INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Joint Iterative Demodulation and Decoding for Turbo-Coded PSK Signals on Flat Fading Channels

By

Christian Dubuc, B. Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Engineering

Ottawa-Carleton Institute For Electrical and Computer Engineering
Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa, Ontario
September, 2000

© Copyright

Christian Dubuc, 1999-2000

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-57725-2

Canada

The undersigned hereby recommend to the faculty of Graduate Studies and Research acceptance of the thesis, "Joint Iterative Demodulation and Decoding for Turbo-Coded PSK Signals on Flat Fading Channels", submitted by Christian Dubuc, B.Eng. in partial fulfillment of the requirements for the degree of Master of Engineering.

Chair, Department of Systems and
Computer Engineering

Professor D. Falconer, Thesis Co-supervisor

Dr. D. Boudreau, Thesis Co-supervisor

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa, Ontario
September, 2000

Abstract

The problem of robust demodulation for turbo-coded PSK signals in flat fading is studied. Iterative joint processing, based on Maximum *A Posteriori* (MAP) demodulation and decoding, is proposed to improve the accuracy of the detection process. Two iterative demodulation and decoding schemes are proposed: one for differentially encoded signals and one for pilot symbol assisted signals. Both schemes are based on linear-predictive symbol-by-symbol MAP demodulation. The proposed iterative demodulation and decoding schemes are derived using iterative decoding principles. In addition, a graphical model approach is used to justify the proposed iterative schemes. It is shown that the application of Pearl's algorithm on the Bayesian graph representing the problem of joint demodulation and decoding of turbo-coded signals yields an iterative structure corresponding precisely to that derived using iterative decoding principles. Simulation results are also presented to evaluate the performance of the proposed iterative demodulation and decoding schemes in Rayleigh fading.

Acknowledgments

First, I am especially grateful to my thesis co-supervisor, Dr. Daniel Boudreau of the Communications Research Centre (CRC), for his ongoing support. His dedication and patience have been truly appreciated. In addition, I would like to thank my thesis supervisor, Dr. David Falconer, for his assistance. I would also like to thank the Communications Signal Processing Group at CRC, for the computational resources required for the completion of this thesis. Finally, I thankfully acknowledge the financial support provided by the National Sciences and Engineering Research Council and Carleton University.

Table of Contents

ABSTRACT	II
ACKNOWLEDGMENTS	III
TABLE OF CONTENTS.....	IV
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF ACRONYMS.....	XIII
LIST OF SYMBOLS.....	XV
CHAPTER 1 – INTRODUCTION	1
1.1 OBJECTIVES	3
1.1.1 Iterative Demodulation and Decoding of Turbo-Coded Signals	3
1.1.2 Application of Bayesian Graph Theory to Joint Demodulation and Decoding	4
1.2 ORGANIZATION OF THE THESIS	4
CHAPTER 2 - TURBO DECODING.....	6
2.1 CONCATENATED CODES	7
2.1.1 General Concept.....	7
2.1.2 Serial Concatenation.....	8

2.1.3 Parallel Concatenation of Recursive Systematic Convolutional Codes	9
2.2 SOFT-IN/SOFT-OUT DECODING.....	12
2.2.1 MAP Decoding Using the BCJR Algorithm	14
2.2.2 MAP Filtering.....	15
2.2.3 Implementation Issues	15
2.3 ITERATIVE PROCESSING	17
2.3.1 Extrinsic Information	18
2.3.2 Iterative Decoding of Turbo Codes	20
2.3.3 Convergence	27
2.4 SUMMARY	28
CHAPTER 3 - SOFT-OUTPUT DEMODULATION	30
3.1 FLAT FADING CHANNEL MODEL.....	31
3.2 CONVENTIONAL DEMODULATION SCHEMES	35
3.2.1 Coherent Detection with Known Channel State Information.....	35
3.2.2 Pilot Symbol-Assisted Modulation and MMSE Channel Estimation	37
3.2.3 Differential Detection	39
3.3 LINEAR-PREDICTIVE MLSE/MAP DETECTION	41
3.3.1 Linear-Predictive MLSE	41
3.3.2 Linear-Predictive Symbol-by-Symbol MAP Estimation.....	45
3.3.3 Coherent PSK with Pilot Symbols	47
3.3.4 Differential PSK.....	48

3.4 RECEPTION DIVERSITY	49
3.4.1 Conventional Demodulation Schemes	50
3.4.2 Linear-Predictive Demodulation	50
3.5 SUMMARY	51
CHAPTER 4 - ITERATIVE DEMODULATION AND DECODING	52
4.1 BASIC STRUCTURE FOR CONVOLUTIONAL CODES.....	55
4.2 EXTENSION TO TURBO CODES	61
4.2.1 Motivations.....	61
4.2.2 Iterative Demodulation and Decoding Algorithm.....	63
4.3 DISCUSSION	71
CHAPTER 5 - BAYESIAN GRAPHS INTERPRETATION	74
5.1 PROBABILISTIC INFERENCE USING BAYESIAN GRAPHS.....	76
5.1.1 Bayesian Graphs (Bayesian Networks)	76
5.1.2 Pearl's Algorithm for Probabilistic Inference	78
5.2 APPLICATION TO CONVOLUTIONALLY ENCODED SIGNALS	83
5.2.1 Bayesian Graph Representation	83
5.2.2 Pearl's Algorithm Messages for the Proposed Graph	86
5.2.3 Discussion	89
5.3 APPLICATION TO TURBO-CODED SIGNALS.....	94
5.3.1 Bayesian Graph Representation	94
5.3.2 Pearl's Algorithm Messages for the Proposed Graph	95

5.3.3 Discussion	97
5.4 SUMMARY.....	105
CHAPTER 6 - SIMULATION RESULTS	107
6.1 CONVOLUTIONAL CODES	108
6.2 TURBO CODES	112
6.2.1 Coding Rate	119
6.2.2 Convergence	122
6.2.3 Iteration Schedules	124
6.2.4 Diversity Reception	127
6.2.5 Effect of the Fading Rate.....	129
6.3 DISCUSSION	132
CHAPTER 7 - CONCLUSIONS.....	135
7.1 SUMMARY.....	135
7.2 SIMULATION RESULTS	136
7.3 CONTRIBUTIONS	137
7.4 SUGGESTIONS FOR FUTURE WORK.....	138
ANNEX A – BCJR ALGORITHM	140
ANNEX B – CHANNEL INTERPOLATION FILTER.....	143
ANNEX C – PEARL’S ALGORITHM FOR JOINT DEMODULATION AND	

DECODING OF TURBO-CODED SIGNALS.....	145
ANNEX D – TURBO CODING SCHEME DEFINITION	151
REFERENCES	155

List of Tables

Table 1 – Messages for the Bayesian Graph Associated with Convolutional Codes	89
Table 2 – Bottom up messages for the Bayesian graph associated with turbo codes	96
Table 3 – Top down messages for the Bayesian graph associated with turbo codes	97
Table 4 – Simulation Parameters	112

