based on a set of reliable survivors.

However, if the channel is not minimum-phase, the symbols associated with the reduced state may not be the ones corresponding to the taps of the channel with most of the energy. In the worst case, if the channel is maximum-phase, they correspond to the taps of the channel with lower energy. The information symbols corresponding to the taps with most of the energy are thus recovered with decision feedback, and error propagation, with severe performance degradation, is very likely. In any case, there is no gap in the completion operation. Multiple completion is also unnecessary, since completion operations at different epochs, used to derive a soft output on a_k, provide the same extrinsic information.

On the other hand, the Bwd-only RS-FB algorithm is obtained by applying RSSD only in the backward recursion, where a *backward* survivor (corresponding to a sequence of "future" decided symbols) is associated with each state at each epoch. In this case, the backward recursion is run first and the forward recursion is then operated with the same metric as the backward recursion. As an example, this algorithm can be obtained from the generic recursions (4.203) and (4.204), by setting $n_\mathrm{f} = n_\mathrm{b} = L - L_1$ and $m_\mathrm{f} = m_\mathrm{b} = 0$. The Bwd-only RS-FB algorithm is a specular version of the Fwd-only RS-FB algorithm. Reasoning as in the case of the Fwd-only RS-FB algorithm, one can conclude that the Bwd-only RS-FB algorithm is the best choice if the ISI channel is maximum-phase. In fact, in this case the taps of the ISI channel, where most of the energy is concentrated, are the ones relative to the information symbols embedded in the reduced state. Error propagation and, consequently, performance degradation increase if the channel is not maximum-phase. The worst case is given by a minimum-phase channel. As for a Fwd-only RS-FB algorithm, in this case as well there is no need to fill the gap and to consider multiple completion.

In BiD RS-FB algorithms, forward and backward decision feedback is used. Either one of the recursions, run first, must be designed in such a way that only the decision feedback in its direction is used. The other recursion, run later, can partly rely on the decision feedback generated in the first recursion. At the extreme, each recursion utilizes only the decision feedback in the corresponding direction. In this case, the two recursions can be run independently—this is the only case, relative to BiD RS-FB algorithms, considered for the numerical results in Section 4.8.6. As an example, this BiD RS-FB algorithm can be obtained from the general recursions (4.203) and (4.204), by setting $n_\mathrm{f} = m_\mathrm{f} = m_\mathrm{b} = 0$ and $n_\mathrm{b} = L_2$ ($0 < L_2 \leq L - L_1$). The special case of independent recursions is obtained by setting $L_2 = L - L_1$. Obviously, for a given number of (reduced) states, BiD RS-FB algorithms have a larger complexity than Fwd-only and Bwd-only RS-FB algorithms. This complexity increase is due to the fact that BiD RS-FB algorithms utilize both forward and backward survivor maps. Filling the gap and multiple completion further increase the complexity. Hence, a trade-off between complexity and performance has to be

properly found.

Performance-wise, since each recursion contains the decision feedback in its direction, it might happen that even if there is severe error propagation in one of the recursions (e.g., the forward recursion using only forward decision feedback for maximum-phase channels), only little error propagation may affect the other recursion. A sort of *diversity* is then provided by the two recursions, and multiple completion can exploit this phenomenon. By properly filling the gap, setting multiple completion, and selecting suitable weight exponents, more reliable soft-input values could dominate over less reliable soft-input values. Unfortunately, an appropriate selection is likely to be problem-dependent and possibly difficult to predict in advance without a trial-and-error process. Nevertheless, intuitively, it is possible to find a BiD RS-FB algorithm that performs relatively well in almost any ISI channel. This is important especially when the ISI channel is mixed-phase (or for any FSM with an equivalent property), in which case Fwd-only and Bwd-only RS-FB algorithms offer unsatisfying performance. Moreover, BiD RS-FB algorithms may yield a better performance, with respect to Fwd-only or Bwd-only RS-FB algorithms, in the case of strong state reduction, especially for mixed-phase ISI channels.

4.8.6 Examples of Application of Generalized RS-FB Algorithms

The performance of the considered receivers is assessed by means of computer simulations in terms of BER versus bit SNR E_b/N_0, E_b being the received signal energy per information bit and N_0 being the one-sided noise power spectral density. In all cases, we assume perfect CSI at the receiver side. All considered FSMs are simple FSMs with memory L. The simple FSM corresponding to a reduced-state trellis has memory L_1. In any component decoder, the min-sum version of the particular RS-FB/FB algorithm is considered.

According to the general structure proposed in Section 4.8.5, a particular structure of RS-FB algorithms can be realized through an assignment of L_1, L_2, m_f, n_f, m_b, n_b, $w_a^c(k,m)$, $w_x^c(k,n)$, and $w_a^o(k,q)$ in (4.209), (4.210), and (4.213). In Table 4.1, a summary of a few possibilities, which are considered in the following, is given. In particular, in all considered schemes $m_\mathrm{f} = n_\mathrm{f} = m_\mathrm{b} = 0$, so that these parameters are not indicated in Table 4.1. The parameter $\theta \in (0,1]$ appearing in Table 4.1 represents the scaling factor embedded in the weight exponents. Note that Fwd-only and BiD-Nfg-sc RS-FB algorithms are equivalent to the existing Fwd-only and BiD RS-FB algorithms in [156] and [151], respectively. The BiD-Ffg-Fmc and BiD-Pfg-Pmc RS-FB algorithms may be viewed as extensions of the BiD-Nfg-sc RS-FB algorithms, where the completion and soft-output combining make use of a larger portion of the available information. The particular structure of the BiD-Pfg-Pmc RS-

Table 4.1: Different parameter settings for RS-FB algorithms.

	L_2	$w_a^c(k,m)$	$w_x^c(k,n)$	$w_a^a(k,q)$	n_b
Fwd-only	0	$\delta_K(m)$	$\delta_K(n)$	$\theta\,\delta_K(q)$	0
BiD-Nfg-sc	$L-L_1$	$\delta_K(m)$	$\delta_K(n)$	$\theta\,\delta_K(q)$	L_2
BiD-Ffg-Fmc	$L-L_1$	$\delta_K(m)$	1	$\frac{\theta}{L_1+1}$	L_2
BiD-Pfg-Pmc	$L-L_1$	$\delta_K(m)$	$\delta_K(n)$ $+\delta_K(n-L_2)$	$\frac{\theta}{2}\delta_K(q)$ $+\frac{\theta}{2}\delta_K(q-L_1)$	L_2

FB algorithm depends on the considered system. The choices in Table 4.1 guarantee good performance and robustness in the considered cases.

Uncoded BPSK Transmission over ISI/AWGN Channels

In this case, we consider uncoded transmission with BPSK over two 12-tap ($L = 11$) and one 8-tap ($L = 7$) ISI/AWGN channels. The sample r_k at the receiver side has the following expression:

$$r_k = \sum_{m=0}^{L} f_m a_{k-m} + n_k \tag{4.215}$$

where $\boldsymbol{f} = (f_0, \ldots, f_L)$ is the channel discrete impulse response and $\{n_k\}$ is a sequence of iid zero-mean Gaussian random variables with variance N_0. The three considered channels are the following:

- Channel A (12-tap): $f_0 = \ldots = f_{11} = c_1$ (equal-weight ISI channel).
- Channel B (12-tap): $f_0 = f_{11} = c_2, f_1 = f_{10} = 2c_2, \ldots, f_5 = f_6 = 6c_2$ (triangular ISI channel).
- Channel C (8-tap): $f_i = c_3(\xi) \cdot v_i(\xi)$ where $v_i(\xi)$, $i = 0, \ldots, 7$ are iid Gaussian random variables with zero mean and unitary variance (random ISI channel[30], where ξ indicates a specific realization).

[30]Channel C can be interpreted as a particular fading channel. This will be confi rmed by the behavior of the BER curves, relative to this channel, shown in Figure 4.29.

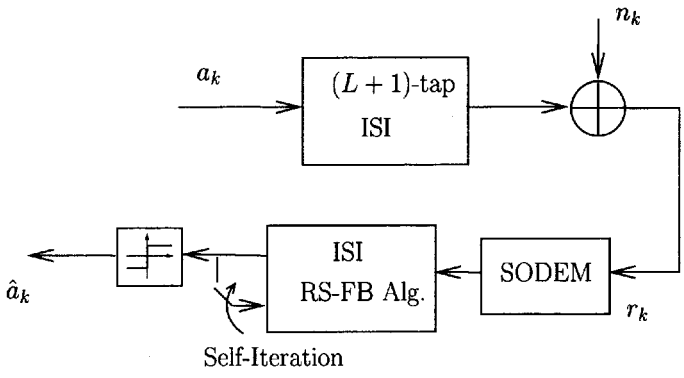


Figure 4.26: Transmitter and receiver for the uncoded BPSK transmission over ISI/AWGN channels.

The constants c_1, c_2 and $c_3(\xi)$ are suitable normalization constants such that $||\boldsymbol{f}||^2 = \sum_{i=0}^{L} |f_i|^2 = 1$. Figure 4.26 shows the transmitter and self-iterative receiver for this isolated ISI system. In all cases, the receiver is constituted by a single RS-FB decoder and the performance results with Channels A, B and C, optimized over θ, are illustrated in Figure 4.27 ($\theta = 0.3$), Figure 4.28 ($\theta = 0.3$), and Figure 4.29 ($\theta = 0.6$), respectively. For BiD RS-FB algorithms the performance for $I = 1$ (solid lines) and $I = 5$ (dashed lines) self-iterations is shown. Note that a single self-iteration corresponds to a single activation of the reduced-state decoder. In this sense, for $I = 1$ there is no "real" self-iteration, i.e., the generated soft-output values are not fed back to the input of the decoder. However, for the Fwd-only RS-FB algorithm, only the performance for $I = 1$ (i.e., no self-iteration) is illustrated, since, based on our results, there is no significant improvement by increasing I. A possible explanation for this phenomenon is that the self-iteration for the Fwd-only RS-FB algorithm is likely to reinforce the decision from the previous iteration, since only forward survivor paths are used in the algorithm.

In Figure 4.27, the performance of the Fwd-only RS-FB algorithm with $L_1 = 3$ ($\zeta' = 8$ states) is about 2 dB away from the performance of the full-state receiver ($\zeta = 2^L = 2048$) and there is no substantial improvement by increasing ζ' up to 128. Considering $L_1 = 1$ ($\zeta' = 2$ states) the performance of the BiD-Ffg-Fmc RS-FB algorithm worsens for an increasing number of iterations at low SNR, whereas the performance of other BiD RS-FB algorithms shows a gain of about 2 dB. This fact suggests that when considering a strong state reduction with Channel A, using all available soft-input information may degrade the performance, while partially using it may help. However, for $\zeta' = 4$, the performance improves for an increasing number

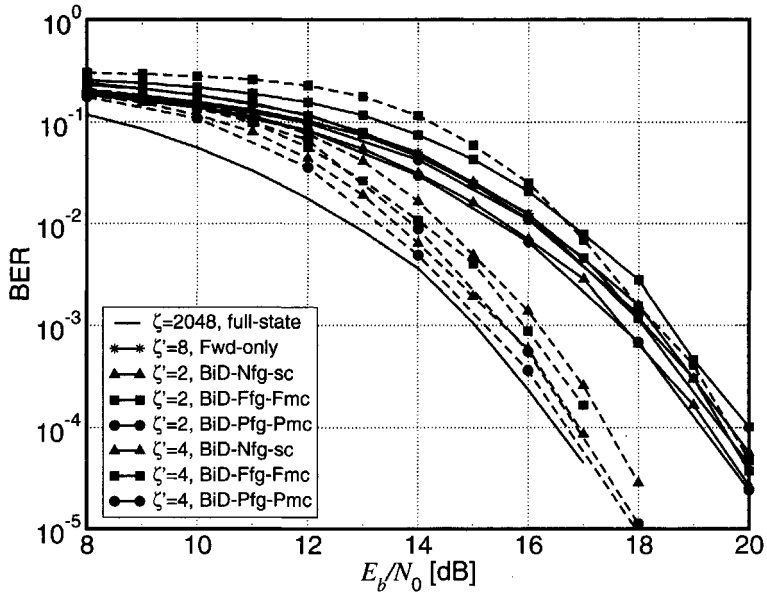


Figure 4.27: Performance comparisons of various self-iterative detection algorithms for Channel A assuming perfect CSI. The number of considered self-iterations I is 1 (solid lines) or 5 (dashed lines), and the number of states is indicated by ζ' in the case of state reduction. For comparison, the performance of the full-state receiver ($\zeta = 2048$) is also shown. Reproduced from [166], ©2002 IEEE, by permission of the IEEE.

of self-iterations in all BiD cases.

In Figure 4.28, the performance results obtained with Channel B are shown. As above, the performance obtained considering the FB algorithm with $\zeta = 2048$ states is compared to the performance obtained considering different RS-FB algorithms. When considering the Fwd-only RS-FB algorithm a performance gain of 1 dB is observed when increasing the number of states from $\zeta' = 32$ ($L_1 = 5$) to $\zeta' = 128$ ($L_1 = 7$). In the latter case, the performance loss, compared to the case with the full-state FB algorithm, is around 2 dB. In both Fwd-only cases, the performance is better than the one obtained with the considered BiD RS-FB algorithms with $\zeta' = 8$, both for $I = 1$ and $I = 5$ self-iterations. The BiD-Pfg-Pmc RS-FB algorithm with $\zeta' = 16$ and $I = 5$ self-iterations shows a gain, in terms of SNR, of 1 dB with respect to the Fwd-only RS-FB algorithm with $\zeta' = 128$, and the performance loss, compared to the full-state case, is reduced to 1 dB. As for Channel A, the best performance was obtained in all reduced-state cases by considering $\theta = 0.3$.

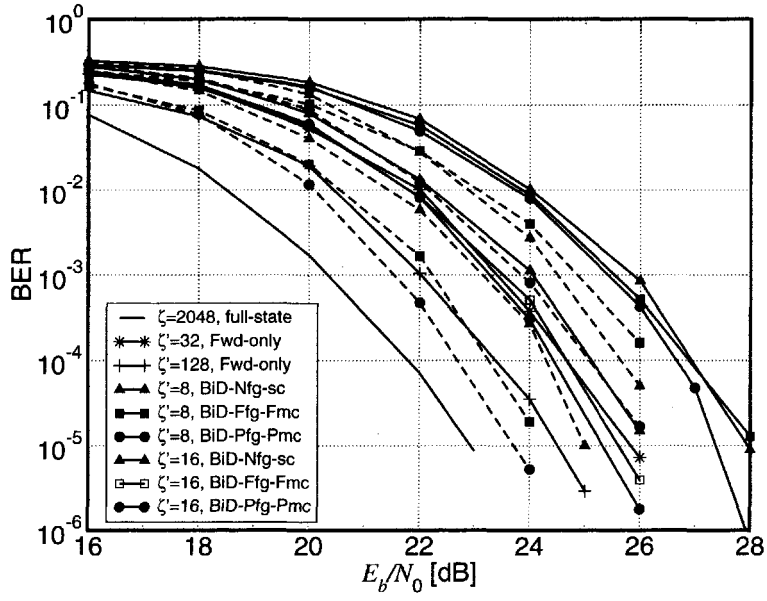


Figure 4.28: Performance comparisons of various self-iterative detection algorithms for Channel B assuming perfect CSI. The number of considered self-iterations I is 1 (solid lines) or 5 (dashed lines), and the number of states, in the case of state reduction, is indicated by ζ'. For comparison, the performance of the full-state receiver ($\zeta = 2048$) is also shown. Reproduced from [166], ©2002 IEEE, by permission of the IEEE.

Finally, in Figure 4.29, the performance with Channel C is analyzed. In this case as well, we compare the performance of the Fwd-only RS-FB algorithm and various BiD RS-FB algorithms to that of the full-state FB algorithm. When considering the Fwd-only RS-FB algorithm with $\zeta' = 64$ states, the performance loss, with respect to the full-state receiver with $\zeta = 128$, is around 2 dB. Considering the BiD-Nfg-sc RS-FB algorithm with $\zeta' = 8$ and $\zeta' = 16$, the performance degrades when increasing the number of self-iterations from $I = 1$ to $I = 5$. The best performance is obtained by considering the BiD-Ffg-Fmc RS-FB algorithm. This is intuitively reasonable, since the channel is random. In fact, due to this randomness, it turns out that the best strategy is using all the available soft-input information.

When considering a minimum-phase channel, the Fwd-only RS-FB algorithm has a very good performance [151, 156]. On the other hand, the performance of this RS-FB algorithm worsens dramatically if the channel is maximum-phase. In contrast, BiD RS-FB algorithms can cope well with different channels through self-iterative decoding, even when the number of states is drastically reduced. However,

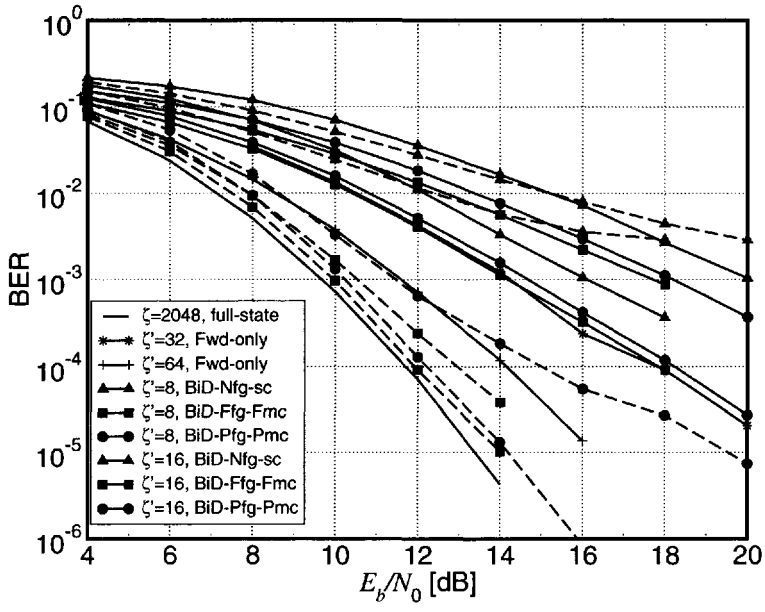


Figure 4.29: Performance comparisons of various self-iterative detection algorithms for Channel C assuming perfect CSI. The number of considered self-iterations I is 1 (solid lines) or 5 (dashed lines), and the number of states is indicated, in the case of state reduction, by ζ'. For comparison, the performance of the full-state receiver ($\zeta = 2048$) is also shown. Reproduced from [166], ©2002 IEEE, by permission of the IEEE.

the implementation complexity of BiD RS-FB algorithms is higher than that of the Fwd-only RS-FB algorithm.

Convolutionally Coded Transmission over an AWGN Channel

Besides an AWGN/ISI channel, we also consider, as an example of an isolated system, a single *convolutional code*, whose output sequence is transmitted over an AWGN channel. In particular we consider the following two NRC codes [17]:

- Convolutional code 1 (CC1): (2,1,9) NRC code[31] with generators (in octal form) $G_1 = (7604)_8$ and $G_2 = (4174)_8$.
- Convolutional code 2 (CC2): (2,1,12) NRC code with generators $G_1 = (42554)_8$ and $G_2 = (77304)_8$.

[31]The notation (n, k, L), used for a convolutional code, indicates that the rate of the code is k/n and the memory is $L + 1$.

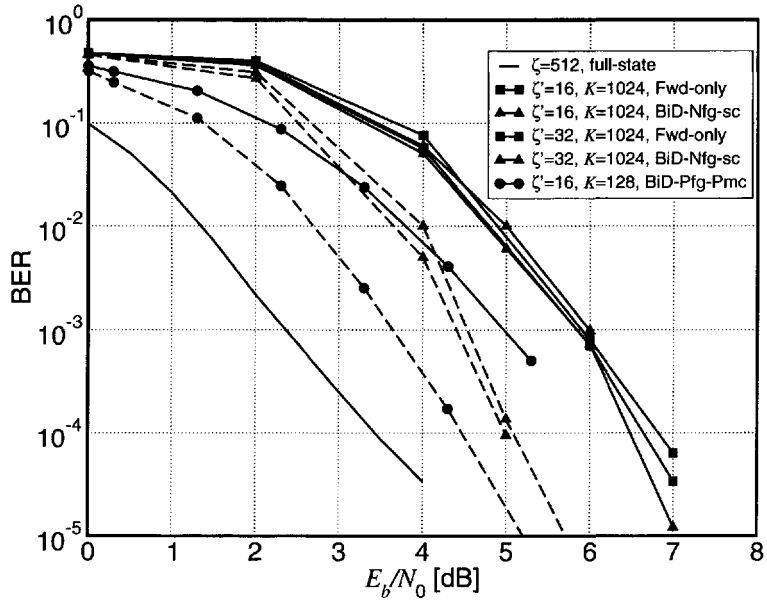


Figure 4.30: Performance of a (2,1,9) NRC code with generators $G_1 = 7604$ and $G_2 = 4174$ with BPSK on an AWGN channel. $I = 1$ (solid lines) and $I = 5$ (dashed lines) self-iterations are considered for different numbers of reduced states ζ' and different packet length K. The performance in the full-state case ($\zeta = 512$ states) is also shown.

In both cases, we evaluate the performance of various RS-FB algorithms considering different transmission lengths, namely $K = 1024$ and $K = 128$. For long transmission length ($K = 1024$), the numerical results show that the performance degrades rapidly when reducing the number of states [168]. A possible explanation of this phenomenon is that the modulo-2 arithmetic used for a binary convolutional code may induce a catastrophic behavior when considering reduced-state decoding. In other words, error propagation is more likely to occur in modulo-2 arithmetic than in the real number arithmetic used in the ISI case. However, by reducing the transmission length ($K = 128$) a better performance can be obtained, probably because of the reduced potential catastrophic behavior. When considering self-iterative decoding jointly with BiD RS-FB algorithms, the best performance is obtained in all cases with $\theta = 0.3$.

In Figure 4.30, the performance of CC1 is shown. When considering $K = 1024$ and the Fwd-only RS-FB algorithm, there is basically no improvement by increasing the number of states from $\zeta' = 16$ to $\zeta' = 32$. Considering BiD RS-FB algorithms,

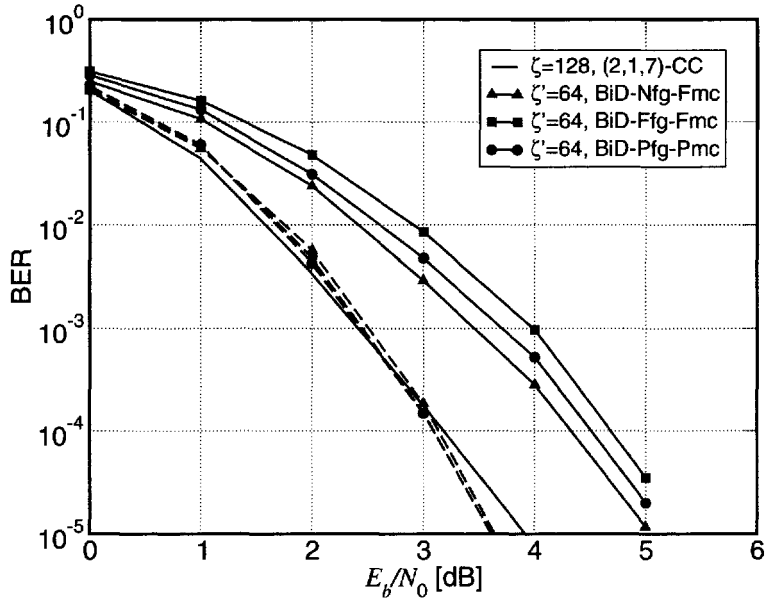


Figure 4.31: Performance of a (2,1,12) NRC code with generators $G_1 = 42554$ and $G_2 = 77304$ with BPSK on an AWGN channel. $I = 1$ (solid lines) and $I = 5$ (dashed lines) self-iterations are considered for different numbers of reduced states ζ'. A packet length $K = 128$ is considered in all cases, and for comparison, the performance of a full-state ($\zeta = 128$) (2,1,7) convolutional code is also shown.

the performance for $I = 5$ decoding iterations (dashed lines), with $K = 1024$, improves by about 1.2 dB with respect to the Fwd-only case. Reducing the packet length to $K = 128$ and considering BiD-Pfg-Pmc, a further improvement of 1 dB is observed. The performance loss, with respect to the full-state receiver, reduces to 0.6 dB at a BER of 10^{-5}. In Figure 4.31, the performance of CC2 with a packet length $K = 128$ is shown, considering various BiD RS-FB algorithms with $\zeta' = 64$ states. As one can see, the performance at $I = 5$ self-iterations (dashed lines) is almost identical in all cases. For comparison, the performance of a (2,1,7) convolutional code is shown. In particular, the performance is basically the same as the one obtained by considering CC2 and BiD RS-FB algorithms.

TCM over an ISI/AWGN Channel

In this system, a sequence of bits is encoded by an 8-state, rate-2/3 Ungerboeck code with 8-PSK (see [9, Figure 9]). The modulated sequence is then interleaved by a 32×32 block interleaver and transmitted over an equal-weight ISI channel with 5

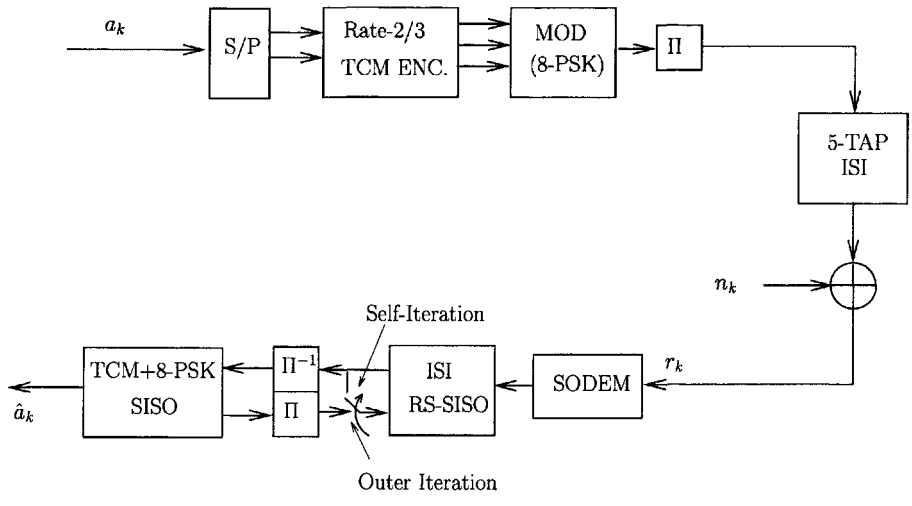


Figure 4.32: Transmitter and receiver for the TCM system over an ISI/AWGN channel.

taps. The noise samples are iid complex-circular Gaussian random variables with zero mean and variance N_0. At the receiver side, the outer decoder, relative to the Ungerboeck code, uses the FB algorithm, while the inner detector, relative to the ISI channel, makes use of an RS-FB algorithm. Figure 4.32 illustrates this communication system. The performance of the system is analyzed by considering different possible settings of the RS-FB algorithm. The parameter L_1 is related to the memory of this algorithm. We denote by $(I_\mathrm{i}, I_\mathrm{o})$ the number of inner self-iterations and outer iterations, respectively, and the parameters θ_i and θ_o refer to the scaling factors considered in the inner self-iterative detection and in the outer iterative decoding.

In Figure 4.33, we compare the performance of the BiD-Nfg-sc RS-FB algorithm proposed in [151] and the BiD-Pfg-Pmc RS-FB algorithm. For $\theta_\mathrm{i} = \theta_\mathrm{o} = 1$ and $(I_\mathrm{i}, I_\mathrm{o}) = (3, 5)$, the performance of the two algorithms is compared in the cases of $L_1 = 1$ ($\zeta' = 8$ states) and $L_1 = 2$ ($\zeta' = 64$ states). The BiD-Pfg-Pmc RS-FB algorithm guarantees a performance gain, with respect to the BiD-Nfg-sc RS-FB algorithm, of the order of 0.3 dB in both cases. The performance of the 64-state BiD-Nfg-sc RS-FB algorithm is comparable to that of the full-state (4096-state) FB algorithm with $(I_\mathrm{i}, I_\mathrm{o}) = (1, 1)$. However, considering $\theta_\mathrm{i} = 0.62$ and $\theta_\mathrm{o} = 0.37$, the performance can be greatly improved. The performance of the BiD-Pfg-Pmc RS-FB algorithm is shown for $L_1 = 1$ and $L_1 = 2$, and considering, in both cases, $(I_\mathrm{i}, I_\mathrm{o}) = (2, 6)$ and $(I_\mathrm{i}, I_\mathrm{o}) = (2, 10)$. As one can see, in both cases there is an improvement of the order of 1 dB with respect to the previous case with $\theta_\mathrm{i} = \theta_\mathrm{o} = 1$.

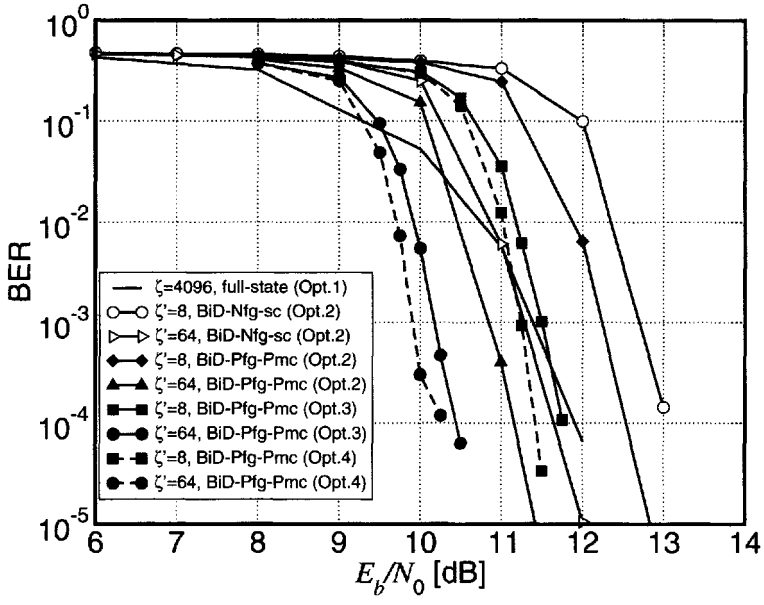


Figure 4.33: Performance comparison of various iterative detection algorithms for the TCM/ISI channel (Opt. 1: $(I_\mathrm{i}, I_\mathrm{o}) = (1,1)$, $(\theta_\mathrm{i}, \theta_\mathrm{o}) = (1,1)$; Opt. 2: $(I_\mathrm{i}, I_\mathrm{o}) = (3,5)$, $(\theta_\mathrm{i}, \theta_\mathrm{o}) = (1,1)$; Opt. 3: $(I_\mathrm{i}, I_\mathrm{o}) = (2,6)$, $(\theta_\mathrm{i}, \theta_\mathrm{o}) = (0.625, 0.375)$; Opt. 4: $(I_\mathrm{i}, I_\mathrm{o}) = (2,10)$, $(\theta_\mathrm{i}, \theta_\mathrm{o}) = (0.625, 0.375)$). Reproduced from [166], ©2002 IEEE, by permission of the IEEE.

4.9 Applications to Wireless Communications

In this section, we discuss a few applications of the general framework for FB algorithms previously introduced. We wish to point out that the considered examples are limited, and the interested reader should try to apply the proposed detection tools to other classes of transmission system, where iterative detection techniques, based on the use of FB algorithms, are attractive.

4.9.1 Noncoherent Iterative Detection of Binary Linear Coded Modulation

Consider the communication system depicted in Figure 4.34. An information sequence $\{a_k\}$, composed of iid symbols belonging to a binary alphabet, is encoded and linearly modulated at the receiver side, and then transmitted as a time-continuous signal. The transmitted signal undergoes a phase rotation $\theta(t)$ and is transmitted over

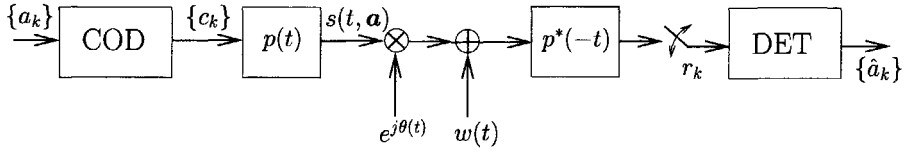


Figure 4.34: Communication system model for transmission over phase-uncertain channels.

an AWGN channel. The phase rotation $\theta(t)$ is initially assumed to be constant during the entire transmission and modeled as a random variable θ with uniform distribution in the interval $[0, 2\pi)$ and independent from the transmitted symbols.

Based on this static phase assumption, it may be easily shown that the sampled output of a filter matched to the shaping pulse is a sufficient statistic for optimal detection of the information sequence [47]. In the absence of ISI and assuming a sampling rate of one sample per modulated symbol, this channel output may be expressed as

$$r_k = c_k e^{j\theta} + n_k \tag{4.216}$$

where c_k is a BPSK (possibly coded) symbol, and $\{n_k\}$ are iid complex noise samples with independent real and imaginary components, each with variance $\sigma^2 = N_0$. The extension to the case of higher order modulation will be considered in the following subsection.

In this subsection, we will consider the noncoherent FB algorithms already partially described in Section 4.7 and referred to as NCSOa and NCSOb algorithms, and another algorithm obtained by applying the principle of SOVA to the noncoherent sequence detection (NSD) algorithm described in Chapter 2, defined as soft-output (SO)-NSD in [162]. The expanded state S_k is defined, as usual, as $(\mu_{k-N}, \boldsymbol{a}_{k-N}^{k-1})$, where μ_k is the state of the encoder/modulator and N is the order of Markovianity. A transition T_k is defined as (S_k, a_k, S_{k+1}).

We first review the considered algorithms. The completion and the two recursions in the NCSOa algorithm can be written as follows:

$$\mathrm{S}[a_k] = \sum_{T_k:a_k} \alpha_k(S_k)\gamma_k(T_k)\beta_{k+1}(S_{k+1}) \tag{4.217}$$

$$\alpha_k(S_k) = \sum_{T_{k-1}:S_k} \alpha_{k-1}(S_{k-1})\gamma_{k-1}(T_{k-1}) \tag{4.218}$$

$$\beta_k(S_k) = \sum_{T_k:S_k} \beta_{k+1}(S_{k+1})\gamma_k(T_k) \tag{4.219}$$

where $\mathrm{S}[a_k]$ indicates the soft-output value on a_k and the exponential metric $\gamma_k(T_k)$ is

$$\gamma_k(T_k) = \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|\right)} \mathrm{SI}[a_k]. \tag{4.220}$$

The NCSOb algorithm is described by the following equations:

$$\mathrm{S}[a_k] = \sum_{T_k:a_k} \mathring{\alpha}_k(T_k)\mathring{\gamma}_k(T_k)\mathring{\beta}_k(T_k) \tag{4.221}$$

$$\alpha_k(T_k) = \sum_{T_{k-1}:S^+(T_{k-1})=S^-(T_k)} \alpha_{k-1}(T_{k-1})\psi_k(T_{k-1},T_k) \tag{4.222}$$

$$\beta_k(T_k) = \sum_{T_k:S^+(T_k)=S^-(T_{k+1})} \beta_{k+1}(T_{k+1})\phi_k(T_k,T_{k+1}) \tag{4.223}$$

where the metrics $\gamma_k(T_k)$, $\psi_k(T_{k-1},T_k)$ and $\phi_k(T_k,T_{k+1})$ are defined as

$$\gamma_k(T_k) = \exp\left(-\frac{1}{2\sigma^2}\sum_{i=0}^{N}|c_{k-i}|^2\right) \mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right) P\{S_k\}\mathrm{SI}[a_k] \tag{4.224}$$

$$\psi_k(T_{k-1},T_k) = \exp\left(-\frac{|c_{k-N-1}|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N+1} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right)} \mathrm{SI}[a_{k-N-1}] \tag{4.225}$$

$$\phi_k(T_{k-1},T_k) = \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N+1} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N+1} r_{k-i}c_{k-i}^*\right|\right)} \mathrm{SI}[a_{k+1}]. \tag{4.226}$$

The SO-NSD algorithm is obtained by applying the principle of SOVA and using a metric at epoch k with the following expression (see Section 3.11.2):

$$\lambda_k(T_k) = \frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right| - \left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|}{\sigma^2} + \mathrm{MI}[a_k] \tag{4.227}$$

where we recall that $\mathrm{MI}[a_k] = \ln \mathrm{SI}[a_k]$. We remark that the SO-NSD generates reliability values corresponding to LLRs (this is expected, since they derive from metric differences). The generated soft outputs are obtained based on successive updatings which depend on the particular channel realization, so that there is no simple expression for the final generated LLR relative to an information symbol a_k. Since an LLR is just a normalized version of a soft output (i.e., in the logarithmic domain it is the reliability of the bit being one assuming that the reliability of the bit being zero is set to 0), we can define:

$$\mathrm{M}[a_k] \triangleq \ln \mathrm{S}[a_k] = \begin{cases} \mathrm{LLR}[a_k] & \text{if } a_k = 1 \\ 0 & \text{if } a_k = 0 \end{cases} \tag{4.228}$$

where $\mathrm{S}[a_k]$ is the reliability, in the probability domain, relative to a_k. From this final reliability, the *extrinsic information*, i.e., the portion of the soft output to be exchanged in an iterative detection process, in the logarithmic domain, can be obtained as follows:

$$\mathrm{MO}[a_k] \triangleq \ln \mathrm{SO}[a_k] = \mathrm{M}[a_k] - \mathrm{MI}[a_k]. \tag{4.229}$$

More details about SO-NSD can be found in [162].

For the purpose of comparison, we consider two receiver schemes.

- In the first scheme, which will be referred to as *separate detection and decoding*, the coding memory is somehow separated, at the receiver side, from the channel memory. This means that there is a block dedicated to *detection* which recovers the impairments introduced by the channel, and a block dedicated to *decoding* which exploits the memory introduced at the transmitter side.

- In the second scheme, referred to as *joint detection and decoding*, the memory introduced at the transmitter side and the memory introduced by the channel are exploited simultaneously.

Separate Detection and Decoding

In the first scheme, shown in Figure 4.35, we consider the serial concatenation of a PCCC and a differential encoder. At the receiver side, noncoherent detection is separated from the decoding of the PCCC. For this purpose, the code sequence at the output of the PCCC is interleaved, by means of the block denoted by Π in Figure 4.35 (a), and differentially encoded. At the receiver, shown in Figure 4.35 (b), the detection is performed by means of one of the described soft-output noncoherent algorithms, designed to take into account differential encoding. The soft output of this noncoherent block is then deinterleaved, by the block denoted by Π^{-1}, and sent to a standard

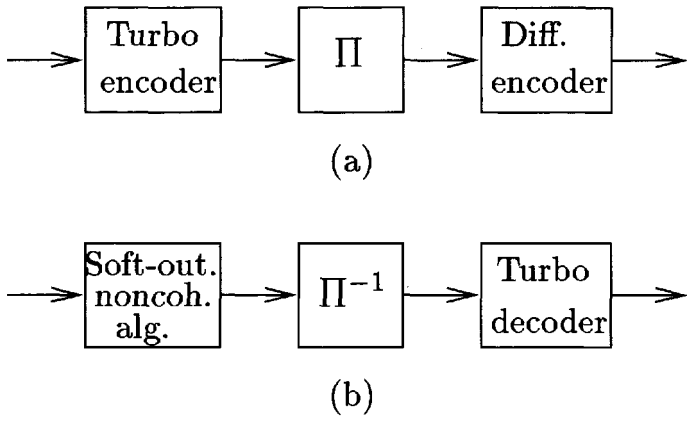


Figure 4.35: Schemes with separate detection and decoding using the proposed soft-output noncoherent algorithms: (a) transmitter and (b) receiver.

turbo decoder which provides symbol decisions after the iterations necessary for the decoding process to converge. These iterations do not involve the soft-output noncoherent predetection block. The interleaver is placed before the differential encoder in order to break up the dependence introduced by the PCCC. Differential encoding is a simple way to obtain a noncoherently noncatastrophic code [150]. We remark that in this case no further performance improvement is obtained by iterating between the outer turbo decoder and the inner differential detector. Numerical results and a comparison with a combined detection and decoding scheme relative to a PCCC will be considered in the next section.

Joint Detection and Decoding

In this case, we first consider a PCCC. The receiver is based on a standard turbo decoder, where each component decoder implements combined noncoherent detection and decoding, using one of the proposed noncoherent soft-output algorithms. In this case, it is not necessary to use differential encoding if the constituent RSC codes are noncoherently noncatastrophic [150]. We consider for example the case of the systematic PCCC with rate 1/2 proposed in [33], where, for each transmitted systematic bit, an additional parity bit is transmitted, obtained by alternately puncturing the coded bits generated by the two component encoders. The receiver is shown in Figure 4.36, in which $r_k^{(1)}$ and $r_k^{(2)}$ denote the channel observations relative to systematic and parity bits, respectively. In Figure 4.36, the soft outputs generated by each component decoder are carried by dashed lines, whereas solid lines carry channel (noncoherent) outputs.

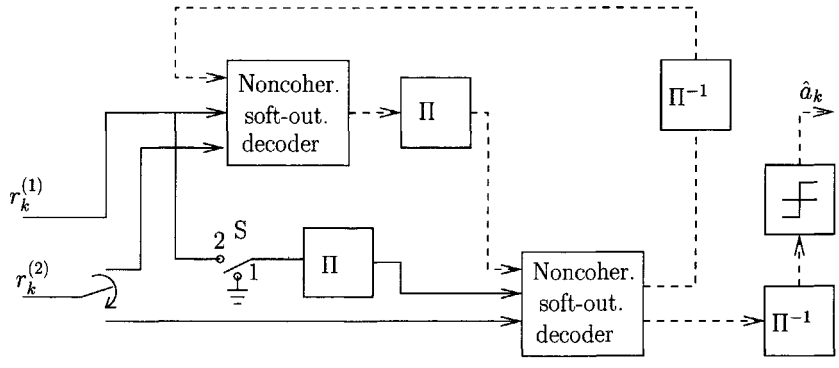


Figure 4.36: Receiver with combined detection and decoding for a PCCC of rate 1/2. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

A remark on the role of switch S in Figure 4.36 is in order. Depending on its position, two schemes for noncoherent iterative decoding can be obtained. In the first one, referred to as *asymmetric*, the second soft-output decoder does not directly use the channel outputs $\{r_k^{(1)}\}$, suitably interleaved, relative to the transmitted systematic bits. This configuration is obtained when the switch S is in position "1." A second scheme, referred to as *symmetric*, may be obtained when the switch S is in position "2." The symmetric and asymmetric schemes are not equivalent for both conceptual and practical reasons. Except for channels with constant or slowly varying phase over each data block, where both symmetric or asymmetric schemes may be adopted, for a time-varying phase, only the asymmetric scheme exhibits good performance. In fact, interleaving the channel outputs $\{r_k^{(1)}\}$ in order to pass them to the second component decoder may increase the phase difference between consecutive observations. Therefore, the performance of the symmetric scheme necessarily degrades. On the contrary, the asymmetric configuration is very robust with respect to a time-varying phase, as will be shown in the numerical results. For a channel phase constant over each data block, the symmetric scheme allows one to obtain a better performance because the second noncoherent component decoder operates with a double number of noncoherent channel outputs and, therefore, with a more refined implicit phase estimation.[32] This second component decoder is the bottleneck in the asymmetric scheme and, for this reason, may require a larger value of the parameter N and thus have larger complexity.

We now show the performance when considering joint noncoherent iterative de-

[32]If puncturing is used, the number of channel outputs used by the second decoder may be more than doubled.

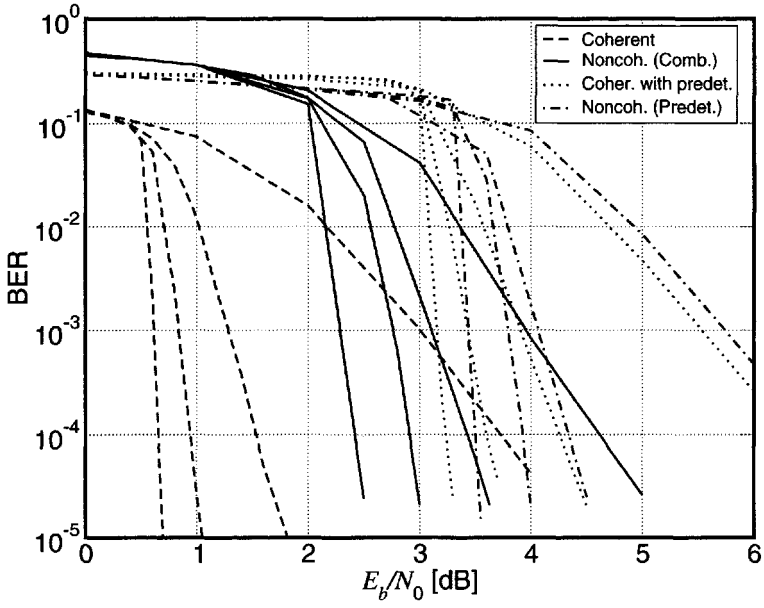


Figure 4.37: BER of the proposed iterative detection schemes using the NCSOb algorithm with predetection (dot-dashed curves), combined detection and decoding (solid curves), coherent decoding (dashed curves), and coherent predetection (dotted curves). The numbers of iterations are 1, 3, 6, and 18 in all cases. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

tection and decoding in the case of a PCCC, and compare the performance with that obtained when considering separate detection and decoding. The performance of the proposed schemes is assessed by means of computer simulations in terms of BER versus E_b/N_0, E_b being the received signal energy per information bit and N_0 the one-sided noise power spectral density. Unless otherwise stated, the channel phase is assumed constant.

We consider the PCCC of rate 1/2, with 16-state RSC constituent codes with generators $G_1 = (37)_8$, $G_2 = (21)_8$, and 256×256 inner interleaver described in [33], and BPSK. The RSC code is noncoherently noncatastrophic. Therefore, differential encoding is not used with combined noncoherent detection and decoding. The performance of the considered schemes using the NCSOb algorithm and SO-NSD has been evaluated. Numerical results for the NCSOb algorithm are shown in Figure 4.37 for various numbers of iterations (1, 3, 6, and 18, in all cases). A value of $N = 2$ is used for the noncoherent predetection block of Figure 4.35 and each noncoherent

soft-output decoder of Figure 4.36. A performance loss of about 3 dB at a BER of 10^{-4} is exhibited by the predetection scheme in Figure 4.35 with respect to coherent combined decoding. This predetection scheme exhibits a loss of only 0.25 dB with respect to a reference scheme which performs soft-output coherent differential predetection followed by turbo decoding. This reference scheme is obtained by substituting the predetection block based on the NCSOb algorithm, with the coherent FB algorithm operating on the differential encoder trellis (see [169–171] for similar decoding schemes). An improvement of about 1 dB is achieved using noncoherent combined detection and decoding and the asymmetric scheme (switch S in position "1"). Considering the performance of a single component decoder, shown in [162], the remaining performance loss of this combined scheme appears to be related to the SNR threshold, above which the single component decoder performs well.