List of Figures

Figure 1 – Transmitter end of a serial concatenation scheme.	9
Figure 2 – Parallel concatenation of recursive systematic convolutional codes.	10
Figure 3 – Markov source interpretation of a convolutional code over a DMC.	13
Figure 4 – Iterative decoding process of a turbo code.	24
Figure 5 – Iterative decoding process of a turbo code in the log-likelihood domain.	26
Figure 6 – Flat fading channel baseband model.	32
Figure 7 – Pseudo-coherent detection using PSAM.	37
Figure 8 – Conventional one-symbol differential detection.	39
Figure 9 – Optimal receiver structure.	52
Figure 10 – Conventional soft-in/soft-out receiver structure.	53
Figure 11 – Iterative demodulation and decoding structure for convolutional codes.	60
Figure 12 – Soft-in/Soft-out turbo decoder for one iteration.	65
Figure 13 – Equivalent MAP filter for one iteration of the turbo decoder.	67
Figure 14 – Iterative demodulation and decoding structure for turbo codes.	70
Figure 15 – Example of a Bayesian graph.	78
Figure 16 – A node Z of a Bayesian graph, with its parents V and its children W	80
Figure 17 – Message flow illustration for a node Z of a Bayesian graph.	81
Figure 18 – Detailed Bayesian graph for convolutionally encoded signals.	84
Figure 19 – Simplified Bayesian graph for convolutionally encoded signals.	85
Figure 20 – Bayesian graph for turbo-coded signal	95

Figure 21 - BER for iterative demodulation and decoding of convolutionally encoded signals	109
Figure 22 – PER for the iterative demodulation and decoding of convolutionally encoded signals	110
Figure 23 - BER for the iterative demodulation and decoding of turbo-coded signals ..	114
Figure 24 - PER for the iterative demodulation and decoding of turbo-coded signals...	115
Figure 25 - PER with and without information about parity and systematic extrinsic information fed back to the demodulator.....	117
Figure 26 - PER for the iterative demodulation and decoding of a rate-2/3 turbo-coded signal	120
Figure 27 - PER for the iterative demodulation and decoding of a rate-3/4 turbo-coded signal	121
Figure 28 – Number of iterations required for convergence.....	123
Figure 29 - PER for different iteration schedules for the iterative demodulation and decoding based on PSAM.....	125
Figure 30 - PER for different iteration schedules for the iterative demodulation and decoding based on differential encoding	126
Figure 31 - PER for the iterative demodulation and decoding schemes with a $D = 2$ diversity reception.....	127
Figure 32 - PER for the iterative demodulation and decoding schemes ($f_d = 0.00125f_s$)	130
Figure 33 - PER for the iterative demodulation and decoding schemes ($f_d = 0.05f_s$)	131

Figure 34 – Impulse response of an interpolation filter for $f_d = 1\%$ and E_s/N_0 10dB.	143
Figure 35 – Frequency response of an interpolation filter for $f_d = 1\%$ and $E_s/N_0 = 10$ dB.	144
Figure 36 – Channel estimation for $f_d = 1\%$ and $E_s/N_0 = 10$ dB.	144
Figure 37 – Recursive systematic convolutional code with polynomials $(23)_8$ and $(35)_8$	151
Figure 38 – BER performance on an AWGN channel, $N = 4096$	154
Figure 39 – PER performance on an AWGN channel, $N = 4096$	154

List of Acronyms

3G	Third-generation wireless systems
APP	<i>A Posteriori</i> Probability
ARQ	Automatic Repeat reQuest
ASIC	Application-Specific Integrated Circuit
AWGN	Additive White Gaussian Noise
BCJR	Bahl Cocke Jelinek Raviv algorithm
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CDMA	Code-Division Multiple Access
CPM	Continuous Phase Modulation
CSI	Channel State Information
CRC	Communications Research Centre
DAG	Directed Acyclic Graph
DPSK	Differential Phase Shift Keying
DQPSK	Differential Quaternary Phase Shift Keying
DSP	Digital Signal Processor
FIR	Finite Impulse Response filter
GF	Galois Field
ISI	Inter-Symbol Interference
LOS	Line-Of-Sight

MAP	Maximum <i>a posteriori</i>
MDD	Multiple Differential Detector
MLSE	Maximum Likelihood Sequence Estimation
MMSE	Minimum Mean-Square Error
MPSK	M-ary Phase Shift Keying
MRC	Maximum Ratio Combining
MSDD	Multiple Symbol Differential Detection
MSE	Mean Square Error
PAM	Pulse Amplitude Modulation
pdf	Probability Density Function
PER	Packet Error Rate
PSAM	Pilot Symbol Assisted Modulation
PSD	Power Spectral Density
PSK	Phase Shift Keying
QPSK	Quaternary Phase Shift Keying
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
TDMA	Time-Division Multiple Access
WSS	Wide Sense Stationary

List of Symbols

c	the speed of light
$c(t)$	the complex gain of a flat fading channel
c_k	the complex gain of a flat fading channel at sampling time k
$\mathbf{c}$	the sequence of complex gains of a flat fading channel
$\tilde{c}_k$	the estimated complex gain of a flat fading channel at sampling time k
$\tilde{\mathbf{c}}$	the sequence of estimated complex gain of a flat fading channel
C	a code
$C(n,k)$	a rate- n/k code
d_k	the differentially encoded symbol at time k
$\mathbf{d}$	the sequence of differentially encoded symbols
$e_k(u_k)$	the extrinsic information about the symbol u_k
$e_{i,k}(u_k)$	the extrinsic information produced by a code C_i about the symbol u_k
$e_{i,k}^{(l)}(u_k)$	the extrinsic information produced by a code C_i about the symbol u_k at iteration l
$\mathbf{e}_i^{(l)}$	the vector of extrinsic information produced by a code C_i at iteration l
E_s	the average received energy per symbol

E_b	the average received energy per bit
$E[X]$	the expected value of a random variable X
E	the evidence event
f_d	the maximum Doppler spread, also termed fading rate
f_c	the carrier frequency
$f(t)$	the combine response of $g(t)$ and $h(t)$
$g(t)$	the pulse-shaping filter
$h(t)$	the receive filter
h	the discrete (FIR) interpolation filter
I	the identity matrix
k	a discrete time index, in sample periods
K	the constraint length of a convolutional code
l	the iteration index
L	the total number of iteration
$L(X)$	the log-likelihood ratio of a binary random variable X
$L_e^{[C]}(x)$	the log-likelihood ratio of extrinsic information added by the decoder for code C about symbol x
$L_e^{[\Psi]}(x)$	the log-likelihood ratio of “extrinsic” information added by the demodulator about symbol x

m	a state in a trellis
m'	a state in a trellis
m''	a state in a trellis
$m_{initial}$	the known initial state of a trellis
m_{final}	the known final state of a trellis
(m', m)	a state transition in a trellis from state m' to state m
N	the message length
$N_0/2$	the noise power spectral density
$p_i^{[j]}$	the i^{th} coefficient of an j^{th} -order linear predictor for the sequence $y'(\hat{x})$
P	the average power of the channel gain $c(t)$
$\mathbf{P}$	the transformation matrix formed by the linear predictors
$\mathbf{P}$	the puncturing operation
$\mathbf{P}^{-1}$	the “un-puncturing” operation
Q	the number of modulation levels
$\mathbf{r}$	the sequence of received pilot symbols separated by all-zero samples
$R_c[k]$	the discrete autocorrelation of the sequence c
$\mathbf{R}_c$	the autocorrelation matrix of the sequence c
$\mathbf{R}_r$	the autocorrelation matrix of the sequence $\mathbf{r}$

$\mathbf{R}_{y'}$	the covariance matrix of the remodulated sequence $\mathbf{y}'(\hat{\mathbf{x}})$
$\mathbf{s}$	the sequence of states an a trellis
s_k	the k^{th} element of the state sequence $\mathbf{s}$
(s_{k-1}, s_k)	the state transition in the sequence $\mathbf{s}$ for time $k-1$ to time k
$s(t)$	the transmitted signal
S	the spreading factor of the spread interleaver
t	the time index
T	the symbol duration
$\mathbf{u}$	the sequence of message symbols
u_k	the k^{th} element of the message symbols sequence $\mathbf{u}$
v	the mobile speed
$\mathbf{w}$	the cross correlation of $\mathbf{c}$ and $\mathbf{r}$
x_k	the k^{th} element of the coded symbols sequence $\mathbf{x}$
$\mathbf{x}$	the sequence of coded symbols
$\hat{\mathbf{x}}$	the hypothesized sequence of transmitted symbols
$y(t)$	the received signal, at the output of the matched filter
$\mathbf{y}$	the sequence of received symbols
y_k	the k^{th} element of the message received sequence $\mathbf{y}$

y_i^j	the sub-sequence of the received symbols y beginning with element the i^{th} element and ending with the j^{th} element
$y'_k(\hat{x}_k)$	the remodulated received sample y_k , conditioned on the hypothesis $\hat{x}_k$
$y'(\hat{x})$	the remodulated received sequence y , conditioned on the hypothesis $\hat{x}$
$y''(\hat{x})$	the prediction error sequence for the remodulated sequence $y'(\hat{x})$
z^{-1}	a unit sample delay
$\alpha_k(m)$	the sum of the product of the weights $\gamma_k(m',m'')$ along all paths which terminate in state m at time k
$\beta_k(m)$	the sum of the product of the weights $\gamma_k(m',m'')$ along all paths to the end of the trellis which begin in state m at time k
$\gamma_k(m',m)$	the joint probability of a state transition from $s_{k-1}=m'$ to $s_k = m$ and of the received sample y_k conditioned on the state $s_{k-1}=m'$
δ	$1/p(y)$
Δ_k	the differentially detected symbol at time k
Δ	the sequence of differentially detected symbols
$\lambda_{vu}(u)$	the message sent by a node V to its parent node U in a Bayesian graph
$\Lambda(X)$	the likelihood ratio of a binary random variable X
$\Lambda_e^{(C)}(x)$	the likelihood ratio of extrinsic information added by the decoder for code C about symbol x

$\Lambda_e^{[\Psi]}(x)$	the likelihood ratio of “extrinsic” information added by the demodulator about symbol x
Ω	the evidence set
$\pi_{i,k}^{(l)}(u_k)$	the <i>refined a priori</i> probability of symbol u_k at the input of the i^{th} decoder at iteration l
$\pi_i^{(l)}$	the <i>refined a priori</i> probabilities at the input of the i^{th} decoder at iteration l
$\pi_{UV}(u)$	the message sent by a node U to its child node V in a Bayesian graph
Π	channel interleaver
Π^{-1}	channel deinterleaver
Π_t	concatenation interleaver
Π_t^{-1}	concatenation deinterleaver
Ψ	the channel/modulation memory

Chapter 1 – Introduction

Since turbo codes offer excellent performance in very low SNR conditions, they are considered as very promising for harsh environments, such as wireless communications. A lot of attention has been given lately to the application of turbo codes to multipath fading channels [1]-[4], particularly since most third-generation (3G) wireless systems are expected to use turbo codes for some services. Furthermore, in the long term, the performance achievable with turbo codes, or with other iteratively decodable concatenated codes, promises significant improvements in the capacity of mobile communications systems. These powerful channel coding schemes, combined with other techniques, will allow current and future mobile communications systems to operate at very low signal-to-noise ratios.

Theoretically, on flat-fading channels, turbo codes and coherent PSK allow very low BER's at low SNR's, if the channel gain, also termed Channel State Information (CSI), is known [1]. However, at such low SNR's, it is difficult to perfectly track both the amplitude and the phase of a rapidly changing channel. Although the problem of unknown amplitude is studied in [1] and [2], the authors assume perfect phase tracking. The frequent phase slips generated by a phase tracking loop in fading conditions generally result in an important additional performance degradation at low SNR's.

The use of a coherent receiver is therefore not practical in such conditions. Non-coherent or pseudo-coherent demodulation techniques, such as differential detection and pilot symbol-assisted detection, are hence more suitable for usage in conjunction with

turbo codes, since they do not require the presence of an external amplitude or phase tracking loop at the receiver side. It is therefore desirable to use robust non-coherent or pseudo-coherent demodulation schemes based on either differential encoding or Pilot Symbol-Assisted Modulation (PSAM).

Well-known sub-optimal demodulation techniques exist for differentially encoded and PSAM signals. One-symbol differential detection is a simple and widely spread sub-optimal demodulation technique for differentially encoded signals [3]. For pilot symbol-assisted signals, Minimum Mean-Square Error (MMSE) filtering can be used to estimate the channel gain, and thus to perform pseudo-coherent detection [5][6]. Nevertheless, these techniques usually show significant performance degradation in fast fading conditions with respect to ideal coherent detection (with known CSI).

For Rayleigh flat fading channels for which the complex gain is unknown, it can be shown that the optimum demodulator, minimizing the probability of symbol error given the received signal, consists in a linear-predictive symbol-by-symbol Maximum *A Posteriori* (MAP) demodulator [7][8]. For coded signals, conventional receiver structures separate demodulation and decoding in two different processes. Ideally, the receiver should minimize the overall probability of error, in a unique process encompassing both demodulation and decoding. This is impractical due to the usual presence of interleaving between the channel encoder and the modulator. A simpler way to estimate the optimum decision, in the minimum error probability sense, is to use iterative techniques, similar to what has been proposed for the decoding of parallel concatenated convolutional (turbo)

codes. It is therefore possible to extend the turbo decoding algorithm to the problem of joint demodulation and decoding in the presence of interleaving. In such an arrangement, the demodulator needs to exploit the memory (correlation) in the signal and the channel, and to generate soft-outputs. Linear-predictive receivers, based on PSAM or differential encoding, possess these characteristics and then appear attractive in iterative joint demodulation and decoding schemes.

Such techniques have been proposed in [8] and [9], for PSAM and differentially encoded signals respectively. In this thesis, the iterative demodulation and decoding techniques of [8] and [9] are extended for the use of turbo-coded PSK signals, on Rayleigh flat fading channels.

1.1 Objectives

The objectives pursued in this thesis are detailed in the following sub-sections.

1.1.1 Iterative Demodulation and Decoding of Turbo-Coded Signals

It has been demonstrated, in [8] and [9], that the performance of the iterative demodulation and decoding of convolutionally encoded PSK signals in Rayleigh fading, approaches the performance of an ideal coherent detector. It is of interest, using the linear-predictive MAP demodulators of [8] and [9] for PSAM or differentially encoded signals, to apply the concept of iterative demodulation and decoding to turbo-coded signals. The main objective of this work is therefore to extend the iterative techniques of [8] and [9] to turbo-coded signals.

1.1.2 Application of Bayesian Graph Theory to Joint Demodulation and Decoding

A Bayesian graph is a graphical model that can be used to represent a set of random variables and the associated statistical dependencies. Pearl's algorithm is an iterative algorithm that allows the computation of the *a posteriori* probabilities of random variables represented by a Bayesian graph. It has been shown in [10] that when Pearl's algorithm is applied to the Bayesian graph representing a turbo code, the celebrated turbo decoding algorithm of [11] results. In this thesis, the Bayesian graph theory is applied to the problem of joint demodulation and decoding. The results provide a better insight to the concept of iterative demodulation and decoding, and could eventually allow the extension of this concept to any concatenated code.