The performance of SO-NSD with combined detection and decoding is shown in Figure 4.38 for the same PCCC. Both asymmetric (switch in position "1") and symmetric (switch in position "2") schemes are considered. For the asymmetric scheme, the constituent decoders have[33] $N = 8$ and $\zeta' = 128$ states, whereas in the symmetric scheme these values are $N = 8$ and $\zeta' = 64$ states. For comparison, the performance for coherent decoding and 18 iterations is also shown. At a BER of 10^{-4}, the performance loss with respect to coherent decoding is about 0.7 dB for the symmetric scheme and 1.1 dB for the asymmetric scheme. The specific choices of parameters N and ζ' correspond to levels of complexity such that the computer simulation time is in the order of that necessary when using the NCSOb algorithm with $N = 2$. A comparison of the NCSOb algorithm and SO-NSD for equal values of N shows the superiority of the first scheme. Moreover, by employing the Fwd-only state reduction techniques for FB algorithms introduced in Section 4.8, it is possible to decouple the number of states from the parameter N in the case of the NCSOb algorithm. This is shown in the following example.

We consider a PCCC having the same component encoders as that considered in the previous paragraph, concatenated by a 32×32 pseudorandom interleaver, i.e., we consider an interleaver with reduced length compared with the previous case. At the receiver, each noncoherent component decoder uses the Fwd-only RS-NCSOb algorithm, and Q is the reduced-state parameter. The extrinsic information generated by each component decoder is passed to the other one by following the heuristic method discussed in Section 4.5. By trial and error, we found that the best performance is obtained when the extrinsic information generated by both component decoders is weighted by a parameter equal to 0.3. Figure 4.39 shows the performance of the re-

[33] When considering SO-NSD, the number of reduced states ζ' can be decoupled from the order of Markovianity N by considering standard RSSD techniques [162].

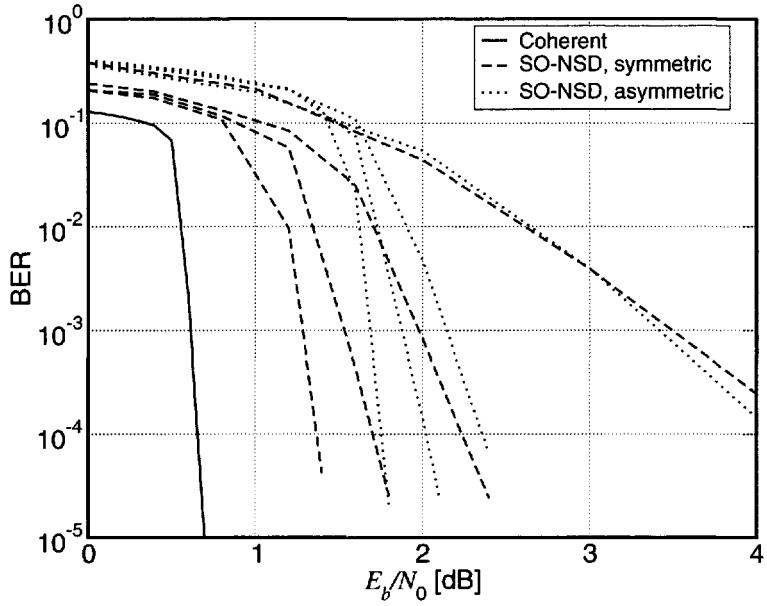


Figure 4.38: BER of the proposed receivers using SO-NSD with combined detection and decoding for asymmetric (dotted curves) and symmetric (dashed curves) schemes. The numbers of iterations are 1, 3, 6 and 18 in both cases. The performance for coherent decoding (solid curve) and 18 iterations is also shown. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

ceiver for various levels of state reduction, specified by the values of N and Q, and compares it to that of a coherent receiver and a noncoherent receiver with $N = Q = 2$ and $\zeta = 64$ (full complexity). In all cases, the considered numbers of iterations are 1, 3 and 6. The RS NCSOb algorithm with $N = 1$ and $\zeta' = 16$ ($Q = 0$) exhibits a performance loss of about 3.2 dB at 6 decoding iterations with respect to the coherent receiver. For $N = 5$ and $\zeta' = 64$ ($Q = 2$), at a BER of 10^{-4} a performance gain of about 1.2 dB is obtained with respect to the full-state case with $N = Q = 2$ and the same number of states ($\zeta = 64$).

The assumption of a channel introducing a phase rotation constant over each data block, used in the derivation of the algorithms, may be removed for the asymmetric scheme. Considering the presence of phase noise, modeled as a time-continuous Wiener process with incremental variance over a signaling interval equal to σ_Δ^2, we investigated the robustness of the proposed algorithms for the system described in the previous paragraph. In Figure 4.40, the performance of an asymmetric scheme with

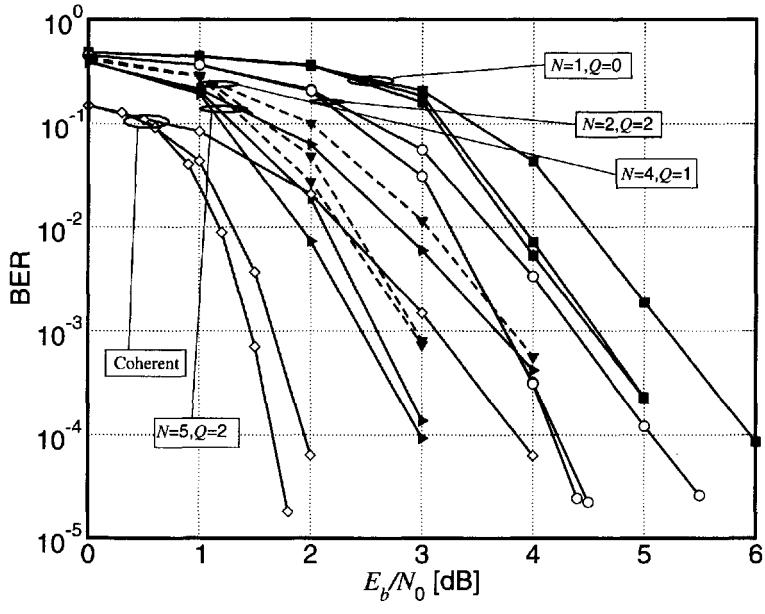


Figure 4.39: Application of the Fwd-only RS NCSOb algorithm to noncoherent decoding of a PCCC. Receivers with various levels of complexity are considered and compared with a full-state receiver (with $N = 3$) and a coherent receiver. The considered numbers of iterations are 1, 3 and 6 in all cases. Reproduced from [156], ©2000 IEEE, by permission of the IEEE.

a 256×256 inner interleaver employing SO-NSD in the constituent decoders characterized by $N = 8$ and $\zeta' = 128$ is shown. Various levels of phase noise with six decoding iterations are considered. As shown in the figure, for a phase noise standard deviation of 5 degrees, the performance loss, with respect to the case of phase noise with zero jitter, is only 0.3 dB at a BER of about 10^{-4}. It is interesting to note that, for the considered range of BER values, the phase noise only affects the E_b/N_0 threshold above which convergence of the iterative process takes place—a larger phase noise has the effect of increasing this threshold. Incidentally, we note that the robustness of the proposed noncoherent algorithms to phase noise is higher for lower values of the phase memory parameter N. If a specific application requires a high robustness, the NCSOb algorithm may be the proper option. In Section 4.9.3 an analysis of the performance of noncoherent iterative decoding schemes for strong phase variations, both with separate or combined detection and decoding, will be considered, together with the insertion of pilot symbols. In that case, we will consider short block lengths, much shorter than the length $K = 65536$ considered in Figure 4.40, and the robust-

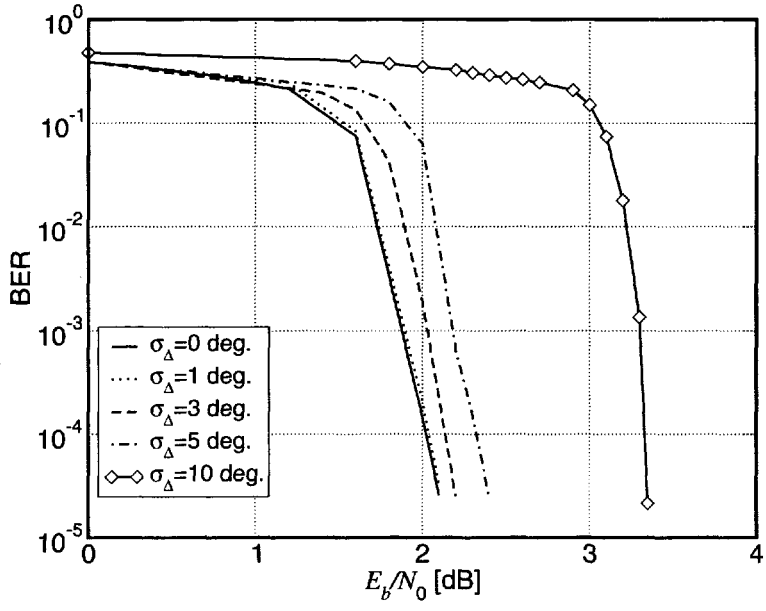


Figure 4.40: BER of the proposed detection scheme using SO-NSD with combined detection and decoding for various levels of phase noise. In all cases, the number of iterations is 6. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

ness will be reduced. This suggests that a long interleaver length, besides improving the performance in the case of perfect CSI at the receiver side, can also increase the robustness of noncoherent iterative decoding to strong phase variations.

We now analyze iterative noncoherent detection of SCCCs. We first consider the concatenation of two interleaved convolutional codes [72]. A special case of this general scheme is given by a serial concatenation of a convolutional code followed by a nonuniform interleaver and a differential code [159, 172]. This allows the separation of the noncoherent detector, which takes into account only the channel memory (together with differential encoding), from the decoder corresponding to the convolutional code. In this sense, this scheme belongs to the class of separate detection and decoding schemes considered in Section 4.9.1. However, as this scheme is a special case of SCCC, we present a few numerical results in the following.

The considered receiver is composed of two decoders concatenated by a deinterleaver as shown in Figure 4.41. The inner decoder is noncoherent and utilizes the proposed algorithms. The outer coherent decoder uses a coherent FB algorithm. Since the inner decoder operates on code symbols, the outer decoder has to provide

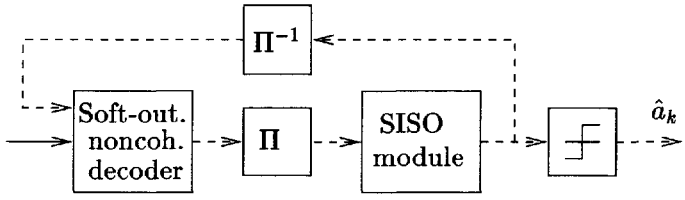


Figure 4.41: Iterative decoding of serially concatenated interleaved codes. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

the *a posteriori* probabilities of these code symbols in order to perform iterative decoding.

We now show some examples relative to noncoherent iterative decoding of SCCCs. In the first case, we consider the concatenation of a 4-state NRNSC code with generators $G_1 = (7)_8$, $G_2 = (5)_8$ and rate 1/2, and a 4-state RSC code with generators $G_1 = (5)_8$, $G_2 = (7)_8$ and rate 1/2. A 256×256 nonuniform interleaver is used [33]. The performance for various numbers of iterations, with $N = 2$ and $N = 4$, is shown in Figure 4.42 and compared with that of the corresponding coherent system. One can again observe that the performance of the noncoherent receiver at each iteration is lower-bounded by that of the corresponding iterative coherent system for the same number of iterations. For 18 iterations, the performance loss of a noncoherent iterative detector, with respect to that of a coherent iterative detector, for $N = 4$ is about 0.7 dB at a BER of 10^{-5}.

In the second case, the overall code is composed of a concatenation of the 16-state RSC code with generators $G_1 = (37)_8$, $G_2 = (21)_8$ and rate-1/2, a 256×256 nonuniform interleaver and a differential code. The corresponding transmission scheme is shown in Figure 4.43. The performance using the NCSOb algorithm in the inner differential detector is shown in Figure 4.44 for various numbers of iterations (1, 3 and 10). The component noncoherent decoder assumes $N = 4$. For comparison, the performance of optimal coherent decoding of the RSC code and iterative coherent decoding of the overall code is also shown. As one may observe, the performance of the noncoherent receiver is obviously lower-bounded by that of the corresponding iterative coherent receiver, for an equal number of iterations. For 10 iterations, the performance loss of the noncoherent receiver with respect to that of the coherent is about 0.3 dB at a BER of 10^{-5}. At low BER, the iterative noncoherent system outperforms the optimal coherent detector for the single RSC code. This behavior, observed also in [159, 169–171, 173], is related to the fact that the concatenation of a convolutional code and a differential code, through interleaving, may generate a new code with better performance, even if the differential encoder has a unitary coding rate. This is due to the recursive nature of differential encoding. In fact,

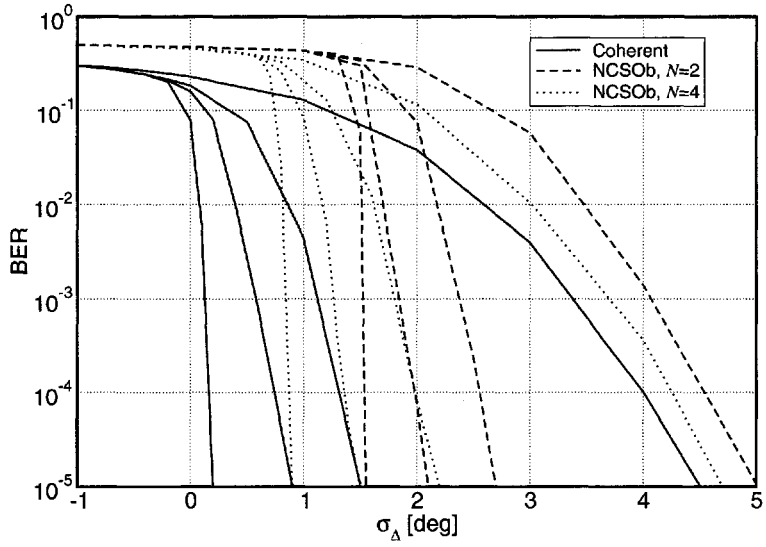


Figure 4.42: BER of the proposed detection scheme using the NCSOb algorithm with $N = 2$ (dashed curves) and $N = 4$ (dotted curves) for the serial concatenation of two interleaved convolutional codes. For comparison, the performance of iterative coherent decoding (solid curves) is also shown. The numbers of iterations are 1, 3, 6 and 18. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

in a serial concatenation, an interleaver gain is possible only for a recursive inner code [72, 170, 171].

The performance of the predetection scheme considered in Figure 4.37 is now further investigated in view of the results in Figure 4.44. As shown in Figure 4.37, the receiver with prediction and inner noncoherent detection exhibits a loss of only 0.25 dB with respect to a reference scheme which performs soft-output coherent predetection followed by turbo decoding. This result, theoretically analyzed in [174], shows the intrinsic limit of schemes based on predetection and was not recognized in [175], where the performance loss was erroneously interpreted as being due to the noncoherent approach. An improved reference receiver, in which the iteration process incorporates a coherent predetection block, was also considered. This scheme, where extrinsic information is exchanged between the predetection block and the two component decoders, exhibits a further performance improvement of only 0.25 dB compared with the previously described reference coherent scheme and confirms the intrinsic limit of predetection schemes. There are two possible alternative interpreta-

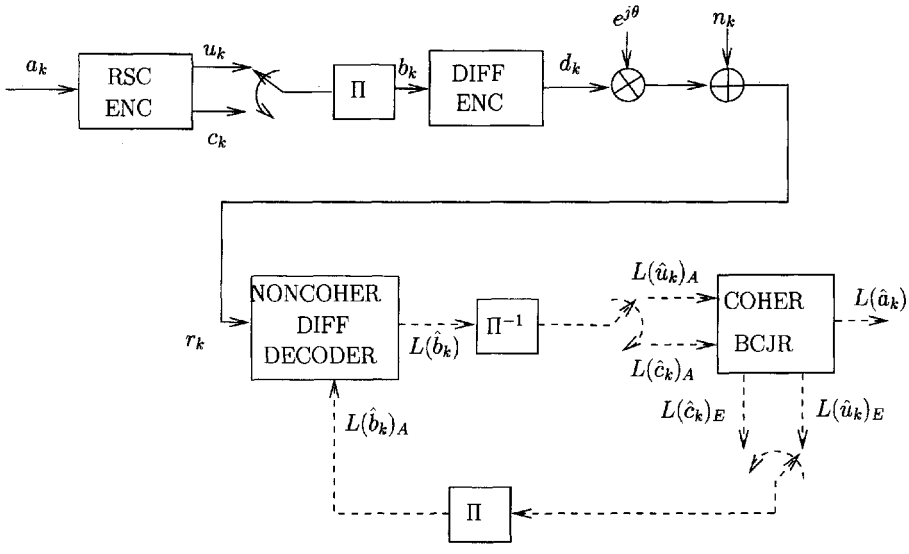


Figure 4.43: Transmission scheme relative to a concatenated code constituted by an outer rate-1/2 RSC code and an inner differential encoder.

tions for this result.

1. According to the first interpretation, differential encoding may render the overall concatenated code significantly worse than the original PCCC. In fact, by adding an interleaver and a differential encoder a new code is obtained, whose performance may be intuitively worse than that of the "practically optimal" PCCC described in [33]. This interpretation does not contradict the results obtained in Figure 4.44 and [169–171] for the concatenation of a differential code and a convolutional code, where the resulting code is better than the simple convolutional code.

2. According to the second interpretation, the overall concatenated code may indeed be no worse[34] than the original PCCC, but for this specific concatenation the iterative decoding process may fail. This second interpretation is suggested in [176].

[34]The overall code cannot be appreciably better than the original PCCC, whose performance is close to the theoretical limit.

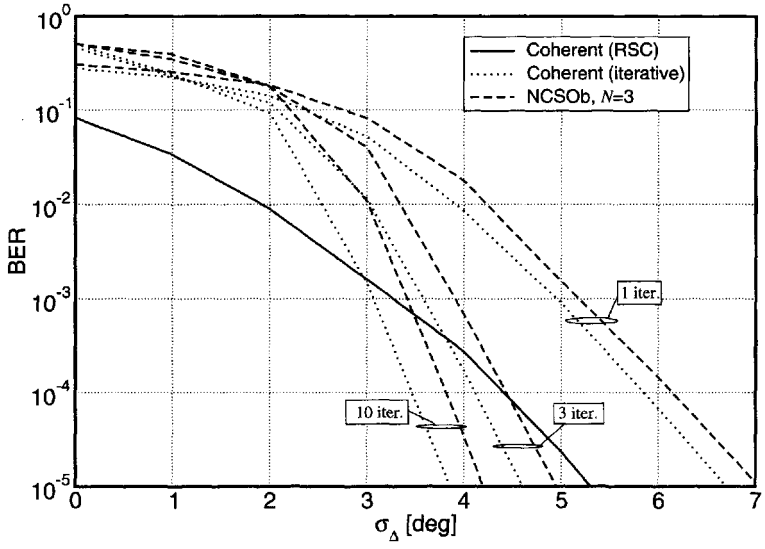


Figure 4.44: BER of the proposed detection scheme using the NCSOb algorithm with $N = 3$ (dashed curves) for the serial concatenation of a convolutional code, an interleaver and a differential encoder. For comparison, the performance of iterative coherent decoding (dotted curves) and optimal coherent decoding of the single convolutional code (solid curve) is also shown. In the cases with iterative detection, the numbers of iterations are 1, 3, and 10. Reproduced from [162], ©2000 IEEE, by permission of the IEEE.

4.9.2 Noncoherent Iterative Detection of Spectrally Efficient Linear Coded Modulation

In this subsection we consider linearly modulated concatenated schemes with high spectral efficiency, obtained by using TCM. As in Section 4.9.1, we distinguish two groups of transmission scheme, namely a first group, where separate detection and decoding is considered, and a second one, where combined detection and decoding is considered.

Separate Detection and Decoding

The first considered class of spectrally efficient schemes uses coding structures based on the concatenation of a turbo TCM (T-TCM) block followed by an inner differential encoder. At the receiver side, a noncoherent differential detector computes *a*

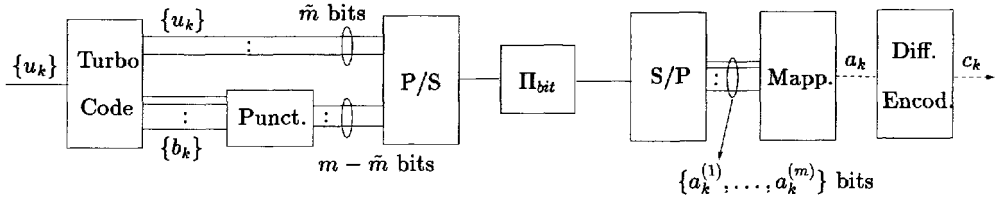


Figure 4.45: Berrou-type PCCC followed by differential encoding on the modulated symbols. Reproduced from [163] by permission of GET/Hermes Science.

posteriori bit probabilities which are passed to the following coherent turbo decoder. As pointed out in the previous section, the introduction of inner differential encoding allows one to obtain noncoherently noncatastrophic coding schemes [47, 150].

The scheme proposed in [177] is basically a systematic PCCC of rate 1/3 followed by a puncturer and a mapper. An immediate extension of this scheme to noncoherent decoding is shown in Figure 4.45, where a sequence of independent bits $\{u_k\}$ undergoes systematic turbo encoding. The code bits $\{b_k\}$ at the output of the turbo encoder are punctured according to a suitable pattern [72]. The systematic and code bits, after being serialized, are interleaved. After interleaving, they are grouped into $m = \log_2 M$ bits and mapped to M-ary complex symbols, undergoing differential encoding. In all block diagrams describing the proposed schemes, we associate solid lines with binary symbols and dashed lines with complex symbols. Furthermore, we use the symbols a_k and c_k to denote, respectively, the input and output symbols of the component encoders which are noncoherently decoded according to the described algorithm. Note that the symbols $\{a_k\}$ are made independent by the interleaver.

The performance of the considered separate scheme is assessed by means of computer simulations in terms of BER versus E_b/N_0. The considered noncoherent soft-output algorithm is the NCSOb algorithm, with the max-log approximation [137]. The generated extrinsic information is weighted by a coefficient as described in Section 4.5: the value of this coefficient, obtained by trial and error, is about 0.6 in all schemes considered hereafter.

In Figure 4.46, the performance in the case of the code shown in Figure 4.45 is shown. The code is that proposed in [177], with internal random 32×32 interleaver. The component RSC codes of the PCCC have generators[35] $G_1 = (37)_8$ and $G_2 = (21)_8$. The PCCC has rate 1/2: every 2 information bits ($\tilde{m} = 2$) two code bits ($m - \tilde{m} = 2$) are retained, with the puncturing pattern considered in [177]. After

[35] In the case of binary codes, for example RSC codes, we refer to the generators of the code as $\{G_i\}$. When referring to Ungerboeck codes, we indicate the generators of the code as $\{h_i\}$.

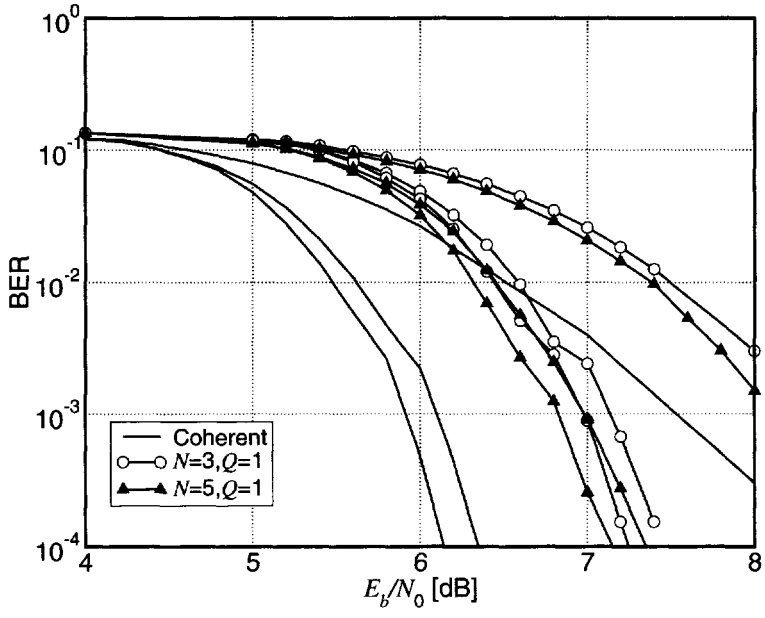


Figure 4.46: Performance of the system shown in Figure 4.45. The considered numbers of inner iterations are 1, 3 and 5 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

random bit interleaving, groups of $m = 4$ bits are mapped to a 16-QAM symbol. It is important to observe that the particular chosen mapping (Gray, reordered, natural, etc.) does not seem to noticeably influence the performance of the noncoherent system. This may be due to the presence of bit interleaving followed by differential encoding. The spectral efficiency of this system is 2 b/s/Hz. The inner noncoherent differential detector at the receiver side uses the RS-NCSOb algorithm, by reducing the number of states to $\zeta' = 16$. The order of Markovianity N is set equal to 3 and 5. For comparison, we also show the performance of the equivalent coherent system (i.e., considering differential encoding after the PCCC). In all cases, the iterations are carried out in the outer coherent turbo decoder, and the numbers of considered iterations are 1, 3 and 5. It is evident that there is a slight improvement in the performance of the noncoherent system by increasing N from 4 to 6, and the loss, with respect to the noncoherent case, is about 1 dB at BER values below 10^{-4}.

A similar scheme derived from one of the structures proposed in [72] is shown in Figure 4.47. This scheme is basically composed of two parallel concatenated Ungerboeck codes, and puncturing on information bits is considered before map-

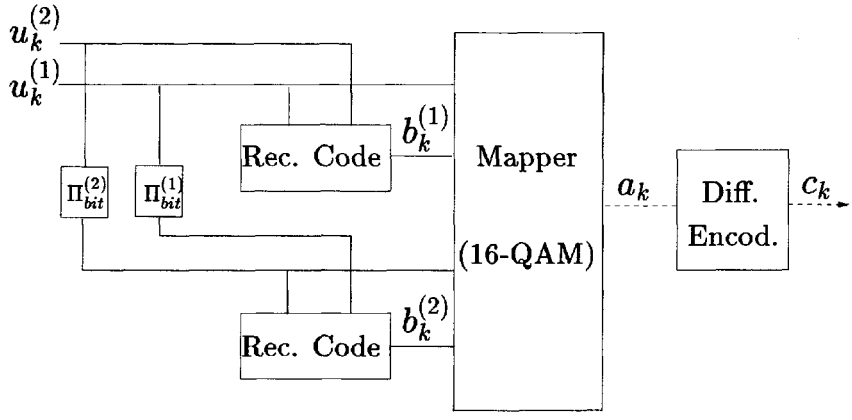


Figure 4.47: Benedetto-type PCCC followed by differential encoding on the modulated symbols. Reproduced from [163] by permission of GET/Hermes Science.

ping. In the figure, a sequence of couples of information bits $(u_k^{(1)}, u_k^{(2)})$ enter the encoder. Both encoders receive this sequence and generate two sequences of coded bits $(b_k^{(1)}, b_k^{(2)})$, but the systematic bits are punctured symmetrically in the two codes, as shown in Figure 4.47. We simply consider differential encoding after mapping. Strictly speaking, symbols $\{a_k\}$ are not independent as assumed in the derivation of a noncoherent decoding algorithm. However, one can observe, through simulations, that breaking this dependence by means of an interleaver (both bit-wise before mapping or symbol-wise after mapping) does not yield substantial performance improvement. This behavior may be related to the implicit puncturing considered in the outer PCCC, which, in a certain sense, decorrelates the bits carried by a modulated symbol.

In Figure 4.48, the performance of noncoherent decoding of the code proposed in Figure 4.47 is shown. The component 16-state recursive Ungerboeck codes of the PCCC have generators $h_0 = (23)_8$, $h_1 = (16)_8$, and $h_2 = (27)_8$ [72], and there are two different 32×32 random bit-interleavers. A 16-QAM modulation format is considered. The system has an efficiency of 2 b/s/Hz. As for the previous scheme, we also consider the inner noncoherent detector with the number of states reduced to $\zeta' = 16$ and order of Markovianity N equal to 3 and 5. For comparison, the performance of the equivalent coherent system is also shown. The numbers of iterations are 1, 3 and 6 in all cases. The performance loss of the noncoherent system with $N = 5$ with respect to the coherent system is about 1 dB.

We remark that the schemes considered in this subsection have a very low complexity, since the inner noncoherent detector is relative to simple differential encod-

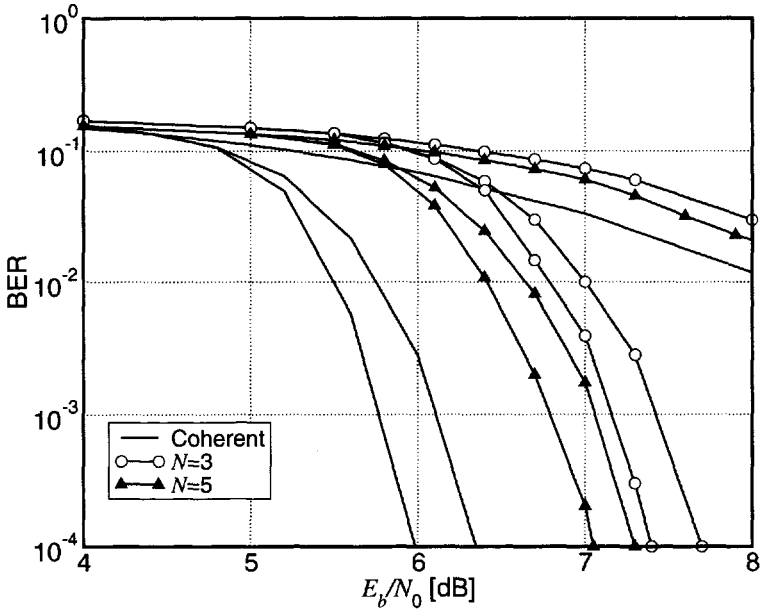


Figure 4.48: Performance of the system shown in Figure 4.47. The considered numbers of iterations are 1, 3 and 6 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

ing. In particular, the proposed schemes show a much lower complexity with respect to the schemes illustrated in the following section, where joint detection and decoding of trellis codes is considered.

Joint Detection and Decoding

In this case, we consider coding structures which do not employ differential encoding. The proposed schemes perform noticeably well in the case of ideal coherent decoding, i.e., assuming perfectly known phase at the receiver side.

As shown in Section 4.9.1, SCCCs have remarkable performance (even better than that of PCCCs) with very simple component codes [72]. However, this performance is obtained at the expense of the spectral efficiency of the code. For example, with rate-1/2 inner and outer convolutional codes, the overall code rate is 1/4. In order to increase the efficiency of the serial code, an inner Ungerboeck code can be considered, as shown in Figure 4.49. A similar structure was also considered in [178], where an outer Reed Solomon code and an inner Ungerboeck code were used. Vari-

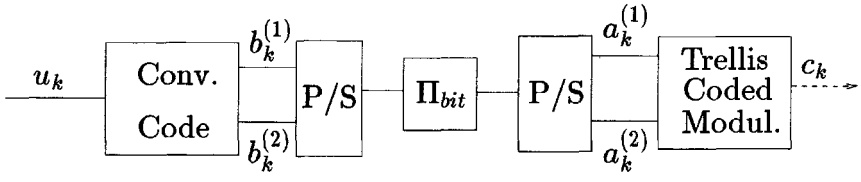


Figure 4.49: SCCC constituted by an outer convolutional code and an inner Ungerboeck code. Reproduced from [163] by permission of GET/Hermes Science.

ous combinations of serial codes are considered, where the outer convolutional code is a simple nonrecursive code [17, 72], whereas the inner Ungerboeck code may be a recursive systematic code [57] or a nonrecursive one [47]. It is worth noting that interleaving is bit-wise. In fact, the coded bits generated by the outer encoder are serialized and then interleaved. Figure 4.49 refers to the case of an outer rate-1/2 code and inner rate-2/3 code. After interleaving, the bits feed the inner encoder in groups of two. The receiver is based on an inner noncoherent decoder of the inner Ungerboeck code, which gives reliability values on the systematic bits of each modulated symbol (bits $a_k^{(1)}$ and $a_k^{(2)}$ in Figure 4.49) by using a noncoherent soft-output algorithm. Obviously, the overall serial code is noncoherently noncatastrophic depending on the characteristics of the inner Ungerboeck code. Hence, particular care has to be used in choosing this code as noncoherently noncatastrophic [47, 150].

In Figure 4.50, the performance of the serial scheme shown in Figure 4.49 is presented. The outer code is NRNSC, with generators $G_1 = (7)_8$ and $G_2 = (5)_8$ and rate 1/2. The inner Ungerboeck code is recursive and systematic, with generators $h_0 = (23)_8$, $h_1 = (16)_8$ and $h_2 = (27)_8$ [72]. The inner interleaver is a 32×32 pseudorandom bit-interleaver. The bits at the output of the inner code are mapped to an 8-PSK symbol, considering reordered mapping [72]. The spectral efficiency of this system is 1 b/s/Hz. The inner noncoherent decoder at the receiver side makes use of the RS NCSOb algorithm. Various complexity reduction levels, denoted by the couple (N, Q), are considered. The order of Markovianity N ranges from 3 to 15, while Q is kept fixed to 2 ($\zeta' = 64$ states). For comparison, the performance of the equivalent coherent system is also shown, i.e., assuming perfect knowledge of the channel phase at the receiver side. In all cases, the number of considered iterations is 10. As one can see, for increasing values of the order of Markovianity N, the performance of the noncoherent scheme approaches that of the coherent scheme. For $N = 15$ the performance loss at a BER of 10^{-5} is around 1 dB.

In Figure 4.51, a coding structure as in Figure 4.49 is considered, with the same inner Ungerboeck code of Figure 4.50 but considering an outer NRNSC code, with

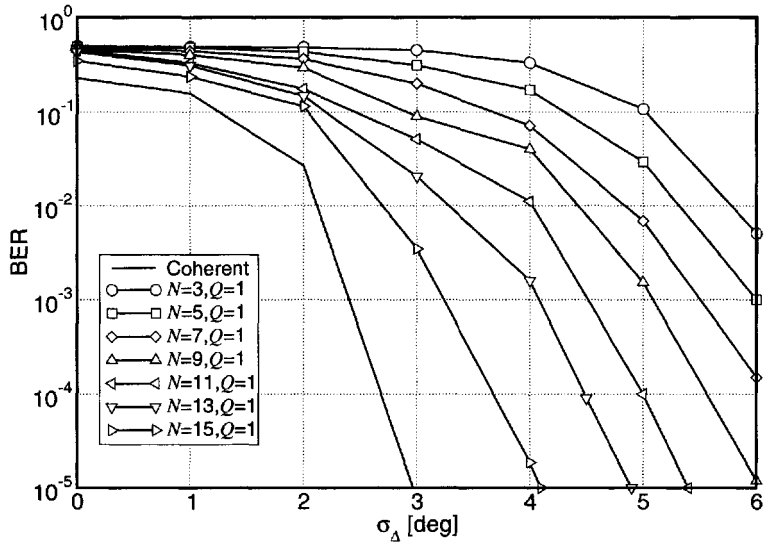


Figure 4.50: Performance of the system shown in Figure 4.49. The outer code has 8 states and the number of iterations is 10 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

generators $G_1 = (15)_8$ and $G_2 = (13)_8$ and rate 1/2. Hence, we replace an outer 8-state code with a 16-state code. As in the previous case, the noncoherent inner decoder is identified by the couple (N, Q). The order of Markovianity N ranges from 3 to 15, and $Q = 1$. The number of considered iterations for both the coherent and noncoherent systems is 10. For $N = 16$ the performance loss of the noncoherent scheme with respect to that of the coherent scheme is only 0.5 dB at a BER of 10^{-4}.

In Figure 4.52, the performance of a serial structure as in Figure 4.49 is evaluated, considering an outer rate-2/3 nonrecursive code with 16 states and generators $G_1 = (17)_8$, $G_2 = (06)_8$ and $G_3 = (15)_8$ [17] and an inner rate-3/4 nonrecursive code with 8 states and generators $G_1 = (040)_8$, $G_2 = (402)_8$, $G_3 = (240)_8$ and $G_4 = (100)_8$ [47]. The inner random interleaver is bit-wise, with length 1536. The spectral efficiency is 2 b/s/Hz and we consider a 16-QAM modulation format at the output of the inner encoder. The inner noncoherent decoder at the receiver side uses the Fwd-only RS NCSOb algorithm. Various complexity reduction levels, denoted by the couple (N, Q), are considered. The numbers of iterations are 1, 5 and 10 in all cases, and, for comparison, the performance of the equivalent coherent system is also shown. The performance of this scheme is comparable with the performance of

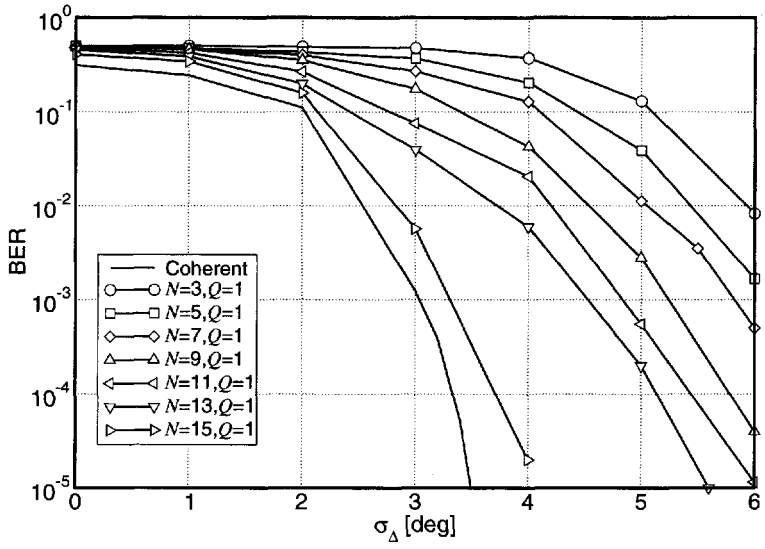


Figure 4.51: Performance of the system shown in Figure 4.49. The outer code has 16 states and the number of iterations is 10 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

the schemes of Figure 4.46 and Figure 4.48. A further performance improvement is expected considering an inner recursive code [72].

Besides serially concatenated coding structures, it is interesting to explore the possibility of deriving parallel concatenated coding structures suitable for combined noncoherent detection and decoding. The scheme proposed in [179], employing 8-PSK as modulation format at the output of each encoder, cannot be used when considering a noncoherent detection strategy. In fact, because of puncturing, the proposed noncoherent soft-output algorithms fail. A simple explanation of this failure can be given. It is easy to show that in this case the exponential metric $\gamma_k(T_k)$ for the NCSOa algorithm, the metrics $\psi_k(T_{k-1}, T_k)$ and $\phi_k(T_{k-1}, T_k)$ for the NCSOb algorithm and the metric $\lambda_k(T_k)$ for SO-NSD reduce to 1 every other epoch. Hence, for every other transition in the decoder trellis the forward and backward (only forward for SO-NSD) recursions cannot be correctly extended. This problem obviously affects the reduced-state version of these algorithms. On the other hand, the scheme proposed in [72] can be directly employed for transmissions over noncoherent channels, provided that the overall code is noncoherently noncatastrophic. With respect to the scheme in [72], the only proposed modification consists of considering a single bit-

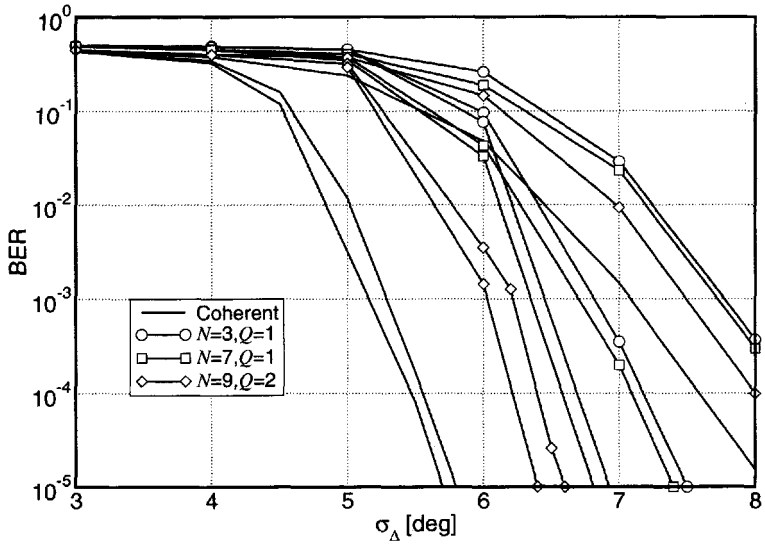


Figure 4.52: Performance of the system shown in Figure 4.49. The modulation format is 16-QAM and the number of iterations is 1, 5 and 10 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

interleaver between the two Ungerboeck codes (as shown in Figure 4.53), instead of considering a different bit interleaver for each bit stream. The two input bit streams are serialized in a single bit stream before being interleaved. The interleaved bit stream is then parallelized and undergoes trellis encoding. It can be verified that using a single interleaver instead of separate interleavers for each bit stream improves the performance, at least at high SNR [180]. This is intuitively related to the fact that low reliability values associated with the couple of bits embedded in the same symbol may be better spread over the whole bit sequence. Hence, the receiver has a structure similar to that of a turbo decoder, where each component decoder uses the reduced-state noncoherent soft-output decoding algorithm previously introduced. This scheme may be considered as a direct extension to spectrally efficient modulation schemes of the noncoherent schemes proposed in [162] for binary modulations.

In Figure 4.53, we consider, for simplicity, the case of a turbo trellis encoder where each of the component Ungerboeck encoders receives a sequence of couples of information bits $(a_k^{(1)}, a_k^{(2)})$ and generates a parity bit ($c_k^{(0)}$ in the upper encoder and $d_k^{(0)}$ in the lower encoder). Puncturing may be considered for one of the two information bits (symmetrically in the two encoders): in the upper encoder the systematic

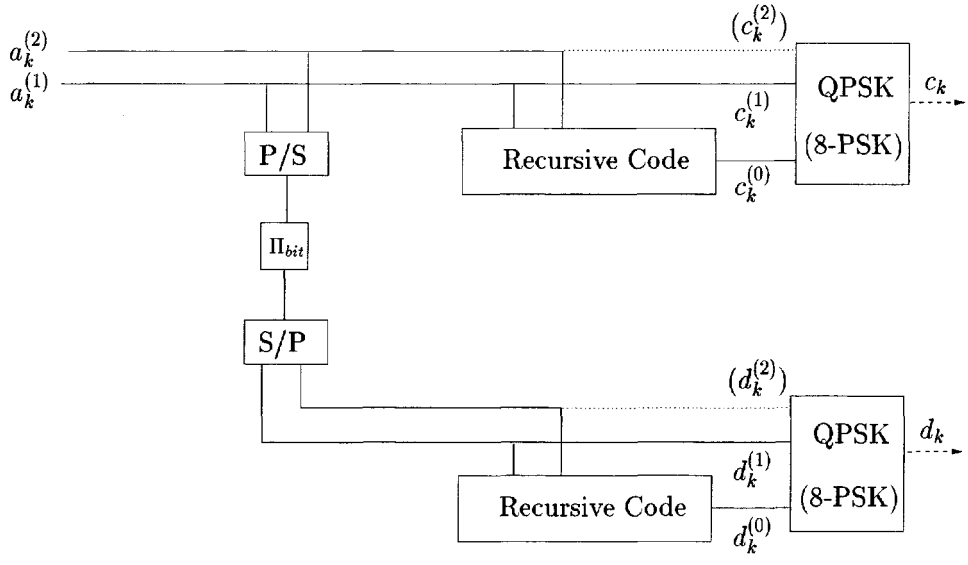


Figure 4.53: Turbo trellis-coded scheme by Benedetto *et al.* with 8-PSK modulation. Puncturing may be embedded in the component Ungerboeck codes to take into account QPSK modulation. Reproduced from [163] by permission of GET/Hermes Science.

bit $c_k^{(1)} = a_k^{(1)}$ is transmitted, whereas in the lower encoder the bit $d_k^{(1)} = a_{i_k}^{(2)}$ is transmitted.[36] As shown in Figure 4.53, after interleaving the two original bit streams have to be separated in order to consider proper puncturing on $a_{i_k}^{(2)}$. This is possible if the single interleaver is odd odd, i.e., if it maps the bits stored in odd positions (bits $\{a_k^{(1)}\}$) to odd positions, so that they can be recovered after interleaving. In this case, the single odd odd interleaver is equivalent to two separate interleavers. A QPSK symbol is generated at the output of each component encoder. The spectral efficiency in this case is 1 b/s/Hz.

Although the above scheme with QPSK has remarkable performance with coherent decoding, it can be observed that the performance noticeably degrades when considering noncoherent decoding, due to the catastrophic nature of the code. This motivates the following modification. The spectral efficiency remains unchanged by eliminating puncturing, hence transmitting an 8-PSK symbol at the output of each component encoder. In this case, both systematic bits at the input of each encoder are mapped to the corresponding generated complex symbol (in Figure 4.53 we indi-

[36]The time instant of the second encoded bit is denoted by i_k because of the presence of interleaving.

cate by dotted lines the supplementary connections which must be considered). This means adding redundancy, at the cost of decreasing the robustness of the modulation constellation. In the coherent case, the performance worsens, whereas in the noncoherent case it improves. Combining modulation and coding when dealing with a noncoherent channel cannot be carried out as in the case of an AWGN channel, because the noncoherent catastrophic nature must be taken into account. Moreover, based on an exhaustive search using different constellation mappings, it can be shown that the receiver performance in the noncoherent case does not seem to be appreciably influenced by the particular mapping rule (Gray, reordered, etc. [72]).

The last considered parallel scheme deserves some remarks about its noncoherent catastrophic nature. By reducing the modulation constellation from 8-PSK to QPSK, the code properties, in terms of modulated output symbols, may change. Hence, a code may not be simultaneously noncoherently noncatastrophic with and without puncturing. An open problem is the design of a good code for such a transmitter structure when considering puncturing and QPSK. An important aspect to be considered is the rotational invariance of the component codes, taking into account puncturing and mapping. The methods proposed in [119, 181, 182] may be considered. A relevant analysis concerning the rotational invariance of T-TCM schemes is addressed in [183], while in [184] a similar analysis is considered in the case of serially concatenated TCM (SCTCM).

In Figure 4.54, we show the performance of noncoherent iterative decoding of the code proposed in Figure 4.53. The component 16-state recursive Ungerboeck codes of the considered scheme have generators $h_0 = (23)_8$, $h_1 = (16)_8$ and $h_2 = (27)_8$ [72] and there is a single 64×64 pseudorandom bit-interleaver [33]. At the output of each component encoder, both systematic bits are retained and mapped, together with the parity bit, to an 8-PSK symbol. Reordered mapping is considered. The system efficiency is 1 b/s/Hz. The two component noncoherent decoders have a number of states reduced to $\zeta' = 64$ and order of Markovianity N equal to 3 and 5. For comparison, we also show the performance of the equivalent coherent system. The numbers of iterations are 1, 3 and 6 in all cases. Considering $N = 5$ and 6 decoding iterations, the performance loss of the noncoherent scheme with respect to the coherent scheme is about 1.5 dB.