1.2 Organization of the Thesis

First, Chapter 2 provides some background information about turbo codes and the iterative decoding principle. The concepts of extrinsic information and of MAP filtering are emphasized, since they play a central role in the remainder of the work.

Chapter 3 describes several soft-output demodulation schemes, as required by the use of turbo decoders. A particular attention is given to linear-predictive MAP demodulators, since these techniques exploit the channel and modulation memory, and are good candidates for iterative demodulation and decoding. Two linear-predictive demodulation schemes are presented: one for pilot symbol-assisted PSK signals, and one for differentially encoded PSK signals.

In Chapter 4, the concept of iterative demodulation and decoding is presented, based

on the linear-predictive demodulation schemes described in Chapter 3. The iterative demodulation and decoding structure is first presented for convolutionally encoded signals. The same concept is extended to the case of turbo-coded signals, yielding a novel iterative demodulation and decoding structure.

A different insight to the problem of joint demodulation and decoding, based on Bayesian graph theory, is proposed in Chapter 5. The basic concepts related to Bayesian graph theory and Pearl's algorithm are presented, and applied to the problem of joint demodulation and decoding of convolutionally encoded and turbo-coded signals. It is shown that in the case of convolutionally encoded signals, Pearl's algorithm yields an iterative structure corresponding precisely to the already proposed iterative demodulation and decoding structures. It is also demonstrated that when Pearl's algorithm is applied to the case of turbo-coded signals, the resulting iterative demodulation and decoding scheme corresponds to the new algorithm derived in Chapter 4.

The performance of this algorithm, in Rayleigh flat fading conditions, is evaluated through simulations in Chapter 6. The results of this thesis are finally summarized in Chapter 7, and suggestions for future research are described.

Chapter 2 - Turbo Decoding

Shannon's information theory showed that there exists channel codes making it possible to transmit information over an Additive White Gaussian Noise (AWGN) channel at a binary information rate C_b or lower with an error rate as low as desired, given the noise power, the signal power and the bandwidth of the signal [12]. This information rate is known as the *channel capacity*, and it allows the computation of a theoretical lower bound for the Signal-to-Noise Ratio (SNR) required to achieve reliable communications at given information rate and signal bandwidth. In order to reach the SNR limit for reliable communications predicted by Shannon's information theory, the construction of a channel code with very high decoding computational complexity and infinite block length is required.

However, such a code is unusable in a practical system. Although the rapid development of the microelectronic industry has increased considerably the available computational power deliverable by ASICs, DSPs and other microprocessors, cost and power consumption issues are still limiting the computational resources available in current communications systems. Furthermore, using very long codes could cause an important transmission latency, which is not desirable for most communications systems. The most important challenge in channel coding theory is thus to design error-correcting codes with high distance properties, and hence high error-correction capabilities, within a reasonable decoding complexity and delay.

Concatenated codes offer a very efficient solution to this problem. Moreover, the

iterative decoding of concatenated codes shows very impressive performances. For instance, the iterative decoding of parallel concatenated recursive systematic convolutional codes, also known as *turbo codes*, can have performance very close to the channel capacity [11].

In this chapter, the basic concepts related to iterative decoding are presented. These concepts are essential to the understanding of turbo decoding and of iterative demodulation and decoding, as treated in subsequent chapters. First, the notion of concatenated coding is explained in Section 2.1. In Section 2.2, optimal soft-in/soft-out decoding is presented, since it is required by iterative decoding schemes. The concept of MAP filtering, which is fundamental for iterative demodulation and decoding schemes, is also defined in this section. Finally, the iterative decoding principle is explained in Section 2.3. A particular emphasis is given to the notion of extrinsic information, since it plays a central role in iterative techniques.

2.1 Concatenated Codes

2.1.1 General Concept

Forney has first introduced the concept of concatenated coding in [13]. The basic idea behind concatenated codes is to build a powerful code by concatenating two (or more) less powerful codes with simple decoding algorithms and short decoding delays. Although the individual codes forming a concatenated code are not necessarily very powerful, the resulting code can be. Since the constituent codes require a reasonable

amount of decoding complexity and delay, the overall complexity and delay of the concatenated code remains practical.

2.1.2 Serial Concatenation

Several concatenation techniques are possible. The most obvious is serial concatenation, where the constituent codes are applied to the message symbols in a sequential manner, as illustrated in Figure 1. First, the outer code is applied to the message bits. The resulting symbols go through a concatenation interleaver, and are encoded using the inner code. The inner and outer codes are either convolutional codes or block codes, or a combination of the two methods. In the block coding case, one block of the concatenated code usually spans more than one block of the outer block code, otherwise the concatenation interleaver would be inefficient. This is so because the goal of the concatenation interleaver is to allow the corresponding de-interleaver to spread the error bursts at the output of the inner decoder over several outer blocks in order to improve the probability of correct decision of the outer decoder. If interleaving is performed over a single block of the outer block code, then the interleaver is useless. Since each block has a limited error-correcting capability, spreading an error burst between the different outer blocks increases the overall probability of correct decision.

Figure 1 – Transmitter end of a serial concatenation scheme.

The most popular implementation of a serially concatenated code is the concatenation of a convolutional inner code and a Reed-Solomon outer code [14]. Convolutional codes, due to their memory, produce bursts of error when a decoding error occurs. Since Reed-Solomon codes are symbol-based codes, they are very efficient at correcting error bursts. When the concatenated code spans more than one Reed-Solomon block, the symbol-based concatenation interleaver can spread the error bursts produced by the convolutional decoder between the different Reed-Solomon blocks. The resulting concatenated code can be very powerful. This concatenated structure has thus been exploited extensively for space communications, where the transmitted power is crucial.

2.1.3 Parallel Concatenation of Recursive Systematic Convolutional Codes

An interesting alternative to serial concatenation is parallel concatenation. With this latter approach, the message bits are encoded twice using two systematic codes. Therefore, the systematic bits are sent with the parity bits produced by the two constituent codes. These constituent codes can be systematic convolutional codes. In this case, the systematic bits are interleaved before being encoded with the second constituent

code, as illustrated in Figure 2. The concatenation interleaver is represented as Π_t , while the puncturing operation is represented as P. The puncturing operation on the coded symbols is required in order to obtain the desired coding rate.

If these concatenated codes are decoded using the serial concatenation principles, the two constituent codes must be decoded sequentially. One of the constituent convolutional codes is decoded first using a soft-output decoder, in order to avoid the hard-input decoding penalty for the second constituent convolutional code. Afterwards, the other convolutional code can be decoded using the soft outputs of the first constituent decoder.

Figure 2 – Parallel concatenation of recursive systematic convolutional codes.

However, the simple sequential decoding by the constituent decoders of such a code does not exploit its strength. Although the second decoder can use the information produced by the first decoder, the first decoder does not do the same with the information

produced by the second decoder. Since both codes are systematic and encoding the same message bits, the constituent codes can be decoded without a specific order. There is no particular reason to decode one code before the other. Furthermore, even though both codes have been decoded, it is possible to decode again using any of the constituent codes to improve the decisions. A possible decoding scheme is thus to iterate between the two constituent decoders to improve the decisions at each iteration.

Note that this iterative scheme is efficient if the constituent decoders accept soft-inputs and produce soft-output decisions. Otherwise, the iterative gain is cancelled by the performance degradation produced by the hard-input decoding of the convolutional codes. It is also necessary to define a method to combine the information about the systematic bits observed from the channel to that produced by a constituent decoder in order to feed it to the other decoder.

Such a decoding scheme has been proposed in [11], a 1993 conference paper by Berrou, Glavieux and Thitimajshima. In this celebrated paper, the iterative decoding algorithm is called turbo decoding, in analogy with turbo engines. Note that the idea of using iterative decoding schemes with soft-in/soft-out decoders was also proposed during the same period by other authors [15]. However, the idea of using the parallel concatenation of recursive systematic convolutional codes with pseudo-random interleaving was first proposed in [11]. The impressive results obtained with the concatenation scheme of [11] are probably responsible for the fact that this paper is usually referred as the origin of the iterative decoding algorithm.

In the remaining of this chapter, the basics of iterative decoding are presented using the parallel concatenation scheme proposed in [11]. Note that the same principles can also be applied to other concatenation schemes, such as serial concatenated codes and product codes.

2.2 Soft-In/Soft-Out Decoding

In this section, the soft-output decoding of convolutional codes is presented. Note that soft-in/soft-out decoding schemes can also be derived for linear block codes, and hence, that such codes can also be used in iterative decoding schemes [16]-[18].

For convolutional codes, it is first important to notice that the k^{th} output x_k of a convolutional encoder can be interpreted as the output of a Markov source, as illustrated in Figure 3. Either a state transition diagram or a trellis diagram can represent such a source [19]. For a convolutional code C , the state of the Markov source is associated with the memory of the encoder. The output x_k of the Markov source is associated with the state transition from s_{k-1} to s_k , denoted (s_{k-1}, s_k) . Therefore, the message sequence $\mathbf{u}$ (and the corresponding output sequence $\mathbf{x}$) of a convolutional encoder can be uniquely associated with a state sequence $\mathbf{s}$ of the Markov source, or equivalently, with a path through the trellis. Also, a unique message symbol u_k (or output symbol x_k) can be associated with a state transition (s_{k-1}, s_k) . In other words, knowledge of the state transition (s_{k-1}, s_k) also implies the knowledge of the message symbol u_k and the output symbol x_k . On the other hand, a message symbol u_k (or output symbol x_k) can be associated to more than one of the possible state transitions. Note that in the case of a recursive systematic

convolutional encoder, the message symbol u_k is included in the output symbol $x_k = (u_k, x_{1k})$, where x_{1k} is the non-systematic output symbol of the convolutional encoder.

Figure 3 – Markov source interpretation of a convolutional code over a DMC.

In conventional non-iterative decoding schemes, the probability of the observed noisy output sequence $\mathbf{y}$ conditioned on the transmitted message symbol sequence $\mathbf{u}$ and the knowledge of the structure of code C , $p(\mathbf{y} | \mathbf{u}, C)$, is used to perform the decoding decision. The expression $p(\mathbf{y} | \mathbf{u}, C)$ denotes the probability of observing $\mathbf{y}$ knowing the message $\mathbf{u}$ and the structure of code C . Equivalently, the probability of the observed output sequence $\mathbf{y}$ conditioned on the state sequence $\mathbf{s}$, $p(\mathbf{y} | \mathbf{s})$, can also be used. The rule consisting in finding the state transition sequence that maximizes the probability $p(\mathbf{y} | \mathbf{u}, C)$ (or $p(\mathbf{y} | \mathbf{s})$) is usually referred as Maximum-Likelihood Sequence Estimation (MLSE). An efficient way to find the maximum-likelihood sequence of a Markov source is given by the Viterbi algorithm [20]. It is worth noting that for a uniformly distributed memoryless sequence $\mathbf{u}$, the hard decisions obtained in the MLSE sense are equivalent to the optimal hard decisions, in the minimum probability of symbol error sense, based on

the *A Posteriori* Probabilities (APP) $p(u_k | \mathbf{y}, C)$. In this case, since

$$p(\mathbf{y} | \mathbf{u}, C) = \prod_k \frac{p(u_k | \mathbf{y}, C)}{p(u_k)} \cdot p(\mathbf{y}), \quad (1)$$

finding the sequence $\mathbf{u}$ that maximizes $p(\mathbf{y} | \mathbf{u}, C)$ corresponds to finding the symbols u_k that maximizes the APP $p(u_k | \mathbf{y}, C)$ for all time indices k .

Note however that the MLSE rule identifies the most likely transmitted message, and thus, it only produces hard decisions. Hard outputs are sufficient for non-iterative decoding schemes, particularly when there is no outer code or when the outer code does not suffer significantly from hard inputs. In the case of iterative decoding schemes, the true APP $p(u_k | \mathbf{y}, C)$ must be computed for each message bits. An optimal algorithm to compute the APP's is presented in Section 2.2.1, while lower-complexity sub-optimal algorithms will be briefly proposed in Section 2.2.3.

2.2.1 MAP Decoding Using the BCJR Algorithm

An efficient algorithm to compute the true APP's of a Markov source is the BCJR algorithm, also known as the forward-backward algorithm, developed by Bahl et al. [16]. It can thus be used to compute the APP's of the message symbols encoded with a convolutional code. In this case, the APP of the transmitted symbol u_k can also be expressed as

$$p(u_k = q | \mathbf{y}, C) = \sum_{(m', m): u_k = q} p(s_{k-1} = m', s_k = m | \mathbf{y}), \quad (2)$$

where $(m', m) : u_k = q$ denotes all the state transitions (m', m) for which $u_k = q$. Thus,

$$p(u_k = q | y, C) = \frac{\sum_{(m', m): u_k = q} p(s_{k-1} = m', s_k = m, y)}{p(y)} = \frac{\sum_{(m', m): u_k = q} p(s_{k-1} = m', s_k = m, y)}{\sum_{(m', m)} p(s_{k-1} = m', s_k = m, y)}. \quad (3)$$

The BCJR algorithm provides an efficient way to compute the joint probability $p(s_{k-1} = m', s_k = m, y)$. The complete algorithm is detailed in Annex A.

Using this algorithm, the true APP's can be computed for the message symbols $\mathbf{u}$. These true APP's can be interpreted as the optimal soft outputs of the decoder. Furthermore, the optimal decision, in terms of minimum probability of error, can also be derived from the true APP.

2.2.2 MAP Filtering

It is also possible to compute the APP's for the coded symbols $\mathbf{x}$. Knowing that

$$p(x_k = q | y, C) = \sum_{(m', m): x_k = q} p(s_{k-1} = m', s_k = m | y), \quad (4)$$

the BCJR algorithm can also be applied to compute these APP's. Such a computation will be termed *MAP filtering*, as in [15], while computing the APP of the message symbols $\mathbf{u}$ will be referred as *MAP decoding*.

2.2.3 Implementation Issues

As shown in the previous sections, the BCJR algorithm allows the computation of the true APP's of the message symbols $\mathbf{u}$ or the coded symbols $\mathbf{x}$. However, the BCJR algorithm operates in the probability domain, which makes it difficult to implement using

conventional precision-limited floating-point representations. With conventional floating-point representations, probabilities very close to 0 can be represented with good precision. On the other hand, probabilities very close to 1 cannot be represented with the same precision, and hence, such a probability would be mapped to the value of 1. This could result in division by zero errors. Adding a very small but non-zero value ε to each probability is a possible solution to this problem. Nevertheless, the BCJR algorithm will suffer from the floating-point representation precision limitation. This is particularly true when very low error rates are required.