Comments on Noncoherent Iterative Decoding of Spectrally Efficient Linear Coded Modulations

In the previous subsections, the communication schemes have been presented according to the detection strategy, i.e., separate or combined detection and decoding. The same schemes could have also been organized in terms of their spectral efficiency. In

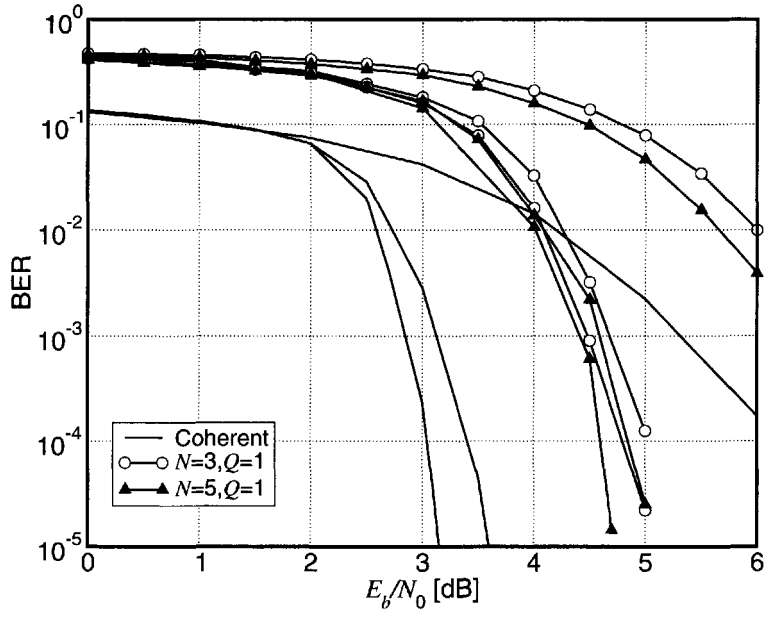


Figure 4.54: Performance of the system proposed in Figure 4.53. The modulation format is 8-PSK and the number of iterations is 1, 3 and 6 in all cases. Reproduced from [163] by permission of GET/Hermes Science.

fact, schemes with spectral efficiency of 1 b/s/Hz or 2 b/s/Hz have been considered. The performance results may be summarized as follows.

- Schemes with spectral efficiency of 2 b/s/Hz: the coherent receivers show a BER of 10^{-4} at an SNR between 5 and 6 dB, while the corresponding noncoherent schemes show a performance degradation of about 1 dB.

- Schemes with spectral efficiency of 1 b/s/Hz: the coherent receivers show a BER of 10^{-4} at an SNR between 3 and 4 dB, while the corresponding noncoherent schemes exhibit a performance loss of less than 1 dB.

Taking into consideration the complexity of the receiver, it turns out that simple schemes with separate detection and decoding offer a very good compromise.

4.9.3 Pilot Symbol-Assisted Iterative Detection for Phase-Uncertain Channels

In this subsection, we are concerned with noncoherent iterative detection for channels characterized by strong phase variations. Due to the underlying CMP in all the considered soft-output noncoherent algorithms, even if they are derived assuming a time-invariant channel phase modeled as a random variable uniformly distributed in $[0, 2\pi)$, they show a very good performance when considering a time-varying channel phase, as partially remarked in Section 4.9.2.

In the following, the proposed schemes will be divided into two classes as well: separate detection and decoding schemes and combined detection and decoding schemes. We also introduce a suitable classification for the soft-output algorithms—which can also be applied to other cases previously analyzed. The considered noncoherent soft-output algorithms can be generally defined as *feed-forward* or *open-loop* (OL) algorithms, since there is an implicit nonrecursive phase estimate based on a window of consecutive channel observations—in particular, these algorithms are direct extensions of the NCSOa algorithm introduced in Section 4.7. In the following, we will compare the performance of the proposed noncoherent algorithms with the performance obtained considering *closed-loop* (CL) soft-output algorithms [185, 186], where the phase is recursively estimated. Before giving more details on OL and CL soft-output algorithms, we note that in the remainder of this subsection, the following model for the discrete-time observable is considered:

$$r_k = c_k e^{j\theta_k} + n_k \tag{4.230}$$

where $\{\theta_k\}$ represents the discrete-time phase process,[37] and $\{n_k\}$ are iid complex noise samples with independent real and imaginary components of equal variance $\sigma^2 = N_0$.

Closed-Loop Adaptive FB Algorithms

Although the theoretical framework developed in [185,186] for adaptive soft-decision algorithms is quite general, we describe here only one instance of these algorithms that is relevant to this specific application and to the proposed comparison. The basic idea of the CL approach is to view the set of all possible sequences $\boldsymbol{a} = \boldsymbol{a}_1^K$ as paths of a depth-K tree. Searching over all these sequences is required to obtain the exact solution of the detection problem. The CL algorithm approximates

[37] It is assumed that the continuous-time phase process is slow compared to the symbol duration, and thus a symbol-spaced matched filter provides sufficient statistics for detection.

this solution by reducing the "optimal" tree search over all possible sequences $\boldsymbol{a}$ to a limited number of sequences, according to a tree-pruning strategy. In particular, by applying the CMP, the sequence tree reduces to a trellis where the generic state is, as usual, $S_k \triangleq (\mu_{k-N}, \boldsymbol{a}_{k-N}^{k-1})$, in which the state μ_{k-N} is relative to the encoder/modulator FSM. The parameter $N \geq 0$ is, in this case, a design choice and determines the expanded trellis size.[38] A transition in the expanded trellis is indicated by $T_k = (S_k, a_k) = (\mu_{k-N}, \boldsymbol{a}_{k-N}^{k})$. In this case, the APP can be approximated as follows [186]:

$$P\{a_k|\boldsymbol{r}\} \simeq \sum_{T_k:a_k} p[\boldsymbol{r}_1^{k-1}|S_k, \hat{\theta}_k^{\mathrm{f}}(S_k)]P\{S_k\}p[r_k|T_k, \hat{\theta}_k^{\mathrm{f}}(S_k), \hat{\theta}_{k+1}^{\mathrm{b}}(S_{k+1})]P\{a_k\} \cdot p[\boldsymbol{r}_{k+1}^{K}|S_{k+1}, \hat{\theta}_{k+1}^{\mathrm{b}}(S_{k+1})] \tag{4.231}$$

where $\hat{\theta}_k^i(S_k)$, $i \in \{\mathrm{f}, \mathrm{b}\}$ are forward and backward phase estimates, respectively, which are obtained as described in the following. For the sake of simplicity, we will simply use the notation $\hat{\theta}_k^i$ to indicate an estimate of θ_k—the dependence on a state S_k should be clear from the context. By defining

$$\alpha_k(S_k) \triangleq p(\boldsymbol{r}_1^{k-1}|S_k, \hat{\theta}_k^{\mathrm{f}})P\{S_k\} \tag{4.232}$$

$$\gamma_k^{\mathrm{CL},c}(T_k, \hat{\theta}_k^{\mathrm{f}}, \hat{\theta}_{k+1}^{\mathrm{b}}, r_k) \triangleq p(r_k|T_k, \hat{\theta}_k^{\mathrm{f}}, \hat{\theta}_{k+1}^{\mathrm{b}})P\{a_k\} \tag{4.233}$$

$$\beta_{k+1}(S_{k+1}) \triangleq p(\boldsymbol{r}_{k+1}^{K}|S_{k+1}, \hat{\theta}_{k+1}^{\mathrm{b}}) \tag{4.234}$$

the soft output on the right-hand side of (4.231) generated by an adaptive CL algorithm, which we denote by $S^{\mathrm{CL}}[a_k]$, can be written as

$$S^{\mathrm{CL}}[a_k] = \sum_{T_k:a_k} \alpha_k(S_k)\gamma_k^{\mathrm{CL},c}(T_k, \hat{\theta}_k^{\mathrm{f}}, \hat{\theta}_{k+1}^{\mathrm{b}}, r_k)\beta_{k+1}(S_{k+1}). \tag{4.235}$$

It is important to remark that we explicitly consider the dependence on the observation and the phase estimate only in the expression of the exponential metric $\gamma_k^{\mathrm{CL},c}$, assuming an implicit dependence on it in the expressions of the quantities α_k and β_{k+1}. In [186], it is shown that the metric $\gamma_k^{\mathrm{CL},c}(T_k)$ characterizing the completion operation (4.235) can be written as

$$\gamma_k^{\mathrm{CL},c}(T_k, \hat{\theta}_k^{\mathrm{f}}, \hat{\theta}_{k+1}^{\mathrm{b}}, r_k) = \gamma_k^{\mathrm{CL}}(T_k, \hat{\theta}, r_k)\, b(\hat{\theta}_k^{\mathrm{f}}, \hat{\theta}_{k+1}^{\mathrm{b}}), \tag{4.236}$$

[38] $N = 0$ corresponds to the case $S_k = \mu_k$, i.e., there is no trellis expansion.

where $\hat{\theta}$ is either $\hat{\theta}_k^{\mathrm{f}}$ or $\hat{\theta}_{k+1}^{\mathrm{b}}$, and (see [186] for details)

$$\gamma_k^{\mathrm{CL}}(T_k, \hat{\theta}, z_k) = P\{a_k\} \exp\left(\frac{2\Re(r_k e^{-j\hat{\theta}} c_k^*) - |c_k|^2}{2\sigma^2}\right) \tag{4.237}$$

$$b(\hat{\theta}_{k-1}^{\mathrm{f}}, \hat{\theta}_k^{\mathrm{b}}) = \exp\left(-\frac{1-\lambda}{\lambda(2-\lambda)} \left| e^{j\hat{\theta}_{k-1}^{\mathrm{f}}} - e^{\hat{\theta}_k^{\mathrm{b}}} \right|^2\right) \tag{4.238}$$

where the meaning of the parameter λ, which is related to the recursive phase estimation strategy, is clarified below. The quantities $\alpha_k(S_k)$ and $\beta_{k+1}(S_{k+1})$ in (4.235) can be computed by means of forward and backward recursions, during which forward and backward phase estimates are computed, respectively.

Let us consider the forward recursion. In general, it can be written as follows:

$$\alpha_{k+1}(S_{k+1}) = \sum_{T_k : S_{k+1}} \alpha_k(S_k) \gamma_k^{\mathrm{CL}}(t_k, \hat{\theta}_k^{\mathrm{f}}, z_k). \tag{4.239}$$

The phase estimate $\hat{\theta}_k^{\mathrm{f}}$ is updated in a per-survivor processing (PSP) fashion, using a first order phase-locked loop (PLL). The PLL update equation is given by

$$\hat{\theta}_{k+1}^{\mathrm{f}} = \hat{\theta}_k^{\mathrm{f}}(S_k^{\max}) + \lambda \Im\{z_k c_k^* e^{-j\hat{\theta}_k^{\mathrm{f}}(S_k^{\max})}\}, \tag{4.240}$$

where the transition $T_k^{\max}$ (and the corresponding beginning state $S_k^{\max}$) is the one determined by an add-compare-select operation similar to the one in (4.239), derived by exchanging the summation operation $\sum_{T_k:S_{k+1}}$ by a maximization operation $\max_{T_k:S_{k+1}}$. The backward recursion, during which a backward phase estimate $\hat{\theta}_k^{\mathrm{b}}$ is updated, is similar to the forward recursion.

At this point, we would like to emphasize that the CL algorithm only keeps and updates a limited number of phase estimates. Each of these estimates corresponds to each of the survivors, i.e., each of the tree paths that are explored in the limited tree search procedure. However, due to the *recursive* nature of the parameter update equations, the entire memory of the channel is retained in each of these estimates (for this reason, this class of adaptive algorithms is referred to as *closed-loop* adaptive algorithms). This is a unique feature of the CL algorithm which allows almost coherent performance for low phase dynamics. This feature is, however, one of its drawbacks for fast phase dynamics, as will be evident from the results shown in the following. In all considered transmission schemes where a CL detection strategy is used, pilot symbols are inserted in the output modulated symbols, as indicated in Figure 4.55, Figure 4.57, Figure 4.58.

Open-Loop Adaptive FB Algorithms

We recall, for ease of comparison, the expression of the exponential metric γ_k of the NCSOa algorithm:[39]

$$\gamma_k^{\mathrm{OL}}(T_k, \boldsymbol{r}_{k-N+1}^k) \sim P\{a_k\} \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|\right)}. \tag{4.241}$$

Comparing the expressions in (4.237) and (4.241), it can be observed that there is no explicit phase estimate involved in the metric γ_k^{OL}. However, by proper manipulations and slight approximations it is possible to interpret this metric as *implicitly* performing phase estimation, based on a window of consecutive observations. More precisely, observing that $\mathrm{I}_0(x) \simeq e^x$ for sufficiently large x, and that, for a generic complex number c, $|c| = ce^{-j\angle c} = \Re(ce^{-j\angle c})$, (4.129) can be approximated as follows:

$$\begin{aligned}
\gamma_k^{\mathrm{OL}}(T_k, \boldsymbol{r}_{k-N+1}^k) &\simeq P\{a_k\} \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \exp\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right| - \left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|}{\sigma^2}\right) \\
&= P\{a_k\} \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \\
&\quad \cdot \exp\left(\frac{\sum_{i=0}^{N} \Re\{r_{k-i}e^{-j\hat{\theta}_k^{(N)}} c_{k-i}^*\} - \sum_{i=1}^{N-1} \Re\{r_{k-i}e^{-j\hat{\theta}_k^{(N)}} c_{k-i}^*\}}{\sigma^2}\right)
\end{aligned} \tag{4.242}$$

where the implicit phase estimates are defined as

$$\hat{\theta}_k^{(N+1)} \triangleq \arg\left(\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right) \qquad \hat{\theta}_k^{(N)} \triangleq \arg\left(\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right). \tag{4.243}$$

[39] The current notation for the exponential metric is slightly different with respect to that used in the description of the NCSOa algorithm in Section 4.7. This is expedient to make the comparison with CL FB algorithms clearer.

If $\hat{\theta}_k^{(N+1)} \simeq \hat{\theta}_k^{(N)}$, then (4.242) can be further approximated as

$$\gamma_k^{\mathrm{OL}}(T_k, \boldsymbol{r}_{k-N}^k) \simeq P\{a_k\} \exp\left(\frac{2\Re\{r_k e^{-j\hat{\theta}_k^{(N)}} c_k^*\} - |c_k|^2}{2\sigma^2}\right). \quad (4.244)$$

For a time-invariant channel phase rotation, the last approximation is reasonable for values of N large enough [163, 187], while for a time-varying channel phase rotation an optimal value of N exists as will be evident from the numerical results. Moreover, the final expression in (4.244) is formally identical to the corresponding metric for the CL approach in (4.237), and clearly shows the connection between $\gamma_k^{\mathrm{OL}}(T_k, \boldsymbol{r}_{k-N}^k)$ and the implicit OL phase estimate $\hat{\theta}_k^{(N+1)}$ in (4.243). Since the OL phase estimate depends on the current observation window, two consecutive estimates $\hat{\theta}_{k-1}^{(N)}$ and $\hat{\theta}_k^{(N)}$ are not explicitly related by a recursive formula. In this sense, it is possible to interpret the phase estimation strategy embedded in (4.244) as *open-loop* phase estimation, as opposed to the CL estimation in CL algorithms. In all the considered transmission schemes with OL detection strategy, pilot symbols are inserted in the input information stream, as indicated in Figure 4.55, Figure 4.57 and Figure 4.58.

Separate Detection and Decoding

In Sections 4.9.1 and 4.9.2, the schemes with separate detection and decoding were based on the use of differential encoding before transmission over the channel. Hence, the terminology "separate" is not formally correct since, in the detection process, besides taking into account the memory of the channel, the memory introduced by the differential encoder was also considered. A differential code is the simplest noncoherently noncatastrophic code [150], and its use allows one to overcome the unknown phase rotation introduced by the channel. However, the insertion of *pilot symbols* is an alternative simple way to combat the unknown channel phase rotation, making the overall code noncoherently noncatastrophic. In Figure 4.55, a possible scheme is shown. The output bits of a binary PCCC are grouped and mapped, through a memoryless mapper, to modulated symbols $\{c_k\}$. The pilot symbols can be inserted either before mapping (for example N_p binary pilot symbols could be inserted every N_d information symbols) or after mapping (as complex pilot symbols). In either case, the noncoherent adaptive soft demodulator (A-SODEM), which generates soft-output reliability values based on the received observations, must be given knowledge of the pilot symbols insertion strategy. The considered noncoherent soft-output algorithms have simply to take advantage of this knowledge. For example, if the pilot symbols are inserted among the information bits as zero bits, the soft input $\mathrm{SI}[a_k]$ (in the nat-

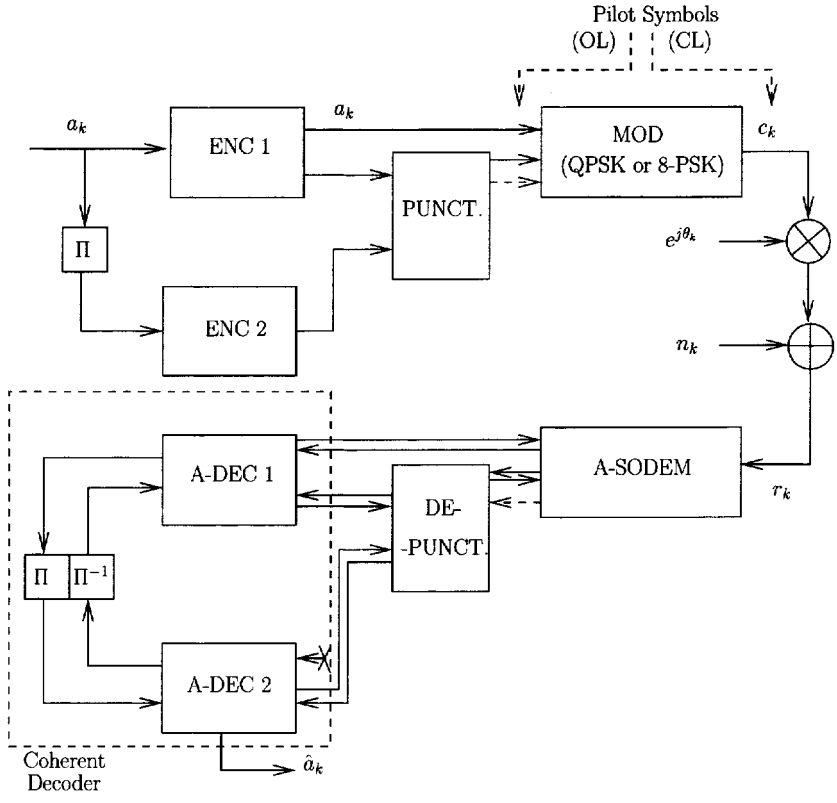


Figure 4.55: Parallel concatenated coding scheme with separate detection and decoding: transmitter, channel and iterative decoder constituted by the concatenation of an A-SODEM and a coherent turbo decoder. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

ural or probabilistic domain) at an epoch k corresponding to the insertion of a pilot bit is set as

$$\mathrm{SI}[a_k] = \begin{cases} 1 & \text{if } a_k = 1 \\ 0 & \text{if } a_k = 0. \end{cases} \tag{4.245}$$

If, instead, we consider insertion of pilot symbols among the modulated output symbols $\{c_k\}$, a simple way to modify the basic exponential metric γ_k is that of assuming that pilot symbols are sent as supplementary complex symbols. Considering a vector notation, it is very simple to take into account the inserted pilot symbols. Basically, a generic correlation sum $\sum_{i=0}^{N} r_{k-i}c_{k-i}^*$ involved in this metric is extended by adding an extra term relative to the inserted pilot symbol.

The performance of the proposed receivers is assessed by means of computer

simulations mainly in terms of BER and frame-error rate (FER) versus E_b/N_0, E_b being the received energy per information bit. The SNR loss due to the insertion of pilot symbols is accounted for in all the results presented herein. In all cases, $N_p = 1$ pilot symbol is inserted every N_d symbols. The performance of the considered systems under dynamic channel conditions is investigated. The time-varying phase process $\{\theta_k\}$ used in the simulations is a random walk with independent Gaussian increments, with variance over a signaling interval equal to σ_Δ^2. In the following, we will also assume that any adaptive FB algorithm is in the min-sum form, while any coherent FB algorithm is the standard FB algorithm [54]. In all presented results, we assume that the initial forward and backward phase estimates (only forward in the OL case and both in the CL case) are ideal. This assumption is justified, since insertions of an initial and a final training sequence result in negligible bandwidth efficiency and energy loss for the considered codeword lengths. In all systems examined in this work, a number of simulations were run to roughly optimize the different system parameters (e.g., N, Q, N_d, λ, etc.). However, for conciseness, only a limited number of these results are presented in order to demonstrate the main conclusions.

In Figure 4.56, the performance of the PCCC-based separate detection and decoding scheme considered in Figure 4.55 is shown. The purpose of these simulations is not to compare *separate* and *combined* strategies (see [186] for such a comparison), but rather to compare CL and OL strategies in the specific case of *separate* detection/decoding. In this case, the PCCC is almost identical to the combined scheme [33] described earlier. The only difference is that at every epoch the two output bits are mapped to a QPSK symbol with Gray mapping, resulting in an overall spectral efficiency of 1 bit/s/Hz. At the receiver side, the inner A-SODEM uses either the OL FB algorithm or the CL FB algorithm. The iterative detection and decoding process can be characterized by I_e external iterations between the A-SODEM and the inner turbo decoder, and by I_i internal iterations in the turbo decoder. The performance of the proposed adaptive algorithms is compared with the performance of the corresponding coherent scheme. In all cases, the number of external iterations I_e is set to 5. In the CL case, for $\sigma_\Delta = 5$ degrees, the performance for $N = 0$ (single-state decoder) is shown—it is the same either for $I_i = 2$ or $I_i = 3$ internal decoding iterations. The best performance is obtained in this case by considering $N_d = 4$. When increasing the phase jitter to $\sigma_\Delta = 10$ degrees, the performance for $N = 1$ and $I_i = 2$ internal decoding iterations is shown. As one can see, the loss with respect to the coherent limit is significant. Increasing the pilot insertion rate to $N_d = 2$, a performance improvement around 3 dB is observed at a BER of 10^{-4}. In the OL case the A-SODEM uses the proposed noncoherent algorithm with $(N, Q) = (7, 3)$: $I_i = 2$ and $I_i = 3$ internal decoding iterations are considered. For $\sigma_\Delta = 5$ degrees the best performance is obtained with $N_d = 16$, while for $\sigma_\Delta = 10$

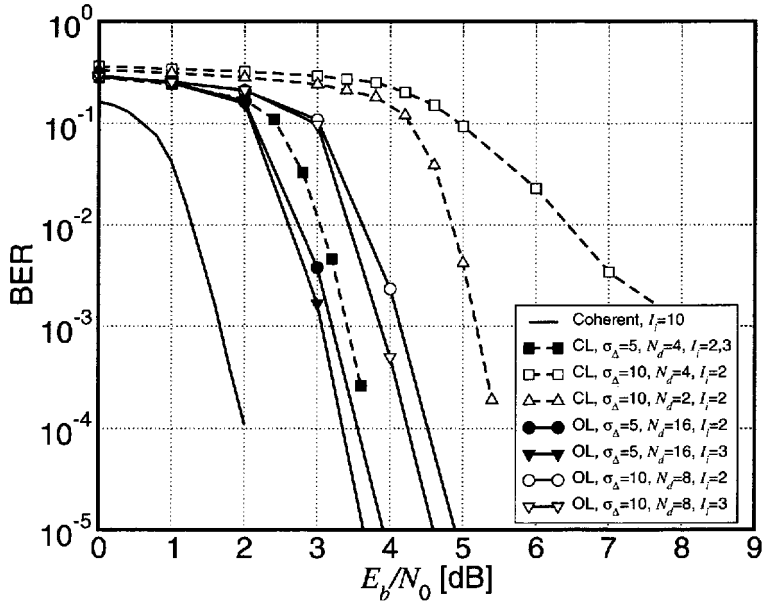


Figure 4.56: BER of the separate scheme with rate-1/2 PCCC and QPSK output modulation. In all cases, $I_\mathrm{e} = 5$ external iterations between the A-SODEM and the turbo decoder are considered. Various numbers I_i of inner decoding iterations are considered. In the CL case, the adaptive algorithm is characterized by $N = 0$ for $\sigma_\Delta = 5$ degrees and by $N = 1$ for $\sigma_\Delta = 10$ degrees. In the OL case, the detection algorithm is characterized by $(N, Q) = (7, 3)$. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

degrees with $N_d = 8$. Both for $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees, increasing I_i from 2 to 3 leads to a performance improvement of less than 0.3 dB. The complexity with $(I_\mathrm{e}, I_\mathrm{i}) = (5, 2)$ is roughly comparable to the performance of the perfect CSI scheme with $I_i = 10$ decoding iterations. As one can see, the OL approach is more robust to strong phase variations with a reduced insertion rate with respect to the CL case, i.e., with a reduced bandwidth expansion. However, this comes at the expense of an increased number of states in the A-SODEM ($Q = 3$ corresponds to $\zeta' = 64$ states).

Combined Detection and Decoding

We consider both an SCCC and a PCCC. The equivalent base-band discrete-time transmission system of an SCCC is shown in Figure 4.57. The bit sequence $\{b_k\}$

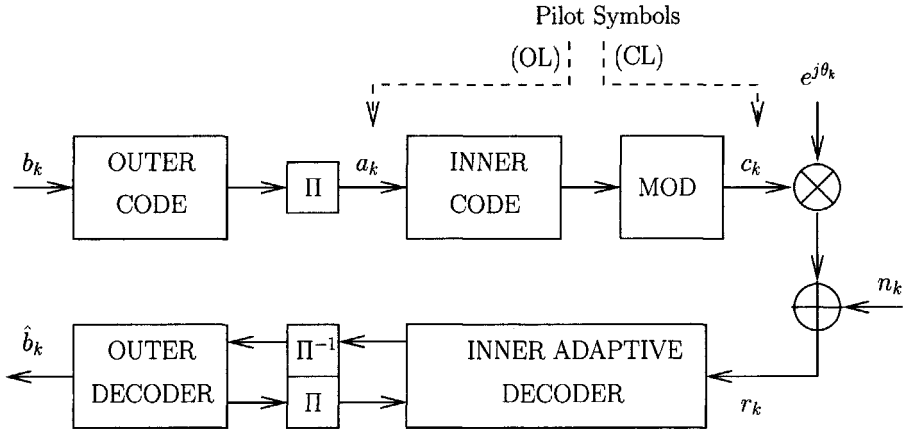


Figure 4.57: Serially concatenated coding scheme with combined detection and decoding: transmitter, channel and adaptive iterative decoder. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

is encoded using an outer code and interleaved using a symbol-wise or bit-wise interleaver. The resulting sequence of M-ary symbols $\{a_k\}$ is coded by an inner code, producing the coded sequence $\{c_k\}$. The resulting coded symbols are further mapped to the complex symbols $\{c_k\}$ and transmitted over an AWGN channel, which, in addition, introduces an unknown carrier phase offset. The statistics of the phase process need not be specified at this point. The complex equivalent signal can be written, after a suitable discretization process, as in (4.230). The receiver consists of an adaptive inner block, that jointly estimates the phase and produces soft information on symbols $\{a_k\}$, and a nonadaptive outer block that produces soft decisions on $\{a_k\}$, as well as hard decisions for $\{b_k\}$. The details of the inner adaptive decoder, which can be either a CL or an OL FB algorithm, are discussed in the next section.

In Figure 4.58, a transmission scheme employing a PCCC is shown. In this case, the PCCC is constituted by two component RSC codes. For simplicity, rate-1/2 RSC codes are considered. After possible puncturing, the sequences of information and coded bits are serialized, mapped to a BPSK constellation and transmitted over the channel. At the receiver side, the turbo decoder consists of two adaptive component decoders. It should be emphasized at this point that a *combined* detection scheme for PCCCs, such as the one described here, can become quite complicated when higher order constellations are used, as demonstrated in [186]. Therefore, only BPSK modulation is considered in this case, while higher modulations are considered in conjunction with *separate* detection and decoding, as described in the following subsection.

The CL and OL algorithms are first compared considering iterative decoding of

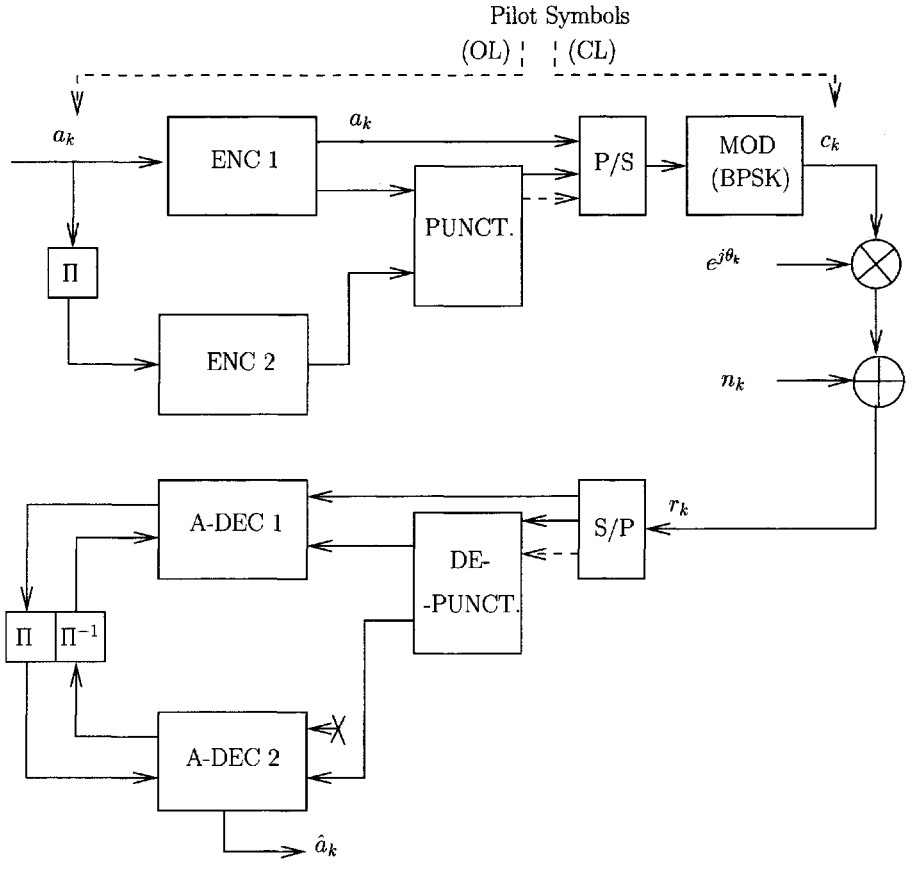


Figure 4.58: Parallel concatenated coding scheme with combined detection and decoding: transmitter, channel and iterative decoder constituted by two adaptive component decoders. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

two SCCCs using *combined* detection and decoding. The first SCCC consists of an outer 4-state, rate-1/2 convolutional code connected through a length-1024 pseudo-random interleaver to an inner 4-state, rate-2/3 convolutional code.[40] The respective generator matrices are given by

$$\boldsymbol{G}_{\mathrm{o}}(D) = \left[1 \quad \frac{1+D^2}{1+D+D^2}\right] \qquad \boldsymbol{G}_{\mathrm{i}}(D) = \begin{bmatrix} 1 & 0 & \frac{1+D^2}{1+D+D^2} \\ 0 & 1 & \frac{1+D}{1+D+D^2} \end{bmatrix}. \tag{4.246}$$

The output symbols are mapped to an 8-PSK constellation with natural mapping,

[40]The constituent codes in all SCCC and PCCC schemes examined herein are properly terminated using tail bits.

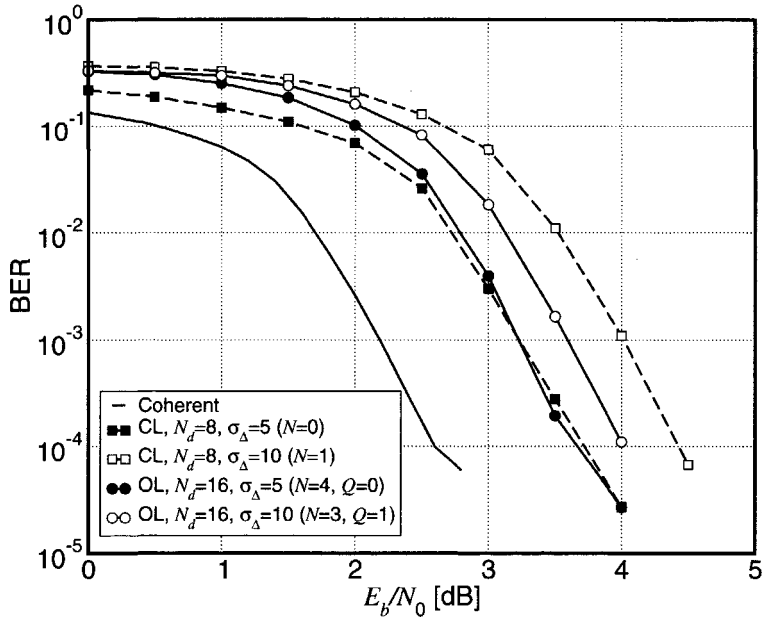


Figure 4.59: BER of an SCCC with OL and CL inner decoding algorithm, for phase jitter standard deviation $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees. The spectral efficiency is 1 bit/s/Hz. For comparison, the performance of the equivalent coherent scheme is shown. In all cases, 10 decoding iterations are considered. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

resulting in an overall code of spectral efficiency of 1 bit/s/Hz. The numerical results for this system are shown in Figure 4.59, in terms of BER. For comparison, the idealized performance of the equivalent coherent receiver, i.e., with perfect phase reference, is also shown. In all cases, 10 decoding iterations are considered. In the OL case, for increasing phase jitter the performance improves by reducing the order of Markovianity N ($N = 4$ for $\sigma_\Delta = 5$ degrees and $N = 3$ for $\sigma_\Delta = 10$ degrees) and the state reduction level ($Q = 0$ for $\sigma_\Delta = 5$ degrees and $Q = 1$ for $\sigma_\Delta = 10$ degrees). For the case of CL phase estimation, the considered number of decoder states is $\zeta' = 4$ and $\zeta' = 16$, for $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees, respectively, so that the overall complexity of the CL and OL receivers is roughly the same. Simulation results show that for $\sigma_\Delta = 5$ degrees, the performance of the CL-based and the OL-based receivers is almost identical at BER values of practical importance. When increasing the phase jitter to 10 degrees, the CL scheme shows a performance loss of 0.5 dB with respect to the OL scheme. A possible explanation for this degradation is that CL estimation may result in losses of lock which are difficult

to recover from. We remark that the CL scheme requires a double number of pilot symbols to obtain a performance similar to that of the OL scheme. However, with the considered values of N_d in the two cases (16 for the OL scheme and 8 for the CL scheme), in order to support the same information rate the CL scheme requires a bandwidth expansion of only 7% with respect to the OL scheme. Similar conclusions can be drawn by looking at the FER curves (see Problem 4.10).

The second considered SCCC consists of an outer 4-state, rate-2/3 nonrecursive convolutional code and an inner 4-state, rate-3/3 recursive convolutional code, connected through a length-1024 symbol interleaver [188]. The outer encoder is obtained by parallelizing two identical encoders with generator $\boldsymbol{G}_o(D)$ as in (4.246), and puncturing every other coded bit. The inner code is essentially the antirotational invariant version of Code 1 in [189]. As in the previously considered SCCC, the three output bits are mapped to an 8-PSK symbol with natural mapping. The spectral efficiency of the overall code is then 2 bits/s/Hz. The performance of the CL and OL schemes is shown in Figure 4.60 and compared with the performance of the equivalent coherent system. In all cases, 10 decoding iterations are considered. In the CL case the best performance is obtained with $N_d = 4$, while in the OL case, the best performance is obtained with $N_d = 8$. In the latter case, for increasing phase dynamics the best performance is obtained by considering a reduced observation window ($N = 4$ for $\sigma_\Delta = 5$ degrees and $N = 3$ for $\sigma_\Delta = 10$ degrees). Results on the FER (not shown here) revealed similar behavior.

The performance of the two considered detection strategies for this communication system has been further analyzed by evaluating the SNR necessary to obtain a prescribed BER of 10^{-3}, at 10 decoding iterations, as a function of the jitter standard deviation σ_Δ. The results are presented in Figure 4.61. Two curves for each of the two detection strategies are shown. In the CL case, one curve corresponds to $N_d = 4$, while the other curve is obtained by optimizing the insertion rate for each specific value of the phase noise jitter standard deviation σ_Δ. In the OL case, both curves correspond to the optimized value $N_d = 8$: one of the curves corresponds to the case with $(N, Q) = (5, 2)$, while the other curve refers to the case with $(N, Q) = (4, 1)$. As one can see, the CL detection strategy is better than the OL detection strategy for low phase jitter standard deviation, while it worsens for increasing standard deviation. On the other hand, the OL approach is more robust at high phase dynamics, and the SNR required to attain the desired BER is almost constant for $\sigma_\Delta \leq 5$ degrees. Moreover, the optimized value N_d in the OL case does not seem to depend on the phase jitter standard deviation σ_Δ. As shown in Figure 4.61, for increasing phase jitter the OL scheme with a short observation window ($N = 4$) outperforms the one with a larger observation window ($N = 5$). The same analysis was carried out with the other considered communication systems, as well as with

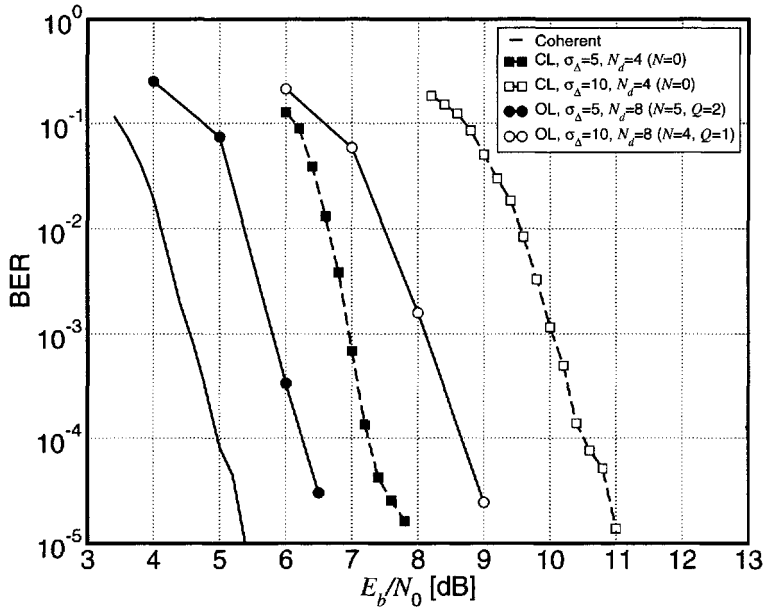


Figure 4.60: BER of an SCCC with OL and CL inner decoding algorithms, for phase jitter standard deviation $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees. The spectral efficiency is 2 bits/s/Hz. For comparison, the performance of the equivalent coherent scheme is shown. In all cases, 10 decoding iterations are considered. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

FER as the performance measure, and the same conclusion was reached: there is a threshold value for σ_Δ of the order of 1 to 2 degrees, such that CL is better than OL for σ_Δ values lower than the threshold, and OL is better than CL for σ_Δ values higher than the threshold.

In Figure 4.62, the performance of the PCCC scheme with *combined* detection and decoding is shown. The component encoders are as in [33], with an inner pseudo-random interleaver of length 1024. BPSK modulation is used, resulting in an overall rate of 1/2, and spectral efficiency of 0.5 bit/s/Hz. In all cases, 10 decoding iterations are considered. It can be seen that the performance of the CL receiver is 1.5 dB worse than that of the OL receiver (at a BER of 10^{-3}) for $\sigma_\Delta = 5$ degrees, while the CL-based receiver does not work at all for $\sigma_\Delta = 10$ degrees. To further investigate this difference in performance between the OL and CL schemes, the FER is shown in Figure 4.63. Note that the CL scheme is slightly better than the OL scheme (for $\sigma_\Delta = 5$ degrees) in terms of FER, while the situation is reversed in terms of BER. This fact shows that a catastrophic behavior is observed each time a frame error oc-

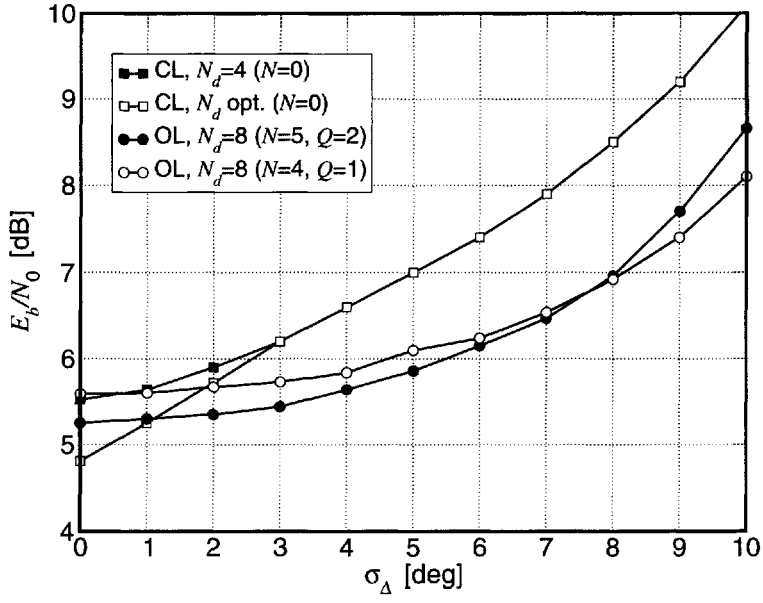


Figure 4.61: E_b/N_0 required to obtain a BER of 10^{-3} at 10 decoding iterations versus the phase jitter standard deviation σ_Δ. Both the OL and CL strategies are considered. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

curs in the CL scheme, which is very likely related to the loss of lock in the PLL. This behavior was not noticed when considering SCCCs, in which case there is substantial agreement between BER and FER curves. A possible explanation is based on the fact that the first component decoder of the turbo receiver can use a reduced number of channel observables (due to puncturing). Hence, the very first iteration is less effective than in the SCCC case, so that, if the observations are noisy, the CL-based iterative receiver for a PCCC does not recover (in other words, the embedded PLL cannot recover the channel phase rotation).

4.9.4 Linear Predictive Iterative Detection for Phase-Uncertain Channels

The linear predictive detection schemes proposed in Chapter 3 for sequence detection can be straightforwardly extended to the case of symbol detection. We review a few preliminaries in order to derive linear predictive FB algorithms. We consider the lowpass complex equivalent system depicted in Figure 4.34. We assume that a sequence of K independent and uniformly distributed M-ary symbols $\{a_k\}_{k=0}^{K-1}$,

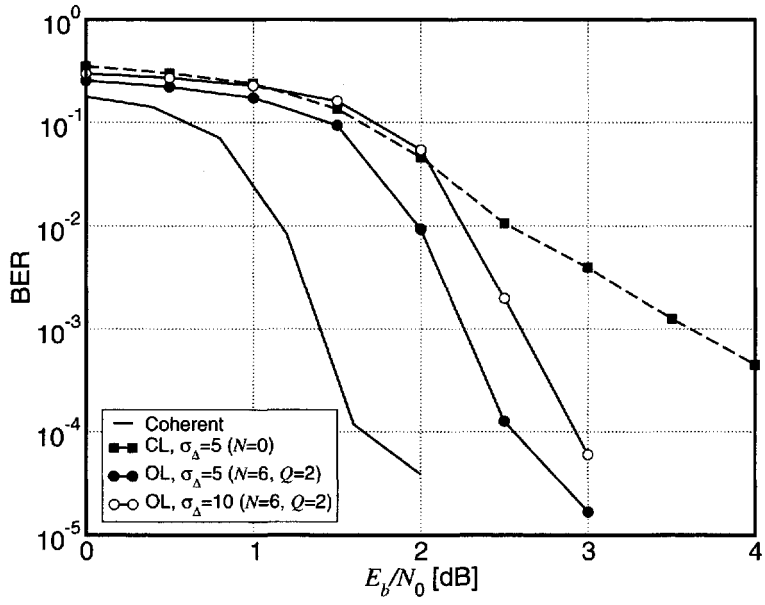


Figure 4.62: BER of a PCCC with OL and CL component decoding algorithms, for phase jitter standard deviation $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees. For comparison, the performance of the equivalent coherent scheme is shown. In all cases, 10 decoding iterations are considered. $N_d = 16$ in all the adaptive cases. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

denoted by the vector $\boldsymbol{a}$ in the figure, feeds an encoder/modulator, which can be modeled as an FSM with state $\mu_k$and transition $t_k = (a_k, \mu_k)$. The linearly modulated continuous-time signal $s(t, \boldsymbol{a})$ is obtained by letting the code symbol c_k be carried by a suitable shaping pulse $p(t)$. Although suboptimal in the presence of a time-varying channel, a matched-filter front-end with sampling rate of one sample per symbol can be practically used, provided that the phase process is not affected by very strong variations [114]. The resulting observation model is

$$r_k = c_k e^{j\theta_k} + n_k \tag{4.247}$$

where $\{n_k\}$ is an iid complex additive Gaussian noise sequence with variance per component equal to $\sigma^2 = N_0$. The channel phase process θ_k is assumed stationary and zero mean, and the autocorrelation sequence of the phasor process $e^{j\theta_k}$ is denoted by $R_\theta(n) \triangleq \mathrm{E}\{e^{j\theta_{k+n}} e^{-j\theta_k}\}$, and is assumed known at the receiver side. In particular,

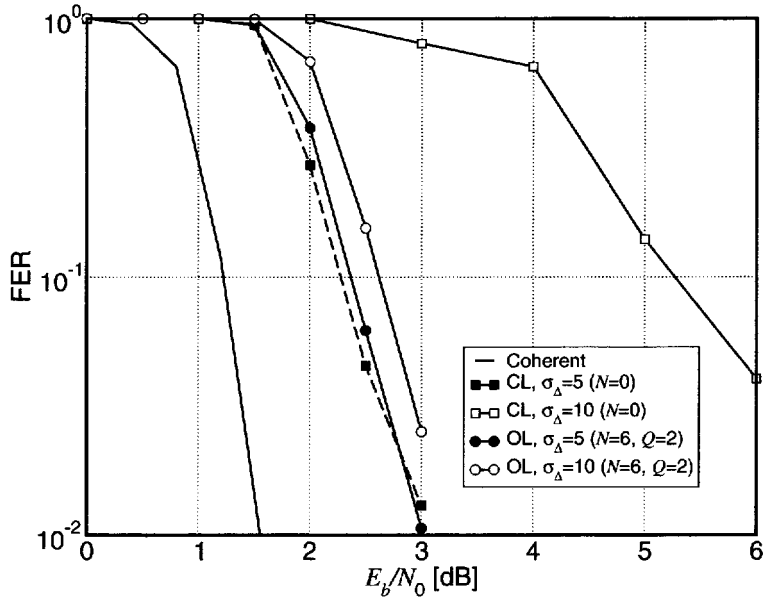


Figure 4.63: FER of a PCCC with OL and CL component decoding algorithms for phase jitter standard deviation $\sigma_\Delta = 5$ degrees and $\sigma_\Delta = 10$ degrees. For comparison, the performance of the equivalent coherent scheme is shown. In all cases, 10 decoding iterations are considered. $N_d = 16$ in all adaptive cases. Reproduced from [129], ©2004 IEEE, by permission of the IEEE.

in the case of linear prediction with order[41] N, the metric can be expressed as (for details, see Chapter 3)

$$\gamma_k(T_k) \sim \exp\left(\frac{|c_k|^2}{2\sigma^2}\right) \exp\left(\frac{\left|r'_k - \frac{\sum_{i=1}^{N} p_i r'_{k-i}}{\left|\sum_{i=1}^{N} p_i r'_{k-i}\right|}\right|^2}{2\sigma^2}\right) P\{a_k\} \qquad (4.248)$$

where the prediction coefficients $\{p_i\}$ are obtained by solving a suitable Yule Walker system, depending on the phasor autocorrelation and the state S_k (see Chapter 3). In particular, in the case of equal-energy modulation the prediction coefficients become independent from the state S_k.