An elegant way to avoid the latter implementation problem is to work in the log-probability domain. Since the channel is usually corrupted by Additive White Gaussian Noise (AWGN), the logarithm of $p(y_k | x_k)$ can be easily computed in the log-probability domain. Note that in log-probability domain, probabilities very close to either 0 or 1 can be represented with a much larger precision than using the probability domain. Such a log-based MAP decoding algorithm is referred as Log-MAP decoding. It can be shown that it is possible to implement an optimal MAP algorithm operating in the log domain [21].

Another factor limiting the use of the BCJR algorithm in a practical system is the amount of complexity involved. Usually, the BCJR algorithm is considered as too complex for most practical implementations. As a result, several sub-optimal algorithms, using APP approximation and requiring less computational complexity than the BCJR algorithm, have been developed. A low-complexity algorithm can be obtained by

modifying the Viterbi algorithm to produce approximated APP's [22][23]. This algorithm is generally referred to as the Soft-Output Viterbi Algorithm (SOVA). Another popular sub-optimal algorithm is the Max-Log-MAP algorithm [21][23]. This algorithm is slightly more complex than the SOVA, but shows much less approximation degradation with respect to the optimal APP's.

2.3 Iterative Processing

Unlike for a single code, there is no efficient way to compute the APP's in the case of parallel concatenated codes with interleaving. The presence of a concatenation interleaver prevents the use of the BCJR algorithm, since the coded symbols can no more be viewed as the output of a Markov source. In addition, the brute force computation of the APP's for the equivalent code requires a prohibitive computational complexity. Iterative decoding of such codes, as proposed in [11], is an interesting solution to the problem since it allows the approximation of the true APP in a computationally simple, yet precise fashion.

In this section, the principle of turbo decoding will be presented for the case of parallel concatenated recursive systematic convolutional codes (also known as turbo codes). Similar iterative decoding schemes can be derived for different concatenated codes [15][17]. The iterative processing principle presented in this section for turbo codes will also be extended to the case of iterative demodulation and decoding in Chapter 4.

2.3.1 Extrinsic Information

Let us take a closer look at the APP's for a recursive systematic convolutional encoder C_1 . First, the APP, expressed by equation (2), could also be written

$$\begin{aligned}
 p(u_k = q | y, C_1) &= \frac{p(y, u_k = q | C_1)}{p(y)} \\
 &= \delta \sum_{u: u_k = q} p(y, u | C_1) \\
 &= \delta \sum_{u: u_k = q} p(y | u, C_1) p(u),
 \end{aligned} \tag{5}$$

where δ is equal to $1/p(y)$. In the case of a single recursive systematic convolutional encoder, such as code C_1 in Figure 2, the message symbols u are included in the output symbol sequence $x = (u, x_1)$, where x_1 is the non-systematic output symbol sequence of the convolutional encoder C_1 . Therefore,

$$p(y | u, C_1) = p(y | x, C_1) = p(y_s | u) p(y_1 | x_1, C_1), \tag{6}$$

where y_s and y_1 are the sequence of noisy observations of u and x_1 respectively over a memoryless channel. Then, for the first decoder in Figure 2, equation (5) becomes [15][10]

$$\begin{aligned}
 p(u_k = q | y, C_1) &= \delta \sum_{u: u_k = q} p(y_1 | x_1, C_1) p(y_s | u) p(u) \\
 &= \delta \sum_{u: u_k = q} \left[p(y_1 | x_1, C_1) \prod_j p(y_{s,j} | u_j) p(u_j) \right] \\
 &= \delta p(y_{s,k} | u_k) \cdot p(u_k) \cdot \underbrace{\left(\sum_{u: u_k = q} \left[p(y_1 | x_1, C_1) \prod_{j: j \neq k} p(y_{s,j} | u_j) p(u_j) \right] \right)}_{\text{Extrinsic information}}
 \end{aligned} \tag{7}$$

for a memoryless source and a memoryless channel.

The first term in equation (7), $p(y_{s,k} | u_k)$, is the soft-output of the channel. This information is known prior to the decoding process. The second term of this equation, $p(u_k)$, is the *a priori* probability about the message symbols, which is also known before decoding. Conversely, the third term of the equation is dependent on the code structure, and hence, is unknown before decoding. It can therefore be interpreted as a refinement information added by the decoder [15]. This information is usually termed *extrinsic* information [11], and is defined as

$$e_{l,k}(u_k) = \sum_{u: u_k=q} \left[p(y_l | x_l, C_l) \prod_{j: j \neq k} p(y_{s,j} | u_j) p(u_j) \right], \quad (8)$$

for code C_l .

Note that the first two terms of (7) are used in the computation of (144) for the BCJR algorithm. They can thus be viewed as the input to the MAP decoding algorithm for the message symbol u_k . If the output of the MAP decoding algorithm is expressed as in equation (7), it is obvious that the extrinsic information produced by a MAP decoding algorithm can be computed knowing the APP produced by this MAP decoder and the input to the decoder using, from (7) and (8),

$$e_{l,k}(u_k) = \frac{p(u_k = q | y, C_l)}{\delta p(y_{s,k} | u_k) p(u_k)}. \quad (9)$$

Since the BCJR algorithm is the most efficient way to compute the true APP, it is easier to compute the extrinsic information produced by a MAP decoder from its input and the

computed APP's using (9), than it is to compute the exhaustive summation of (8). A MAP decoder can thus be modified to compute efficiently both the APP's and the extrinsic information.

Note that equations (8) and (9) are not applicable only to recursive systematic convolutional codes. Any systematic code produces extrinsic information about the message symbols $\mathbf{u}$. The concept of extrinsic information can also be extended to the non-systematic coded symbols of a systematic code or, more generally, to the coded symbols of a non-systematic code. In this case, the extrinsic information of the coded symbol x_k can be defined as

$$e_{1,k}(x_k) = \frac{p(x_k = q | \mathbf{y}, C_1)}{\delta p(y_k | x_k) p(x_k)} \quad (10)$$

for code C_1 . A MAP filter can be used to compute the APP's of the coded symbols (equation (4)).

2.3.2 Iterative Decoding of Turbo Codes

Using a similar development as in Section 2.3.1, the APP for a message symbol u_k estimated from a turbo code C , formed by constituent codes C_1 and C_2 , can be expressed as

$$\begin{aligned}
p(u_k = q | \mathbf{y}, C) &= \frac{p(\mathbf{y}, u_k = q | C)}{p(\mathbf{y})} \\
&= \delta \sum_{\mathbf{u}: u_k = q} p(\mathbf{y}, \mathbf{u} | C) \\
&= \delta \sum_{\mathbf{u}: u_k = q} p(\mathbf{y} | \mathbf{u}, C) p(\mathbf{u}) \\
&= \delta \sum_{\mathbf{u}: u_k = q} p(y_1 | \mathbf{x}_1, C_1) p(y_2 | \mathbf{x}_2, C_2) p(y_s | \mathbf{u}) p(\mathbf{u}) \\
&= \delta \sum_{\mathbf{u}: u_k = q} \left[p(y_1 | \mathbf{x}_1, C_1) p(y_2 | \mathbf{x}_2, C_2) \prod_j p(y_{s,j} | u_j) p(u_j) \right] \\
&= \delta p(y_{s,k} | u_k) \cdot p(u_k) \cdot \underbrace{\left(\sum_{\mathbf{u}: u_k = q} \left[p(y_1 | \mathbf{x}_1, C_1) p(y_2 | \mathbf{x}_2, C_2) \prod_{j: j \neq k} p(y_{s,j} | u_j) p(u_j) \right] \right)}_{\text{Extrinsic information}} \quad (11)
\end{aligned}$$

where the symbol sequence $\mathbf{x}$ is formed by $(\mathbf{u}, \mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{y}$ is formed by (y_s, y_1, y_2) . Here again, the third term of the multiplication in (11) is the extrinsic information added by the code structure. Note that evaluating this summation would require extensive computations.