[41] Note that the prediction order coincides with the order of Markovianity if one sample per symbol period is considered. In the case of oversampling, the prediction order is larger than the order of Markovianity. This will be illustrated by considering a system where CPM is used.

A very common model for the phase process $\{\psi_k\}$ is a discrete-time Wiener process [19] described by the following recursion:

$$\psi_k = \psi_{k-1} + \Delta_k \tag{4.249}$$

where $\{\Delta_k\}$ are iid Gaussian increments with zero mean and variance σ_Δ^2, descriptive of the phase noise intensity. In this case, the autocorrelation of the process $\{e^{j\psi_k}\}$ becomes $R_\psi(n) = \exp(-|n|\sigma_\Delta^2/2)$. The case of a time-invariant random frequency offset can also be incorporated by indicating the channel phase as $\theta_k = \psi_k + 2\pi\nu kT$, where ν is a random variable uniformly distributed in $(-\alpha/T, \alpha/T)$, α is the normalized frequency offset intensity, and T denotes the signaling interval. For this phase model, assuming that the phase jitter and the frequency offset are independent, it follows that

$$R_\theta(n) = \exp(-|n|\sigma_\Delta^2/2)\,\mathrm{sinc}(2\alpha n) \tag{4.250}$$

where $\mathrm{sinc}(x) \triangleq \sin(\pi x)/\pi x$.

As an example of application, we consider linear predictive iterative detection of an SCCC. In particular, the considered SCCC is constituted by an outer 4-state, rate-1/2 code connected through a length-1024 pseudorandom bit-interleaver to an inner 4-state, rate-2/3 code—this is the code considered in Section 4.9.3. The output symbols are mapped to an 8-PSK constellation with natural mapping. Pilot symbols are introduced with a rate of 1 pilot every 16 information symbols. At the receiver side, the inner decoder uses the proposed linear predictive FB algorithm, performing joint detection and decoding. The relevant numerical results are shown in Figure 4.64. For comparison, the performance of the corresponding coherent system is also shown. In all cases, five decoding iterations are considered. We evaluated the performance of the system for increasing phase jitter standard deviation σ_Δ (from 2 degrees to 10 degrees) and for various complexity levels, indicated by the couple (N, Q). In particular, the performance loss, with respect to the coherent system, considering $(N, Q) = (6, 2)$ and $\sigma_\Delta = 2$ degrees is about 0.5 dB at a BER of 10^{-4}. The loss increases to 1.4 dB for $\sigma_\Delta = 10$ degrees.

The performance of this system is further analyzed in Figure 4.65, in terms of BER versus phase noise standard deviation, for increasing values of the frequency offset. The SNR is fixed to 4 dB in all cases. For $N = 5$ and $Q = 2$, the proposed iterative detection scheme is very robust to phase instabilities up to $\sigma_\Delta = 10$ degrees and $\alpha = 0.01$. A less complex receiver, with $N = 3$ and $Q = 1$, has still acceptable performance for low values of σ_Δ and α. The performance of the proposed scheme degrades significantly for $\alpha \geq 0.02$.

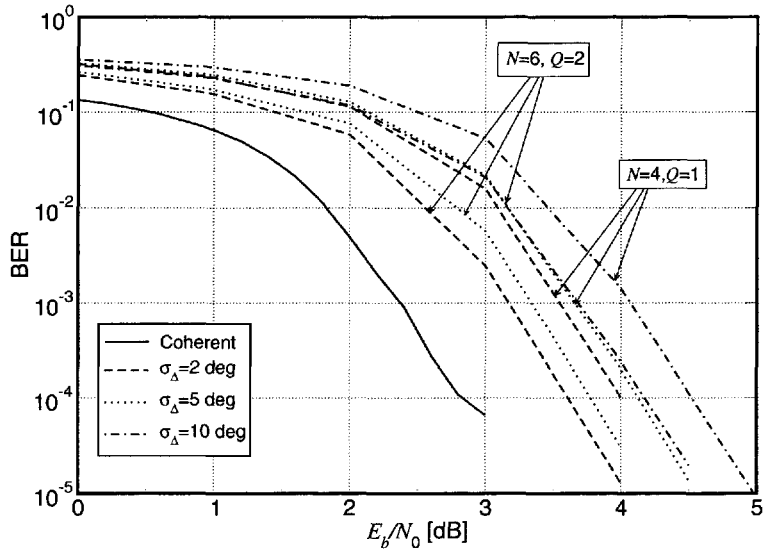


Figure 4.64: BER of an SCCC with 8-PSK and inner linear prediction at the receiver side. Various receiver complexity levels are considered. For comparison, the performance of the coherent system is also shown. In all cases, 5 decoding iterations are considered.

As an interesting example, we consider the application of linear predictive symbol detection techniques to the case of continuous phase modulation (CPM) transmission. The decomposition approach to CPM proposed in [61] clearly shows that any CPM can be interpreted as a serial concatenation of a continuous phase encoder (CPE) and a memoryless mapper (MM). In particular, the CPE is a rate-$1/L$ recursive encoder with state $s_k = (a_{k-1}, \ldots, a_{k-L+1}, \mu_{k-L+1})$, where L is the duration of the frequency impulse characterizing the CPM signal and μ_k is a p-ary variable which is recursively updated according to $\mu_k = R_p[\mu_{k-1} + a_k]$, where $R_p[*]$ indicates the integer remainder of * in base p. The integer p is related to the CPM modulation index h: it holds indeed that $h = k/p$, with k and p relatively prime numbers. The equivalent base-band CPM signal $s_{\mathrm{bb}}(t, \boldsymbol{a})$ can be written as [61]

$$s_{\mathrm{bb}}(t, \boldsymbol{a}) = e^{j\overline{\psi}(t,\boldsymbol{a})} \tag{4.251}$$

where the *tilted phase* $\overline{\psi}$ has the following expression:

$$\begin{aligned}\overline{\psi}(\tau + kT, \boldsymbol{a}) &= R_{2\pi}[2\pi h \mu_{k-L+1} \\ &\quad + 4\pi h \sum_{i=0}^{L-1} a_{k-i} f(\tau + iT) + w(\tau)] \quad 0 \le \tau \le T \end{aligned} \tag{4.252}$$

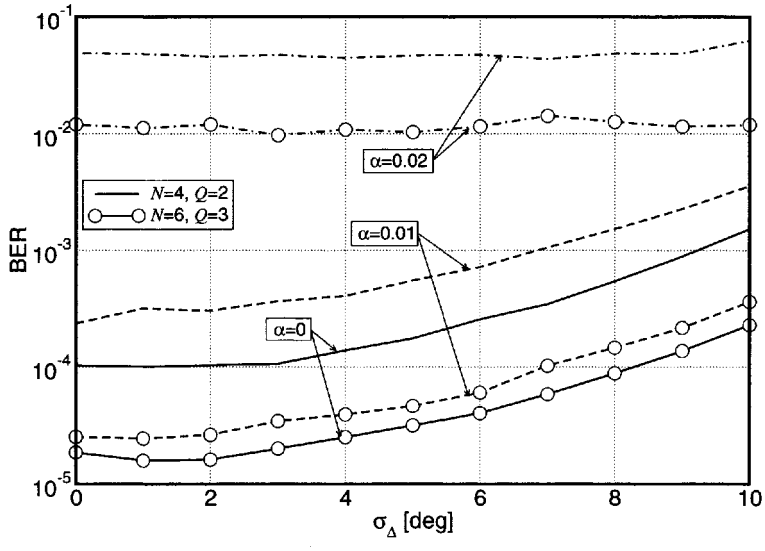


Figure 4.65: BER as a function of the phase noise standard deviation σ_Δ for an SCCC with 8-PSK, an inner linear predictive detector, $E_b/N_0 = 4$ dB, various values of frequency offset intensity and levels of receiver complexity. In all cases, 5 decoding iterations are considered. Reproduced from [128], ©2003 IEEE, by permission of the IEEE.

where $f(t)$ is the CPM phase pulse and $w(\tau)$ is a function of time only, not depending on information symbols [61].

A simple and almost optimal receiver structure consists of a bandpass filter followed by oversampling (see Appendix A for more details). Considering β samples per symbol interval, the samples relative to the time interval at epoch k have the following expression:

$$r_k^{(i)} = e^{j\theta_k^{(i)}} c_k^{(i)} + n_k^{(i)} = e^{j[\theta_k^{(i)} + \overline{\psi}(iT/\beta + kT, \boldsymbol{a})]} + n_k^{(i)} \qquad i = 0, \ldots, \beta - 1. \tag{4.253}$$

The bandwidth expansion, due to oversampling, in the front-end bandpass filter makes the variance of the Gaussian noise sample $n_k^{(i)}$ equal to $\beta 2\sigma^2$. As the channel noise is white, it follows that $\{n_k^{(i)}\}$ are independent. In this case, we assume that $e^{j\theta_k^{(i)}}$ is estimated based on the sequence $(r_{k-N}^{(i),\prime}, \ldots, r_{k-N}^{(\beta-1),\prime}, \ldots\ldots, r_{k-1}^{(0),\prime}, \ldots, r_{k-1}^{(\beta-1),\prime}, r_k^{(0),\prime}, \ldots, r_k^{(i-1),\prime})$. Figure 4.66 highlights the sliding window prediction strategy considered in the case of oversampling. Different strategies, based on an expanding window of observations, could be considered. With the current estimation strategy, the prediction

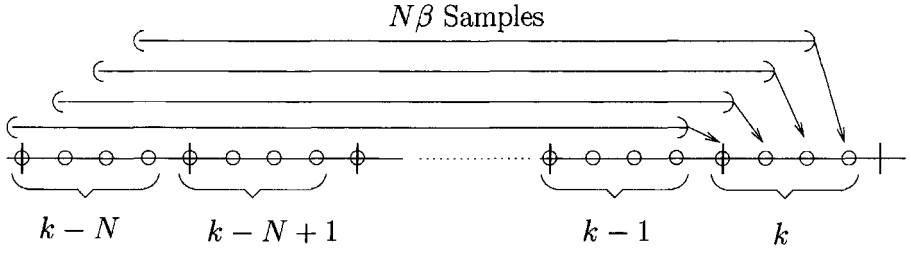


Figure 4.66: Sliding window linear prediction strategy in the case of oversampling with $\beta = 4$.

order becomes $N\beta$ and the basic exponential metric can be expressed as

$$\gamma_k(T_k) = \exp\left(-\frac{\sum_{i=0}^{\beta-1} |c_k^{(i)}|^2 |r_k^{(i),'} - \widehat{e^{\theta_k^{(i)}}}|}{(\epsilon_k^{\circ})^2} \right) P\{a_k\} \tag{4.254}$$

where $(\epsilon_k^{\circ})^2$ is the minimum mean square error (MMSE) in the current case with oversampling. In particular, denoting by $R_{\theta}^{\circ-\beta}(j)$ the autocorrelation of the oversampled phase process $\{\theta_k^{(i)}\}$, it is possible to extend the MMSE problem in (3.172) as follows:

$$\min_{\{p_i^{\circ}\}_{i=1}^{N\beta}} \mathrm{E}\left\{ \left| r_k^{(i),'} - \sum_{j=0}^{i-1} p_{i-j}^{\circ} r_{k-i}^{(j),'} - \sum_{l=0}^{N-2} \sum_{j=0}^{\beta-1} p_{l\beta+j}^{\circ} r_{k-1-l}^{(j),'} - \sum_{j=i}^{\beta-1} p_{N\beta+i-j}^{\circ} r_{k-N}^{(j),'} \right|^2 \middle| \boldsymbol{c}_{k-N\beta}^{k-1} \right\} \tag{4.255}$$

where $\{p_i^{\circ}\}$ represent the prediction coefficients for the considered oversampled detection strategy. Solving the MMSE problem (4.255) in order determine the prediction coefficients $\{p_i^{\circ}\}$ leads to a Yule Walker system in the form $\boldsymbol{R}^{\circ}\boldsymbol{p}^{\circ} = \boldsymbol{b}^{\circ}$, where $\boldsymbol{R}^{\circ}$ is a square $N\beta \times N\beta$ matrix whose elements have, owing to the fact that $|c_k^{(i)}| = 1$, the following expression:

$$[\boldsymbol{R}^{\circ}]_{l,m} = \begin{cases} R_{\theta}^{\circ-\beta}(|l-m|) & \text{if } l \neq m \\ R_{\theta}^{\circ-\beta}(0) + 2\sigma^2 & \text{if } l = m. \end{cases} \tag{4.256}$$

Finally, the MMSE in the case of oversampling is formally identical to the case without oversampling, and has the following expression:

$$(\epsilon_k^{\circ})^2 = R_\theta^{\circ-\beta}(0) + 2\sigma^2 - \sum_{i=1}^{N\beta} p_i^{\circ} R_\theta^{\circ-\beta}(i). \tag{4.257}$$

As an example of application of linear predictive detection to the case of CPM signaling, we consider a serially concatenated scheme obtained by concatenating an outer convolutional code and an inner Gaussian minimum shift keying (GMSK) modulator [17]. In particular, we refer to the GSM standard [190], where the outer code is a 16-state NRNSC code with rate-1/2 and generator matrix

$$\boldsymbol{G}_{\circ}(D) = \left[1 + D^3 + D^4 \qquad 1 + D + D^3 + D^4\right]. \tag{4.258}$$

The outer code and the GMSK modulator are connected through a length-1024 pseudorandom bit-interleaver [33]. The spectral efficiency of the overall code is 0.5 bit/s/Hz. At the receiver side, we consider $\beta = 2$ samples per symbol interval. The numerical results are shown in Figure 4.67. As one can see, for sufficiently large N and Q the performance loss, with respect to the coherent receiver, is within 1.2 dB for $\sigma_\Delta \leq 10$.

4.9.5 Noncoherent Iterative Detection for Slow Frequency Nonselective Fading Channels

Considering, as a first approximation, the case of a time-invariant frequency nonselective fading channel, an observation can be written as

$$r_k = f e^{j\theta} c_k + n_k \tag{4.259}$$

where f is modeled as a complex Gaussian random variable with mean η_f and variance σ_f^2, θ is modeled as a random variable with uniform distribution in the interval $[0, 2\pi)$ and independent of the multiplicative fading coefficient, and n_k is a complex AWGN sample with variance per component equal to $\sigma^2 = N_0$. The exponential metric γ_k can be written as follows:

$$\gamma_k(T_k) = \frac{\mathrm{E}_{f,\theta}\{p(\boldsymbol{r}_{k-N}^{k} | \boldsymbol{c}_{k-N}^{k})\}}{\mathrm{E}_{f,\theta}\{p(\boldsymbol{r}_{k-N}^{k-1} | \boldsymbol{c}_{k-N}^{k-1})\}} P\{a_k\}. \tag{4.260}$$

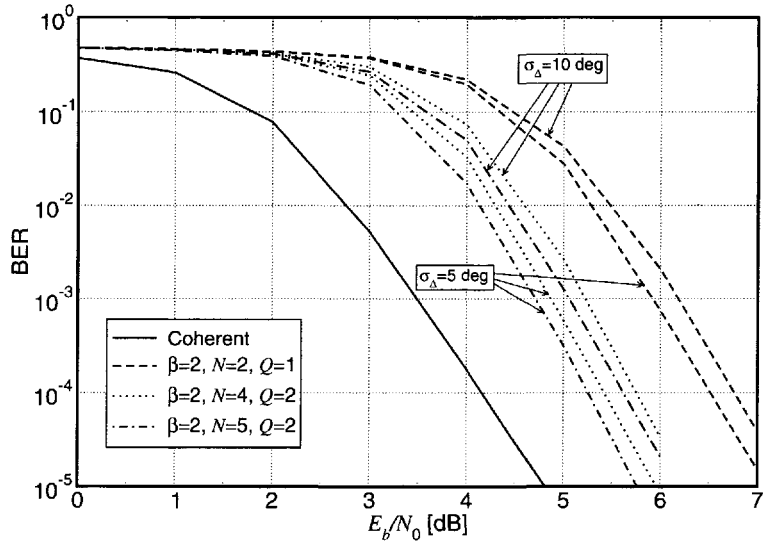


Figure 4.67: BER of a serially concatenated scheme given by an outer convolutional code and inner GMSK modulator. At the receiver side there is inner linear prediction with a sampling rate of $\beta = 2$ samples per symbol. Various receiver complexity levels are considered. For comparison, the performance of the coherent system is also shown. In all cases, 5 decoding iterations are considered. Reproduced from [128], ©2003 IEEE, by permission of the IEEE.

Assuming that σ_f^2 and $|\eta_f|$ are known, following the derivation in Section 3.11.3, it is possible to obtain

$$\mathrm{E}_{f,\theta}\{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{c}_{k-N}^k)\} \sim \frac{\omega}{1+\omega\sum_{i=0}^{N}|c_{k-i}|^2}\mathrm{I}_0\left(\frac{1}{\sigma^2}\left[\frac{|\eta_f|\sum_{i=0}^{N}|c_{k-i}|^2}{1+\omega\sum_{i=0}^{N}|c_{k-i}|^2}\right]\right)$$
$$\cdot\exp\left(\frac{1}{2\sigma^2}\left[\frac{\omega\left|\sum_{i=0}^{N}r_{k-i}c_{k-i}^*\right|^2-|\eta_f|^2\sum_{i=0}^{N}|c_{k-i}|^2}{1+\omega\sum_{i=0}^{N-1}|c_{k-i}|^2}\right]\right) \tag{4.261}$$

where $\omega \triangleq \sigma_f^2/2\sigma^2$. By assuming a Rayleigh fading channel ($\eta_f = 0$) and large SNR ($\omega \gg 1$), the metric (4.260) can be further expressed as follows:

$$\gamma_k(T_k) \sim \frac{\sum_{i=1}^{N}|c_{k-i}|^2}{\sum_{i=0}^{N}|c_{k-i}|^2} \exp\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|^2}{2\sigma^2\sum_{i=0}^{N}|c_{k-i}|^2} - \frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|^2}{2\sigma^2\sum_{i=1}^{N}|c_{k-i}|^2}\right) P\{a_k\}. \tag{4.262}$$

Considering finally equal-energy signaling ($|c_k| = 1$), the metric (4.260) can be further simplified as

$$\gamma_k(T_k) \sim \exp\left(\frac{\left|\sum_{i=0}^{N} r_{k-i}c_{k-i}^*\right|^2}{2\sigma^2(N+1)} - \frac{\left|\sum_{i=1}^{N} r_{k-i}c_{k-i}^*\right|^2}{2\sigma^2 N}\right). \tag{4.263}$$

As an example of application, we consider transmission of an SCCC over a Rayleigh flat fading channel with normalized Doppler rate $f_D T = 0.01$. The code consists of an outer 4-state, rate-1/2 convolutional code connected through a length-1024 pseudorandom interleaver to an inner 4-state, rate-1/2 convolutional code. The respective generator matrices are given by

$$\boldsymbol{G}_o(D) = \left[1 + D + D^2 \quad 1 + D^2\right] \qquad \boldsymbol{G}_i(D) = \left[1 \quad \frac{1 + D^2}{1 + D + D^2}\right]. \tag{4.264}$$

The output symbols are mapped to a QPSK constellation with Gray mapping, resulting in an overall code of spectral efficiency 0.5 bit/s/Hz. In Figure 4.68, the performance, in terms of BER versus E_b/N_0, E_b being the received energy per information bit, is shown. In particular, the inner adaptive decoder/detector (using linear prediction) makes use of an FB algorithm in the logarithmic domain (based on the max-log approximation), whereas the outer coherent decoder is the coherent FB algorithm. Various complexity levels, in terms of the couple (N, Q), are considered.

4.9.6 Linear Predictive Iterative Detection for Fading Channels

In this case, the observation r_k is modeled as follows:

$$r_k = c_k f_k + n_k \tag{4.265}$$

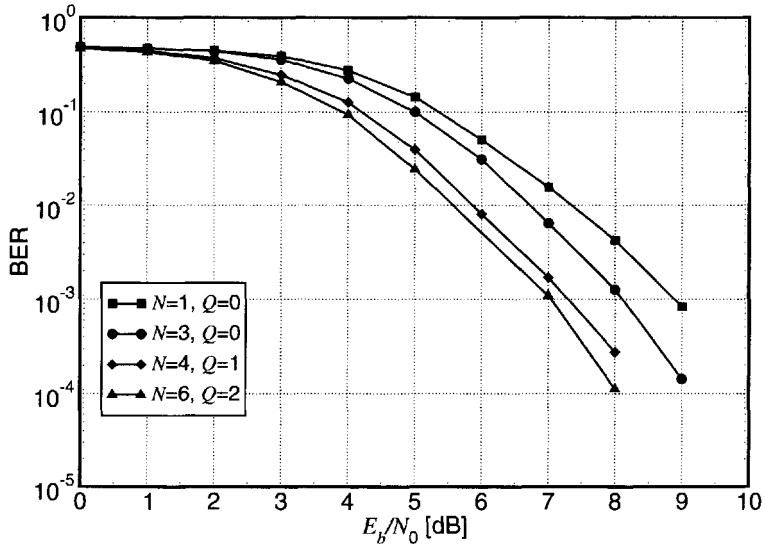


Figure 4.68: BER of an SCCC with inner noncoherent combined detection and decoding over a Rayleigh flat fading channel with normalized bandwidth $f_{\mathrm{D}}T = 0.01$. In all cases, 5 decoding iterations are considered. Reproduced from [149], ©2003 IEEE, by permission of IEEE.

where $\{c_k\}$ is a sequence of linearly modulated (and possibly coded) symbols, $\{f_k\}$ is a sequence of complex random variables, jointly Gaussian and with mean η_f, and $\{n_k\}$ is a complex AWGN process with variance per component equal to $\sigma^2 = N_0$. The autocorrelation of this process follows the isotropic scattering model [21]

$$R_f(n) = \mathrm{E}\{f_k f_{k-n}^*\} = \mathrm{J}_0(2\pi f_{\mathrm{D}} T n) \tag{4.266}$$

where $\mathrm{J}_0(\cdot)$ is the zero-th order Bessel function of the first kind and $f_{\mathrm{D}}T$ is the normalized fading bandwidth. As r_k is conditionally Gaussian, a possible approach to the computation of γ_k consists of estimating r_k based on the previous N observations. By introducing the modified observation[42]

$$r'_k \triangleq \frac{r_k}{c_k} = f_k + \frac{n_k}{c_k} \tag{4.267}$$

the metric γ_k can be expressed as follows

$$\gamma_k(T_k) \sim \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \exp\left(-\frac{|r'_k - \hat{f}_k|^2}{\epsilon_k}\right) P\{a_k\} \tag{4.268}$$

[42]This is an extension of the approach proposed in [101] and was considered in [160].

where

$$\hat{f}_k \triangleq \sum_{i=1}^{N} p_i r'_{k-i} = \sum_{i=1}^{N} p_i \frac{r_{k-i}}{c_{k-i}} \tag{4.269}$$

and

$$\epsilon_k \triangleq \mathrm{E}\left\{|r'_k - \hat{f}_k|^2\right\} = \mathrm{E}\left\{\left|\frac{r_k}{c_k} - \sum_{i=1}^{N} p_i \frac{r_{k-i}}{c_{k-i}}\right|^2\right\}. \tag{4.270}$$

The prediction coefficients $\{p_i\} = \{p_i(S_k)\}$ correspond to the solution of a Wiener Hopf linear system $\boldsymbol{Rp} = \boldsymbol{b}$ where $\boldsymbol{R}$ is a square $N \times N$ matrix whose elements have the following expression

$$[\boldsymbol{R}]_{l,m} = \begin{cases} R_f(|l-m|) & \text{if} \quad l \neq m \\ R_f(0) + \frac{2\sigma^2}{|c_{k-l}|^2} & \text{if} \quad l = m \end{cases} \tag{4.271}$$

$\boldsymbol{p} \triangleq [p_1 \cdots p_N]^{\mathrm{T}}$ is the unknown vector and $\boldsymbol{b} = [R_f(1), R_f(2), \cdots, R_f(N)]^{\mathrm{T}}$.

As an example of application, we consider iterative detection of the same SCCC considered in Section 4.9.5. The considered modulation format is QPSK. The numerical results are shown in Figure 4.69. Comparing the BER performance in Figure 4.69, relative to linear predictive iterative detection, with that in Figure 4.68, relative to noncoherent iterative detection, one can immediately recognize minimal differences. This suggests that the performance in the case of finite memory detection depends ultimately on the finite memory parameter N, rather than on the specific considered detection strategy. For more details, see [149].

4.10 Summary

This chapter has been dedicated to MAP symbol detection based on the use of *forward backward algorithms*. After deriving the forward backward algorithm in the case of a memoryless channel for the exact computation of symbol *a posteriori* probabilities, *iterative detection and decoding* has been introduced and described. The concept of *extrinsic information*, exchanged in an iterative decoding process, has been introduced and its impact on the process has been evaluated. Trellis-based finite memory symbol detection, based on the forward backward algorithm, has then been described as a particular instance of the general framework developed in Chapter 2. State reduction techniques for forward backward algorithms have been thoroughly

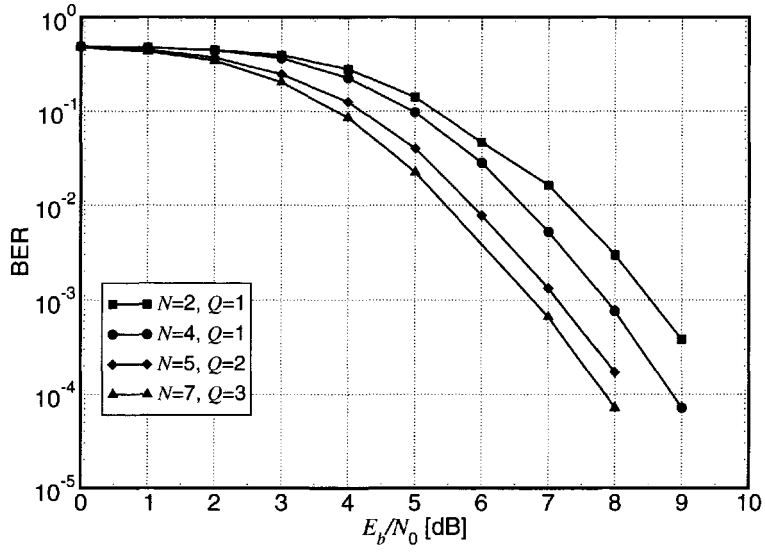


Figure 4.69: BER of an SCCC with inner linear prediction-based combined detection and decoding over a Rayleigh flat fading channel with normalized bandwidth $f_D T = 0.01$. In all cases 5 decoding iterations are considered. Reproduced from [149], ©2003 IEEE, by permission of IEEE.

investigated and demonstrated, especially for ISI channels. Several examples of applications to wireless communications have been illustrated: noncoherent iterative detection for phase-uncertain and fading channels; pilot symbol-assisted iterative detection for phase-uncertain channels; linear predictive iterative detection for phase-uncertain and fading channels.

4.11 Problems

Problem 4.1: Consider the linear block code considered in Problem 1.6. Derive an FB algorithm to be run over the time-varying trellis diagram relative to the linear block code. Compare your solution with the general results presented in [54, 55, 191].

Problem 4.2: Re-do the same derivation of Fwd-only RS-FB algorithms in Section 4.8.1 considering an FMC rather than a CMP.

Problem 4.3: In [98, Section 3.2], the authors propose a method to alter the convergence rate of iterative decoders, by *filtering* the soft information. Con-

sidering the turbo decoder in Figure 4.3, this soft information filtering, when applied to the values at the input of the second component decoder, consists of replacing the extrinsic information $S_n^{(E,1)}[a_k]$ on a_k generated by the first component decoder at the n-th iteration by the value $c_{\text{scal}} \times S_n^{(E,1)}[a_k] + (1 - c_{\text{scal}}) \times S_{n-1}^{(E,1)}[a_k]$, where the parameter $c_{\text{scal}} \in (0, 1]$ can be chosen to adjust the bandwidth of the corresponding filter. Compare this result with the heuristic approach, based on multiplying the generated reliability value, in the logarithmic domain, with a multiplicative parameter $\theta \in (0, 1]$ (use of this parameter is considered, for example, for SOVA-based iterative detection in Section 4.5.3 and in reduced-state detection techniques proposed in Section 4.8). In particular, discuss the relation between θ and c_{scal}.

Problem 4.4: Verify by computer simulations that adding an interleaver between the 16-QAM mapper and the differential encoder in Figure 4.47 does not modify the performance results shown in Figure 4.48. Try to justify this result.

Problem 4.5: Considering a serial structure as in Figure 4.49, find a *recursive* inner convolutional code such that, by keeping the spectral efficiency equal to 2 b/s/Hz, the performance is better than that of the code considered in Figure 4.52. Use computer simulations.

Problem 4.6: Show that for sufficiently large values of SNR and N, assuming that $\widehat{\theta}^{(N+1)} \simeq \widehat{\theta}^{(N)}$ in (4.243) is a good approximation for a time-invariant channel phase rotation.

Problem 4.7: Verify by computer simulations that the performance results, in terms of BER, shown in Figure 4.59 and relative to iterative detection of an SCCC are qualitatively the same when considering the FER performance. Compare this behavior with that of a PCCC shown in Figure 4.62 and Figure 4.63: in this case, performance results, in terms of BER and FER, are different. Try to justify these different behaviors of PCCCs and SCCCs.

Problem 4.8: Let $\max*\{x_1, \ldots, x_n\} \triangleq \ln(e^{x_1} + \ldots + e^{x_n})$.

A. Show that $\max*\{\max*\{x_1, x_2\}, x_3\} = \max*\{x_1, x_2, x_3\}$.

B. Show that $\max*\{x_1, x_2\} = \max\{x_1, x_2\} + \ln(1 + e^{-|x_1 - x_2|})$.

C. Show that the *exact* FB algorithm can be formulated replacing the max operator with $\max*$ in the max-log approximation.

D. Discuss the computational complexity of the FB algorithm and compare it with the complexity of the max-log FB algorithm.

Problem 4.9: Consider uncoded transmission of binary symbols $a_k \in \{\pm 1\}$ through a static dispersive channel with white noise discrete equivalent impulse response $\boldsymbol{f} = (1, 2, 1)/\sqrt{6}$. Assume that the received sequence and the initial state are as in Problem 3.14. Use the max-log FB algorithm to approximately detect the sequence of MAP symbols $\{\hat{a}_k^{(\mathrm{FB})}\}_{k=0}^{7}$ and compare this result with the MAP sequence found in Problem 3.14. What can you conclude? Compare this result with that obtained in [148].

Problem 4.10: Consider the serially concatenated scheme in Figure 4.57. Using the same code as in Figure 4.59, verify that, when OL-based and CL-based iterative detection with the algorithms described in Section 4.9.3 is carried out, the performance in terms of FER is basically the same. Compare your results with those in Figure 4.63 and comment on this.

Problem 4.11: In Section 4.9.4, linear predictive iterative detection of a serial scheme given by the concatenation of an outer convolutional code and an inner GMSK modulator is considered. At the receiver side, the inner linear predictive detector is based on the decomposed representation of a CPM signal into a sequence of a CPE and an MM [61]. Is it possible to extend this approach if the CPM signal is represented as a superposition of pulse amplitude modulation (PAM) waveforms [59, 60]? If so, comment on the relative complexity of this solution with that proposed in Section 4.9.4.

5

Graph-Based Detection: Algorithms and Applications

5.1 Introduction

In this chapter, a graph-based approach to maximum *a posteriori* (MAP) symbol detection will be applied to derive new algorithms for joint detection and decoding. Namely, we introduce a factor graph (FG)-based approach to represent the joint *a posteriori* probability (APP) of the transmitted sequence, and we apply the sum-product (SP) algorithm to derive the desired symbol APPs necessary to implement MAP symbol detection. After a brief description of FGs and the SP algorithm (for more insights, the interested reader may refer to [79]), we will focus on the application of the general framework introduced in Chapter 2 to perform graph-based detection. By means of a factorization of the joint APP of the transmitted symbols and application of the finite memory condition (FMC) introduced in Chapter 2, we derive an FG representing both the code constraints and the channel model. In this FG, the channel parameters *are not* explicitly represented since they are *a priori* averaged out. The application of the SP algorithm to this FG leads to an iterative scheme for joint detection and decoding. The above mentioned factorization is exact in the case of channels with *finite* memory, such as a channel with known inter-symbol interference (ISI), and approximate for channels with *infinite* memory. This latter case includes a noncoherent channel and a flat correlated Rayleigh fading channel. For these channels, the factorization is based on the application of the conditional Markov property (CMP) introduced in Chapter 2. In the derived general FG, the

Detection Algorithms for Wireless Communications - G. Ferrari, G. Colavolpe and R. Raheli
 ISBN: 0-470-85828-1

function associated with a generic factor node modeling the channel will be the same basic metric (γ_k) used in the previous chapters. As already pointed out in Chapter 2, detection strategies considered for Viterbi algorithm (VA)-based sequence detection algorithms (in Chapter 3) and forward backward (FB)-based symbol detection algorithms (in Chapter 4) can be systematically extended to derive graph-based detection algorithms.

When the SP algorithm is run over the derived FGs, the computational complexity is concentrated at the factor nodes modeling the channel. In fact, the computation complexity at these nodes is, in general, *exponential* in the finite memory parameter (or Markovianity order). Techniques for complexity reduction are therefore studied and illustrated in the following. In a few particular cases of practical relevance, the complexity of the proposed graph-based detection algorithms becomes *linear* in the finite memory parameter. This does not happen in the same cases when detection is performed by using the VA or the FB algorithm, suggesting that the use of FGs and the SP algorithm might be computationally more appealing.

An alternative approach to graph-based detection for phase-uncertain channels will be also considered at the end of this chapter. This approach is inspired by [192] where a general framework to solve the problem of joint decoding and estimation, in the presence of unknown parameters, is described. The approach is Bayesian, i.e., the channel parameters are modeled as stochastic processes with known statistics and the use of FGs that include both code constraints and channel statistics is advocated in a very general setting. The SP algorithm is then used to implement the MAP symbol detection strategy. Since the channel parameters, which are continuous random variables, are explicitly represented in the graph, the application of the SP algorithm becomes impractical. To solve this problem, the method of *canonical distributions* is adopted. We will show that the choice of the used canonical distribution becomes crucial in determining the performance and the complexity of the derived algorithms. In particular, we will show that this approach can produce, in some cases, algorithms with better performance or lower complexity.

The graph-based algorithms described in this chapter are inherently iterative. As a consequence, although they can be used to decode a general channel code, they are particularly suited to turbo codes and low density parity check (LDPC) codes whose decoding is iterative even when they are transmitted over a memoryless channel. In almost all numerical results shown to illustrate the proposed algorithms, we will consider LDPC codes since the methods described in Chapter 4 cannot be applied to them unless trellis-based detection, performed by means of FB algorithms, is used.

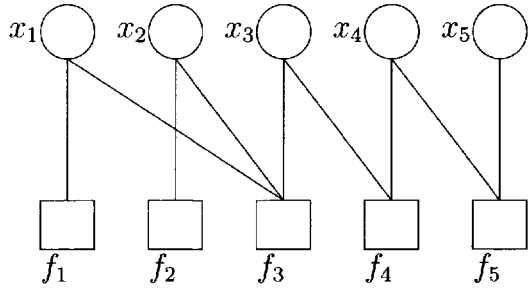


Figure 5.1: The FG corresponding to the factorization (5.1).

5.2 Factor Graphs and the Sum-Product Algorithm

We now briefly recall the basics of FGs and the SP algorithm. Let $\boldsymbol{x} = \{x_1, \ldots, x_N\}$ denote a set of variables and $f(\boldsymbol{x})$ a multivariate function. Let $\mathcal{B}_1, \ldots, \mathcal{B}_m$ denote subsets of $\boldsymbol{x}$. We say that $f(\boldsymbol{x})$ admits a factorization with supports $\mathcal{B}_1, \ldots, \mathcal{B}_m$, if $f(\boldsymbol{x})$ can be written as the product of the functions $\{f_j : j = 1, \ldots, m\}$, where f_j has the variables in $\mathcal{B}_j$ as arguments. The FG representing the factorization $f = \prod_j f_j$ is a *bipartite* graph $\mathcal{G} = \{\mathcal{V}, \mathcal{F}, \mathcal{E}\}$, such that nodes in $\mathcal{V}$ (variable nodes) are associated with the variables $x_i \in \boldsymbol{x}$, nodes in $\mathcal{F}$ (factor nodes) are associated with the functions f_j, and there exists an edge $e \in \mathcal{E}$ joining x_i and f_j if and only if $x_i \in \mathcal{B}_j$ (i.e., if x_i is an argument of f_j). A *cycle* is a closed path in the graph and its *length* is defined as the corresponding number of path edges. The length of the smallest cycle is the *girth* of the graph.

Example 1. Let f be a function of five variables, and suppose that it can be expressed as the product

$$f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) f_4(x_3, x_4) f_5(x_4, x_5). \tag{5.1}$$

In this case, $m = 5$, $\mathcal{B}_1 = \{x_1\}$, $\mathcal{B}_2 = \{x_2\}$, $\mathcal{B}_3 = \{x_1, x_2, x_3\}$, $\mathcal{B}_4 = \{x_3, x_4\}$, and $\mathcal{B}_m = \{x_4, x_5\}$. The corresponding FG is shown in Figure 5.1. It can be observed that in this graph there are no cycles. □

Example 2. Let f be a function of five variables, and suppose that it can be expressed as the product

$$f(x_1, x_2, x_3, x_4, x_5) = f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) f_4(x_3, x_4, x_5) f_5(x_4, x_5). \tag{5.2}$$

In this case, $m = 5$, $\mathcal{B}_1 = \{x_1\}$, $\mathcal{B}_2 = \{x_2\}$, $\mathcal{B}_3 = \{x_1, x_2, x_3\}$, $\mathcal{B}_4 = \{x_3, x_4, x_5\}$, and $\mathcal{B}_m = \{x_4, x_5\}$. The corresponding FG is shown in Figure 5.2. As one can see,

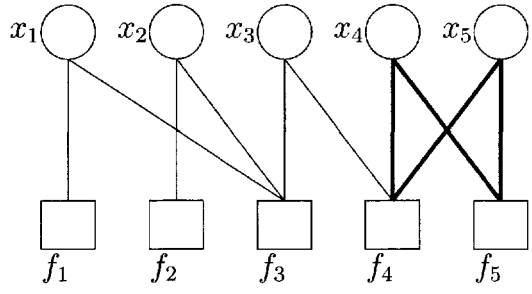


Figure 5.2: The FG corresponding to the factorization (5.2).

there is a cycle of length 4 highlighted with thicker lines. Hence, the graph has girth 4. □

We now specialize the concepts introduced above regarding FGs to the derivation of SP algorithms. Let now $f(\boldsymbol{x})$ be a probability mass function (pmf). If the FG corresponding to the factorization of f has no cycles, the marginal pmfs can be computed exactly in a finite number of steps by the SP algorithm [79]. The SP algorithm is defined by the computation rules at variable and factor nodes, and by a suitable node activation schedule. Denoting by $\mu_{x_i \to f_j}(x_i)$ a message sent from the variable node x_i to the factor node f_j, by $\mu_{f_j \to x_i}(x_i)$ a message in the opposite direction, and by $\mathcal{A}_i$ the set of functions f_j having x_i as argument, the message computations performed at variable and factor nodes are, respectively [79]

$$\mu_{x_i \to f_j}(x_i) = \prod_{h \in \mathcal{A}_i \setminus \{f_j\}} \mu_{h \to x_i}(x_i) \tag{5.3}$$

$$\mu_{f_j \to x_i}(x_i) = \sum_{\sim\{x_i\}} \left[f_j(\{y \in \mathcal{B}_j\}) \prod_{y \in \mathcal{B}_j \setminus \{x_i\}} \mu_{y \to f_j}(y) \right] \tag{5.4}$$

where, following the notation of [79], we indicate[1] by $\sum_{\sim\{x_i\}}$ the *summary* operator, i.e., a sum over all variables excluding x_i. It can be observed that the message sent on an edge does not depend on the message previously received on the same edge, i.e., only *extrinsic information* is exchanged [33].

If the FG contains cycles, convergence to the exact marginal pmfs is not guaranteed in general. Moreover, the SP algorithm is inherently iterative. Nevertheless, for many relevant problems characterized by FGs with cycles, the SP algorithm has been

[1]The symbol $\sim$ usually indicates a proportionality relation throughout the book. The context eliminates any ambiguity.

found to provide very good results and therefore it represents a viable solution to the approximated marginalization of multivariate pmfs when exact calculation is not feasible because of complexity. Remarkably, the ubiquitous *belief propagation* (BP) iterative decoding algorithm used to decode LDPC codes [80], the turbo-decoding algorithm [33, 79, 193], turbo multiuser detection/decoding [194] and turbo equalization algorithms [73, 74] are instances of the SP algorithm or approximations thereof.

The messages in (5.3) and (5.4) may be also computed in the logarithmic domain. Defining $\overline{\mu}_{f_j \to x_i}(x_i) \triangleq \ln \mu_{f_j \to x_i}(x_i)$ and $\overline{\mu}_{x_i \to f_j}(x_i) \triangleq \ln \mu_{x_i \to f_j}(x_i)$, the message computations performed at variable and factor nodes become

$$\overline{\mu}_{x_i \to f_j}(x_i) = \sum_{h \in \mathcal{A}_i \setminus \{f_j\}} \overline{\mu}_{h \to x_i}(x_i) \tag{5.5}$$

$$\overline{\mu}_{f_j \to x_i}(x_i) = \ln \left\{ \sum_{\sim \{x_i\}} \exp \left[\ln f_j(\{y \in \mathcal{B}_j\}) + \sum_{y \in \mathcal{B}_j \setminus \{x_i\}} \overline{\mu}_{y \to f_j}(y) \right] \right\}. \tag{5.6}$$

The implementation of (5.6) does not require multiplications but only additions and the evaluation of a nonlinear function. In fact, by using the Jacobian logarithm [137, 195, 196], it is well known that, if x_1 and x_2 are real numbers then

$$\ln(e^{x_1} + e^{x_2}) = \max(x_1, x_2) + \ln(1 + e^{-|x_2 - x_1|}) \tag{5.7}$$

and $\ln(1 + e^{-|x_2 - x_1|})$ is the nonlinear function whose evaluation requires a look-up table. Evaluation of $\ln(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$ can be done recursively [137]. A further simplification of the updating rule (5.6) may be approximated by using the so-called *max-log* approximation (see Section 4.3):

$$\overline{\mu}_{f_j \to x_i}(x_i) \approx \max_{\sim \{x_i\}} \left[\ln f_j(\{y \in \mathcal{B}_j\}) + \sum_{y \in \mathcal{B}_j \setminus \{x_i\}} \overline{\mu}_{y \to f_j}(y) \right] \tag{5.8}$$

having denoted by $\max_{\sim \{x_i\}}$ a maximization over all variables excluding x_i.

A *message passing schedule* in the SP algorithm is the specification of the order in which messages are updated. In general, especially for graphs with cycles, the so-called *flooding schedule* is adopted [197]: in each iteration, all variable nodes and subsequently all factor nodes pass new messages to their neighbors. This schedule is well suited for a fully parallel implementation of the iterative detectors/decoders presented in this chapter. Other schedules may be adopted, serial or mixed serial parallel, according to the specific implementation requirements.

FGs and the SP algorithm can be also used to describe and marginalize a joint probability density function (pdf) or even a joint pdf with some discrete probability masses. In this case, in the computation of the messages at factor nodes, the summary operator involves integration with respect to the continuous random variables. Obviously, in this case the implementation complexity of the exact SP algorithm is impractical. A solution for this problem is suggested in [192] and consists of the use of *canonical distributions*. In Section 5.7, we will consider several canonical distributions, yielding different algorithms for joint detection and decoding over a channel which introduces a strong phase noise.

FGs are a generalization of other remarkable graphs proposed in the literature. In 1981, Tanner introduced the so-called *Tanner graphs*, which can be used to describe the LDPC codes [81]. LDPC codes are linear codes specified by a *sparse* parity check matrix and were first introduced by Gallager [32] in their original *regular* version. A (d_v, d_c)-regular LDPC code, is a binary linear code such that every code bit participates in exactly d_v parity check equations and every check equation involves exactly d_c code bits. In other words, the corresponding parity check matrix $\boldsymbol{H}$ has d_v ones in each column and d_c ones in each row. As originally suggested by Tanner [81], LDPC codes are well represented by *bipartite graphs* in which the variable nodes represent the elements of a codeword and the factor nodes, also called in this case *check nodes*, correspond to the set of parity check constraints which define the code. This graph can be drawn by direct inspection of a parity check matrix $\boldsymbol{H}$ of the code. Regular LDPC codes are such that all nodes of the same type have an equal number of edges. On the contrary, for *irregular* LDPC codes, the edge degree of each node in each set is not equal but chosen according to an optimized distribution [146, 198, 199]. In terms of performance, regular LDPC codes are only slightly inferior to parallel and serial concatenated convolutional codes. However, in their irregular version, they exhibit an impressive performance outperforming the best known turbo codes [146, 198, 199].

The decoding algorithm for LDPC codes, the so-called BP working on the code Tanner graph, is an instance of the SP algorithm [79]. We now review the BP decoding algorithm. Let us denote by Λ a suitable *message* propagating on an edge of the code Tanner graph. This message represents the logarithmic likelihood ratio (LLR) related to the code bit corresponding to the variable node from which the considered edge originates. At the m-th iteration, we denote by $\Lambda^{(m,v\to c)}$ a message sent from a variable node to a check node and by $\Lambda^{(m,c\to v)}$ a message in the opposite direction. A variable node of degree d_v receives and processes the messages $\Lambda_i^{(m-1,c\to v)}$, $i = 1, \ldots, d_v$, and sends back to its j-th ($j = 1, \ldots, d_v$) neighboring check node the

message [32, 80]

$$\Lambda_j^{(m,v\to c)} = \Lambda^0 + \sum_{\substack{i=1 \\ i\neq j}}^{d_v} \Lambda_i^{(m-1,c\to v)} \tag{5.9}$$

where Λ^0 is the initial message received by the considered variable node as a function of the channel output corresponding to the considered code bit. When $m = 1$, the variable node simply propagates its initial received message Λ^0.

A check node of degree d_c receives and processes the messages $\Lambda_i^{(m,v\to c)}$, $i = 1, \ldots, d_c$, and sends back to its j-th ($j = 1, \ldots, d_c$) neighboring variable node the message [32, 80]

$$\Lambda_j^{(m,c\to v)} = 2 \tanh^{-1} \left\{ \prod_{\substack{i=1 \\ i\neq j}}^{d_c} \tanh \left\{ \frac{1}{2} \Lambda_i^{(m,v\to c)} \right\} \right\}. \tag{5.10}$$

It can be easily shown that the message computations (5.9) and (5.10) derive from (5.3) and (5.4) (see Problem 5.2). The decoding algorithm proceeds iteratively until the code parity check constraints are all verified or a maximum number of iterations is reached. Although this decoding algorithm is provably optimal for bipartite graphs without cycles [80], in practice it is necessary to only avoid cycles of length up to 4 to attain good performance [199].

In [200], in his PhD dissertation, Wiberg proposed the application of FGs beyond coding and the introduction of *hidden* variable nodes representing the states of the demodulator trellis. The resulting FGs, denoted as Wiberg graphs, do not have cycles. The FB algorithm described in Chapter 3 can be interpreted as an instance of the SP algorithm applied to these graphs with a natural forward backward schedule (see Problem 5.3). Wiberg graphs will not be further considered in this chapter since trellis-based MAP symbol detection was profusely analyzed in Chapter 4.

5.3 Finite Memory Graph-Based Detection

Without loss of generality, we assume that the code $\mathcal{C}$ admits an encoding function $\mu_{\mathcal{C}}$, mapping information sequences $\boldsymbol{a}$ into the codewords $\boldsymbol{c}$. As shown in Chapter 4, the optimal MAP symbol detection rule minimizing the average symbol-error probability is given by

$$\widehat{a}_k = \text{argmax} P\{a_k | \boldsymbol{r}\} \tag{5.11}$$

where $P\{a_k|\boldsymbol{r}\}$ denotes the *a posteriori* pmf for the k-th information symbol given the received signal vector $\boldsymbol{r} = (r_0, \ldots, r_{K-1})$. The application of the SP algorithm [79] to an FG representing the joint APP of the transmitted information sequence $\boldsymbol{a}$ conditioned on a given observation sequence $\boldsymbol{r}$, allows the exact or approximate computation of the marginal APPs $P\{a_k|\boldsymbol{r}\}$ [79]. Therefore, this algorithm may be used to perform MAP symbol detection. From the definition of the encoding function $\mu_{\mathfrak{C}}$, one obtains the following factorization[2]:

$$\begin{aligned} P\{\boldsymbol{a}|\boldsymbol{r}\} &\sim P\{\boldsymbol{a}\}p(\boldsymbol{r}|\boldsymbol{a}) \\ &\sim P\{\boldsymbol{a}\}\chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})]p(\boldsymbol{r}|\boldsymbol{c}) \\ &\sim \chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})] \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k)P\{a_k\} \end{aligned} \tag{5.12}$$

where (i) the causality condition (2.16), (ii) the fact that the output signal pdf $p(\boldsymbol{r})$ does not depend on $\boldsymbol{a}$, and (iii) the fact that the information symbols are independent, have been used. The notation $\chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})]$ indicates the code indicator function, equal to 1 if $\boldsymbol{c}$ is the codeword corresponding to $\boldsymbol{a}$ and to zero otherwise.

If the probability density function $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k)$, which appears in (5.12), satisfies the FMC (2.17) introduced in Chapter 2, one can write that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k) = p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_{k-C}^k) \tag{5.13}$$

where C is the *finite memory parameter*. Substituting (5.13) into (5.12), the information sequence joint *a posteriori* pmf $P\{\boldsymbol{a}|\boldsymbol{r}\}$ may be expressed as

$$P\{\boldsymbol{a}|\boldsymbol{r}\} \sim \chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})] \prod_{k=0}^{K-1} p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_{k-C}^k)P\{a_k\}. \tag{5.14}$$

The corresponding FG is shown in Figure 5.3 for $C = 2$, and represents both the code constraints (described by $\chi(\boldsymbol{c})$) and the channel behavior. In general, the code constraint function can be also represented by means of an FG. This graph is a portion of the overall graph and is connected with the graph representing the channel behavior. With respect to SP-based decoding schemes for LDPC codes over a memoryless channel, *additional factor nodes* must be added at the bottom of the graph, as shown in Figure 5.3. These additional factor nodes perform a marginalization, based on the channel model, without taking into account the code constraints. The complexity of

[2]In this chapter, we use extensively the proportionality relationship $f \sim g$, indicating that $f = ag$ for some real constant a, since the SP algorithm is defi ned up to scaling its messages by positive factors, independent of the variables represented in the graph.

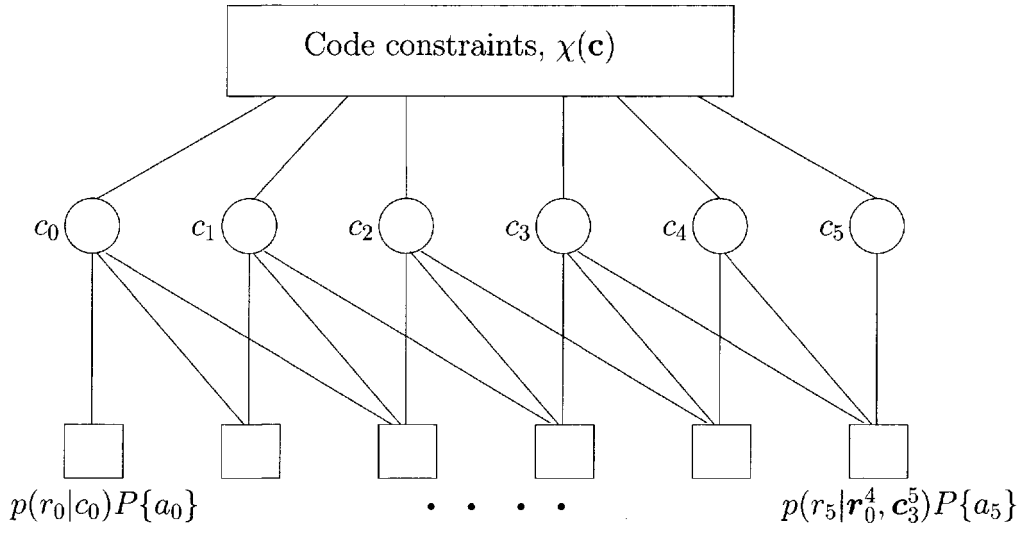


Figure 5.3: Factor graph corresponding to the factorization (5.14) for $C = 2$. Reproduced from [201], ©2004 IEEE, by permission of the IEEE.

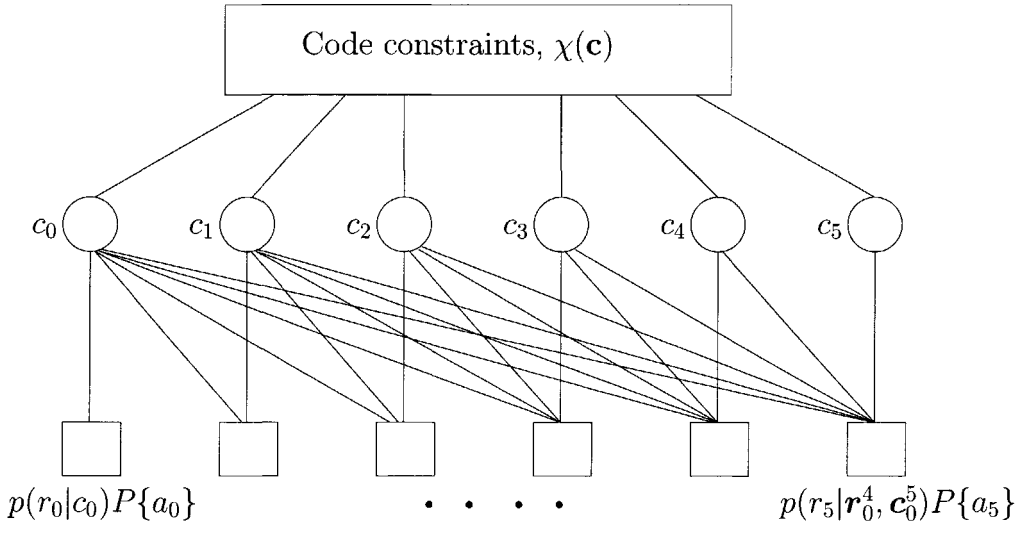


Figure 5.4: Factor graph corresponding to the factorization (5.12).

this marginalization is, in general, exponential in C. Note that in the derived FG, the channel parameters *are not* explicitly represented since they are *a priori* averaged out.

The FMC (5.13) is exactly verified in the case of channels with known ISI. In

fact, in this case it holds that

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k) = p(r_k|\boldsymbol{c}_{k-L}^k) \tag{5.15}$$

where L is the length of the discrete-time channel impulse response. If the FMC (5.13) is not verified in an exact sense, as for a noncoherent or a fading channel (channels with *infinite* memory), an FG may still be devised. In other words, we may build the FG corresponding to the factorization (5.12). This FG is shown in Figure 5.4. However, the complexity of the message computation at the generic factor node $p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k)P\{a_k\}$ would grow exponentially with k and thus become impractical. Assuming that the CMP introduced in Chapter 2 holds, then one can write:

$$p(r_k|\boldsymbol{r}_0^{k-1}, \boldsymbol{c}_0^k) = p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{c}_{k-C}^k) \tag{5.16}$$

where N denotes the order of Markovianity and is such that $N \leq C$. This property, in general adopted in all practical detection schemes, is intuitive in the case of time-varying channels. In fact, in this case the conditional observations are asymptotically independent for increasing epoch difference. Hence, although an approximation, (5.16) can be applied in many practical cases with little discrepancy from the non-truncated version. The resulting (approximate) expression of $P\{\boldsymbol{a}|\boldsymbol{r}\}$ becomes

$$P\{\boldsymbol{a}|\boldsymbol{r}\} \sim \chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})] \prod_{k=0}^{K-1} \gamma_k(\boldsymbol{c}_{k-C}^k) \tag{5.17}$$

where

$$\gamma_k(\boldsymbol{c}_{k-C}^k) \triangleq p(r_k|\boldsymbol{r}_{k-N}^{k-1}, \boldsymbol{c}_{k-C}^k)P\{a_k\}. \tag{5.18}$$

The quantity $\gamma_k(\boldsymbol{c}_{k-C}^k)$ is the equivalent of the basic exponential metric $\gamma_k(T_k)$ introduced in Chapter 2, where $T_k = (S_k, a_k)$ is a transition in a suitable trellis diagram. In this case, there is no need to explicitly defined a state[3] S_k, and we simply indicate the sequence of consecutive coded symbols. The FG associated with (5.17) is shown in Figure 5.5 for $C = 2$ (for simplicity, the arguments of $\{\gamma_i\}$ are omitted).

The function (5.18) associated with the generic factor node modeling the channel is the same basic exponential metric used in the Viterbi algorithm when MAP sequence detection is applied or in the FB algorithm implementing MAP symbol detection (see the previous chapters). This fact suggests that all the solutions previously proposed for Viterbi and FB algorithms simply extend to graphical models. An

[3]The trellis diagram associated with a linear block code is usually time-variant [54], so that trellis-based detection would be more complicated than graph-based detection with the SP algorithm. In particular, the set of possible states could be time-dependent.

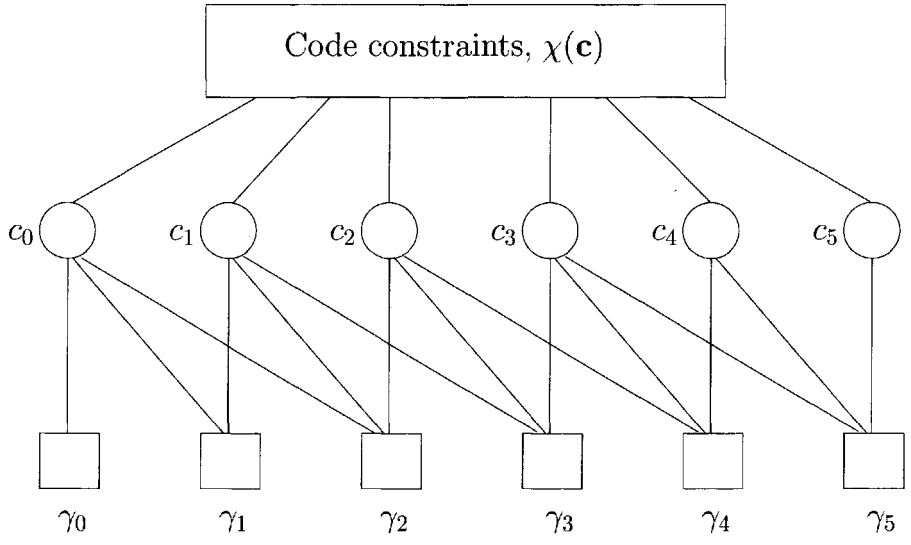


Figure 5.5: Factor graph corresponding to the factorization (5.17) for $C = 2$. Reproduced from [201], ©2004 IEEE, by permission of the IEEE.

important application of the proposed class of graph-based detection algorithms is related to stochastic channels. As in Section 2.3.2, we consider the following general parametric model:

$$r_k = g(\boldsymbol{a}_{k-L}^k, \mu_{k-L}, \boldsymbol{\theta}_0^k) + n_k \tag{5.19}$$

where L is an integer, $\boldsymbol{\theta}_0^k$ is a sequence of stochastic parameters independent from $\boldsymbol{a}$, and n_k is an additive noise sample. As shown in Chapter 2, application of a CMP with order of Markovianity equal to N is sufficient to guarantee an FMC where finite memory parameter $C = N + L$. In this case, the function $\gamma_k(\boldsymbol{c}_{k-C}^k)$ which appears in (5.17) may be computed as

$$\gamma_k(\boldsymbol{c}_{k-C}^k) = \frac{\mathrm{E}_{\boldsymbol{\theta}_0^k}\{p(\boldsymbol{r}_{k-N}^k|\boldsymbol{c}_{k-C}^k, \boldsymbol{\theta}_0^k)\}}{\mathrm{E}_{\boldsymbol{\theta}_0^{k-1}}\{p(\boldsymbol{r}_{k-N}^{k-1}|\boldsymbol{c}_{k-C}^{k-1}, \boldsymbol{\theta}_0^{k-1})\}} P\{a_k\}. \tag{5.20}$$

The quality of the convergence of the SP algorithm to the exact marginal probabilities is, in general, determined by the girth of the graph. As an example, in designing LDPC codes, cycles of length 4 must be avoided to ensure decoding convergence. The FG derived from the proposed factorization has, in general, girth 4 (as one can see in Figure 5.5). However, we verified by computer simulations that these length-4

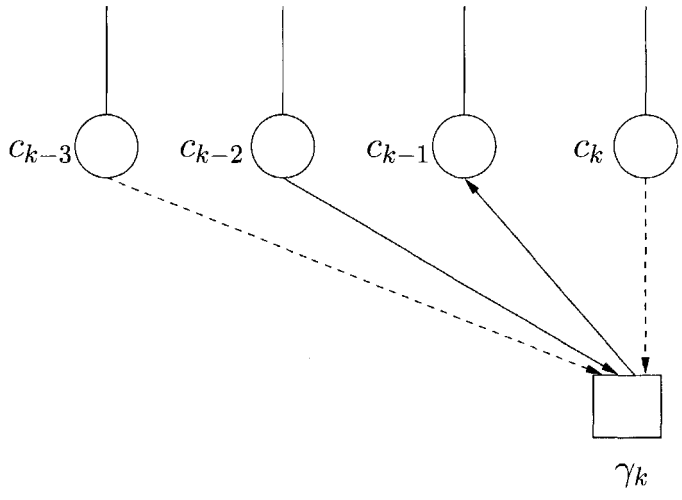


Figure 5.6: Complexity reduction in graph-based detection. The messages in some of the incoming branches (dashed lines) are hard-quantized, so only the remaining branches carry soft information to be used. The considered case corresponds to $C = 3$ and $Q = 1$.

cycles involving two factor nodes which model the channel behavior often do not affect the convergence of the algorithm. If this is not the case, as for transmissions over ISI channels, FG transformations can be adopted, as will be shown in Section 5.5.

5.4 Complexity Reduction for Graph-Based Detection Algorithms

The complexity of the proposed graph-based detection algorithms may be reduced by applying techniques similar to those described in Chapter 3 for VA-based detection algorithms or in Chapter 4 for FB algorithms. In fact, by choosing an integer $Q < C$, the updating rule (5.4) at factor nodes modeling the channel can be simplified as follows: the $C - Q$ symbols with highest reliability are hard-quantized based on the messages on the graph, and the sum is performed over the Q symbols with lowest reliability.[4] In this way, the complexity becomes exponential in Q. With respect to VA and FB algorithm, the complexity of graph-based detection algorithms can be reduced more efficiently. In fact, symbols with the highest reliability can be

[4] A partial representation by using set partitioning can be also adopted.

hard-quantized regardless of their position. On the contrary, when the VA or the FB algorithm is used, the constraints imposed by the trellis structure are more stringent (see Chapter 3 and Chapter 4 for more details). Figure 5.6 shows a graphical example. A channel factor node (square) is associated with $C+1$ variable nodes (circles). The messages in $C-Q=2$ of the incoming branches (dashed lines) are hard-quantized and only $Q=1$ incoming branch (solid line) carries soft information.

For equal-energy signals, a modified version of the described FG for noncoherent and flat fading channels may be devised. In fact, in Section 5.6.3 we will show that for fading channels the function $p(\boldsymbol{r}|\boldsymbol{c})$ can be factorized into the product of functions of two code symbols. For noncoherent channels this factorization is not exact but involves a simple approximation. The SP algorithm on these modified graphs has a complexity *linear* in C, allowing a low-complexity receiver implementation for all practical values of C: in other words, no complexity reduction is needed.

5.5 Strictly Finite Memory: Inter-Symbol Interference Channels

In this section, we consider graph-based detection for known ISI channels [202]. By representing on an FG the joint APP of the transmitted symbols and applying to this graph the SP algorithm, we derive a simple soft-input soft-output (SISO) algorithm that can be used for turbo equalization [73]. The same approach and the resulting detection algorithm is also considered in [203]. However, from the observation that in general the FG describing the channel has cycles, the authors of [203] conclude that the derived algorithm is suboptimal and therefore they discard this approach. Unlike [203], it is possible to verify by computer simulations that when the girth of the graph is at least 6, the performance is practically optimal. In addition, for graphs of girth 4, we apply the SP algorithm to a new FG obtained by transforming the original one. The resulting new algorithm has a negligible performance loss with respect to optimal detection.

The proposed algorithms have a complexity which depends on the number of non-zero interferers only. As a consequence, with respect to the FB algorithm, whose complexity depends on the channel memory, the proposed algorithms are more suited for *sparse* ISI channels, i.e., channels characterized by a large memory but a small number of interferers, as may occur in high frequency transmissions due to multi-path [204]. With respect to other algorithms devised for sparse channels [204, 205], the proposed graph-based detection algorithms are more general.

Another advantage with respect to trellis-based FB algorithms consists of the

intrinsic parallel structure of the SP algorithm with *flooding* schedule, allowing very high-speed detection. This aspect is very important when an LDPC code is used and turbo detection is performed. In fact, if detection is performed by using the proposed algorithms, an *overall* graph taking into account both the code and the channel model can be built, allowing combined detection and decoding in a fully parallel manner without a complexity increase with respect to separate detection and decoding. On the contrary, when the BCJR algorithm is used for detection, combined detection and decoding is not possible and the serial structure of the BCJR algorithm prevents the possibility of a parallel implementation of the receiver.

In the technical literature relative to applications of LDPC codes for magnetic recording, the FB algorithm is often considered for soft-output detection relative to the single channel [206–210]. In order to overcome intrinsic speed limitations of the FB algorithm, several schemes have been proposed in which the FB algorithm is run only once or one time each K_I iterations of the LDPC decoder [208–210], although it has been shown that the best schedule is obtained for $K_I = 1$ [203]. In order to allow parallel joint detection and decoding, in [203] a version of the BCJR algorithm with parallel schedule has been proposed. A very fast implementation of the BCJR algorithm, based on the construction of a suitable tree structure, is proposed in [211]. As already mentioned, the graph-based detection schemes considered in the following represent an alternative solution to these problems.

Finally, as described in Section 5.4, the complexity of the proposed graph-based detection algorithms can be reduced more efficiently. In fact, the interfering symbols with highest reliability can be truncated regardless of their position.

5.5.1 Factor Graph

For an ISI channel, the observation at the output of a whitened matched filter (WMF) [100] can be expressed as (see also Chapter 3 and Chapter 4):

$$r_k = \sum_{\ell=0}^{L} f_\ell c_{k-\ell} + n_k \tag{5.21}$$

in which $\{n_k\}$ are independent complex Gaussian random variables with zero-mean and variance σ^2 per component, L is the channel memory, and $\boldsymbol{f} = \{f_\ell\}_{\ell=0}^{L}$ represents the equivalent discrete-time channel impulse response, assumed perfectly known at the receiver. The joint APP distribution of the transmitted symbols may be expressed as

$$P\{\boldsymbol{a}|\boldsymbol{r}\} \sim \chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})] \prod_{k=0}^{K-1} \gamma_k(\boldsymbol{c}_{k-L}^{k}) \tag{5.22}$$

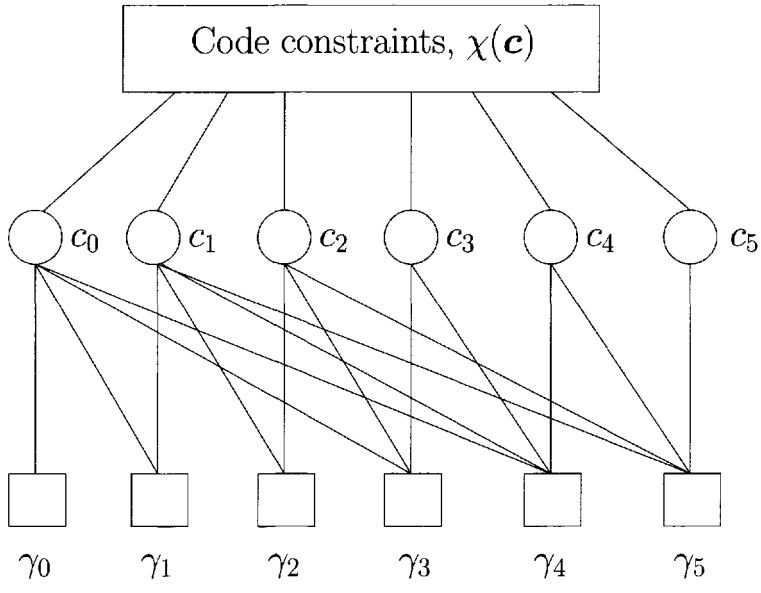


Figure 5.7: Factor graph for an ISI channel with $L = 4$ and $f_2 = 0$.

where

$$\gamma_k(\boldsymbol{c}_{k-L}^k) = p(r_k|\boldsymbol{c}_{k-L}^k)P\{a_k\} \sim \exp\left(-\frac{|r_k - \sum_{\ell=0}^{L} f_\ell c_{k-\ell}|^2}{2\sigma^2}\right) P\{a_k\}. \quad (5.23)$$

For an ISI channel with $C = L = 4$ and $f_2 = 0$, the FG corresponding to the global function given in (5.17) is shown in Figure 5.7. The graph representing the ISI channel may have cycles. Hence, the application of the SP algorithm to this graph leads to an iterative detection process. Since we are interested in the performance analysis of the proposed graph-based detection algorithm, we limit ourselves to the case of absence of coding. Hence, symbols $\{c_k\}$ can be assumed independent and identically distributed and the upper part of the graph in Figure 5.7, representing the code constraints, is not present.

As already mentioned, the graph representing an ISI channel may have cycles. Let us introduce another parameter that is relevant for an ISI channel: the number J of non-zero interferers. When $J \ll L$, the channel is called *sparse*. The following three possible cases can be distinguished.

1. If there is only one interferer ($J = 1$), the graph is cycle-free.[5] In a cycle-free graph, the SP algorithm computes exactly the APP for each variable, regardless

[5]In this case, if $L > 1 = J$, the original graph is composed of L independent subgraphs [205] on which L independent instances of the algorithm can be run.

of the considered schedule [212]. The use of a flooding schedule, rather than a serial schedule starting from leaf vertices and with a natural termination [79], has the advantage of allowing parallel detection at the expense of a complexity increase, due to the need for iterative processing.

2. If the differences between the indexes of the channel interferers are all different, the graph has girth 6.

3. If the previous conditions are not verified, the graph has girth 4.

In cases 2. and 3., the SP algorithm is approximate. However, as verified through extensive computer simulations, if the girth of the graph is 6 and the channel is minimum- or maximum-phase, after a few iterations the algorithm always converges to the performance of the optimal FB algorithm. For mixed-phase channels, we verified that convergence is obtained when there is only a dominant interferer. However, a mixed-phase channel can always be converted into an equivalent minimum- or maximum-phase channel. For graphs with girth 4, the proposed algorithm usually does not converge. However, as will be shown in Section 5.5.2, the original girth-4 graph may be transformed into an equivalent girth-6 graph on which the SP algorithm converges. In addition, we will see that the presence of a powerful channel code often helps the convergence even in the presence of cycles of length 4 in the part of the graph describing the channel behavior.

Since the most demanding computation is that performed at factor nodes (compare the updating rules (5.5) and (5.6)), we may define a cost per coded symbol of the considered algorithms, indicated as cost_s as related to the above mentioned *summary* operations (see (5.6)). Hence, we may say that for the proposed algorithm $\text{cost}_\text{s}^{prop} = I(J+1)M^{J+1}$, where I is the number of iterations performed. In fact, for the computation of the M values of a message sent on one of the $J+1$ edges coming out from a factor node, a *summary* operation involving M^J terms is needed. Note that the same unit can be used to measure the complexity of the BCJR algorithm. In fact, as already mentioned in Section 5.2, this algorithm can be viewed as the application of the SP algorithm to the cycle-free Wiberg graph of the channel [200]. Hence, functions associated with factor nodes are of the same form, and a complexity comparison is fair. For the BCJR algorithm, the cost per coded symbol is $\text{cost}_\text{s}^{\text{BCJR}} = 3M^{L+1}$. In fact, for the forward (backward) recursion, the computation of the M^L values of the propagating message requires a *summary* operation involving M terms, whereas the computation of the M values of the marginal APP of the k-th symbol requires a *summary* operation involving M^L terms. As may be observed comparing $\text{cost}_\text{s}^{\text{prop}}$ with $\text{cost}_\text{s}^{\text{BCJR}}$, the proposed algorithm is a valid alternative to the BCJR for sparse ISI channels or when parallel detection is preferred.

The described algorithm is similar to that, called *bit-based message passing*, developed in [203] for partial response channels. However, the authors of [203] observed that the graph has cycles and therefore opted for the *state-based message passing* which is the SP algorithm with a parallel schedule applied to the cycle-free Wiberg graph [200] of the channel, that is a BCJR algorithm with parallel schedule. This algorithm, which will be denoted in the following as p-BCJR, has a cost given by $\text{cost}_{\text{s}}^{\text{p-BCJR}} = 3IM^{L+1}$.

The complexity of the proposed algorithm may be reduced following the technique described in Section 5.4. However, in this case we have the further alternative of computing (5.6) summing over the Q interfering symbols with strongest weights and substituting the remaining $J - Q$ interfering symbols with a decision made on the basis of the messages on the graph. In both cases, the cost becomes $\text{cost}_{\text{s}}^{\text{red}} = I(Q+1)M^{Q+1}$.

The performance of the proposed detection schemes is assessed by computer simulations in terms of bit error rate (BER) versus E_b/N_0, E_b being the received signal energy per information bit and N_0 the one-sided noise power spectral density. The performance of the BCJR algorithm for the same channel is given as a benchmark. Although formally incorrect, since BCJR and p-BCJR algorithms are also instances of the SP algorithm, the BER curves corresponding to the proposed algorithm are labeled with SP in the following figures.

In Figure 5.8, we consider a sparse ISI channel characterized by $L = 5, J = 2$, and the following discrete-time impulse response:

$$\boldsymbol{f} = (0.408, 0, 0, 0, 0.816, 0.408).$$

The FG corresponding to the channel has girth 6. In fact, the set of differences between the indexes of the channel interferers is $\{1, 4, 5\}$ and they are all different (see condition 2.). Uncoded binary phase-shift keying (BPSK) and 8-PSK modulations are considered. It can be observed that the performance of the proposed algorithm converges in 5 iterations to the optimal performance of the BCJR for both modulation formats. In addition, one can note the dramatic performance improvement from the first and the third iteration. Similar considerations also hold for multiamplitude modulation formats such as quadrature amplitude modulation (QAM) schemes.

In Figure 5.9, the impact of the application of complexity reduction techniques to the BCJR and the proposed algorithms is analyzed. The considered girth-6 minimum-phase channel is characterized by $L = 5, J = 2$, and the following coefficients:

$$\boldsymbol{f} = (0.83, 0, 0, 0, 0.25, 0.5).$$

Also in this case, an uncoded BPSK modulation is adopted. A reduced-state 16-state BCJR algorithm can be obtained by applying the state reduction techniques consid-

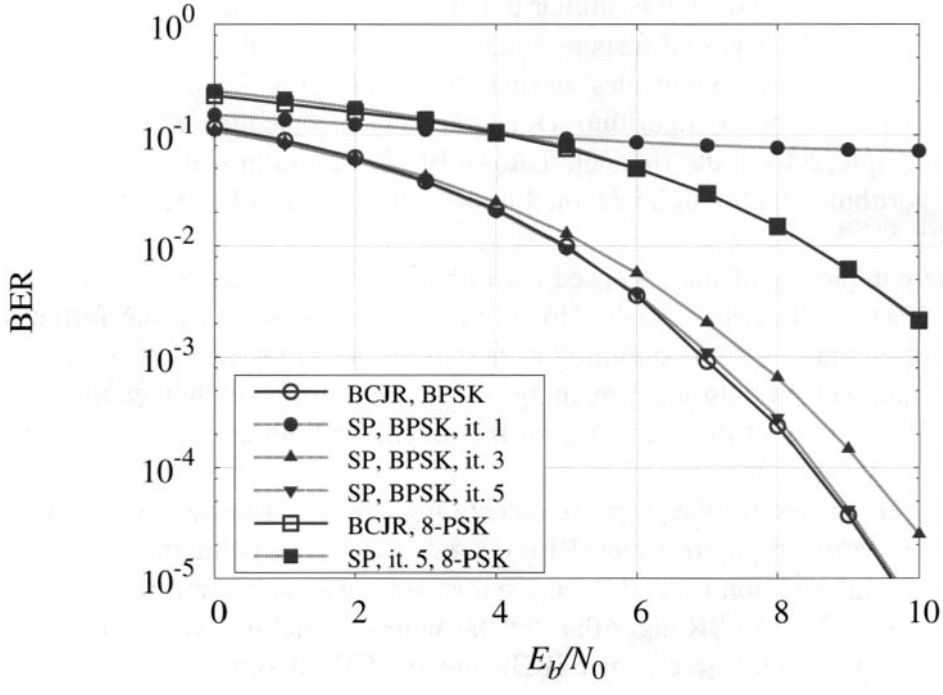


Figure 5.8: Performance for a sparse ISI channel.

ered in Chapter 4. In particular, one can define a reduced state as $(a_{n-1}, a_{n-2}, a_{n-3}, a_{n-4})$ and recover the symbol a_{n-5} in the survivor history. Hence, the only possibility we have is the truncation, in the trellis definition, of the farthest symbol. The proposed graph-based detection algorithm, instead, is not constrained by a trellis structure and the interfering symbol with lowest weight may therefore be truncated, regardless of its position (for the ISI channel in this example, symbol a_{n-4}). In the computation of the messages at factor nodes, this symbol can be substituted with a decision made on the basis of the messages on the graph. The advantage with respect to a reduced-state BCJR algorithm, in terms of signal-to-noise ratio required for a BER equal to 10^{-4}, is about 1 dB, as shown in Figure 5.9.

5.5.2 Modified Graph

In [79], an FG transformation, called *stretching*, is introduced to obtain a cycle-free graph. Denoting by $\mathcal{N}_2(x_i)$ the set of variable nodes that can be reached from node x_i through a path of length 2, one can replace any node $x_\ell \in \mathcal{N}_2(x_i)$ with a node representing the pair (x_i, x_ℓ). In this way, it is possible for an edge (or a variable

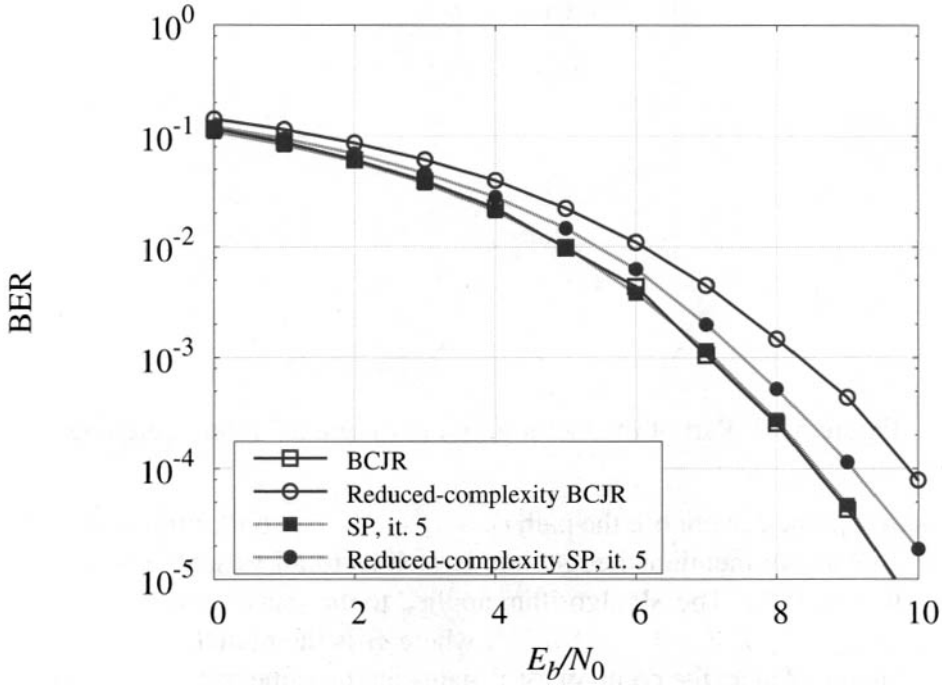


Figure 5.9: Performance in the case of complexity reduction.

node) to become redundant. Hence, it can be removed "without damage."

Our goal is to remove a minimum number of edges so that the girth of the transformed graph becomes 6 instead of 4, and the complexity of the resulting SP algorithm remains the same. As a consequence, we adopt the following stretching rule.

- Given an edge e connecting a variable node x_i to a factor node f_j, in order to preserve the information lost by cutting this edge, we choose an arbitrary path $\mathcal{P}(x_i, f_j)$ connecting x_i and f_j and not involving e and stretch the variable node x_i to all variable nodes $x_\ell \in \mathcal{P}(x_i, f_j)$.

Obviously, for a given girth-4 graph, there are different choices of edges such that their removal can lead to a girth-6 graph. However, we verified that the application of the SP algorithm to the resulting girth-6 graphs gives similar performance, even if the minimum required number of iterations can be different, depending on the weight of the interfering symbol whose corresponding edge has been removed.

An example of a girth-6 graph, obtained by transforming the girth-4 graph in Figure 5.7, is shown in Figure 5.10 (considering the relevant part describing only the channel behavior). In order to remove the edge connecting the variable node c_0 to

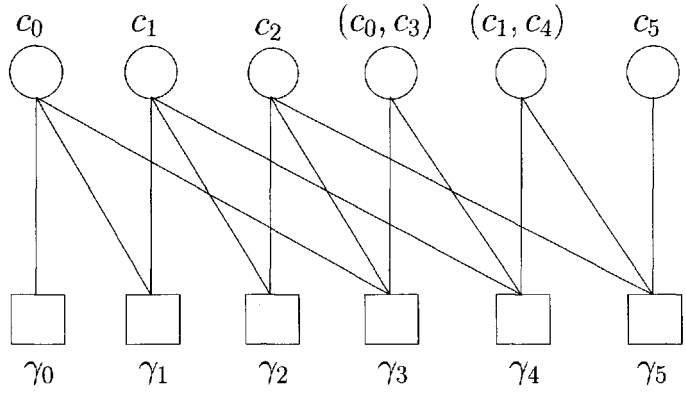


Figure 5.10: Part of the factor graph in Figure 5.7 after stretching.

factor node γ_4, one can choose the path $c_0 - \gamma_3 - c_3 - \gamma_4$ connecting c_0 to γ_4 and not involving the above mentioned edge, and stretch c_0 to all variable nodes belonging to this path, i.e., to c_3. The SP algorithm applied to the transformed graph has a cost given by $\text{cost}_{\text{s}}^{\text{mod}} = I(J+1-R)M^{J+1}$, where R is the number of edges removed per factor node. Hence, the complexity remains on the same order of magnitude.[6]

A further application of the described stretching rule to obtain a cycle-free graph leads to a sort of Wiberg graph of the channel—the SP algorithm applied to this graph becomes the state-based message passing algorithm described in [203]. We do not further pursue this approach since it gives an unnecessary increase in complexity.

As an example, we now consider the so-called $\text{E}^2\text{PR4}$ channel encountered in magnetic storage systems [213,214], i.e., an ISI channel with $L = 4$, $J = 3$, and

$$\boldsymbol{f} = \frac{1}{\sqrt{10}}(1, 2, 0, -2, -1).$$

An uncoded binary pulse amplitude modulation (PAM) with symbols $a_k \in \{\pm 1\}$ is considered. The corresponding graph, shown in the lower part of Figure 5.7, has girth 4. In this case, the SP algorithm, if directly applied to this graph, does not converge to the performance of the BCJR algorithm, as shown in Figure 5.11, where a loss of about 1 dB is observed, even for a large number of iterations. Therefore, a graph transformation is expedient. By using the described stretching technique, we obtain the graph in Figure 5.10. When run over this graph, the SP algorithm converges in about 10 iterations to the optimal performance. In Figure 5.11, the performance of

[6]We are neglecting the increase in computational complexity at variable nodes related to the need for a marginalization of the messages associated with the pair of symbols, since this computation is limited with respect to that associated with the factor nodes.

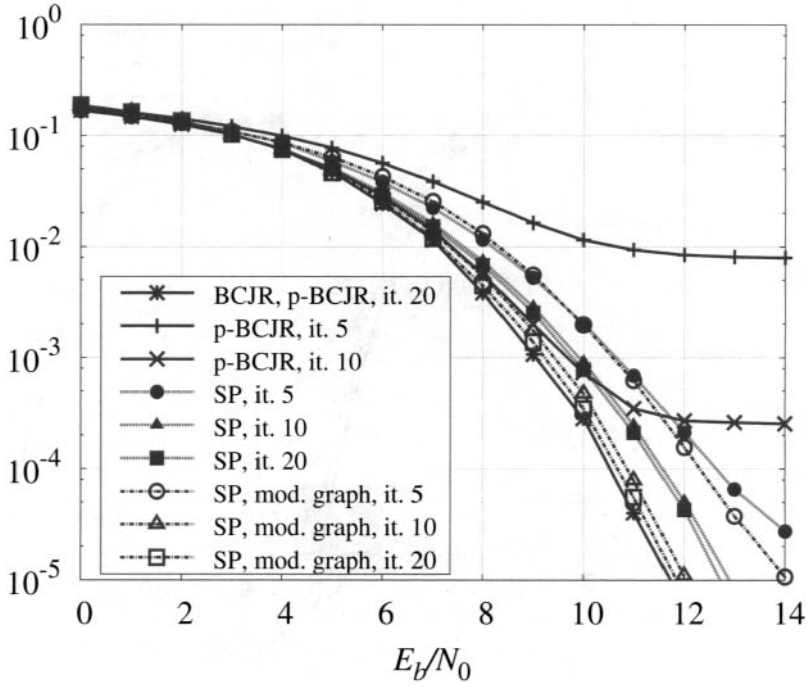


Figure 5.11: Performance of the SP algorithm on a modified graph.

the p-BCJR algorithm is also shown. We may observe that, even if for a sufficiently large number of iterations this algorithm has an optimal performance, at least in the considered BER range, for 5 and 10 iterations it performs worse than the SP algorithm and exhibits an error floor. This floor has been already observed in [203], where a precoder has been introduced to overcome this problem. Its presence is due to the structure of the FGs. In fact, in the graphs of Figure 5.7 and Figure 5.10, the message relative to a variable node rapidly propagates to other nodes, whereas in the Wiberg graph this propagation, in the case of a parallel schedule, is slower.

We now consider the application of the derived algorithms to combined detection and decoding (the so-called "turbo equalization") in the presence of an LDPC code. Due to the structure of an LDPC code, an interleaver is not necessary. The considered channel is still the E^2PR4 magnetic channel and the modulation format is the above mentioned binary PAM. Two LDPC codes are considered, namely (i) a regular (3,6) code of rate 0.5 and codeword length 4000 and (ii) an irregular code of rate 0.82 and codeword length 4095 [215]. Considering the SP algorithm on the original or the modified graph, a maximum number of 150 iterations is allowed. When detection is performed considering an inner block using the BCJR algorithm relative to an ISI

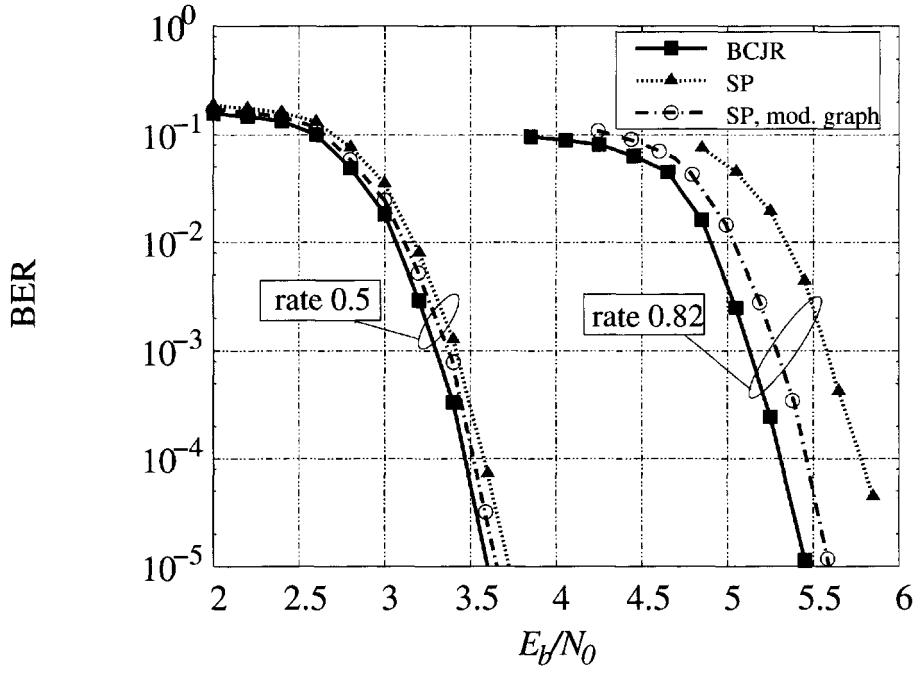


Figure 5.12: BER in the case of application to LDPC codes.

channel, an inner detection step is executed at each outer LDPC decoder iteration (for a maximum allowed number of iterations of 150). This choice has been shown to be optimal in [203]. The BER performance, shown in Figure 5.12, demonstrates that, in the case of a powerful channel code, the SP algorithm guarantees optimal performance even on the original girth-4 graph. For the rate-0.82 LDPC code, the SP algorithm run over the original graph exhibits a performance loss of about 0.5 dB, whereas this loss is reduced to about 0.15 dB for the SP algorithm working on the modified graph. Hence, side information provided in the iterative detection/decoding process by the decoder of a powerful code helps the detector convergence, even in the presence of cycles of length 4 in the part of the graph describing the ISI channel.

5.6 Applications to Wireless Communications

In this section, we consider the application of the described framework to the case of noncoherent and flat correlated Rayleigh fading channels [201].

5.6.1 Noncoherent Graph-Based Detection

The channel phase is assumed to be a time-invariant random variable θ with uniform distribution in $[0, 2\pi)$. However, application of the CMP leads to a detection algorithm that can be used for slowly varying channels also. In this case, the order of Markovianity N coincides with the finite memory parameter C and the basic exponential metric $\gamma_k(\boldsymbol{c}_{k-C}^k)$ which appears in (5.17) can be expressed, based on (5.20), as

$$\gamma_k(\boldsymbol{c}_{k-C}^k) = \frac{\mathrm{E}_\theta\{p(\boldsymbol{r}_{k-C}^k|\boldsymbol{c}_{k-C}^k,\theta)\}}{\mathrm{E}_\theta\{p(\boldsymbol{r}_{k-C}^{k-1}|\boldsymbol{c}_{k-C}^{k-1},\theta)\}} P\{a_k\} \sim \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{C} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right|\right)} e^{-\frac{|c_k|^2}{2\sigma^2}} P\{a_k\} \tag{5.24}$$

where $\mathrm{I}_0(x)$ is the zero-th order modified Bessel function of the first kind. Unless complexity reduction techniques are considered, such as those introduced in Section 5.4, at the corresponding factor node the SP algorithm performs a marginalization whose computational burden grows exponentially with C.

In Figure 5.13, the performance of the described detection algorithm, for different values of C, is shown in the case of transmission of a (3,6)-regular LDPC code with codewords of length 4000 [215]. A comparison with the performance of the ideal coherent receiver is also shown (curve labeled "perfect channel state information (CSI)"). The considered modulation format is BPSK and the maximum allowed number of decoding iterations of the SP algorithm, with *flooding* schedule, is 200. A pilot symbol every 19 code bits is added for ambiguity problems. This corresponds to a decrease in the effective transmission rate, resulting in an increase in the required signal-to-noise ratio of about 0.223 dB which has been arbitrarily introduced in the curve labeled "perfect CSI" for the sake of comparison. Hence, the gap between the "perfect CSI" curve and the others is uniquely due to the need for phase estimation/compensation, and not to the rate decrease due to pilot symbols insertion. Complexity reduction techniques are also applied to increase the phase memory C with no complexity increase. Considering $Q = 1$ (i.e., fixing a low complexity level), the performance of the ideal coherent receiver is approached for increasing values of C.

5.6.2 Linear Predictive Graph-Based Detection for Phase-Uncertain Channels

For a general time-varying phase process θ_k, assumed stationary, zero-mean and described by a given autocorrelation sequence of the phasor process $\{e^{j\theta_k}\}$, denoted by

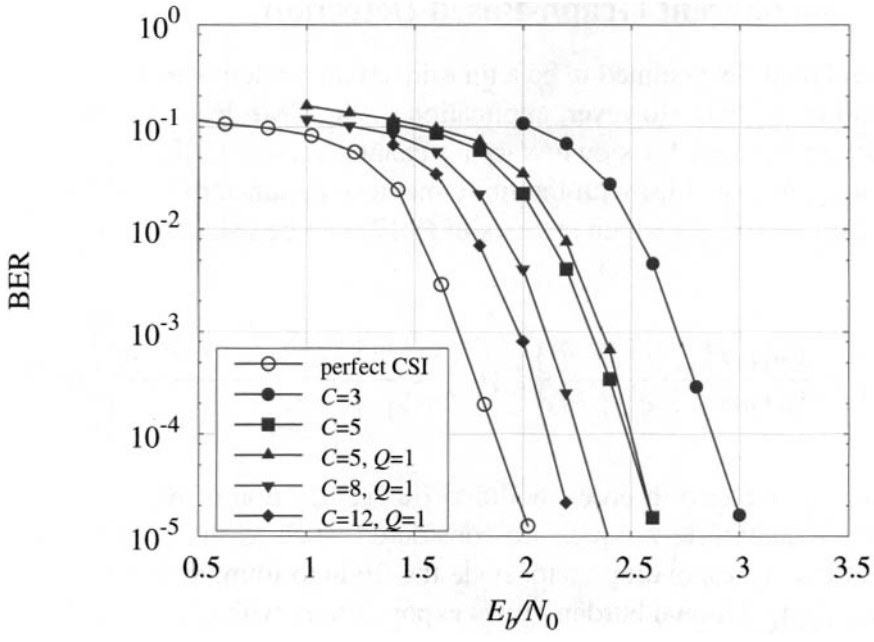


Figure 5.13: BER performance of an LDPC code transmitted over a noncoherent channel. Reproduced from [201], ©2004 IEEE, by permission of the IEEE.