A simple way to estimate that extrinsic information for a turbo code would be to combine the extrinsic information produced by the two constituent decoders, since the extrinsic information can be easily computed for the individual constituent codes. However, estimating the extrinsic information of the turbo code by the product of the individual extrinsic information of the constituent codes would be insufficient since neither code could take advantage of the information added by the other code, i.e. the extrinsic information produced by the other decoder. In order to allow both decoders to exploit the extrinsic information produced by the other decoder, an iterative decoding scheme is required.

The two constituent codes can thus be decoded alternatively. In this thesis, one iteration is defined as one complete round of decoding for both constituent codes. At iteration l , the extrinsic information produced by the first decoder can be used by the second decoder as a refinement factor to the *a priori* distribution of the message symbols $\mathbf{u}$. The refined estimate of the *a priori* probability at the input of the second decoder can be expressed as

$$\pi_{2,k}^{(l)}(u_k) = p(u_k)e_{1,k}^{(l)}(u_k), \quad (12)$$

where $e_{1,k}^{(l)}(u_k)$ is the extrinsic information about the message symbol u_k at iteration l from the first constituent code. The extrinsic information produced by the second decoder at iteration l is then

$$e_{2,k}^{(l)}(u_k) = \sum_{u: u_k = q} \left[p(\mathbf{y}_2 | \mathbf{x}_2, C_2) \prod_{j: j \neq k} p(y_{s,j} | u_j) \underbrace{p(u_j)e_{1,j}^{(l)}(u_j)}_{\text{Refined } a \text{ priori probability}} \right], \quad (13)$$

where $e_{2,k}^{(l)}(u_k)$ is the extrinsic information about the message symbol u_k at iteration l from the second constituent code. Note again that this extrinsic information can be computed efficiently using the BCJR algorithm on the trellis produced by the second constituent code, instead of computing the exhaustive summation. In this case, the *a priori* information about the message bits required by the BCJR algorithm (equation (144)) is refined using the extrinsic information produced by the first decoder (equation (12)).

Similarly, the first constituent code can be decoded again in the next iteration ($l+1$) using the extrinsic information produced by the second decoder. The refined *a priori* distribution used by the first decoder at iteration $l+1$ is then

$$\pi_{1,k}^{(l+1)}(u_k) = p(u_k) e_{2,k}^{(l)}(u_k), \quad (14)$$

and the extrinsic information produced by the first decoder for this iteration is

$$e_{1,k}^{(l+1)}(u_k) = \sum_{u: u_k = q} \left[p(y_1 | x_1, C_1) \prod_{j: j \neq k} p(y_{s,j} | u_j) \underbrace{p(u_j) e_{2,j}^{(l)}(u_j)}_{\text{Refined } a \text{ priori probability}} \right]. \quad (15)$$

Once more, in practice this extrinsic information can be computed efficiently using the BCJR algorithm and the expression (similar to (9))

$$e_{1,k}^{(l+1)}(u_k) = \frac{p(u_k = q | y, C_1)}{\delta p(y_{s,k} | u_k) p(u_k) e_{2,k}^{(l)}(u_k)}. \quad (16)$$

The overall iterative decoding scheme is illustrated in Figure 4. The concatenation interleaver and deinterleaver are represented as Π_t and Π_t^{-1} respectively. At the output of the first MAP decoder, the extrinsic information $e_1^{(l)}$ is extracted, knowing the systematic symbol *a priori* distribution $p(u_k)$ and the soft channel decision $p(y_{s,k} | u_k)$. This extrinsic information is used to produce a refined *a priori* distribution $\pi_2^{(l)}$ for the second MAP decoder. Using the same principle, the extrinsic information $e_2^{(l)}$ is used to refine the *a priori* distribution $\pi_1^{(l+1)}$ for the first MAP decoder at next iteration.

Figure 4 – Iterative decoding process of a turbo code.

The iterative process is repeated for $l = \{1, \dots, L\}$, L being the total number of iterations. For the first iteration, the extrinsic information values $e_2^{(0)}$ are instantiated with a unitary value. After the last iteration L , the final estimated APP for symbol u_k is

$$p(u_k = q | y, C) \approx \delta p(y_{s,k} | u_k) \cdot p(u_k) \cdot \underbrace{e_{1,k}^{(L)}(u_k) \cdot e_{2,k}^{(L)}(u_k)}_{\text{Estimated extrinsic information}}. \quad (17)$$

The iterative decoding process can also be defined by using likelihood ratios. For a random variable x_k , defined over GF(2) with elements $\{+1, -1\}$, the likelihood ratio is defined as

$$\Lambda(x_k) = \frac{p(x_k = +1)}{p(x_k = -1)}. \quad (18)$$

This ratio can also be expressed in the log domain with the log-likelihood ratio [17]

$$L(x_k) = \text{Log}[\Lambda(x_k)]. \quad (19)$$

The sign of the log-likelihood ratio corresponds to the hard decision, while its magnitude expresses the reliability of the decision.

Using this likelihood ratio notation, the structure illustrated in Figure 4 can be simplified, since if equally likely information symbols are assumed, then $\Lambda(u_k) = 1$ and $L(u_k) = 0$. The iterative demodulation and decoding process of a turbo code operating in the log domain is illustrated in Figure 5. Note that in the log domain, multiplications and divisions are becoming additions and subtractions.

Figure 5 – Iterative decoding process of a turbo code in the log-likelihood domain.

The likelihood ratios for the extrinsic information $e_{1,k}(u_k)$ and $e_{2,k}(u_k)$ are defined as

$$L_e^{[C_1]}(u_k) = \text{Log} \left[\frac{e_{1,k}(u_k = +1)}{e_{1,k}(u_k = -1)} \right], \quad (20)$$

and

$$L_e^{[C_2]}(u_k) = \text{Log} \left[\frac{e_{2,k}(u_k = +1)}{e_{2,k}(u_k = -1)} \right]. \quad (21)$$

The log-likelihood of the channel soft-outputs $p(y_k | x_k)$ is

$$L(y_k | x_k) = \text{Log} \left[\frac{p(y_k | x_k = +1)}{p(y_k | x_k = -1)} \right]. \quad (22)$$

After the last iteration, the ratio of the estimated APP's is therefore (equation (17))

$$\begin{aligned}
L(u_k | y, C) &\approx \text{Log} \left[\frac{p(y_{s,k} | u_k = +1)}{p(y_{s,k} | u_k = -1)} \cdot \frac{p(u_k = +1)}{p(u_k = -1)} \cdot \frac{e_{1,k}(u_k = +1)}{e_{1,k}(u_k = -1)} \cdot \frac{e_{2,k}(u_k = +1)}{e_{2,k}(u_k = -1)} \right] \\
&\approx L(y_{s,k} | u_k) + L_e^{[C_1]}(u_k) + L_e^{[C_2]}(u_k),
\end{aligned} \tag{23}$$

assuming equally likely information bits. Using this notation, the output of the turbo decoder is thus the sum of the demodulation output and of the extrinsic information added by the component codes C_1 and C_2 .