$R_\theta(n) = \mathrm{E}\{e^{j\theta_{n+k}} e^{-j\theta_k}\}$, the linear predictive approach described in Section 3.11.4 for VA-based approximate MAP sequence detection and in Section 4.9.4 for approximate MAP symbol detection based on FB algorithms can be adopted. In this case, the probability density function $\gamma_k(\boldsymbol{c}_{k-C}^k)$ may be approximated, omitting irrelevant constant terms, as

$$\gamma_k(\boldsymbol{c}_{k-C}^k) \simeq \exp\left(-\frac{1}{\sigma_e^2}\left|r_k - c_k \frac{\sum_{i=1}^C p_i \frac{r_{k-i}}{c_{k-i}}}{\left|\sum_{i=1}^C p_i \frac{r_{k-i}}{c_{k-i}}\right|}\right|^2\right) P\{a_k\} \tag{5.25}$$

where, in this case, C corresponds to the *prediction order*, $\{p_i\}_{i=1}^C$ are the prediction coefficients and σ_e^2 is the mean square prediction error. The prediction coefficients $\{p_i\}_{i=1}^C$ can be computed by solving a Wiener Hopf linear system $\boldsymbol{R}\boldsymbol{p} = \boldsymbol{b}$, where $\boldsymbol{R}$ is a square $C \times C$ matrix whose elements have the following expression

$$[\boldsymbol{R}]_{\ell,m} = \begin{cases} R_h(|\ell - m|) & \text{if} \quad \ell \neq m \\ R_h(0) + \frac{2\sigma^2}{|c_{k-\ell}|^2} & \text{if} \quad \ell = m \end{cases} \tag{5.26}$$

$\boldsymbol{p} \triangleq [p_1 \cdots p_C]^{\mathrm{T}}$ is the unknown vector, $\boldsymbol{b} = [R_h(1), R_h(2), \cdots, R_h(C)]^{\mathrm{T}}$, and $2\sigma^2$ is the variance of the discrete-time complex AWGN samples. The mean square

prediction error may be expressed as (see Section 3.11.4)

$$\sigma_e^2 = R_\theta(0) + \frac{2\sigma^2}{|c_k|^2} - \sum_{i=1}^{C} p_i R_\theta(i). \tag{5.27}$$

For PSK signals, the prediction coefficients and the mean square prediction error become independent of the considered sequence. In addition, approximating[7] $|\sum_{i=1}^{C} p_i \frac{r_{k-i}}{c_{k-i}}| \simeq |\sum_{i=1}^{C} p_i|$, and taking into account that $|c_k| = 1$, one obtains

$$\begin{aligned}
\gamma_k(\boldsymbol{c}_{k-C}^k) &\sim \exp\left(\frac{2}{\sigma_e^2|\sum_{i=1}^{C} p_i|}\Re\left\{r_k c_k^* \sum_{i=1}^{C} p_i r_{k-i}^* c_{k-i}\right\}\right) P\{a_k\} \\
&= \prod_{i=1}^{C} \exp\left(\frac{2\Re\left\{p_i r_k c_k^* r_{k-i}^* c_{k-i}\right\}}{\sigma_e^2|\sum_{i=1}^{C} p_i|}\right) P\{a_k\} \\
&= \prod_{i=1}^{C} g_{k-i,k}(c_{k-i}, c_k)
\end{aligned} \tag{5.28}$$

where

$$g_{k-i,k}(c_{k-i}, c_k) \triangleq P\{a_k\} \exp\left(\frac{2\Re\left\{p_i r_k c_k^* r_{k-i}^* c_{k-i}\right\}}{\sigma_e^2|\sum_{i=1}^{C} p_i|}\right). \tag{5.29}$$

This further factorization has a direct impact on the graph structure. In fact, each factor node can be decomposed into C simpler degree-2 factor nodes. As an example, for $C = 2$, the corresponding FG is shown in Figure 5.14 (for brevity, the arguments of functions $g_{k-i,k}(c_{k-i}, c_k)$ are omitted). Hence, for increasing values of C, the number of factor nodes increases linearly but the computational burden at each factor node remains the same. In addition, in this modified FG there are no cycles of length 4 in the part of the graph modeling the channel. This approach can also be used when the phase is time-invariant. In this case, as shown in Section 3.11.4 $p_i = 1/C$ and $\sigma_e^2 = 2\sigma^2$.

In Figure 5.15, we compare the performance of this algorithm, denoted as "linear predictive," with that of the algorithm described in Section 5.6.1, denoted as "noncoherent," in the case of a time-varying channel. The same code and modulation format as in Figure 5.13 are adopted. The phase noise is modeled as a discrete-time Wiener process with incremental variance over a signaling interval equal to σ_Δ^2. The performance of the algorithms, with the corresponding values of C optimized by computer

[7]This approximation is valid for large signal-to-noise ratios.

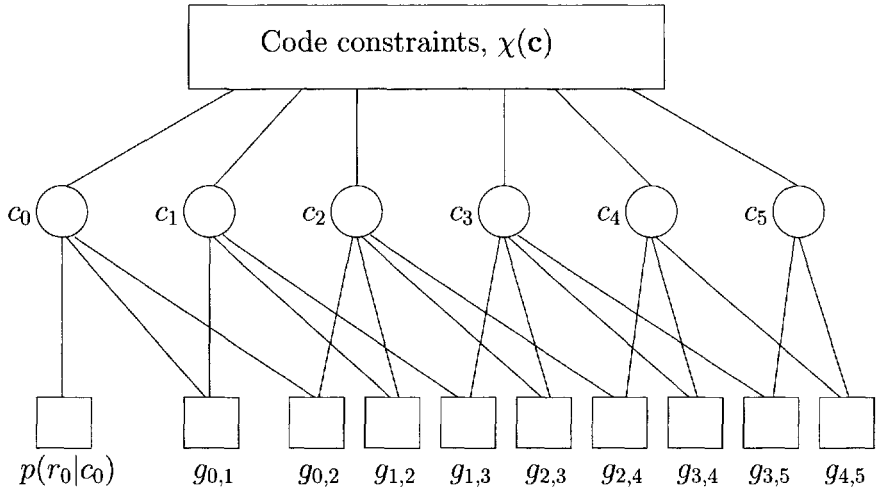


Figure 5.14: Simplified overall factor graph for PSK signals and $C = 2$. Reproduced from [201], ©2004 IEEE, by permission of the IEEE.

simulations, is shown for $\sigma_\Delta = 6$, 12, and 16 degrees. Both the proposed detection algorithms are very robust, especially the linear predictive-based one which has been designed taking into account the channel statistics.

5.6.3 Linear Predictive Graph-Based Detection for Frequency Flat Fading Channels

We now consider the case of transmission over a flat Rayleigh fading channel. The system model is the same as that considered in Section 3.11.5 and Section 4.9.6. As in Section 5.6.2, the Markovianity order N coincides with the finite memory parameter C. The basic exponential metric $\gamma_k(\boldsymbol{c}_{k-C}^k)$ which appears in (5.17) can be expressed as

$$\gamma_k(\boldsymbol{c}_{k-C}^k) \sim \exp\left(-\frac{1}{\sigma_e^2}\left|r_k - c_k \sum_{i=1}^{C} p_i \frac{r_{k-i}}{c_{k-i}}\right|^2\right) P\{a_k\}. \tag{5.30}$$

As for the detection algorithm in Section 5.6.2, the parameter C can be interpreted as the *prediction order*, $\{p_i\}_{i=1}^C$ are the prediction coefficients, and σ_e^2 represents the mean square prediction error. Coefficients $\{p_i\}$ and the mean square prediction error can be computed by solving a Wiener Hopf linear system, and in general depend on the considered code sequence.

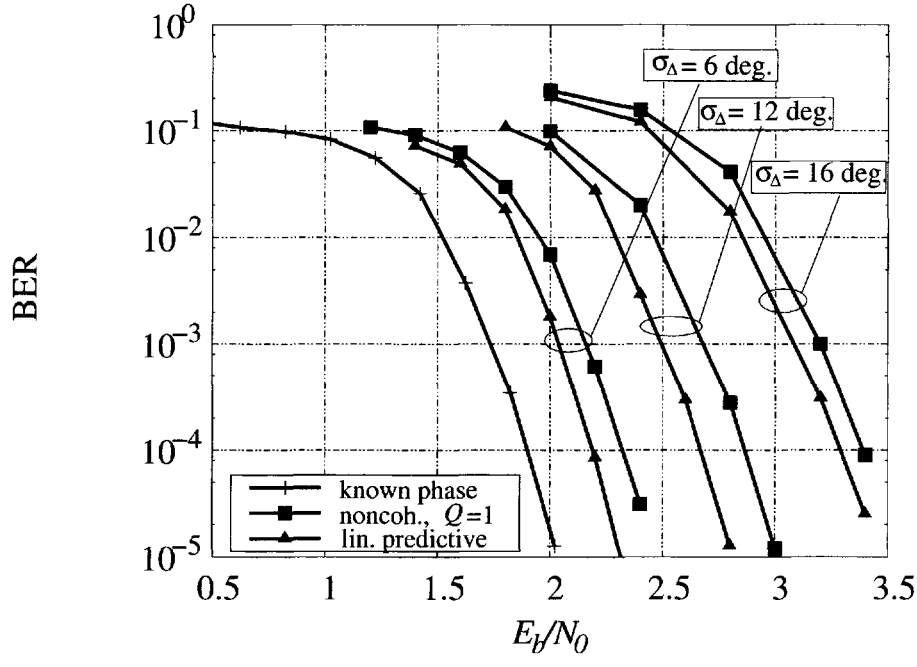


Figure 5.15: Performance in the case of a time-varying channel phase. The value of C is optimized in each individual case. Reproduced from [201], ©2004 IEEE, by permission of the IEEE.

For PSK signals, the prediction coefficients and the mean square prediction error become independent of the considered sequence. Taking into account that $|c_k| = 1$, after straightforward manipulations, one obtains:

$$\gamma_k(\boldsymbol{c}_{k-C}^k) \sim \exp\left(\frac{2}{\sigma_e^2}\sum_{i=1}^{C}\Re\{p_i r_k r_{k-i}^* c_k^* c_{k-i}\}\right) \cdot \exp\left(-\frac{2}{\sigma_e^2}\sum_{i=1}^{C}\sum_{\ell=i+1}^{C}\Re\{p_i p_\ell r_{k-i} r_{k-\ell}^* c_{k-i}^* c_{k-\ell}\}\right) P\{a_k\}. \tag{5.31}$$

Substituting (5.31) into (5.17), it can be easily shown that the resulting joint *a posteriori* pmf of the information symbols can be expressed as (see Problem 5.7)

$$P\{\boldsymbol{a}|\boldsymbol{r}\} \simeq \chi[\boldsymbol{c} = \mu_{\mathfrak{C}}(\boldsymbol{a})] \prod_{k=0}^{K-1} P\{a_k\} \prod_{i=1}^{C} \exp\left(\frac{2}{\sigma_e^2}\Re\{q_i r_k r_{k-i}^* c_k^* c_{k-i}\}\right) \tag{5.32}$$

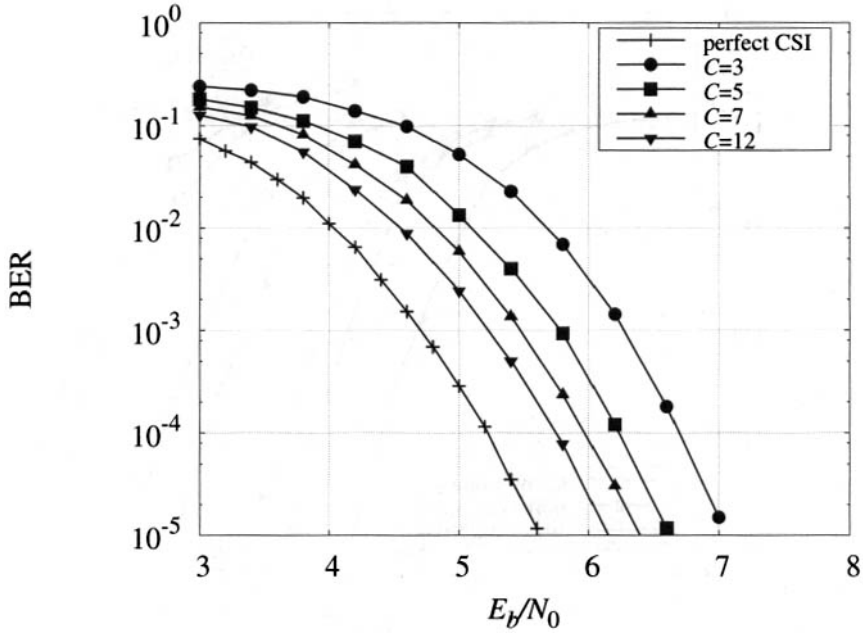


Figure 5.16: Performance of graph-based finite memory detection in the case of a flat correlated Rayleigh fading channel with $f_{\mathrm{D}}T = 10^{-2}$.

where

$$q_i \triangleq p_i - \sum_{\ell=1}^{C-i} p_\ell p_{\ell+i}. \tag{5.33}$$

The corresponding FG is similar to that depicted in Figure 5.14 and the complexity thus becomes linear in C.

The performance of the described algorithm, considering the same LDPC code and modulation format used for Figure 5.13, is shown in Figure 5.16. The normalized Doppler rate of the flat fading process is $f_{\mathrm{D}}T = 10^{-2}$. For comparison, the performance in the case of perfect channel state information (CSI) is considered. Obviously, for increasing values of the finite memory parameter C, the performance improves. In addition, due to the linear complexity of the detection algorithm, it is possible to implement receivers with values of C higher than that used in Chapter 3 and Chapter 4 for the VA or FB algorithm, respectively, and this allows a better approach to the performance of a receiver with perfect CSI.

5.7 Strong Phase Noise: An Alternative Approach to Graph-Based Detection

In the case of communication over a phase-uncertain channel, possibly affected by strong phase noise, we now develop new algorithms expanding upon the framework described in [192]. As in the approach described in Section 5.3, the channel parameters are modeled as stochastic processes with known statistics and the use of FGs and the SP algorithm will be considered to implement the MAP symbol detection strategy. The difference with respect to the approach in Section 5.3 is that the channel parameters are now explicitly represented in the graph. Therefore, since these channel parameters are continuous random variables, the application of the exact SP algorithm becomes impractical. To solve this problem, the method of canonical distributions is adopted. By specializing the approach of [192] to particular channel phase statistics and canonical distributions, several algorithms for noncoherent detection of LDPC codes have been proposed [216–219]. As will be shown, the choice of the canonical distribution becomes crucial in determining the performance and the complexity of the derived algorithms.

In the literature, other approaches have been proposed to perform LDPC decoding taking explicitly into account the stochastic channel parameters. In [50], the authors consider a noncoherent channel model where the unknown carrier phase is constant over a block of N symbols and independent from block to block (the so-called block noncoherent channel), and detection algorithms for LDPC codes based on this model are developed. A non-Bayesian approach is adopted in [220]: the channel parameters are estimated by using the expectation-maximization (EM) algorithm, as originally proposed in [221,222] for turbo codes, and the estimation algorithm is embedded into the LDPC iterative decoding process. On the contrary, in [93] a class of problems is identified for which the optimal (in the sense of the generalized-likelihood ratio test) computation of the symbol *a posteriori* probabilities can be performed with polynomial complexity and the application to LDPC codes transmitted over a noncoherent channel is discussed. A comparison of the proposed graph-based detection algorithm with some of these algorithms will be considered in Section 5.7.3.

5.7.1 System Model and Exact Sum-Product Algorithm

As considered in Chapter 3 and Chapter 4, the observation at the output of a generic phase-uncertain channel can be expressed as follows:

$$r_k = c_k e^{j\theta_k} + n_k \tag{5.34}$$

where $\{\theta_k\}$ is a channel phase noise process and $\{n_k\}$ are the independent and identically distributed Gaussian noise samples with variance σ^2 per component. As already considered in Section 5.6.2, a common model for the phase noise process $\{\theta_k\}$ is the random-walk (Wiener) model described by

$$\theta_k = \theta_{k-1} + \Delta_k \tag{5.35}$$

where $\{\Delta_k\}$ is a discrete-time real white Gaussian process with $\Delta_k \in \mathcal{N}(0, \sigma_\Delta^2)$.[8] Assuming further that θ_0 has uniform distribution in $[0, 2\pi)$, it follows that

$$p(\theta_k|\theta_{k-1}, \theta_{k-2}, \ldots, \theta_0) = p(\theta_k|\theta_{k-1}) = p_\Delta(\theta_k - \theta_{k-1}) \tag{5.36}$$

where we define $p_\Delta(\phi)$ as the pdf of the increment $\Delta_k \bmod [0, 2\pi)$, i.e.,

$$p_\Delta(\phi) \triangleq \begin{cases} \displaystyle\sum_{\ell=-\infty}^{\infty} g\left(0, \sigma_\Delta^2, \phi - \ell 2\pi\right) & \phi \in [0, 2\pi) \\ 0 & \text{elsewhere.} \end{cases} \tag{5.37}$$

The Wiener phase noise model will be considered in the following as a working assumption in order to derive efficient iterative detection and decoding algorithms. This assumption will be relaxed in Section 5.7.3, where the proposed detection algorithms will be applied to a DVB-S2-compliant model provided by the European Space Agency (ESA) and described in [223, 224]. This phase noise model considers $\{\theta_k\}$ as the sum of the outputs of two infinite impulse response filters driven by white Gaussian noise with unit variance, where the filters are chosen to fit an experimental phase noise mask. The transfer functions of these filters are [223, 224]:

$$H_1(z) = \frac{1}{\sqrt{2T}} \frac{-4.7 \cdot 10^{-11}}{(z - 0.999975)^2} \tag{5.38}$$

$$H_2(z) = \frac{1}{\sqrt{2T}} \frac{2.8 \cdot 10^{-6}(z - 0.992015)(z - 1.103181)}{(z - 0.991725)(z - 0.9999985)(z - 0.563507)} \tag{5.39}$$

where T is the symbol interval.

Let $P(\boldsymbol{a}, \boldsymbol{\theta}|\boldsymbol{r})$ denote the joint *a posteriori* probability distribution function[9] of the information symbols and of the phase noise vector $\boldsymbol{\theta}$, given $\boldsymbol{r}$. Clearly, the desired

[8] In the following, a complex circularly symmetric (real) Gaussian random vector $\boldsymbol{v}$ with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$ will be denoted by $\boldsymbol{v} \in \mathcal{N}_{\mathbb{C}}(\boldsymbol{m}, \boldsymbol{\Sigma})$ (and by $\boldsymbol{v} \in \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$). We will denote the multivariate complex circularly symmetric (real) Gaussian pdf with mean $\boldsymbol{m}$, covariance matrix $\boldsymbol{\Sigma}$ and argument $\boldsymbol{x}$ by $g_{\mathbb{C}}(\boldsymbol{m}, \boldsymbol{\Sigma}, \boldsymbol{x})$ (and by $g(\boldsymbol{m}, \boldsymbol{\Sigma}, \boldsymbol{x})$).

[9] We use the term probability distribution function to denote a continuous pdf with some discrete probability masses. For a probability distribution function we will use the symbol $P(.)$.

$P\{a_k|\boldsymbol{r}\}$ can be obtained by marginalizing $P(\boldsymbol{a},\boldsymbol{\theta}|\boldsymbol{r})$ with respect to $\boldsymbol{\theta}$ and to all a_j for $j \neq k$. This can be accomplished in an approximate (but low-complexity) way by running the SP algorithm on the FG of $P(\boldsymbol{a},\boldsymbol{\theta}|\boldsymbol{r})$, as illustrated in the following.

From the definition of the encoding function $\mu_{\mathfrak{C}}$ and the channel model (5.34), one obtains the factorization

$$\begin{aligned} P(\boldsymbol{a},\boldsymbol{\theta}|\boldsymbol{r}) &\sim P\{\boldsymbol{a}\}p(\boldsymbol{\theta})p(\boldsymbol{r}|\boldsymbol{\theta},\boldsymbol{a}) \\ &\sim \chi[\boldsymbol{c}=\mu_{\mathfrak{C}}(\boldsymbol{a})]P\{\boldsymbol{a}\}p(\boldsymbol{\theta})p(\boldsymbol{r}|\boldsymbol{\theta},\boldsymbol{c}=\mu_{\mathfrak{C}}(\boldsymbol{a})) \\ &\sim \chi[\boldsymbol{c}=\mu_{\mathfrak{C}}(\boldsymbol{a})]P\{\boldsymbol{a}\}p(\boldsymbol{\theta})\prod_{k=0}^{K-1}p(r_k|c_k,\theta_k) \\ &\sim \chi[\boldsymbol{c}=\mu_{\mathfrak{C}}(\boldsymbol{a})]P\{\boldsymbol{a}\}p(\boldsymbol{\theta})\prod_{k=0}^{K-1}f_k(c_k,\theta_k) \end{aligned} \tag{5.40}$$

where we have used the fact that the output signal pdf $p(\boldsymbol{r})$ does not depend on $\boldsymbol{a}$, that the AWGN channel for a given $\boldsymbol{\theta}$ is memoryless, and we have defined the functions

$$f_k(c_k,\theta_k) \triangleq \exp\left(\frac{1}{\sigma^2}\Re\{r_k c_k^* e^{-j\theta_k}\} - \frac{|c_k|^2}{2\sigma^2}\right) \sim \exp\left(-\frac{1}{2\sigma^2}|r_k - c_k e^{j\theta_k}|^2\right). \tag{5.41}$$

In the following, we will also assume that the information symbols are independent, identically and uniformly distributed. As a consequence, $P\{\boldsymbol{a}\} = 1/M^K$, and this term can be discarded. The FG corresponding to (5.40) is shown in Figure 5.17.

Assuming a first order Markov model for the phase noise, we can further factorize the term $p(\boldsymbol{\theta})$ as $p(\boldsymbol{\theta}) = p(\theta_0)\prod_{k=1}^{K}p_\Delta(\theta_k - \theta_{k-1})$, obtaining

$$P(\boldsymbol{a},\boldsymbol{\theta}|\boldsymbol{r}) \sim \chi[\boldsymbol{c}=\mu_{\mathfrak{C}}(\boldsymbol{a})]p(\theta_0)\prod_{k=1}^{K-1}p_\Delta(\theta_k-\theta_{k-1})\prod_{k=0}^{K-1}f_k(c_k,\theta_k). \tag{5.42}$$

The corresponding FG, sketched in Figure 5.18, represents the starting point for the derivation of the proposed algorithms.

The SP algorithm applied to the FG in the upper box, corresponding to the code constraints, consists of the well-known standard BP whose efficient implementation depends on the structure of the code $\mathfrak{C}$ and needs no details here. Hence, we shall concentrate on the SP algorithm message propagation in the lower part of the graph. Omitting, for notational simplicity, the explicit reference to the current iteration, let us denote by $P_d\{c_k\}$ the message from variable node c_k to factor node f_k, and by $P_u\{c_k\}$ the message in the opposite direction (see Figure 5.18)—note that $P_d\{c_k\}$

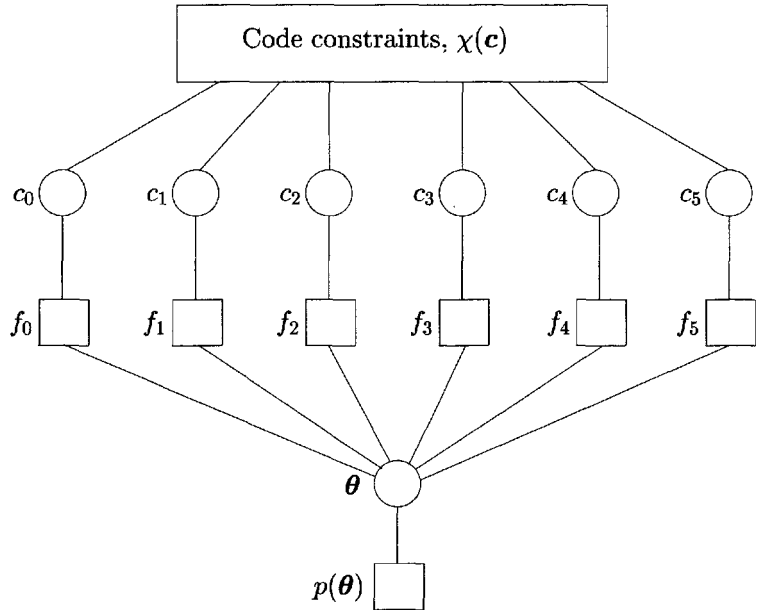


Figure 5.17: Factor graph corresponding to (5.40).

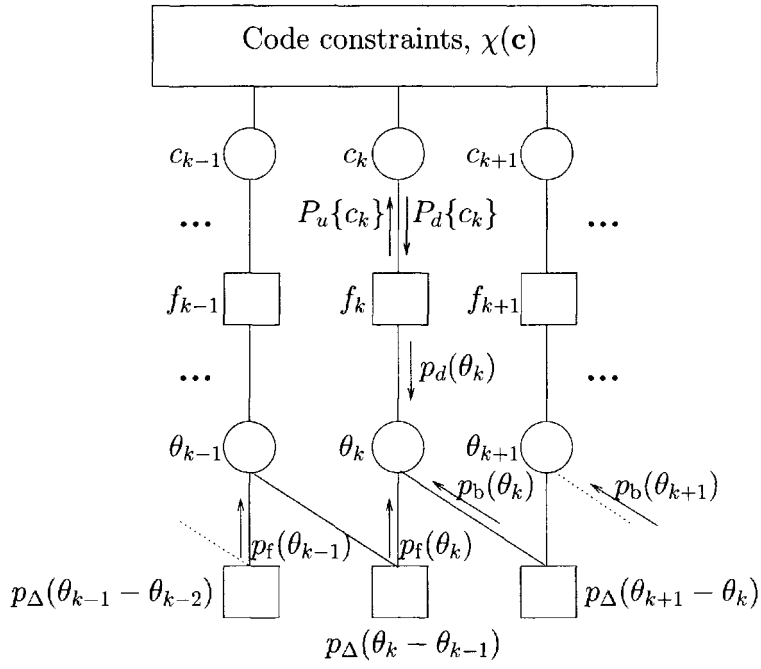


Figure 5.18: Factor graph corresponding to (5.42).

and $P_u\{c_k\}$ are, in general, reliability values which approximate the APPs. The message $p_d(\theta_k)$ from factor node f_k to variable node θ_k can be expressed as

$$p_d(\theta_k) \sim \sum_{c \in \mathcal{C}} P_d\{c_k = c\} f_k(c_k = c, \theta_k) \tag{5.43}$$

where $\mathcal{C}$ denotes the constellation of the code symbols. We also assume that in the lower part of the FG, describing the phase-noise evolution, a forward backward node activation schedule is adopted. Therefore, messages $p_{\mathrm{f}}(\theta_k)$ from factor node $p_\Delta(\theta_k - \theta_{k-1})$ to variable node θ_k, and $p_{\mathrm{b}}(\theta_k)$ from factor node $p_\Delta(\theta_{k+1} - \theta_k)$ to variable node θ_k, can be recursively computed as follows:

$$p_{\mathrm{f}}(\theta_k) \sim \int_0^{2\pi} p_d(\theta_{k-1}) p_{\mathrm{f}}(\theta_{k-1}) p_\Delta(\theta_k - \theta_{k-1}) \, \mathrm{d}\theta_{k-1} \tag{5.44}$$

$$p_{\mathrm{b}}(\theta_k) \sim \int_0^{2\pi} p_d(\theta_{k+1}) p_{\mathrm{b}}(\theta_{k+1}) p_\Delta(\theta_{k+1} - \theta_k) \, \mathrm{d}\theta_{k+1} \tag{5.45}$$

with uniform pdfs as initial conditions:

$$p_{\mathrm{f}}(\theta_0) = p_{\mathrm{b}}(\theta_K) = \frac{1}{2\pi} \qquad \theta_0, \theta_K \in [0, 2\pi). \tag{5.46}$$

The message $P_u\{c_k\}$ from f_k to c_k is given by

$$P_u\{c_k\} \sim \int_0^{2\pi} p_{\mathrm{f}}(\theta_k) p_{\mathrm{b}}(\theta_k) f_k(c_k, \theta_k) \, \mathrm{d}\theta_k. \tag{5.47}$$

The vector of messages $\{P_u\{c_k\} : k = 0, \ldots, K-1\}$ represents the observation (in the form of a sequence of approximate APPs) of the coded symbols "seen" through a virtual memoryless channel, and is processed by the upper part of the graph according to the standard BP algorithm. At each iteration, this produces updated messages $\{P_d\{c_k\} : k = 0, \ldots, K-1\}$ and updated estimates of the APPs $\{P\{a_k|\boldsymbol{r}\}\}$.

Equations (5.44), (5.45), and (5.47) form the main part of the SP algorithm for iterative detection and decoding in the presence of phase noise.

5.7.2 Proposed Algorithms

It is clear that the implementation complexity of the exact SP algorithm is impractical, since the messages from and to the variable nodes $\{\theta_k\}$ are continuous pdfs. In

order to obtain practical algorithms, we follow the canonical distribution approach proposed in [192]. This consists of constraining the messages from/to the continuous variables to take values in a prescribed family of pdfs, that admits a compact parametric representation. Hence, the message computation reduces to the computation of the pdf parameters. This representation can be exact or, more often, may involve some approximations. In the latter case, finding good choices of the pdf parameterization, such that the resulting algorithm yields good performance and low computational complexity is generally nontrivial. In the following, we discuss different options to attack this problem.

Discretization of the Channel Parameters

This case corresponds to letting the canonical distribution be a weighted sum of impulses. This approach has been adopted for Viterbi- and BCJR-like receivers in [173,217,225], respectively. We assume that the channel phase θ_k may take on the following L values: $\Theta = \{0, 2\pi/L, \ldots, 2\pi(L-1)/L\}$.[10] Obviously, this approach becomes "optimal" (in the sense that it approaches the performance of the exact SP algorithm) for a sufficiently large number of discretization levels, at the expense of an increasing computational complexity.

Fourier Parameterization

The function $f_k(c_k, \theta_k)$ defined in (5.41) is periodic in θ_k. Hence, it can be expanded in Fourier series. We use the well-known identity [110, equation (9.6.34)]

$$e^{x\cos\theta} = \mathrm{I}_0(x) + 2\sum_{\ell=1}^{\infty} \mathrm{I}_\ell(x)\cos(\ell\theta) \tag{5.48}$$

where $\mathrm{I}_\ell(x)$ is the modified Bessel function of the first kind of order ℓ. Letting, for a complex number z, $\phi(z) \triangleq \arg(z)$, after some straightforward manipulations one obtains

$$f_k(c_k, \theta_k) \sim e^{-\frac{|c_k|^2}{2\sigma^2}} \sum_{\ell=-\infty}^{\infty} \mathrm{I}_\ell\left(\frac{|r_k||c_k|}{\sigma^2}\right) e^{-j\ell\phi(r_k c_k^*)} e^{j\ell\theta_k}. \tag{5.49}$$

Substituting (5.49) into (5.43), we may express

$$p_d(\theta_k) \sim \sum_{\ell=-\infty}^{\infty} A_k^{(\ell)} e^{j\ell\theta_k} \tag{5.50}$$

[10] In [173], the authors state that for M-PSK signals, $L = 8M$ values are suffi cient to have negligible performance loss.

having defined

$$\begin{aligned} A_k^{(\ell)} &\triangleq \sum_{c\in\mathcal{C}} P_d\{c_k = c\} e^{-\frac{|c|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c|}{\sigma^2}\right) e^{-j\ell\phi(r_k c^*)} \\ &= e^{-j\ell\phi(r_k)} \sum_{c\in\mathcal{C}} P_d\{c_k = c\} e^{-\frac{|c|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c|}{\sigma^2}\right) e^{j\ell\phi(c)} \\ &= e^{-j\ell\phi(r_k)} \sum_{c\in\mathcal{C}} P_d\{c_k = c\} e^{-\frac{|c|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c|}{\sigma^2}\right) \frac{c^\ell}{|c|^\ell}. \end{aligned} \tag{5.51}$$

Note that for PSK signals, the expression of coefficient $A_k^{(\ell)}$, neglecting irrelevant terms, simplifies to

$$A_k^{(\ell)} = e^{-j\ell\phi(r_k)} \mathrm{I}_\ell\left(\frac{|r_k|}{\sigma^2}\right) \sum_{c\in\mathcal{C}} P_d\{c_k = c\}\, c^\ell. \tag{5.52}$$

The pdfs $p_{\mathrm{f}}(\theta_k)$ and $p_{\mathrm{b}}(\theta_k)$ take on the same form, i.e., they are periodic as well and can be expanded in Fourier series as

$$p_{\mathrm{f}}(\theta_k) = \sum_{\ell=-\infty}^{\infty} B_{\mathrm{f},k}^{(\ell)} e^{j\ell\theta_k} \tag{5.53}$$

$$p_{\mathrm{b}}(\theta_k) = \sum_{\ell=-\infty}^{\infty} B_{\mathrm{b},k}^{(\ell)} e^{j\ell\theta_k}. \tag{5.54}$$

Substituting (5.50) and (5.53) into (5.44), one obtains

$$\begin{aligned} \sum_{\ell=-\infty}^{\infty} B_{\mathrm{f},k}^{(\ell)} e^{j\ell\theta_k} &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} A_{k-1}^{(m)} B_{\mathrm{f},k-1}^{(n)} \int_0^{2\pi} e^{j(m+n)\theta_{k-1}} p_\Delta(\theta_k - \theta_{k-1})\, \mathrm{d}\theta_{k-1} \\ &= \sum_{\ell=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{\mathrm{f},k-1}^{(\ell-m)} \int_0^{2\pi} e^{j\ell\theta_{k-1}} p_\Delta(\theta_k - \theta_{k-1})\, \mathrm{d}\theta_{k-1}. \end{aligned} \tag{5.55}$$

Note that, for practical values of σ_Δ, the pdf $p_\Delta(\phi)$ is essentially zero for argument ϕ outside an interval centered on 0 of size much smaller than 2π. Hence, one can write

$$\begin{aligned} \int_0^{2\pi} e^{j\ell\theta_{k-1}} p_\Delta(\theta_k - \theta_{k-1})\, \mathrm{d}\theta_{k-1} &\simeq \int_{-\infty}^{\infty} e^{j\ell\theta_{k-1}} g(0, \sigma_\Delta^2, \theta_k - \theta_{k-1})\, \mathrm{d}\theta_{k-1} \\ &= D_\ell(\sigma_\Delta) e^{j\ell\theta_k} \end{aligned} \tag{5.56}$$

where

$$D_\ell(\sigma_\Delta) = e^{-\frac{\sigma_\Delta^2 \ell^2}{2}}. \tag{5.57}$$

By using (5.56) in (5.55), it follows that

$$\sum_{\ell=-\infty}^{\infty} B_{\mathrm{f},k}^{(\ell)} e^{j\ell\theta_k} = \sum_{\ell=-\infty}^{\infty} \left[D_\ell(\sigma_\Delta) \sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{\mathrm{f},k-1}^{(\ell-m)} \right] e^{j\ell\theta_k}. \tag{5.58}$$

The above result leads to a forward recursion for the computation of the Fourier coefficients $\{B_{\mathrm{f},k}^{(\ell)}\}$. The step at epoch k becomes:

$$B_{\mathrm{f},k}^{(\ell)} = D_\ell(\sigma_\Delta) \sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{\mathrm{f},k-1}^{(\ell-m)} = D_\ell(\sigma_\Delta)[A_{k-1}^{(\ell)} \otimes B_{\mathrm{f},k-1}^{(\ell)}] \tag{5.59}$$

where $\otimes$ denotes convolution of sequences. From condition (5.46), the following initial condition can be derived:

$$B_{\mathrm{f},0}^{(\ell)} = \delta_{\mathrm{K}}(\ell) \tag{5.60}$$

where $\delta_{\mathrm{K}}(\ell)$ denotes the Kronecker delta function, defined in Chapter 4. Similarly, a step in the backward recursion to compute the coefficients $\{B_{\mathrm{b},k}^{(\ell)}\}$ is given by

$$B_{\mathrm{b},k}^{(\ell)} = D_\ell(\sigma_\Delta)[A_{k+1}^{(\ell)} \otimes B_{\mathrm{b},k+1}^{(\ell)}] \tag{5.61}$$

with initial condition

$$B_{\mathrm{b},K-1}^{(\ell)} = \delta_{\mathrm{K}}(\ell). \tag{5.62}$$

Note that the computation of these coefficients can be simplified taking into account the following symmetries:

$$A_k^{(-\ell)} = A_k^{(\ell)*} \tag{5.63}$$

$$B_{\mathrm{f},k}^{(-\ell)} = B_{\mathrm{f},k}^{(\ell)*} \tag{5.64}$$

$$B_{\mathrm{b},k}^{(-\ell)} = B_{\mathrm{b},k}^{(\ell)*}. \tag{5.65}$$

Finally, substituting (5.49), (5.53), and (5.54) into (5.47) and defining

$$E_k^{(\ell)} \triangleq e^{-\frac{|c_k|^2}{2\sigma^2}} \left\{ B_{\mathrm{f},k}^{(\ell)} \otimes B_{\mathrm{b},k}^{(\ell)} \otimes \left[\mathrm{I}_\ell \left(\frac{|r_k||c_k|}{\sigma^2} \right) e^{-j\ell\phi(r_k c_k^*)} \right] \right\} \tag{5.66}$$

one obtains

$$P_u\{c_k\} \sim \sum_{\ell=-\infty}^{\infty} E_k^{(\ell)} \int_0^{2\pi} e^{j\ell\theta_k} \, \mathrm{d}\theta_k = E_k^{(0)}. \tag{5.67}$$

The convolution of the infinite-duration Fourier coefficients can be effectively implemented by truncation. In fact, a reduced number N of coefficients must be taken into account due to the fact that, for a given x, the sequence $\{\mathrm{I}_\ell(x)\}$ is monotonically decreasing for increasing values of ℓ. Standard smoothed truncation methods (windowing) can be applied [155]. In particular, by means of computer simulations, one can show that the Kaiser window [155] with an optimized parameter yields good results, as will be shown in Section 5.7.3.

Tikhonov Parameterization

Consider (5.43). If the messages $\{P_d\{c_k\}\}$ were the exact probabilities of the code symbols, it would hold that

$$p_d(\theta_k) \sim \sum_{c\in\mathcal{C}} P_d\{c_k = c\} f_k(c_k = c, \theta_k) \sim p(r_k|\theta_k). \tag{5.68}$$

We approximate $p(r_k|\theta_k)$ by the Gaussian pdf with the same mean and variance (first and second moment matching). It can be shown that this is equivalent to an approximation of $p(r_k|\theta_k)$ by the Gaussian pdf at minimum *divergence* [35] (see Problem 5.10). By direct computation, one obtains:

$$\mathrm{E}\{r_k|\theta_k\} = \mathfrak{a}_k e^{j\theta_k} \tag{5.69}$$

$$\mathrm{VAR}\{r_k|\theta_k\} = 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2 \tag{5.70}$$

where $\mathfrak{a}_k$ and $\mathfrak{b}_k$ are the first and second order moments of the discrete random variable c_k with pmf $P_d\{c_k\}$, and are given by

$$\mathfrak{a}_k \triangleq \sum_{c\in\mathcal{C}} c P_d\{c_k = c\} \tag{5.71}$$

$$\mathfrak{b}_k \triangleq \sum_{c\in\mathcal{C}} |c|^2 P_d\{c_k = c\}. \tag{5.72}$$

Hence, we approximate $p(r_k|\theta_k) \simeq g_{\mathbb{C}}(\mathfrak{a}_k e^{j\theta_k}, 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2, r_k)$. Under this Gaussian approximation, it follows that

$$\begin{aligned} p_d(\theta_k) &\sim p(r_k|\theta_k) \\ &\simeq g_{\mathbb{C}}\left(\mathfrak{a}_k e^{j\theta_k}, 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2, r_k\right) \\ &\sim \exp\left(2\frac{\Re\{r_k \mathfrak{a}_k^* e^{-j\theta_k}\}}{2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2}\right). \end{aligned} \tag{5.73}$$

Substituting (5.73) into the forward recursion (5.44), one obtains

$$p_{\mathrm{f}}(\theta_k) \simeq \int_0^{2\pi} \exp\left(2\frac{\Re\{r_{k-1}\mathfrak{a}_{k-1}^* e^{-j\theta_{k-1}}\}}{2\sigma^2 + \mathfrak{b}_{k-1} - |\mathfrak{a}_{k-1}|^2}\right) p_{\mathrm{f}}(\theta_{k-1}) p_\Delta(\theta_k - \theta_{k-1})\,\mathrm{d}\theta_{k-1}. \quad (5.74)$$

When the channel phase is slowly varying, i.e., for $\sigma_\Delta \to 0$, $p_\Delta(\theta_k - \theta_{k-1}) = \delta(\theta_k - \theta_{k-1})$. In this case, the solution of the recursion given by (5.74) with initial condition (5.46) is a sequence of Tikhonov pdfs, given by

$$p_{\mathrm{f}}(\theta_k) \sim \exp\left(\Re\{a_{\mathrm{f},k} e^{-j\theta_k}\}\right) \quad (5.75)$$

where $\{a_{\mathrm{f},k}\}$ can be recursively computed as

$$a_{\mathrm{f},k} = a_{\mathrm{f},k-1} + 2\frac{r_{k-1}\mathfrak{a}_{k-1}^*}{2\sigma^2 + \mathfrak{b}_{k-1} - |\mathfrak{a}_{k-1}|^2} \quad (5.76)$$

with initial condition $a_{\mathrm{f},0} = 0$. Similarly, the solution of the backward recursion (5.45) under the above approximations is the sequence of Tikhonov pdfs

$$p_{\mathrm{b}}(\theta_k) \sim \exp\left(\Re\{a_{\mathrm{b},k} e^{-j\theta_k}\}\right) \quad (5.77)$$

where $\{a_{\mathrm{b},k}\}$ can be recursively computed as

$$a_{\mathrm{b},k} = a_{\mathrm{b},k+1} + 2\frac{r_{k+1}\mathfrak{a}_{k+1}^*}{2\sigma^2 + \mathfrak{b}_{k+1} - |\mathfrak{a}_{k+1}|^2} \quad (5.78)$$

with initial condition $a_{\mathrm{b},K-1} = 0$. From (5.75), (5.77) and (5.47), it follows that

$$P_u\{c_k\} \sim \exp\left(-\frac{|c_k|^2}{2\sigma^2}\right) \mathrm{I}_0\left(\left|a_{\mathrm{f},k} + a_{\mathrm{b},k} + \frac{r_k c_k^*}{\sigma^2}\right|\right). \quad (5.79)$$

When the phase is rapidly varying, the approximation $p_\Delta(\theta_k - \theta_{k-1}) \simeq \delta(\theta_k - \theta_{k-1})$ does not hold any longer. However, one can show that good approximations of the functions $p_{\mathrm{f}}(\theta_k)$ and $p_{\mathrm{b}}(\theta_k)$ are still given by (5.75) and (5.77), where the coefficients $\{a_{\mathrm{f},k}\}$ and $\{a_{\mathrm{b},k}\}$ are updated by the following modified forward and backward recursions

$$a_{\mathrm{f},k} = \left[a_{\mathrm{f},k-1} + 2\frac{r_{k-1}\mathfrak{a}_{k-1}^*}{2\sigma^2 + \mathfrak{b}_{k-1} - |\mathfrak{a}_{k-1}|^2}\right] \cdot\Gamma\left(\sigma_\Delta^2, \left|a_{\mathrm{f},k-1} + 2\frac{r_{k-1}\mathfrak{a}_{k-1}^*}{2\sigma^2 + \mathfrak{b}_{k-1} - |\mathfrak{a}_{k-1}|^2}\right|\right) \quad (5.80)$$

$$a_{\mathrm{b},k} = \left[a_{\mathrm{b},k+1} + 2\frac{r_{k+1}\mathfrak{a}_{k+1}^*}{2\sigma^2 + \mathfrak{b}_{k+1} - |\mathfrak{a}_{k+1}|^2}\right] \cdot\Gamma\left(\sigma_\Delta^2, \left|a_{\mathrm{b},k+1} + 2\frac{r_{k+1}\mathfrak{a}_{k+1}^*}{2\sigma^2 + \mathfrak{b}_{k+1} - |\mathfrak{a}_{k+1}|^2}\right|\right) \quad (5.81)$$

where the real function $\Gamma(x_1, x_2)$, of real arguments x_1 and x_2 can be numerically computed and stored in a look-up table. The motivation of (5.80) and (5.81) and a closed-form approximated expression of the correction factor Γ can be derived as described in the following. Consider the function

$$f(y) \triangleq \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_0^{2\pi} e^{\Re\{ze^{-jx}\}} e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}} \, \mathrm{d}x = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_{-\pi}^{\pi} e^{\Re\{ze^{-j(x+y)}\}} e^{-\frac{x^2}{2\sigma_\Delta^2}} \, \mathrm{d}x \tag{5.82}$$

where z is a complex number and x and y are real numbers. By discarding irrelevant multiplicative factors, we shall show that $f(y) \simeq e^{\Gamma(\sigma_\Delta^2, |z|)\Re\{ze^{-jy}\}}$, where $\Gamma(\sigma_\Delta^2, |z|)$ is a real function of $|z|$ and σ_Δ^2. This can be seen using the following approximation which holds for large positive values of the real parameter a (in practice, $a > 5$):

$$\frac{e^{a\cos(x-y)}}{2\pi \mathrm{I}_0(a)} \simeq \frac{1}{\sqrt{2\pi/a}} e^{-\frac{a}{2}(x-y)^2} = g\left(y, \frac{1}{a}, x\right). \tag{5.83}$$

In fact, for sufficiently large values of a, the Tikhonov pdf $\exp(a\cos(x-y))/2\pi\mathrm{I}_0(a)$ has its support in a small interval around y. Hence, by using a second-order Taylor expansion, it follows that $\cos(x-y) \simeq 1 - (x-y)^2/2$. A normalization constant has been further added to obtain a pdf.

The correction term Γ in (5.80) and in (5.81) can be derived using the approximation (5.83). In fact, we let

$$\begin{aligned}
f(y) &= \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_0^{2\pi} e^{\Re\{ze^{-jx}\}} e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}} \, \mathrm{d}x \\
&\overset{(a)}{\simeq} \int_{-\infty}^{\infty} e^{\Re\{ze^{-jx}\}} g(x, \sigma_\Delta^2, y) \, \mathrm{d}x \\
&\overset{(b)}{\simeq} 2\pi \mathrm{I}_0(|z|) \int_{-\infty}^{\infty} g\left(\phi(z), \frac{1}{|z|}, x\right) g(x, \sigma_\Delta^2, y) \, \mathrm{d}x \\
&\overset{(c)}{\simeq} g\left(\phi(z), \frac{1}{|z|} + \sigma_\Delta^2, y\right) \\
&\overset{(d)}{\simeq} \frac{1}{2\pi \mathrm{I}_0\left(\frac{|z|}{1+\sigma_\Delta^2|z|}\right)} \exp\left(\frac{1}{1+\sigma_\Delta^2|z|} \Re\{ze^{-jy}\}\right) \\
&\sim \exp\left(\frac{1}{1+\sigma_\Delta^2|z|} \Re\{ze^{-jy}\}\right)
\end{aligned} \tag{5.84}$$

where (a) follows from the observation that, for $\sigma_\Delta \ll 1$, the function $\exp[-(x-y)^2/2\sigma_\Delta^2]$ has its support in a small interval around y, (b) and (d) follow from the approximation (5.83) and (c) follows from the following property of a Gaussian distribution (see Problem 5.12):

$$\int g(A_1, \Sigma_1, x) g(A_2 x, \Sigma_2, y) \mathrm{d}x \sim g\left(A_1 A_2, \Sigma_2 + |A_2|^2 \Sigma_1, y\right). \tag{5.85}$$

The same relationship holds for complex Gaussian pdfs. Hence,

$$\Gamma(\sigma_\Delta^2, |z|) = \frac{1}{1 + \sigma_\Delta^2 |z|}.$$

If pilot symbols are used and arranged in bursts (training sequences) separated by long blocks of code symbols, as in the case of the DVB-S2 system [226], it is necessary to slightly modify the algorithm in order to speed up the convergence process and to avoid the risk of a phase ambiguity. In fact, consider the recursive integral equation (5.44) from the second iteration on. If the product

$$p_d(\theta_k) p_{\mathrm{f}}(\theta_k) = \left(\sum_{c \in \mathcal{C}} P_d\{c_k = c\} f_k(c_k = c, \theta_k) \right) p_{\mathrm{f}}(\theta_k)$$

contains a dominant exponential term, i.e., if there exists a value $\overline{c} \in \mathcal{C}$ such that

$$\ln P_d\{c_k = \overline{c}\} + \left| a_{\mathrm{f},k} + \frac{r_k \overline{c}^*}{\sigma^2} \right| > \epsilon + \ln P_d\{c_k = c\} + \left| a_{\mathrm{f},k} + \frac{r_k c^*}{\sigma^2} \right|, \ \forall c \in \mathcal{C}, c \neq \overline{c} \tag{5.86}$$

where ϵ is a real parameter to be optimized by computer simulation, it is preferable to let $\mathfrak{a}_k = \overline{c}$ and $\mathfrak{b}_k = |\overline{c}|^2$. Otherwise, one can choose $\mathfrak{a}_k$ and $\mathfrak{b}_k$ as in (5.71) and (5.72). This corresponds to using a decision-aided scheme based on hard decisions for some symbols $\{c_k\}$. Similar considerations also hold for the recursive integral equation (5.45). In the numerical results relative to the DVB-S2 system, we found that $\epsilon = 1.5$ yields satisfactory results.

Gaussian Parameterization

Another application of the canonical distribution approach consists of modeling the phasor process $h_k \stackrel{\triangle}{=} e^{j\theta_k}$ as a complex circularly symmetric Gauss Markov process and treating $\boldsymbol{h} = (h_0, \ldots, h_{K-1})$ and the observation sequence $\boldsymbol{r}$ as jointly Gaussian. This assumption yields the forward and backward recursions (5.44) and (5.45) in the form of a Kalman smoother [227].

As for the Tikhonov parameterization of the previous section, we impose a jointly Gaussian structure on the observation $\{r_k\}$ and the phasor process $\{h_k\}$ by the approximation based on first and second order moment matching of the pdf of r_k given h_k, i.e., we let $p(r_k|h_k) \simeq g_{\mathbb{C}}(\mathfrak{a}_k h_k, 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2, r_k)$ where the conditional mean and variance of r_k given h_k are given by $\mathrm{E}\{r_k|h_k\} = \mathfrak{a}_k h_k$ and by $\mathrm{VAR}\{r_k|h_k\} = 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2$, with $\mathfrak{a}_k$ and $\mathfrak{b}_k$ given in (5.71) and (5.72), respectively.

Under the Gauss Markov assumption for $\{h_k\}$ and the above joint Gaussianity, we can define the "state" and "observation" equations by

$$h_{k+1} = \rho h_k + v_k \tag{5.87}$$

$$r_k = \mathfrak{a}_k h_k + w_k \tag{5.88}$$

where $\{v_k\}$ and $\{w_k\}$ are independent Gaussian processes with independent components such that $v_k \in \mathcal{N}_{\mathbb{C}}(0, 1-\rho^2)$ and $w_k \in \mathcal{N}_{\mathbb{C}}(0, 2\sigma^2 + \mathfrak{b}_k - |\mathfrak{a}_k|^2)$. For the Wiener phase noise model, one obtains explicitly that $\rho = e^{-\sigma_\Delta^2/2}$. The time-reversal process $(h_{K-1}, \ldots, h_1, h_0)$ is also Gauss Markov [227] with state and observation equations given by $h_{k-1} = \rho h_k + v'_k$ and by $r_k = \mathfrak{a}_k h_k + w'_k$, respectively, where $\{v'_k\}$ and $\{w'_k\}$ have the same statistics as $\{v_k\}$ and $\{w_k\}$.

Under this model, we have $p(h_k|\{r_j : j \neq k\}) = g_{\mathbb{C}}(m_k, \Sigma_k, h_k)$, where the conditional mean and variance can be computed iteratively using the Kalman smoother [227, 228]. The derivation of the Kalman filter via the SP algorithm is given in [79]. The forward and backward recursions (5.44) and (5.45) are evaluated explicitly by a repeated application of (5.85) and this following further property of Gaussian distributions (see Problem 5.13):

$$g_{\mathbb{C}}(A_1, \Sigma_1, x) g_{\mathbb{C}}(A_2, \Sigma_2, x) \sim g_{\mathbb{C}}\left(\frac{\Sigma_2}{\Sigma_1+\Sigma_2}A_1 + \frac{\Sigma_1}{\Sigma_1+\Sigma_2}A_2, \frac{\Sigma_1\Sigma_2}{\Sigma_1+\Sigma_2}, x\right). \tag{5.89}$$

Let $m_{k|k-1}, \Sigma_{k|k-1}$ be the conditional mean and variance of h_k given $\{r_j : j = 0, \ldots, k-1\}$ (prediction) and $m_{k|k}$ and $\Sigma_{k|k}$ be the conditional mean and variance of h_k given $\{r_j : j = 0, \ldots, k\}$ (filtering). Similarly, let $\mu_{k|k+1}, \Xi_{k|k-1}$ be the conditional mean and variance of h_k given $\{r_j : j = k+1, \ldots, K-1\}$ (backward prediction) and $\mu_{k|k}$ and $\Xi_{k|k}$ be the conditional mean and variance of h_k given

$\{r_j : j = k, \ldots, K-1\}$ (backward filtering). The resulting recursions are given by

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + \frac{\Sigma_{k|k-1}\mathfrak{a}_k^*}{|\mathfrak{a}_k|^2(\Sigma_{k|k-1}-1)+2\sigma^2+\mathfrak{b}_k}\left(r_k - \mathfrak{a}_k m_{k|k-1}\right) \\ \Sigma_{k|k} &= \frac{2\sigma^2+\mathfrak{b}_k-|\mathfrak{a}_k|^2}{|\mathfrak{a}_k|^2(\Sigma_{k|k-1}-1)+2\sigma^2+\mathfrak{b}_k}\Sigma_{k|k-1} \\ m_{k+1|k} &= \rho m_{k|k} \\ \Sigma_{k+1|k} &= \rho^2\Sigma_{k|k}+1-\rho^2 \end{aligned} \tag{5.90}$$

for $k = 0, \ldots, K-1$, with initial conditions $\Sigma_{0|-1} = 1$ and $m_{0|-1} = 0$, and by

$$\begin{aligned} \mu_{k|k} &= \mu_{k|k+1} + \frac{\Xi_{k|k+1}\mathfrak{a}_k^*}{|\mathfrak{a}_k|^2(\Xi_{k|k+1}-1)+2\sigma^2+\mathfrak{b}_k}\left(r_k - \mathfrak{a}_k \mu_{k|k+1}\right) \\ \Xi_{k|k} &= \frac{2\sigma^2+\mathfrak{b}_k-|\mathfrak{a}_k|^2}{|\mathfrak{a}_k|^2(\Xi_{k|k+1}-1)+2\sigma^2+\mathfrak{b}_k}\Xi_{k|k+1} \\ \mu_{k-1|k} &= \rho\mu_{k|k} \\ \Xi_{k-1|k} &= \rho^2\Xi_{k|k}+1-\rho^2 \end{aligned} \tag{5.91}$$

for $k = K-1, \ldots, 0$, with initial conditions $\Xi_{K-1|K} = 1$ and $\mu_{K-1|K} = 0$.

Finally, for each k we obtain

$$\begin{aligned} m_k &= \frac{\Xi_{k|k+1}}{\Sigma_{k|k-1}+\Xi_{k|k+1}} m_{k|k-1} + \frac{\Sigma_{k|k-1}}{\Sigma_{k|k-1}+\Xi_{k|k+1}}\mu_{k|k+1} \\ \Sigma_k &= \frac{\Sigma_{k|k-1}\Xi_{k|k+1}}{\Sigma_{k|k-1}+\Xi_{k|k+1}} \end{aligned} \tag{5.92}$$

It remains to find an expression for the message $P_u(c_k)$, that is, the probability of the code symbol c_k given the observation r_k and the phasor estimate $h_k \in \mathcal{N}_{\mathbb{C}}(m_k, \Sigma_k)$. We let $h_k = R_k e^{j\theta_k}$ and, after some manipulations, we obtain (see Problem 5.14):

$$\begin{aligned} P_u\{c_k\} &\sim \int \exp\left(-\frac{1}{2\sigma^2}|r_k - c_k e^{j\theta_k}|^2\right) g_{\mathbb{C}}\left(m_k, \Sigma_k, R_k e^{j\theta_k}\right) R_k \mathrm{d}R_k \mathrm{d}\theta_k \\ &\sim e^{-\frac{|c_k|^2}{2\sigma^2}} \int_0^\infty e^{-z} I_0\left(\sqrt{\frac{|r_k c^*|^2}{\sigma^4} + 4\frac{z|m_k|^2}{\Sigma_k} + 4\frac{\sqrt{z}}{\sqrt{\Sigma_k}\sigma^2}\Re\{r_k c^* m_k^*\}}\right) \mathrm{d}z. \end{aligned} \tag{5.93}$$

The above integral can be easily computed using Gauss Laguerre quadrature rules [23].

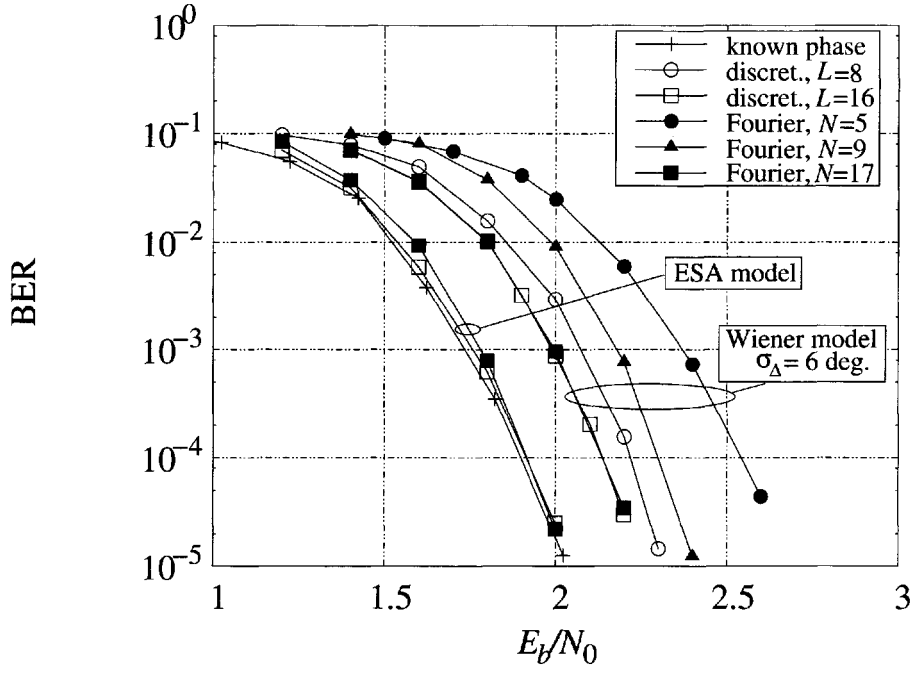


Figure 5.19: Performance of the algorithms based on discretization of channel parameters and Fourier parameterization. BPSK and two different phase models are considered.

5.7.3 Numerical Results

In this section, we assess the performance of the proposed schemes. Unless otherwise stated, the considered code is a regular (3,6) LDPC code with codewords of length 4000 [215]. A maximum of 200 iterations for the SP algorithm on the overall graph is allowed, and BPSK is the considered modulation format. Pilot symbols are inserted in the transmitted codeword in order to make the iterative decoding algorithms bootstrap. As previously observed, the gap between the "known phase" curve and the others is uniquely due to the need for phase estimation/compensation, and not to the rate decrease caused by the insertion of pilot symbols.

In Figure 5.19, two of the algorithms described in Section 5.7.2 (discretization of channel parameters and Fourier parameterization) have been considered, assuming the insertion of a pilot symbol every 20 transmitted symbols. The ESA phase noise model and a more severe Wiener model (5.35) with $\sigma_\Delta = 6$ degrees have been considered. This latter case has been used to stress the robustness of the described schemes to a strong phase noise and to select the best algorithm, from a performance-

complexity trade-off point of view, to be used for high order modulations. In the case of the ESA model, all the receivers were designed by assuming a Wiener phase noise model with $\sigma_\Delta = 0.3$ degrees.

In the case of the Wiener model, different values of the number L of discretization levels and different values of the number N of considered Fourier coefficients have been considered. No improvement has been observed for $L > 16$ and this is in agreement with a result in [173]. Similarly, values of $N > 17$ are not considered since they do not produce any performance improvement. Therefore, the value of $N = 17$ (i.e., $-8 \leq \ell \leq 8$ in all the equations of Section 5.7.2) can be considered as optimal for $\sigma_\Delta = 6$ degrees. Hence, a gap of about 0.2 dB with respect to the curve labeled "known phase" is only due to the loss in channel capacity for a time-varying channel phase.

In the binary case considered in the previous figure, the proposed algorithms have a practically optimal performance and a similar complexity. However, for a modulation format characterized by a more dense constellation, if for the discretization-based algorithm the optimal number of discretization levels, and thus the complexity, must be increased, it can be expected that the number N of considered Fourier coefficients in the proposed algorithm remains practically the same. This aspect is shown in Figure 5.20, where QPSK ($M = 4$) is used. The phase noise has $\sigma_\Delta = 6$ degrees and even in this case we have a pilot symbol in every block of 20 transmitted symbols. For the discretization-based algorithm $L = 8 \times M = 32$ quantization levels are considered, whereas for the algorithm based on Fourier parameterization, the number of Fourier coefficients is still $N = 17$.

In Figure 5.21, the performance of the algorithms based on Tikhonov and Gaussian parameterizations is shown under the same conditions as Figure 5.19. One can observe that, despite the very low complexity, these algorithms have practically the same performance of more computationally demanding algorithms based on discretization and Fourier parameterization. This fact can be also observed from Figure 5.22, where all the considered algorithms are compared for a Wiener phase model with $\sigma_\Delta = 6$ degrees. In this figure, the performance of two other algorithms described in the literature is also shown for the sake of comparison: the first one is the "ultra fast" algorithm with overlapped windows described in [93], with the value of N optimized by computer simulation; the second one is based on the EM algorithm [220–222]. In order to adapt the algorithm to a time-varying channel phase, different phase estimates are computed for each code symbol, taking into account the contribution of the adjacent symbols belonging to a window whose dimension is optimized by computer simulation. For this reason, the algorithm is denoted by EM with sliding window (EM-SW). We found that the optimal window has a width of 60 symbols for the considered phase noise. In both cases, the performance loss is

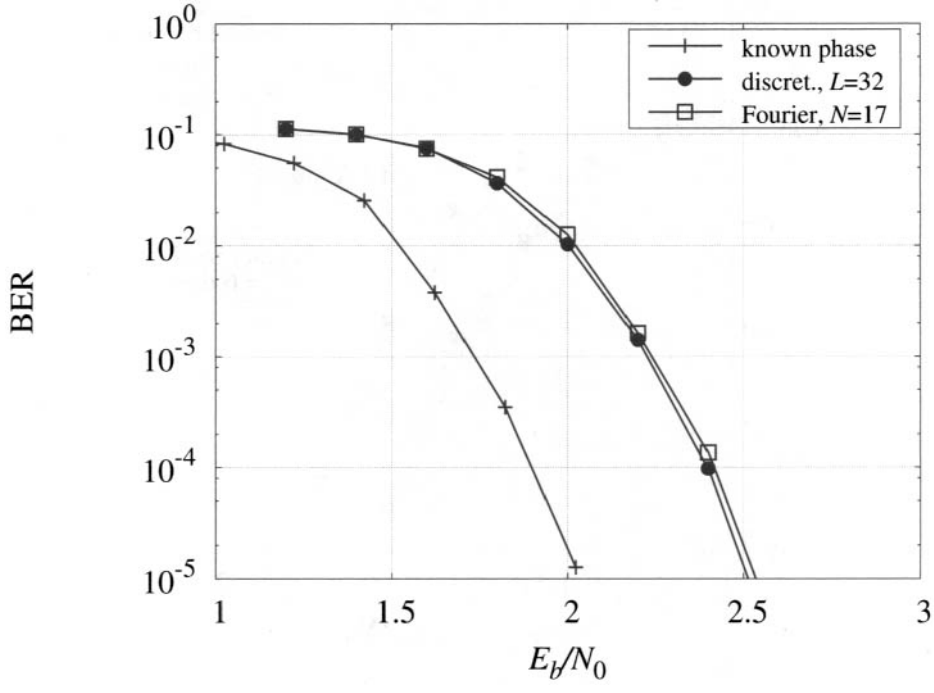


Figure 5.20: Performance of the algorithms based on discretization of channel parameters and Fourier parameterization. QPSK and the Wiener model with $\sigma_\Delta = 6$ degrees are considered.

due to the fact that these two algorithms are designed for a different phase model, i.e., a block-constant phase. Based on the above experiments and on extensive numerical evidence, it is possible to conclude that all the algorithms proposed in Section 5.7.2 exhibit a practically optimal performance (i.e., they perform as well as the discretization approach). Among them, those based on Tikhonov and Gaussian parameterizations, because of their low complexity (roughly equivalent to that of the EM-SW algorithm), represent the best candidates. For this reason, these two algorithms will be considered in the remaining results. By comparing the results in Figure 5.22 with those in Figure 5.15, one can also observe that, for $\sigma_\Delta = 6$ degrees, the linear-predictive algorithm described in Section 5.6.2 has a practically optimal performance, although with a complexity higher than that of the algorithm based on Tikhonov parameterization.

The sensitivity to distributions of the pilot symbols is considered in Figure 5.23. In the case of the Wiener model with $\sigma_\Delta = 6$ degrees, two different distributions are considered, namely 1 pilot symbol in each block of 20 consecutive bits and 20

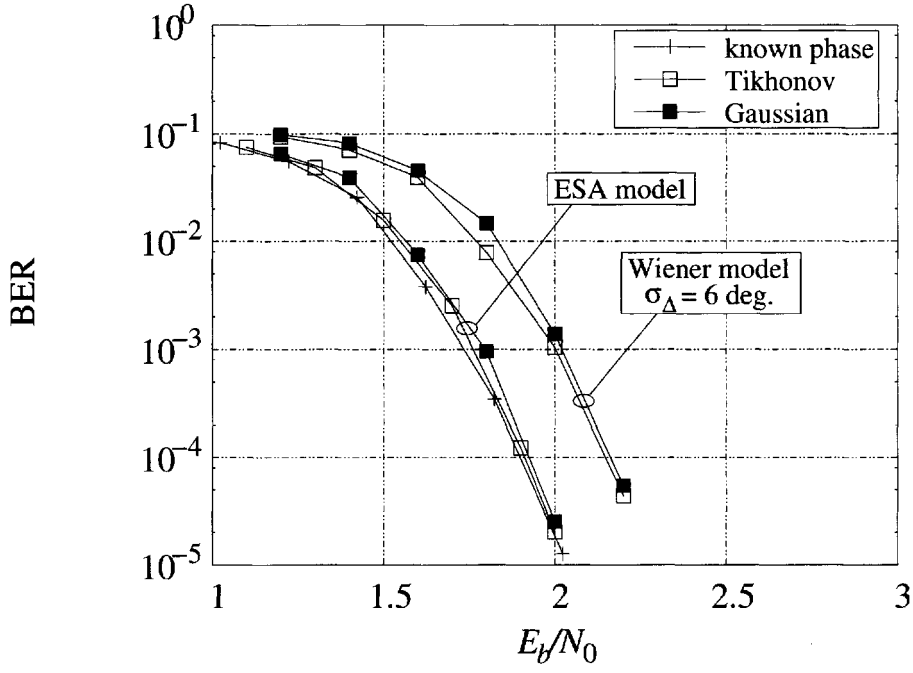


Figure 5.21: Performance of the algorithms based on Tikhonov and Gaussian parameterizations. BPSK and two different phase models are considered.

pilots in each block of 400 consecutive bits (hence, the effective transmission rate is the same). We may observe that the algorithm based on Tikhonov parameterization is almost insensitive to the pilot symbol insertion strategy thanks to the algorithm modification described in Section 5.7.2. A similar modification is not possible in the case of the algorithm based on Gaussian parameterization, since it can be shown that the choice of a dominant term corresponds to a hard decision based uniquely on the decoder outcome $P_d\{c_k\}$. We verified that this modification of the algorithm does not provide any performance improvement. Note that, in general, the distribution of the pilots has to be optimized for the specific detection algorithm employed.

Finally, we consider the application of the algorithms to the DVB-S2 system. We consider two standardized LDPC codes with codeword length equal to 64,800 [226]: the first one has rate 2/3 and is mapped onto an 8-PSK modulation; the second one has rate 4/5 and is mapped onto a 32-amplitude phase shift keying (APSK) modulation.[11] A maximum number of 50 iterations is considered and 36 pilot symbols

[11] In this case, the constellation is composed of three concentric PSK constellations with different radii, namely a QPSK, a 12-PSK and a 16-PSK. See Problem 2.2.

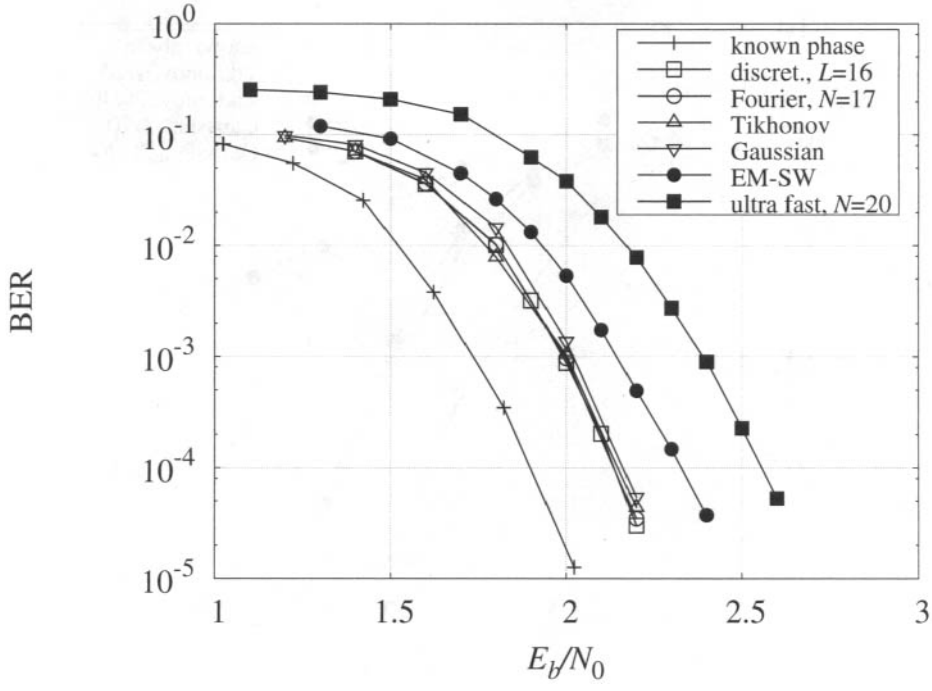


Figure 5.22: Performance of all the proposed algorithms and comparison with other algorithms proposed in the literature. BPSK and the Wiener phase model with $\sigma_\Delta =$ 6 degrees are considered.

are inserted every 1476 symbols, as prescribed by the existing standard [226]. The above mentioned phase noise ESA model is considered. The performance is shown in Figure 5.24. For the algorithm based on Tikhonov parameterization, the loss due to phase noise is less than 0.1 dB in both cases. Notice that a further improvement in performance may be obtained if the maximum number of iterations is not limited to 50. The Kalman smoother (Gaussian parameterization) does not perform as well mainly because of the bursty allocation of pilot symbols.

5.8 Summary

In this chapter, we have extended the general framework developed in the previous chapters for VA-based MAP sequence detection and FB-based MAP symbol detection, to the case of graph-based MAP symbol detection. In particular, using the FMC or the CMP, we have described how to build a factor graph representing the joint a

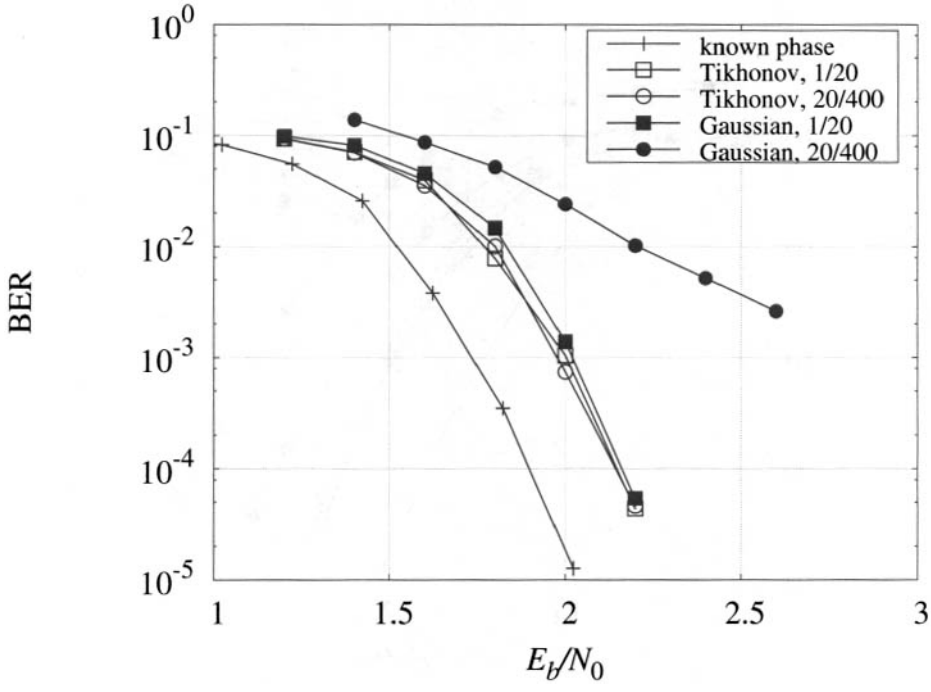


Figure 5.23: Performance of the algorithms based on Tikhonov and Gaussian parameterizations. BPSK and two different pilot distributions are considered.

posteriori probability of the information symbols and taking into account channel statistics and code constraints. In this FG, the channel parameters are not explicitly represented. The desired marginals necessary to implement MAP symbol detection are approximately derived using the SP algorithm. The cases of ISI, phase-uncertain and flat-fading channels have been considered in the numerical results, showing the effectiveness of the described approach.

An alternative approach inspired by [192] has also been considered for phase-uncertain channels. In this case, the channel parameters are explicitly represented in the FG and the method of canonical distributions has been used to simplify the implementation of the SP algorithm. We have shown that a clever choice of the used canonical distribution allows one to derive graph-based detection algorithms with very low complexity and practically optimal performance.

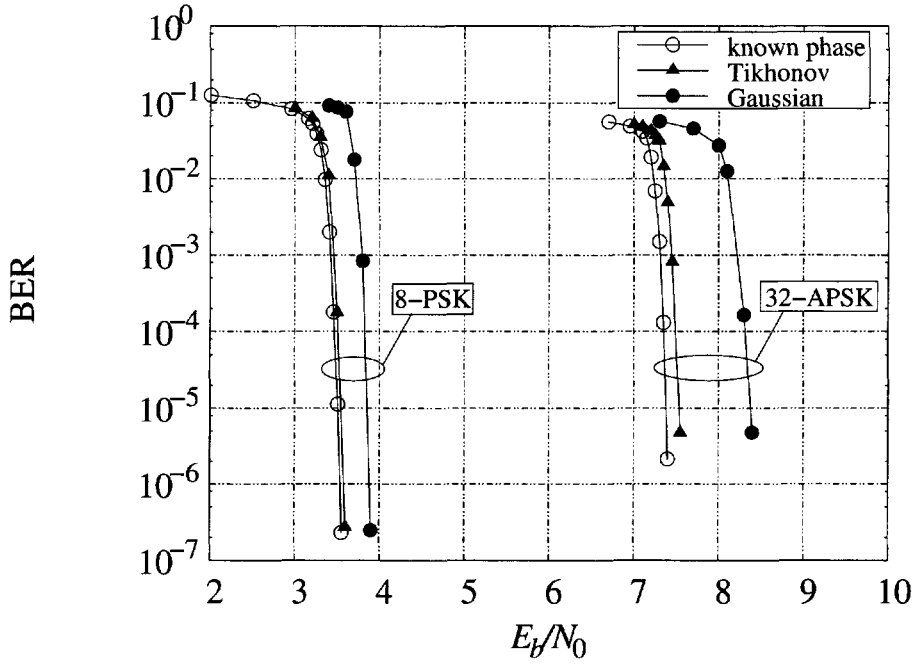


Figure 5.24: Performance of the algorithms based on Tikhonov and Gaussian parameterizations. The ESA phase model is considered along with 8-PSK and 32-APSK modulations.

5.9 Problems

Problem 5.1: Consider a rate-2/3 binary block code with parity check matrix

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

and recall that $\boldsymbol{c}$ is a codeword iff $\boldsymbol{H}\boldsymbol{c}^{\mathrm{T}} = 0$, where $\boldsymbol{c} \triangleq (c_1, c_2, c_3)$. Assume that this code is transmitted over a channel which introduces AWGN noise and let $\boldsymbol{r} = (r_1, r_2, r_3)$ denote the received sequence.

A. Draw the factor graph of the joint APP $P(\boldsymbol{c}|\boldsymbol{r})$ and verify that the graph does not contain cycles.

B. Using the sum-product algorithm, compute the marginals $P(c_i|\boldsymbol{r})$, $i = 1, 2, 3$.

C. Compute the LLRs

$$\mathrm{LLR}_i = \log \frac{P\{c_i = 0|\boldsymbol{r}\}}{P\{c_i = 1|\boldsymbol{r}\}} \quad i = 1, 2, 3.$$

Problem 5.2: Derive the message computations (5.9) and (5.10) from (5.3) and (5.4).

Problem 5.3: Show that the FB algorithm described in Chapter 3 can be interpreted as an instance of the SP algorithm applied to Wiberg graphs with a natural forward backward schedule.

Problem 5.4: For an ISI channel, demonstrate that if the differences between the indexes of the channel interferers are all different, the graph has girth 6.

Problem 5.5: For an ISI channel with $L = 4$ and $f_2 = 0$ and an uncoded transmission (lower part of the graph in Figure 5.7), identify all possible stretching transformations which allow one to obtain a girth-6 graph.

Problem 5.6: For an ISI channel with $L = 4$ and $f_2 = 0$ and an uncoded transmission (lower part of the graph in Figure 5.7), transform the original graph into a cycle-Wiberg graph.

Problem 5.7: Substituting (5.31) into (5.17), show that for frequency flat fading channels the joint *a posteriori* pmf of the information symbols $P(\boldsymbol{a}|\boldsymbol{r})$ has the expression (5.32).

Problem 5.8: Compute the phase noise power spectral density for the ESA model described in Section 5.7.1.

Problem 5.9: Restate the algorithm based on Fourier parameterization in Section 5.7.2 assuming the following phase model:

$$\theta_n = \theta_{n-1} + 2\pi\nu_n T$$

where T is the symbol interval and $\{\nu_n\}$ is a sequence of independent and identically distributed frequency offsets with uniform distribution in $[-\alpha/T, \alpha/T]$ (α is a known constant).

Problem 5.10: Let $p(x)$ and $q(x)$ be two probability density functions. The divergence $D(p||q)$ (also known as cross-entropy, or Kullbach Leibler distance [35]) is defined by

$$D(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x.$$

A. Let $f(x)$ be a probability density function of a real random variable with mean μ_f and variance σ_f^2. Show that the solution to the divergence minimization problem $\min_{g \in \mathcal{G}} D(f\|g)$, where $\mathcal{G}$ is the set of all real Gaussian pdfs, is given by $g(\mu_f, \sigma_f^2, x)$, i.e., it is the Gaussian pdf with the same mean and variance.

B. Let $f(x)$ be the probability density function of a complex random variable with mean μ_f and variance σ_f^2. Show that the solution to the divergence minimization problem $\min_{g \in \mathcal{G}'} D(f\|g)$, where $\mathcal{G}'$ is the set of all complex circularly-symmetric Gaussian pdfs, is given by $g_{\mathbb{C}}(\mu_f, \sigma_f^2, x)$, i.e., it is the complex circularly-symmetric Gaussian pdf with the same mean and variance.

Problem 5.11: Reformulate the algorithm based on Tikhonov parameterization derived in Section 5.7.2 in the case of a flooding schedule.

Problem 5.12: Prove (5.85).

Problem 5.13: Given two Gaussian distributions $g_{\mathbb{C}}(A_1, \Sigma_1, x)$ and $g_{\mathbb{C}}(A_2, \Sigma_2, x)$, show that (5.89) is true.

Problem 5.14: Show that (5.93) is true.

Appendix A

Discretization by Sampling

A.1 Introduction

In several chapters of this book, *sampling* has been considered at the receiver side to extract a sufficient statistic for data detection. Most of the time, it has been simply assumed that one sample per symbol interval is "sufficient." In this appendix, we discuss statistical *sufficiency* of a sequence of discrete-time signal samples, and we derive relevant sufficient conditions. This derivation provides a theoretical justification of many signal discretization schemes considered in the previous chapters. As a representative example of application of the concept of sufficient statistics, throughout the appendix we will consider the case of a frequency and time selective fading channel.

We first consider a continuous-time model of the communication system, and we derive conditions for the receiver filter to provide sufficient statistics at its output. We finally consider the corresponding discrete-time model.

A.2 Continuous-Time Signal Model

The considered transmission system is shown in Figure A.1, in which the information sequence $\{a_k\}$ consists of independent, M-ary valued, equiprobable complex symbols with rate $1/T$, T being the symbol interval. We denote by $g(t)$ and $v(t)$ the impulse responses of transmitter and receiver filters, respectively. The physical channel is modeled as a time-varying linear filter with impulse response $h(\tau, t)$, defined as the response at time t to a delta pulse applied at time $t - \tau$. Thermal noise $n(t)$ is assumed to be zero-mean and white with one-sided power spectral density N_0. The output $r(t)$ of the receiver filter is sampled with period T_s related to the

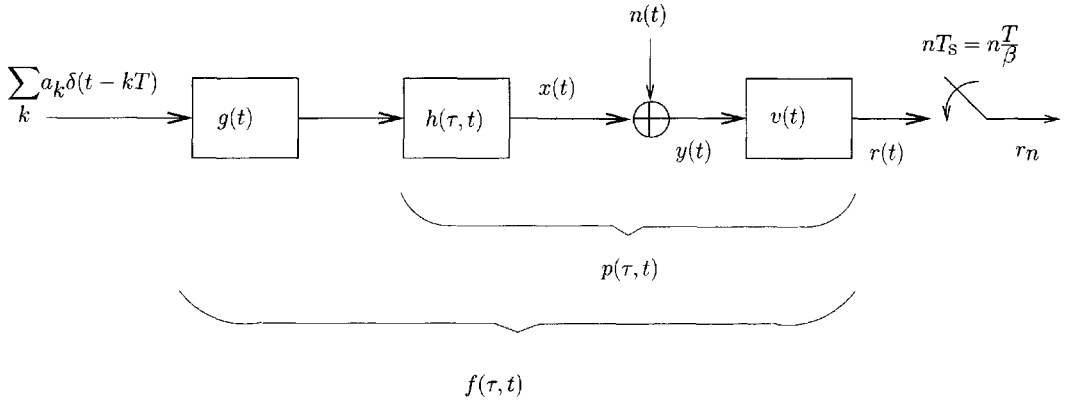


Figure A.1: Transmission system model.

symbol period by $T = \beta T_s$, where β is assumed to be an integer interpretable as an *oversampling factor.*

In order to obtain an expression for signal $r(t)$ at the input of the sampler, we may consider the following time-varying impulse responses: $p(\tau, t)$, denoting the cascade of $g(t)$ and $h(\tau, t)$, and $f(\tau, t)$, denoting the cascade of $p(\tau, t)$ and the receiver filter $v(t)$. By straightforward manipulations, one obtains:

$$p(\tau,t) = \int_{-\infty}^{\infty} g(\alpha)h(\tau-\alpha,t)\,\mathrm{d}\alpha \tag{A.1}$$

$$f(\tau,t) = \int_{-\infty}^{\infty} v(\lambda)p(\tau-\lambda,t-\lambda)\,\mathrm{d}\lambda. \tag{A.2}$$

Substituting (A.1) into (A.2) and changing the integration variables by letting $\xi = \alpha + \lambda$, it follows that

$$f(\tau,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} v(\lambda)g(\xi-\lambda)h(\tau-\xi,t-\lambda)\,\mathrm{d}\xi\mathrm{d}\lambda. \tag{A.3}$$

Hence, the signal $r(t)$ may be expressed as

$$r(t) = \sum_k a_k f(t-kT,t) + n_c(t) \tag{A.4}$$

where $n_c(t) \triangleq n(t) \otimes v(t)$ is the colored noise process at the output of the receiver filter (the symbol $\otimes$ denotes convolution).

In the case of a frequency and time selective fading channel, the time-varying physical channel may be modeled as a transversal filter with the following time-varying impulse response:

$$h(\tau, t) = \sum_{l=0}^{E-1} h_l(t)\ \delta(\tau - \tau_l) \tag{A.5}$$

where E is the number of echos due to multipath propagation, $h_l(t)$ are time-varying tap weights, and τ_l are the echo delays. Note that a frequency nonselective fading channel can be interpreted as a special case with $E = 1$.

In the case of Rayleigh fading, the weights $h_l(t)$ are modeled as zero-mean, independent, complex, Gaussian stochastic processes with variance $\sigma_{h_l}^2$, each with independent real and imaginary components of identical power spectrum. This model, commonly used in applications [58], satisfies the assumptions of wide-sense stationary uncorrelated scattering (WSSUS) and the channel Doppler power spectrum coincides with the power spectrum of the tap weights [229]. Using (A.5), we may specialize (A.3) as follows:

$$f(\tau, t) = \sum_{l=0}^{E-1} \int_{-\infty}^{\infty} v(\lambda) h_l(t - \lambda)\, g(\tau - \tau_l - \lambda)\, \mathrm{d}\lambda. \tag{A.6}$$

In order to describe the assumptions on the transmitter and receiver filters, let us denote with $N_{\epsilon,T}(f)$ a function of f satisfying the Nyquist criterion for absence of inter-symbol interference (ISI) with vestigial symmetry around $1/2T$ and excess bandwidth factor ϵ, such as a raised-cosine function with roll-off ϵ [17]. The Fourier transform of the transmitter filter is $G(f) = \sqrt{N_{\epsilon,T}(f)}$ and that of the receiver filter, denoted by $V(f)$, is specified in the following subsections.

A.2.1 Power Spectrum of a Rayleigh Faded Signal

In the remainder of the appendix we will consider, as a representative communication scenario where oversampling can be used, the case of transmission over a fast Rayleigh fading channel. Before describing in more detail oversampling, we first derive the expression of the signal power spectrum at the output of a Rayleigh fading channel.

In order to determine the power spectrum of the received signal, we refer to the transmission system shown in Figure A.1, where: $a(t) = \sum_k a_k \delta(t - kT)$ is the

signal generated by the source; $g(t)$ is the transmission filter at the input of the time-varying channel; $h(\tau, t)$ is the time-varying channel impulse response and has the expression (A.5); $x(t)$ is the output signal of the time-varying channel filter and $n(t)$ is the AWGN process independent from $x(t)$.

The overall received signal $y(t)$ at the input of the receiver is thus given by the sum of two independent stochastic processes. The autocorrelation and power spectrum of the received signal $y(t)$ is then simply the sum of the corresponding quantities for $x(t)$ and $n(t)$. For this reason, it is sufficient to compute the autocorrelation function and the power spectrum of the process $x(t)$ containing the source message. Recalling the expressions of the transmitted signal and the channel impulse response $h(\tau, t)$, the process $x(t)$ can be expressed as follows:

$$x(t) = \sum_{k=-\infty}^{k=\infty} a_k \sum_{l=0}^{E-1} h_l(t) g(t - kT - \tau_l) = \sum_{l=0}^{E-1} h_l(t) \sum_{k=-\infty}^{k=\infty} a_k g(t - kT - \tau_l). \quad \text{(A.7)}$$

The autocorrelation function of $x(t)$ is:

$$\begin{aligned} R_x(t + \Delta t, t) &= \mathrm{E}\{x(t + \Delta t) x^*(t)\} \\ &= \mathrm{E}\left\{ \sum_{l=0}^{E-1} h_l(t + \Delta t) \sum_{k=-\infty}^{k=\infty} a_k g(t + \Delta t - kT - \tau_l) \right. \\ &\qquad \left. \sum_{m=0}^{E-1} h_m^*(t) \sum_{j=-\infty}^{k=\infty} a_l^* g^*(t - jT - \tau_m) \right\} \\ &= \sum_{l=0}^{E-1} \sum_{m=0}^{E-1} \mathrm{E}\{h_l(t + \Delta t) h_m^*(t)\} \sum_{k=-\infty}^{k=\infty} \sum_{j=-\infty}^{k=\infty} \\ &\qquad \cdot \mathrm{E}\left\{a_k a_j^*\right\} g(t + \Delta t - kT - \tau_l) g^*(t - jT - \tau_m). \quad \text{(A.8)} \end{aligned}$$

Recalling that the transmitted symbols $\{a_k\}$ are independent and assuming[1] that $\mathrm{E}\{a_k\} = 0$, it follows that

$$\mathrm{E}\left\{a_k a_j^*\right\} = \begin{cases} \mathrm{E}\{|a_k|^2\} = \sigma_a^2 & \text{if} \quad k = j \\ 0 & \text{if} \quad k \neq j. \end{cases} \quad \text{(A.9)}$$

According to the WSSUS model of the considered Rayleigh fading channel, the pro-

[1] This is respected for most modulation formats of practical interest. The general case can be easily dealt with by standard methods.

cesses $\{h_i(t)\}_{i=0}^{E-1}$ are independent, so that

$$\mathrm{E}\{h_l(t+\Delta t)h_m^*(t)\} = \begin{cases} R_{h_l}(\Delta t) & \text{if} \quad l=m \\ 0 & \text{if} \quad l \neq m \end{cases} \tag{A.10}$$

where $R_{h_l}(\Delta t)$ is the autocorrelation function of $h_l(t+\Delta t)$. According to the isotropic scattering model (Clarke's model [20, 21]), $R_{h_l}(\Delta t)$ has the following expression:

$$R_{h_l}(\Delta t) = \sigma_{h_l}^2 \, \mathrm{J}_0(2\pi f_\mathrm{D} \Delta t) \tag{A.11}$$

where $\sigma_{h_l}^2$ is the power of the l-th fading process, $\mathrm{J}_0(\cdot)$ is the Bessel function of zero order, and f_D is the maximum Doppler shift.

By using (A.9) and (A.10) in (A.8), the autocorrelation of the transmitted signal can finally be written as follows:

$$R_x(t+\Delta t, t) = \sigma_a^2 \sum_{l=0}^{E-1} R_{h_l}(\Delta t) \sum_{k=-\infty}^{\infty} g(t+\Delta t - kT - \tau_l) g^*(t - kT - \tau_l). \tag{A.12}$$

From expression (A.12), one can immediately conclude that $R_x(t+\Delta t, t)$ is a periodic function with period T in the variable t, so that $x(t)$ is cyclostationary. Using stationarization techniques, e.g., time averaging $R_x(t+\Delta t, t)$ with respect to t, the autocorrelation function $R'_x(\Delta t)$ of the stationarized process becomes:

$$\begin{aligned} R'_x(\Delta t) &= \frac{1}{T} \int_0^T R_x(t+\Delta t, t) \, \mathrm{d}t \\ &= \frac{\sigma_a^2}{T} \sum_{l=0}^{E-1} R_{h_l}(\Delta t) \sum_{k=-\infty}^{\infty} \int_0^T g(t+\Delta t - kT - \tau_l) g^*(\underbrace{t - kT - \tau_l}_{=-\lambda}) \, \mathrm{d}t \\ &= \frac{\sigma_a^2}{T} \sum_{l=0}^{E-1} R_{h_l}(\Delta t) \sum_{k=-\infty}^{\infty} \int_{(k-1)T+\tau_l}^{kT+\tau_l} g(\Delta t - \lambda) g^*(-\lambda) \, \mathrm{d}\lambda \\ &= \frac{\sigma_a^2}{T} \sum_{l=0}^{E-1} R_{h_l}(\Delta t) \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} g(\Delta t - \lambda) g^*(-\lambda) \, \mathrm{d}\lambda \\ &= \frac{\sigma_a^2}{T} \sum_{l=0}^{E} R_{h_l}(\Delta t) \rho_g(\Delta t) \end{aligned} \tag{A.13}$$

where

$$\rho_g(\Delta t) \triangleq g(\Delta t) \otimes g^*(-\Delta t). \tag{A.14}$$

Using (A.11) in (A.13), one obtains:

$$\begin{aligned} R'_x(\Delta t) &= \frac{\sigma_a^2}{T}\rho_g(\Delta t)\mathrm{J}_0(2\pi f_\mathrm{D}\Delta t)\sum_{l=0}^{E-1}\sigma_{h_l}^2 \\ &= \frac{\sigma_h^2\sigma_a^2}{T}\rho_g(\Delta t)\mathrm{J}_0(2\pi f_\mathrm{D}\Delta t) \end{aligned} \tag{A.15}$$

where

$$\sigma_h^2 \triangleq \sum_{l=0}^{E}\sigma_{h_l}^2 \tag{A.16}$$

is the overall power of the fading processes. The power spectrum $S_x(f)$ of the process $x(t)$ can be obtained considering the Fourier transform, denoted by the notation $\mathcal{F}(\cdot)$ of the stationarized autocorrelation function of the process $x(t)$:

$$\begin{aligned} S_x(f) &= \mathcal{F}\{R'_x(\Delta t)\} = \frac{\sigma_a^2\sigma_h^2}{T}\mathcal{F}\{\rho_g(\Delta t)\} \otimes \mathcal{F}\{\mathrm{J}_0(2\pi f_\mathrm{D}\Delta t)\} \\ &= \frac{\sigma_a^2\sigma_h^2}{T}|G(f)|^2 \otimes S_D(f) \end{aligned} \tag{A.17}$$

where $S_D(f)$ is the normalized power spectral density of the fading process according to Clarke's model and has the expression (1.3). Using known properties of convolution, one can immediately conclude that the bandwidth B of the process $x(t)$, i.e., the bandwidth of its power spectrum, is the sum of the bandwidth of the power spectral densities corresponding to the autocorrelation functions $\rho_g(\Delta t)$ and $\mathrm{J}_0(2\pi f_\mathrm{D}\Delta t)$. Considering, for example, a transmitter filter $g(t)$ with transfer function given by a raised-cosine function with roll-off ϵ, the bandwidth of the transfer function corresponding to $\rho_g(\Delta t)$ is $(1+\epsilon)/2T$. The power spectral density of the second term is instead the Doppler spectrum, with maximum Doppler shift equal to f_D. The overall bandwidth of the process $x(t)$ is therefore the following:

$$B = \frac{1+\epsilon}{2T} + f_\mathrm{D}. \tag{A.18}$$

A.2.2 Signal Oversampling

Assume that the received signal component $x(t)$ is a strictly band-limited stationary random process with bandwidth B. As an example, for a Rayleigh faded signal this bandwidth is the result of the Doppler spread introduced by the channel on the (base-band) transmission bandwidth $(1+\epsilon)/2T$ and may be easily shown to have the expression (A.18).

It is well known that the sampled sequence is a sufficient statistic for data detection if:

- the receiver filter frequency response is such that [94]

$$V(f) = \begin{cases} \text{arbitrary but } \neq 0 & \text{for } |f| < B \\ \text{arbitrary} & \text{for } B \leq |f| < B_V \\ 0 & \text{otherwise} \end{cases} \tag{A.19}$$

  in which B_V denotes the filter bandwidth;

- the sampling rate satisfies the condition

$$\frac{1}{T_s} \geq B + B_V. \tag{A.20}$$

The proof that a filter satisfying (A.19) and (A.20) produces, after sampling, sufficient statistics for the estimation of the information sequence can be based on the concept of *reversibility* [92]. According to this concept, any reversible (or invertible) transformation carried out on the received signal does not modify the performance of an optimal receiver [92]. In order to clarify this concept in the context of the considered scenario, we assume that the received signal $y(t)$ is processed in two different ways, and we then show that the results of these processing operations are statistically equivalent. The two different processing schemes are shown in the upper and lower branches, respectively, in Figure A.2.

- As one can observe from Figure A.2, in the upper branch the received signal $y(t)$ passes through an ideal lowpass analog filter with the following transfer function:

$$H_L(f) = \begin{cases} 1 & |f| < B \\ 0 & \text{otherwise} \end{cases} \tag{A.21}$$

  The signal $r_a(t)$ at the output of the filter with transfer function $H_L(f)$ is

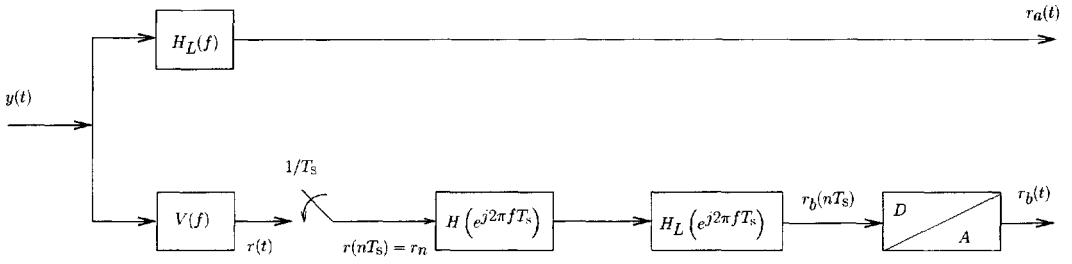


Figure A.2: Application of the concept of reversibility.

given by the sum of the signal component $x(t)$ (see Figure A.1)—unmodified by this filtering operation (since the filter bandwidth coincides with the signal bandwidth)—and a noise component $n_a(t)$, which is now band-limited. The signal $r_a(t)$ is a sufficient statistic for the estimation of the information sequence. In fact, it is possible to invert this lowpass filtering operation, which modifies only the noise component, by adding to $r_a(t)$ a suitable highpass independent Gaussian noise process: the obtained process is statistically equivalent to the original noise process $n(t)$ affecting $y(t)$. Since $r_a(t)$ is band-limited, it can be sampled with no information loss. The resulting samples $\{r_a(nT_s)\}$ at the Nyquist rate $f_s = 1/T_s = 2B$ are discrete sufficient statistics. Note that this solution is simply an idealization of practical cases, since an ideal lowpass filter cannot be implemented in practice.

- The lower branch in Figure A.2 corresponds to a practical receiver structure. In particular, the signal $y(t)$ passes through a filter with transfer function $V(f)$ which satisfies (A.19): after filtering, the output signal is sampled with a suitable period T_s satisfying (A.20). The obtained samples $\{r(nT_s)\}$ are a sufficient statistic for the estimation of the information sequence: in fact, from them it is possible to derive another sufficient statistic, equivalent to the signal $r_a(t)$. This is due to the reversibility of the filtering and sampling operation. To this purpose, the discrete-time process $\{r(nT_s)\}$ can be filtered with a digital filter with frequency response $H(e^{j2\pi fT_s})$ which cancels out the effects of the analog filter $V(f)$ in the bandwidth of the signal component $x(t)$. From a mathematical viewpoint, canceling out the effects of a filter in a given band consists of requiring that the second filter has a transfer function equal to the inverse of the transfer function of the first filter, so that the transfer function obtained as the product of the two is constant in the interval of interest (in our case, the band

$[-B, B]$). In other words:

$$H\left(e^{j2\pi fT_s}\right) = \begin{cases} \dfrac{1}{V(e^{j2\pi fT_s})} & 0 < f < B\,,\ \frac{1}{T_s} - B < f < \frac{1}{T_s} \\ \text{arbitrary} & B < f < \frac{1}{T_s} - B \end{cases} \tag{A.22}$$

where $V(e^{j2\pi fT_s})$ is the frequency response of the digital filter with impulse response $\{v_n\} \triangleq \{v(nT_s)\}$ equivalent to the analog filter with impulse response $v(t)$ (impulse invariance [155]). The effect of $H(e^{j2\pi fT_s})$ is the desired one: within frequency intervals centered around $0, 1/T_s, 2/T_s, \ldots$ and with bandwidth equal to $2B$, the power spectrum of the signal component at the output of this digital filter remains unchanged with respect to the input.

The reconstruction operation is completed by sending the samples $\{r(nT_s)\}$ to an ideal lowpass digital filter with transfer function $H_L(e^{j2\pi fT_s})$—the digital equivalent of (A.21)—which eliminates the noise spectral components outside the bandwidth B. Finally, an analog signal $r_b(t)$ is obtained from the samples $\{r_b(nT_s)\}$ through an analog/digital converter equipped with an analog lowpass filter with the following transfer function:

$$H_C(f) = \begin{cases} 1 & |f| < \frac{1}{2T_s} \\ 0 & \text{otherwise.} \end{cases} \tag{A.23}$$

In [94], the authors show that the two analog signals $r_a(t)$ and $r_b(t)$ at the output of the two processing branches considered in Figure A.2 are such that

$$\mathrm{E}\{|r_a(t) - r_b(t)|^2\} = 0 \qquad \forall t. \tag{A.24}$$

In other words, (A.24) means that the signals $r_a(t)$ and $r_b(t)$ are equal in the mean square error sense. It therefore follows that $r_b(t)$ is a sufficient statistic as well. Since the signal $r_b(t)$ is obtained (indirectly) from the samples $\{r(nT_s)\}$, for the reversibility principle it is immediately concluded that the samples $\{r(nT_s)\}$ also are a sufficient statistic for the estimation of the transmitted information sequence.

Having shown that the samples $\{r(nT_s)\}$ represent a sufficient statistic if the analog filter $V(f)$ has the form given by (A.19) does not necessarily mean that the signal $r(t)$ can be reconstructed from its samples $\{r(nT_s)\}$. In fact, the initial filter has a bandwidth B_V which can be larger than $1/2T_s$, and this implies that the sampling theorem, which imposes the condition $f_s = 1/T_s \geq 2B_V$ cannot be applied to the noise component in $y(t)$. However, the samples $\{r(nT_s)\}$ are sufficient to recover

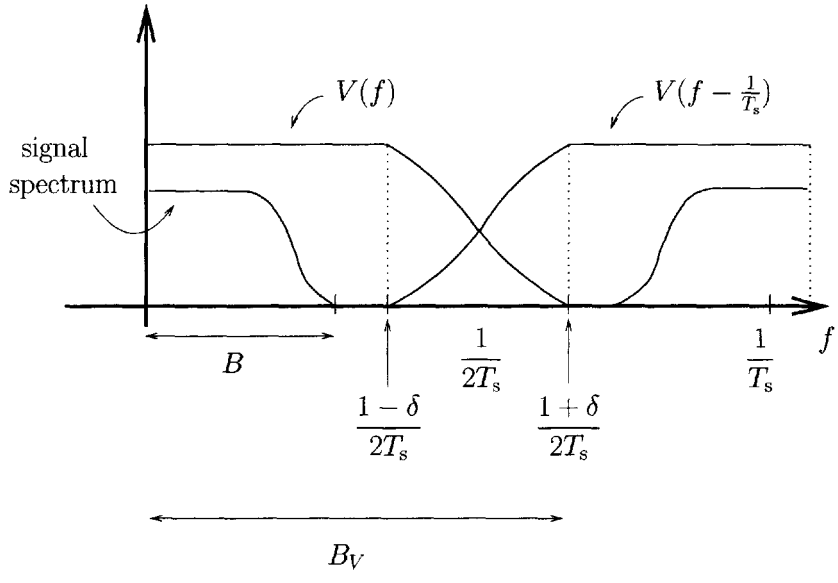


Figure A.3: Sampling and filtering operations.

the information sequence $\{a_n\}$. In other words, aliasing would be present in the recovered noise component. However, this spectral overlap does not impact the signal. A pictorial description of the filter transfer function is shown in Figure A.3.

Summarizing, the filter at the receiver with transfer function $V(f)$ in (A.19) allows one (i) to limit the bandwidth of the noise process $n(t)$, (ii) to leave unmodified the spectrum of the signal $x(t)$ or to modify it in a reversible manner, (iii) to apply the sampling theorem to the signal component of $r(t)$ only, undersampling the noise component of $r(t)$, and (iv) to generate, as sufficient statistics, the samples $\{r(nT_s)\}$.

At this point, one can make interesting remarks on the noise components after sampling. According to the conditions imposed on the filter $v(t)$, since the overall noise spectrum is obtained by adding the noise spectral components placed around multiples of the sampling frequency $f_s = 1/T_s$, these noise spectra overlap in the frequency intervals such that

$$B + if_s < f < B_V + if_s \tag{A.25}$$

where $i \geq 0$ is an integer value. Since the noise process at the input of the receiver is white with bilateral power spectral density $N_0/2$, the noise power spectral density after the filter $v(t)$ is $N_0|V(f)|^2/2$, which, after sampling, becomes

$(N_0/2T_s)\sum_n |V(f - n/T_s)|^2$. Owing to the arbitrary value of $V(f)$ (given that the conditions in (A.19) are satisfied), one can investigate the possibility of obtaining uncorrelated noise samples—if the noise is Gaussian, besides white, these samples would thus be independent. Assuming that $V(f)$ is constant in the band B of the signal, it is sufficient that the behavior of $V(f)$ in its transition band makes the noise spectrum constant, i.e.:

$$|V(f)|^2 + \left|V\left(f - \frac{1}{T_s}\right)\right|^2 = \text{constant} \qquad \text{for} \quad 0 < f < \frac{1}{T_s}. \tag{A.26}$$

If the squared amplitude response of the receiver filter, i.e., $|V(f)|^2$, has vestigial symmetry around $1/2T_s$ in the interval $[B, B_V]$, then condition (A.26) is satisfied: this is the case, for example, of a root raised-cosine filter. Figure A.3 shows a possible power spectrum of the received signal along with a possible receiver filter response of the root raised-cosine type $V(f) = \sqrt{N_{\delta,T_s}(f)}$, in which δ is the roll-off factor, and exemplifies the sampling and filtering operations. The choice of $V(f)$ and $1/T_s$ in the figure is such that (A.19) and (A.20) are satisfied. In addition, (A.27) holds and the noise samples are independent.

An oversampling factor $\beta > 1$ is needed to satisfy (A.20) and in many cases of practical significance $\beta = 2$ may be sufficient. Moreover, if we assume that $V(f)$ has a flat frequency response within the signal band B (as considered in Figure A.3), (A.6) may be simplified by noting that the receiver filter does not introduce any further shaping (i.e., it acts as having impulse response $v(t) = \delta(t)$) but only limits the noise. In the case of a frequency selective fading channel, this implies:

$$f(\tau, t) = \sum_{l=0}^{E-1} h_l(t) g(\tau - \tau_l). \tag{A.27}$$

A.2.3 Signal Symbol-Rate Sampling

When $\beta = 1$, it may not be possible to obtain sufficient statistics. This is particularly true in a communication scenario with fading, especially for *fast* fading variations. However, one can derive an attractive simplified transmission model, assuming *slow* variation of $h_l(t)$. In fact, the factor $h_l(t - \lambda)$ in (A.6) can be moved outside the integral. As a consequence, (A.6) becomes

$$f(\tau, t) = \sum_{l=0}^{E-1} h_l(t) \int_{-\infty}^{\infty} v(\lambda) g(\tau - \tau_l - \lambda)\, \mathrm{d}\lambda = \sum_{l=0}^{E-1} h_l(t) \rho(\tau - \tau_l) \tag{A.28}$$

where $\rho(t) \triangleq v(t) \otimes g(t)$. In this case, a natural assumption is $V(f) = G(f) = \sqrt{N_{\epsilon,T}(f)}$ (i.e., $\epsilon = \delta$), for which $\rho(t)$ is a Nyquist pulse with Fourier transform $N_{\epsilon,T}(f)$.

It is worth remarking that if $h_l(t) = h_l$, i.e., it is time-invariant, then sampling at symbol rate does provide a sufficient statistic.

A.3 Discrete-Time Signal Model

The described models for Nyquist- and symbol-rate sampling are used in the following. Sampling $r(t)$ with period $T_s = T/\beta$, one obtains:

$$r_n \triangleq r(nT_s) = \sum_k a_k f(nT_s - kT, nT_s) + w_n = \sum_i f_{i,n} a_{\lceil \frac{n}{\beta} \rceil - i} + n_n \qquad \text{(A.29)}$$

where, in the last expression, the summation index is $i \triangleq \lceil \frac{n}{\beta} \rceil - k$ ($\lceil \cdot \rceil$ denotes the smallest integer equal to or larger than the argument) and $w_n \triangleq w(nT_s)$ is a discrete-time white noise process.

In the case of a frequency selective fading channel, the coefficients $f_{i,n}$ have the following expression:

$$\begin{aligned} f_{i,n} &\triangleq f\left(\left[n - \left(\left\lceil \frac{n}{\beta} \right\rceil - i\right)\beta\right] T_s, nT_s\right) \\ &= \begin{cases} \sum_{l=0}^{E-1} h_l(nT_s) g\left(\left[n - \left(\left\lceil \frac{n}{\beta} \right\rceil - i\right)\beta\right] T_s - \tau_l\right) & \beta > 1 \\ \sum_{l=0}^{E-1} h_l(nT)\rho(iT - \tau_l) & \beta = 1. \end{cases} \end{aligned} \qquad \text{(A.30)}$$

The resulting discrete-time model (A.29) has the structure of a tapped delay line in which the time-varying coefficients $\{f_{i,n}\}$ are, in general, correlated random processes.

If the following conditions are met: (i) $\beta = 1$, (ii) $\rho(t)$ is a Nyquist pulse, and (iii) $\tau_l = lT$, then (A.30) simplifies to

$$f_{i,n} = h_i(nT) \qquad \text{(A.31)}$$

and the cross-correlation of different tap coefficients ($f_{i,n}$ and $f_{j,n}$ for $i \neq j$) is zero. Although perhaps not realistic, this assumption has been often used in the technical literature because of its simplicity (e.g., see [58]).

Since both $g(t)$ and $\rho(t)$ are physically causal and approximately time-limited pulses, we may consider a finite ISI span L affecting the sampled signal. Hence, (A.29) may be modified by noting that the summation index takes on the values $i = 0, 1, \ldots, L$, resulting in

$$r_n = \sum_{i=0}^{L} f_{i,n} a_{\lceil \frac{n}{\beta} \rceil - i} + n_n. \tag{A.32}$$

References

[1] 3rd Generation Partnership Project, "3GPP," Website: `http://www.3gpp.org`.

[2] H. L. VanTrees, *Detection, Estimation, and Modulation Theory - Part I.* John Wiley & Sons, 1968.

[3] C. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, pp. 379–423, July 1948.

[4] E. R. Berlekamp, *Algebraic Coding Theory.* New York: McGraw-Hill, 1968.

[5] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* New York: North-Holland, 1977.

[6] S. Lin and D. Costello, Jr., *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, 1983.

[7] G. C. Clark and J. B. Cain, *Error Correction Coding for Digital Communications.* New York: Plenum Press, 1988.

[8] J. L. Massey, "Coding and modulation in digital communications," in *Proc. Int. Zurich Seminar on Digital Comm.*, Zurich, Switzerland, March 1974, pp. E2(1)–E2(4).

[9] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, pp. 55–67, January 1982.

[10] W. C. Y. Lee, *Mobile Cellular Telecommunications: Analog and Digital Systems,* 2nd Edition. New York: McGraw-Hill, 1995.

[11] T. Pratt, C. W. Bostian, and J. E. Allnutt, *Satellite Communications,* 2nd Edition. John Wiley & Sons, 2002.

[12] Insitute of Electrical and Electronics Engineers, "IEEE Std 802.11b-1999," 1999.

[13] The Official Bluetooth Membership Site, "Bluetooth," Website: `http://www.bluetooth.org`.

[14] C. E. Perkins, *Ad hoc Networking*. Upper Saddle River, NJ, USA: Addison-Wesley, 2001.

[15] C.-K. Toh, *Ad hoc Mobile Wireless Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[16] T. S. Rappaport, *Wireless Communications. Principles & Practice,* 2nd Edition. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[17] J. G. Proakis, *Digital Communications,* 4th Edition. New York: McGraw-Hill, 2001.

[18] A. B. Carlson, *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*. McGraw-Hill, 1986.

[19] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York, NY: McGraw-Hill, 1991.

[20] R. H. Clarke, "A statistical theory of mobile radio reception," *Bell System Tech. J.*, vol. 47, pp. 957–1000, August 1968.

[21] W. C. Jakes, *Microwave Mobile Communications*. New York: John Wiley & Sons, 1974.

[22] M. J. Gans, "A power spectral theory of propagation in the mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. 21, pp. 27–38, February 1972.

[23] I. S. Gradshtein and I. M. Ryzhik, *Table of Integrals, Series and Products*. New York: Academic Press, 1980.

[24] E. Chiavaccini and G. M. Vitetta, "GQR models for multipath Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 6, 2001.

[25] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *BSTJ*, vol. 57, pp. 2589–2611, September 1978.

[26] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. 29, pp. 1715–1720, November 1981.

[27] T. Aulin, "CPM A power and bandwidth efficient digital constant envelope modulation scheme," Ph.D. dissertation, University of Lund, Lund, Sweden, November 1979.

[28] T. Aulin and C.-E. W. Sundberg, "Continuous phase modulation–Part I: full response signaling," *IEEE Trans. Commun.*, vol. 29, no. 3, pp. 196–209, March 1981.

[29] T. Aulin, N. Rydbeck, and C.-E. W. Sundberg, "Continuous phase modulation–Part II: partial response signaling," *IEEE Trans. Commun.*, vol. 29, no. 3, pp. 210–225, March 1981.

[30] J. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation*. New York: Plenum Press, 1986.

[31] R. G. Gallager, "Low density parity check codes," *IEEE Trans. Inform. Theory*, vol. 8, pp. 21–28, January 1962.

[32] ——, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

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

[34] R. Raheli, A. Polydoros, and C. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, pp. 354–364, February/April 1995.

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

[36] T. Ericson, "A Gaussian channel with slow fading," *IEEE Trans. Inform. Theory*, vol. 16, pp. 353–355, May 1970.

[37] L. Ozarow, S. Shamai, and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 359–378, May 1994.

[38] I. C. Abou-Faycal, M. D. Trott, and S. Shamai, "The capacity of discrete-time memoryless Rayleigh-fading channels," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1290–1301, May 2001.

[39] A. Goldsmith and P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1986–1992, November 1997.

[40] H. Viswanathan, "Capacity of Markov channels with receiver CSI and delayed feedback," *IEEE Trans. Inform. Theory*, vol. 45, pp. 761–771, March 1999.

[41] E. Biglieri, J. Proakis, and S. S. (Shitz), "Fading channels: information-theoretic and communication aspects," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2619–2692, October 1998.

[42] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecommun.*, vol. 10, pp. 585–595, November Dececember 1999.

[43] T. L. Marzetta and B. Hochwald, "Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading," *IEEE Trans. Inform. Theory*, vol. 45, pp. 139–157, January 1999.

[44] G. Colavolpe and R. Raheli, "The capacity of the noncoherent channel," *European Trans. Telecommun.*, vol. 12, no. 4, pp. 289–296, July/August 2001.

[45] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[46] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers (Applications of Communications Theory)*. Plenum Press, 1997.

[47] G. Colavolpe and R. Raheli, "Noncoherent sequence detection," *IEEE Trans. Commun.*, vol. 47, pp. 1376–1385, September 1999.

[48] M. Peleg and S. S. (Shitz), "On the capacity of the blockwise incoherent MPSK channel," *IEEE Trans. Commun.*, vol. 46, pp. 603–609, May 1998.

[49] M. Katz and S. Shamai, "On the capacity-achieving distribution of the discrete-time non-coherent and partially-coherent awgn channels," in *Proc. 22nd Convention of Electrical and Electronics Engineers in Israel*, December 2002, pp. 148 – 150.

[50] R. Nuriyev and A. Anastasopoulos, "Capacity characterization for the non-coherent block-independent AWGN channel," in *Proc. IEEE Symposium on Information Theory* (ISIT), Yokohama, Japan, July 2003, p. 373.

[51] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Comm. Tech.*, vol. 19, pp. 751–772, October 1971.

[52] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. 16, no. 6, pp. 720–738, November 1970.

[53] ——, "Convolutional codes II: Maximum likelihood decoding," *Inf. Control*, vol. 25, pp. 222–266, July 1974.

[54] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, March 1974.

[55] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 4, pp. 1072–1092, July 1996.

[56] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill, 1965.

[57] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications*. New York: Macmillan, 1991.

[58] E. T. S. Institute, "Radio transmission and reception," January 1992, GSM Recommendations No. 05.05-DCS (version 3.0).

[59] P. A. Laurent, "Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 150–160, February 1986.

[60] M. Morelli and U. Mengali, "Decomposition of M-ary CPM signals into PAM waveforms," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1265–1275, September 1995.

[61] B. E. Rimoldi, "A decomposition approach to CPM," *IEEE Trans. Inform. Theory*, vol. 34, pp. 260–270, March 1988.

[62] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. Inform. Theory*, vol. 23, pp. 371–377, May 1977.

[63] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.

[64] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

[65] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May 1999.

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

[67] C. Berrou, "The ten-year-old turbo codes are entering into service," *IEEE Commun. Mag.*, pp. 110–116, August 2003.

[68] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA, USA: MIT Press, 1966.

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

[70] ——, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 408–428, March 1996.

[71] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "Soft-input soft-output modules for the construction and distributed iterative decoding of code networks," *European Trans. Telecommun.*, vol. 9, no. 2, pp. 155–172, Mar/Apr 1998.

[72] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, May 1998.

[73] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, September/October 1995.

[74] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754 – 767, May 2002.

[75] L. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, November 1996.

[76] G. Ferrari and K. M. Chugg, "Linear programming-based optimization of the distance spectrum of linear block codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1794–1800, July 2003.

[77] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEE Electronics Letters*, vol. 33, pp. 457–458, March 1997.

[78] ——, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, February 1999.

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

[80] T. Richardson and R. Urbanke, "The capacity of low density parity check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, February 2001.

[81] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, September 1981.

[82] G. Colavolpe, "Design and performance of turbo Gallager codes," to appear in *IEEE Trans. Commun.*. See also G. Colavolpe, "Performance of turbo Gallager codes," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, September 2003, pp. 547-550.

[83] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, January 1997.

[84] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 891–907, May 2001.

[85] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, February 2001.

[86] S. ten Brink, "Convergence of iterative decoding," *IEE Electronics Letters*, vol. 35, pp. 1117–1119, 24th June 1999.

[87] ——, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727 – 1737, October 2001.

[88] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, March 1998.

[89] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744–765, March 1998.

[90] J. H. Winters, J. Salz, and R. D. Gitlin, "The impact of antenna diversity on the capacity of wireless communication systems," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1740–1751, February/March/April 1994.

[91] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456–1467, July 1999.

[92] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. Waveland Press, 1990, (reprint of 1965 original from John Wiley & Sons).

[93] I. Motedayen-Aval and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. Commun.*, vol. 51, pp. 197–207, February 2003.

[94] 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, December 1994.

[95] K. M. Chugg, "The condition for the applicability of the Viterbi algorithm with implications for fading channel MLSD," *IEEE Trans. Commun.*, vol. 46, no. 9, pp. 1112–1116, September 1998.

[96] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 259–260, April 1967.

[97] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, March 1973.

[98] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications.* Kluwer Academic Publishers, 2001.

[99] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommun. Conf.* (GLOBECOM), Dallas, TX, USA, November 1989, pp. 1680–1686.

[100] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 18, pp. 284–287, May 1972.

[101] J. Lodge and M. Moher, "Maximum likelihood estimation of CPM signals transmitted over Rayleigh flat fading channels," *IEEE Trans. Commun.*, vol. 38, pp. 787–794, June 1990.

[102] H. Meyr and G. Ascheid, *Synchronization in Digital Communications.* John Wiley & Sons, 1990.

[103] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control.* Upper Saddle River, NJ: Prentice-Hall, 1995.

[104] S. Benedetto, E. Biglieri, and V. Castillani, *Digital Transmission Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[105] S. Haykin, *Adaptive Filter Theory,* 4th Edition. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[106] R. Raheli, G. Marino, and P. Castoldi, "Per-survivor processing and tentative decisions: What is in between?" *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 127–129, February 1996.

[107] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey cost analysis," *IEEE Trans. Commun.*, vol. 32, pp. 169–176, February 1984.

[108] T. Aulin, "Breadth-first maximum likelihood sequence detection: basics," *IEEE Trans. Commun.*, vol. 47, pp. 208–216, February 1999.

[109] ——, "Breadth-first maximum likelihood sequence detection: geometry," *IEEE Trans. Commun.*, vol. 51, no. 6, pp. 2071–2080, June 2003.

[110] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*. Dover, 1972.

[111] W. J. Weber, "Differential encoding for multiple amplitude and phase shift keying systems," *IEEE Trans. Commun.*, vol. 26, pp. 385–391, March 1978.

[112] G. D. Forney, Jr., "Lower bounds on error probability in the presence of large intersymbol interference," *IEEE Trans. Commun.*, vol. 20, pp. 76–77, February 1972.

[113] U. Mengali and M. Morelli, "Data-aided frequency estimation for burst digital transmission," *IEEE Trans. Commun.*, vol. 45, pp. 23–25, January 1997.

[114] M. Morelli and U. Mengali, "Feedforward frequency estimation for PSK: a tutorial review," *European Trans. Telecommun.*, vol. 9, pp. 103–116, March/April 1998.

[115] D. C. Rife and R. R. Boorstyn, "Single-tone parameter estimation from discrete-time observations," *IEEE Trans. Inform. Theory*, vol. 20, pp. 591–598, September 1974.

[116] S. Kay, "A fast and accurate single frequency estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1987–1990, December 1989.

[117] G. Colavolpe and R. Raheli, "Detection of linear modulations in the presence of strong phase and frequency instabilities," *IEEE Trans. Commun.*, vol. 50, pp. 1617–1626, October 2002.

[118] Y. Okunev, *Phase and Phase-Difference Modulation in Digital Communications*. Norwood, MA: Artech House, 1997.

[119] L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space–Part I: 180°," *IEEE J. Select. Areas Commun.*, vol. 2, no. 5, pp. 659–671, September 1984.

[120] G. Colavolpe and R. Raheli, "Noncoherent sequence detection in frequency nonselective slowly fading channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 11, pp. 2302–2311, November 2000.

[121] P. Castoldi and R. Raheli, "Recursive optimal detection of linear modulations in the presence of random fading," *European Trans. Telecommun.*, vol. 9, no. 2, pp. 209–220, March/April 1998.

[122] S. L. Marple, *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[123] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, April 1975.

[124] G. M. Vitetta and D. P. Taylor, "Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels," *IEEE Trans. Commun.*, vol. 43, no. 11, pp. 2750–2758, November 1995.

[125] X. Yu and S. Pasupathy, "Innovations-based MLSE for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 43, pp. 1534–1544, February/April 1995.

[126] D. Makrakis, P. T. Mathiopoulos, and D. Bouras, "Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: A combined envelope, multiple differential and coherent detection approach," *IEEE Trans. Commun.*, vol. 42, pp. 63–75, January 1994.

[127] F. Giannetti, M. Luise, and R. Reggianini, "Simple carrier frequency rate-of-change estimators," *IEEE Trans. Commun.*, vol. 47, pp. 1310–1314, September 1999.

[128] G. Ferrari, G. Colavolpe, and R. Raheli, "Linear predictive receivers for phase uncertain channels," in *Intern. Symp. on Signal Process. and Applications*, vol. 1, Paris, France, July 2003, pp. 301–304.

[129] G. Ferrari, A. Anastasopoulos, G. Colavolpe, and R. Raheli, "Adaptive iterative detection for the phase uncertain channel: limited-tree-search versus truncated-memory detection," *IEEE Trans. Veh. Technol.*, vol. 53, no. 2, pp. 433–442, March 2004.

[130] G. Colavolpe and R. Raheli, "On noncoherent sequence detection of coded QAM," *IEEE Commun. Lett.*, vol. 2, pp. 211–213, August 1998.

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

[132] Q. Dai and E. Shwedyk, "Detection of bandlimited signals over frequency selective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 42, pp. 941–950, February/March/April 1994.

[133] M. E. Rollins and S. J. Simmons, "Simplified per-survivor Kalman processing in fast-frequency-selective fading channels," *IEEE Trans. Commun.*, vol. 45, pp. 544–553, May 1997.

[134] R. W. Chang and J. C. Hancock, "On receiver structures for channels having memory," *IEEE Trans. Inform. Theory*, vol. 12, pp. 463–468, October 1966.

[135] K. Abend and B. D. Fritchman, "Statistical detection for communication channels with intersymbol interference," *Proc. IEEE*, vol. 58, pp. 779–785, May 1970.

[136] P. L. McAdam, L. R. Welch, and C. L. Weber, "M.A.P. bit decoding of convolutional codes," *Proc. IEEE Int. Symp. Info. Theory*, 1972.

[137] P. Robertson, E. Villebrun, and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. Telecommun.*, vol. 8, no. 2, pp. 119–125, March/April 1997.

[138] J. Lodge, R. Young, and J. H. P. Hoeher, "Separable MAP "filters" for the decoding of product and concatenated codes," in *Proc. IEEE International Conf. on Commun.* (ICC), Geneva, Switzerland, June 1993, pp. 102–106.

[139] G. Battail, "Pondération des symboles décodé par l'algorithme de Viterbi," *Annals of Telecommun.*, vol. 42, pp. 31–38, January 1987, (in French).

[140] C. Berrou, P. Adde, E. Angui, and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," in *Proc. IEEE International Conf. on Commun.* (ICC), Geneva, Switzerland, June 1993, pp. 737–740.

[141] B. Sklar, "A primer on turbo code concepts," *IEEE Commun. Mag.*, vol. 35, pp. 94–102, December 1997.

[142] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *Proc. IEEE International Conf. on Commun.* (ICC), Dallas, Texas, USA, June 1996, pp. 102–106.

[143] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in *Proc. IEEE Global Telecommun. Conf.* (GLOBECOM), San Francisco, CA, USA, November 1994, pp. 1298–1303.

[144] L. Lin and R. Cheng, "Improvements in SOVA-based decoding for turbo codes," in *Proc. IEEE International Conf. on Commun.* (ICC), Montréal, Canada, June 1997, pp. 1473–1478.

[145] H. E. Gamal and A. R. H. Jr., "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, February 2001.

[146] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, February 2001.

[147] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in iterative decoding: a unified view," *IEEE Trans. Commun.*, vol. 49, pp. 2088–2094, December 2001.

[148] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and max-log-MAP decodings," *IEEE Commun. Lett.*, vol. 2, pp. 137–139, May 1998.

[149] G. Ferrari, G. Colavolpe, and R. Raheli, "On trellis-based truncated-memory detection," in *Proc. IEEE Global Telecommun. Conf.* (GLOBECOM), San Francisco, USA, December 2003.

[150] D. Raphaeli, "Noncoherent coded modulation," *IEEE Trans. Commun.*, vol. 44, pp. 172–183, February 1996.

[151] X. Chen and K. M. Chugg, "Reduced-state soft-in/soft-out algorithms for complexity reduction in iterative and non-iterative data detection," in *Proc. IEEE International Conf. on Commun.* (ICC), New Orleans, LA, USA, 2000.

[152] A. Duel-Hallen and C. Heegard, "Delayed decision feedback estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.

[153] M. V. Eyuboğlu and S. U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 38, pp. 13–20, January 1988.

[154] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 36, pp. 669–676, July 1989.

[155] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[156] G. Colavolpe, G. Ferrari, and R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 848–859, May 2001.

[157] S. Ariyavisitakul, "A decision feedback equalizer with time-reversal structure," *IEEE J. Select. Areas Commun.*, vol. 10, no. 3, pp. 599–613, April 1992.

[158] N. C. McGinty and R. A. Kennedy, "Reduced-state sequence estimator with reverse-time structure," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 265–268, March 1997.

[159] P. Hoeher and J. Lodge, ""Turbo DPSK": Iterative differential PSK demodulation and channel decoding," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 837–843, June 1999.

[160] G. Colavolpe, P. Castoldi, and R. Raheli, "Linear predictive receivers for fading channels," *IEE Electronics Letters*, vol. 34, pp. 1289–1290, 25th June 1998.

[161] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, November 1997.

[162] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) detection," *IEEE Trans. Commun.*, vol. 48, no. 9, pp. 1488–1498, September 2000.

[163] G. Ferrari, G. Colavolpe, and R. Raheli, "Noncoherent iterative decoding of spectrally efficient coded modulations," *Annals of Telecommun.*, vol. 56, pp. 409–421, July/August 2001.

[164] P. Thiennviboon, G. Ferrari, and K. M. Chugg, "On reduced-state forward-backward algorithms," *IEEE Trans. Inform. Theory*, 2004, (Submitted).

[165] G. Ferrari, "On iterative detection for channels with memory," Ph.D. dissertation, University of Parma, Parma, Italy, November 2001. Available upon request.

[166] P. Thiennviboon, G. Ferrari, and K. M. Chugg, "Generalized trellis-based reduced-state soft-input/soft-output algorithms," in *Proc. IEEE International Conf. on Commun.* (ICC), New York, USA, April 2002, pp. 1667–1671.

[167] P. Thiennviboon, "Graphical models for iterative data detection," Ph.D. dissertation, University of Southern California, Los Angeles, CA, August 2002.

[168] J. B. Anderson and E. Offer, "Reduced-state sequence detection with convolutional codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 965–972, May 1994.

[169] I. D. Marsland and P. T. Mathiopoulos, "On the performance of iterative noncoherent detection of coded M-PSK signals," *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 588–596, April 2000.

[170] K. R. Narayanan and G. L. Stüber, "A serial concatenation approach in iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 956–961, July 1999.

[171] M. Peleg, I. Sason, S. S. (Shitz), and A. Elia, "On interleaved, differentially encoded convolutional codes," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2572–2582, November 1999.

[172] M. Peleg and S. Shamai (Shitz), "Iterative decoding of coded and interleaved noncoherent multiple symbol detected DPSK," *IEE Electronics Letters*, vol. 33, no. 12, pp. 1018–1020, June 1997.

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

[174] M. Peleg and S. Shamai (Shitz), "On coded and interleaved noncoherent multiple symbol detected MPSK," *European Trans. Telecommun.*, vol. 10, no. 1, pp. 65–73, January/February 1999.

[175] E. K. Hall and S. G. Wilson, "Turbo codes for noncoherent channels," in *IEEE Commun. Theory Mini-Conf.*, Phoenix, AZ, USA, November 1997, pp. 66–70.

[176] T. M. Duman and M. Salehi, "Performance bounds for Turbo-Coded modulation systems," *IEEE Trans. Commun.*, vol. 47, pp. 511–521, April 1999.

[177] S. Le Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *Proc. IEEE International Conf. on Commun.* (ICC), New Orleans, LA, USA, May 1994, pp. 645–649.

[178] R. H. Deng and D. J. Costello, "High rate concatenated coding systems using bandwidth efficient trellis inner codes," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 420–427, May 1989.

[179] P. Robertson and T. Worz, "A novel bandwidth efficient coding scheme employing turbo codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 206–218, February 1998.

[180] C. Fragouli and R. D. Wesel, "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation," *IEEE Trans. Commun.*, vol. 49, pp. 425–435, March 2001.

[181] L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space–Part II: nonlinear codes," *IEEE J. Select. Areas Commun.*, vol. 2, no. 5, pp. 672–686, September 1984.

[182] S. S. Pietrobon, G. Ungerboeck, L. C. Perez, and D. J. Costello, "Rotationally invariant nonlinear trellis codes for two-dimensional modulation," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1773–1791, November 1994.

[183] W. Liu and S. G. Wilson, "Rotationally-invariant concatenated (turbo) TCM codes," in *Proc. Asilomar Conf. Signals, Systems, Comp.*, vol. 1, 1999, pp. 32–36.

[184] R. Nuriyev and A. Anastasopoulos, "Rotationally invariant and rotationally robust codes for the awgn and the noncoherent channel," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2001 – 2010, 2003.

[185] A. Anastasopoulos and K. M. Chugg, "Adaptive Soft-Input Soft-Output algorithms for iterative detection with parametric uncertainty," *IEEE Trans. Commun.*, vol. 48, no. 10, pp. 1638–1649, October 2000.

[186] ——, "Adaptive iterative detection for phase tracking in turbo coded systems," *IEEE Trans. Commun.*, vol. 49, pp. 2135–2144, December 2001.

[187] G. Colavolpe and R. Raheli, "Theoretical analysis and performance limits of noncoherent sequence detection of coded PSK," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1483–1494, July 2000.

[188] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenation of trellis coded modulation with rate-1 inner code," in *Proc. IEEE Global Telecommun. Conf.* (GLOBECOM), San Francisco, USA, November 2000, pp. 777–782.

[189] R. Nuriyev and A. Anastasopoulos, "Design and robustness analysis of rotationally invariant SCTCM," in *Proc. IEEE International Conf. on Commun.* (ICC), Helsinki, Finland, June 2001.

[190] ETSI, "ETSI EN300 909 v.8.3," July 2000.

[191] A. B. Kiely, S. J. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis decoding complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1687–1697, November 1996.

[192] A. P. Worthen and W. E. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 843–849, February 2001.

[193] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 325–343, March 2000.

[194] J. Boutros and G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.

[195] J. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1661–1671, Feb/Mar/Apr. 1994.

[196] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in *Proc. IEEE Global Telecommun. Conf.* (GLOBECOM), San Diego, CA, USA, December 1990, pp. 807.5.1–5.

[197] F. Kschischang and B. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, pp. 219–231, February 1998.

[198] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569–584, February 2001.

[199] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, February 2001.

[200] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University (Sweden), 1996.

[201] G. Colavolpe, "LDPC codes over channels with memory," in *Proc. IEEE International Conf. on Commun.* (ICC), Paris, France, June 2004, to be published.

[202] G. Colavolpe and G. Germi, "Simple iterative detection schemes for ISI channels," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, September 2003, pp. 283–286.

[203] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1410–1422, June 2002.

[204] N. Benvenuto and R. Marchesani, "The Viterbi algorithm for sparse channels," *IEEE Trans. Commun.*, vol. 44, pp. 287–289, March 1996.

[205] N. C. McGinty, R. A. Kennedy, and P. Hoeher, "Parallel trellis Viterbi algorithm for sparse channels," *IEEE Trans. Commun.*, pp. 143–145, May 1998.

[206] T. V. Souvignier, M. Öberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for partial response channels," *IEEE Trans. Commun.*, vol. 48, no. 8, pp. 1297–1308, August 2000.

[207] T. Mittelholzer, A. Dholakia, and E. Eleftheriou, "Reduced-complexity decoding of low-density parity check codes for generalized partial response channels," *IEEE Trans. Magn.*, vol. 37, pp. 721–728, March 2001.

[208] A. Thangaraj and S. W. McLaughlin, "Thresholds and scheduling for LDPC-coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2307–2309, September 2002.

[209] N. Varnica and A. Kavčić, "Optimized low-density parity-check codes for partial response channels," *IEEE Commun. Lett.*, vol. 7, no. 4, pp. 168–170, April 2003.

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

[211] P. A. Beerel and K. M. Chugg, "A low latency SISO with application to broadband turbo decoding," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 860–870, May 2001.

[212] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[213] H. Thapar and A. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. 23, no. 5, pp. 3666 – 3668, September 1987.

[214] D. J. Tyner and J. G. Proakis, "Partial response equalizer performance in digital magnetic recording channels," *IEEE Trans. Magn.*, vol. 29, no. 6, pp. 4194 – 4208, November 1993.

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

[216] J. Dauwels and H.-A. Loeliger, "Joint decoding and phase estimation: an exercise in factor graphs," in *Proc. IEEE Symposium on Information Theory* (ISIT), Yokohama, Japan, July 2003, p. 231.

[217] ——, "Phase estimation by message passing," in *Proc. IEEE International Conf. on Commun.* (ICC), Paris, France, June 2004.

[218] G. Colavolpe, A. Barbieri, G. Caire, and N. Bonneau, "Bayesian and non-Bayesian methods for iterative joint decoding and detection in the presence of phase noise," in *Proc. IEEE Symposium on Information Theory* (ISIT), Chicago, IL, USA, June-July 2004, p. 131.

[219] G. Colavolpe, A. Barbieri, and G. Caire, "Iterative decoding in the presence of strong phase noise," 2004, submitted to *IEEE J. Select. Areas Commun.*

[220] H. Steendam, N. Noels, and M. Moeneclaey, "Iterative carrier phase synchronization for low-density parity-check coded systems," in *Proc. IEEE International Conf. on Commun.* (ICC), Anchorage, AK, USA, June 2003, pp. 3120–3124.

[221] V. Lottici and M. Luise, "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations," *IEEE Trans. Commun.*, vol. 52, pp. 661–669, April 2004.

[222] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *Proc. IEEE International Conf. on Commun.* (ICC), Anchorage, AK, USA, June 2003, pp. 2933–2937.

[223] A. Ginesi, D. Fittipaldi, A. Bigi, and R. D. Gaudenzi, "Pilot-aided carrier synchronization techniques for broadband satellite transmissions," ESA-ESTEC, Tech. Rep., September 2003.

[224] L. Benvenuti, L. Giugno, V. Lottici, and M. Luise, "Code-aware carrier phase noise compensation on turbo-coded spectrally-efficient high-order modulations," in *8th Intern. Work. on Signal Processing for Space Commun.*, Catania, Italy, September 2003, pp. 177–184.

[225] O. Macchi and L. Scharf, "A dynamic programming algorithm for phase estimation and data decoding on random phase channels," *IEEE Trans. Inform. Theory*, pp. 581–595, September 1981.

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

[227] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*. Prentice-Hall, 2000.

[228] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1994.

[229] P. Hoeher, "A statistical discrete-time model for the WSSUS multipath fading channel," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 461–468, November 1992.

List of Acronyms

8-PSK	8-ary Phase Shift Keying
ACS	Add-Compare-Select
APP	*A Posteriori* Probability
ARMA	Autoregressive Moving Average
A-SODEM	Adaptive SOft DEModulator
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek, Raviv
BER	Bit Error Rate
BICM	Bit Interleaved Coded Modulation
BiD	BiDirectional
BPSK	Binary Phase Shift Keying
Bwd-only	Backward-only
CC	Convolutional Code
CL	Closed-Loop
CMP	Conditional Markov Property
CPE	Continuous Phase Encoder
CPM	Continuous Phase Modulation
CSI	Channel State Information
DDE	Double Differential Encoding
DE	Differential Encoding
DQAM	Differential Quadrature Amplitude Modulation
DQPSK	Differential Quaternary Phase Shift Keying
DVB-S2	Digital Video Broadcasting Second Generation Satellite System
ECMP	Extended Conditional Markov Property
EM	Expectation-Maximization
ESA	European Space Agency
EXIT	EXtrinsic Information Transfer
FB	Forward Backward
FER	Frame Error Rate
Ffg	Fully filling the gap

FG	Factor Graph
FMC	Finite Memory Condition
FS	Forward Survivor
FSM	Finite State Machine
Fwd-only	Forward-only
GMSK	Gaussian Minimum Shift Keying
GQR	Gaussian Quadrature Rule
GSM	Groupe Speciale Mobile
iid	Independent and identically distributed
ISI	Inter-Symbol Interference
LDPC	Low Density Parity Check
LEO	Low Earth Orbit
LLR	Logarithmic Likelihood Ratio
LMS	Least Mean Squares
LTI	Linear Time Invariant
LVA	List Viterbi Algorithm
MAP	Maximum *A Posteriori*
MC	Multilevel Coding
ML	Maximum Likelihood
MLSD	Maximum Likelihood Sequence Detection
MM	Memoryless Mapper
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
NCSO	NonCoherent Soft-Output
Nfg	Not filling the gap
NRC	NonRecursive Convolutional
NRI	NonRotationally Invariant
NRNSC	NonRecursive NonSystematic Convolutional
NSD	Noncoherent Sequence Detection
OL	Open-Loop
PAM	Pulse Amplitude Modulation
PCCC	Parallel Concatenated Convolutional Code
pdf	Probability density function
Pfg	Partially filling the gap
PLL	Phase-Locked Loop
Pmc	Partial multiple completion
pmf	Probability mass function
PSK	Phase Shift Keying
PSP	Per-Survivor Processing

QAM	Quadrature Amplitude Modulation
QPSK	Quaternary Phase Shift Keying
RI	Rotationally Invariant
RS	Reduced-State
RSC	Recursive Systematic Convolutional
RS-FB	Reduced-State Forward Backward
RSSD	Reduced State Sequence Detection
sc	Single completion
SCCC	Serially Concatenated Convolutional Code
SCTCM	Serially Concatenated Trellis Coded Modulation
SER	Symbol Error Rate
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SO-NSD	Soft-Output Noncoherent Sequence Detection
SOVA	Soft-Output Viterbi Algorithm
SP	Sum-Product
TC	Trellis Coded
TCM	Trellis Coded Modulation
T-TCM	Turbo Trellis-Coded Modulation
VA	Viterbi Algorithm
WMF	Whitened Matched Filter
WSSUS	Wide Sense Stationary Uncorrelated Scattering

Index