2.3.3 Convergence

Ideally, the iterative decoding algorithm should be applied until convergence. There are several possible ways to ensure that the iterative decoding has converged. The most obvious is to fix the number of iteration L to a value large enough to ensure the convergence in most conditions. However, this method wastes computational resources since the number of iterations required to obtain convergence might be highly variable from one data block to the other. A block only slightly affected by noise might be decoded correctly with a single iteration. For this reason, it would be advantageous to stop the iterative decoding algorithm as soon as it has converged, in order to reduce the computations required to a minimum. A convergence measure is thus desirable.

It has been demonstrated in [24] that the optimal decoding decision for a concatenated code, based on the channel soft outputs, is the one that minimizes the cross-entropy between the *a posteriori* distributions at the output of the constituent codes. The cross-entropy of two distributions is a measure of the difference between these

distributions. It has been proposed in [17] to use the cross-entropy of the distributions at the outputs of the constituent decoders as a convergence measure. It was observed that, when the cross-entropy measure converges to a minimum value, no more errors could be corrected by the iterative decoding scheme. The cross-entropy can thus be used to detect the convergence of the iterative decoding process.

Simpler solutions to the problem of convergence detection have also been proposed. For instance, it is proposed in [25] to compare the hard decisions corresponding to the maximum likelihood paths in the constituent decoders, as computed by the Viterbi algorithm. Once the hard decisions of the two decoders agree, the iterative decoding process is assumed to have converged. An even simpler scheme would consist in comparing the hard decisions at the output of the constituent MAP decoders. In any case, it has been demonstrated that a convergence test, even though it is approximate, can reduce considerably the amount of iterations required, without affecting significantly the performance of the decoding process [17][25].

2.4 Summary

In this chapter, the concepts related to iterative decoding have been presented, using the iterative decoding of turbo codes as an example. These concepts also apply to other concatenated codes. Furthermore, as proposed in this thesis, the notion of iterative processing can also apply to the problem of joint demodulation and decoding. In order to perform iterative demodulation and decoding of turbo-coded signals, the concepts of soft-in/soft-out decoding, MAP filtering and extrinsic information, as presented in this

chapter, are of first importance. In addition, the concept of soft-output demodulation is essential to the principle of iterative demodulation and decoding. Several soft-output demodulation schemes are proposed in the next chapter.

Chapter 3 - Soft-Output Demodulation

The benefits of soft-output demodulation combined with soft-in decoding are well known [26][27]. Soft-output demodulation is very important for turbo-coded systems, since the MAP decoding of the constituent codes requires the knowledge of the demodulation soft decisions. The optimal soft decision from the demodulator, regarding a symbol x_k , is expressed by the *a posteriori* probability of this symbol conditioned on the observed signal $\mathbf{y}$, $p(x_k | \mathbf{y})$. The symbol x_k maximizing the APP $p(x_k | \mathbf{y})$ is the optimal demodulated symbol decision, in the minimum probability of symbol error sense. For turbo-coded systems, the soft output $p(x_k | \mathbf{y})$ is required, rather than the hard decision corresponding to that APP, and the demodulator used in these systems must be able to compute the APP $p(x_k | \mathbf{y})$.

In this chapter, several soft-output demodulation schemes for flat fading channels are derived within a common framework. The flat fading channel model is first established in Section 3.1. Next, sub-optimal receivers, for PSAM and differentially encoded signals, are described in Section 3.2. These receivers are not suitable for iterative demodulation and decoding, and will be used in the remaining of the thesis as comparison references for the iterative schemes. The linear-predictive receivers, for PSAM and differentially encoded signals, are presented in Section 3.3. These demodulators are good candidates for iterative demodulations and decoding, and they will be used extensively in this thesis. Finally, for all the receivers presented in this chapter, the case of reception diversity is treated in Section 3.4.

Note that the soft-output demodulation schemes are developed for BPSK signals in this chapter. The extension of these schemes for QPSK signals is straightforward, since QPSK signals can be interpreted as two orthogonal BPSK signals. The extension to higher order PSK constellations is also feasible.

3.1 Flat Fading Channel Model

The impulse response of a wireless channel is dependent on the position of the transmitter and the receiver. Thus, depending on the relative speed between the transmitter and the receiver, slow or fast fading will be observed [28]. The characteristics of the channel impulse response result from the sum of all possible paths between the transmitter and the receiver, each one affected by a different attenuation and delay. The time dispersion of the impulse response can be expressed by the delay spread [29]. If the delay spread is larger than the symbol duration, then the channel will introduce Inter-Symbol Interference (ISI). In this case, the channel is said to be time-dispersive, or equivalently, frequency-selective. However, if the delay spread is smaller than the symbol epoch, then the channel impulse response can be modeled as an impulse with variable gain and phase. In that case, the channel shows flat fading characteristics.

Figure 6 – Flat fading channel baseband model.

The complex baseband flat fading channel model used in this thesis is presented in Figure 6. Let x be the sequence of transmitted symbols of a PSK modulator. These symbols are processed by a pulse-shaping filter $g(t)$ limiting the bandwidth of the transmitted signal $s(t)$. The transmitted signal can thus be expressed as

$$s(t) = \sum_{n=1}^{\infty} x_n g(t - nT), \quad (24)$$

where x_n is the n^{th} transmitted symbol, and T is the symbol duration.

The flat fading process $c(t)$ can be modeled as a multiplicative complex gain varying in time with average power P . The additive noise process $n(t)$ is assumed to be Gaussian with a zero mean, and is uncorrelated (or *white*). In order for the received symbols to experience independent noise, the latter assumption can be easily relaxed, as long as the autocorrelation of the noise process is much narrower than the symbol duration. At the receiver, the SNR is maximized using a receive filter $h(t)$ matched to $g(t)$. The cascade of the transmit and receive filters can be expressed as

$$f(t) = g(t) * h(t) = g(t) * g^*(-t). \quad (25)$$

Then, the received signal $y(t)$ can be written as

$$\begin{aligned} y(t) &= \left[c(t) \cdot \sum_{n=1}^{\infty} x_n g(t - nT) \right] * h(t) + n(t) * h(t) \\ &\approx c(t) \cdot \sum_{n=1}^{\infty} x_n f(t - nT) + n(t) * h(t), \end{aligned} \quad (26)$$

where it is assumed that the fading rate is much slower than the symbol rate, such that the channel gain $c(t)$ can be assumed to be constant over the duration of $f(t)$. The conditions under which this assumption is valid are discussed in [30].

The received signal $y(t)$ is sampled once per symbol at the output of the matched filter $h(t)$. The k^{th} sample at the output of the matched filter can thus be expressed as

$$y_k = y(t)|_{t=kT} = c_k \cdot \sum_{n=1}^{\infty} x_n f(kT - nT) + n(t) * h(t)|_{t=kT}, \quad (27)$$

where c_k is the channel gain at sampling time $t = kT$, also termed Channel State Information (CSI). Assuming unit power symbols and filters for the sake of simplicity, the mean power of the channel gain is

$$E[|c_k|^2] = E_s, \quad (28)$$

where E_s is the average received energy per symbol.

The impulse response $f(t)$ is usually designed such that the combination of the filters $g(t)$ and $h(t)$ does not cause ISI. Therefore, $f(t)$ should respect the following conditions for no ISI (Nyquist 1st criterion):

$$f(nT) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

A typical example of a filter $h(t)$ matched to $g(t)$ such that the overall impulse response $f(t)$ satisfies Nyquist 1st criterion is a square-root raised cosine filter [19]. In this case, equation (27) becomes

$$y_k = c_k x_k + n(t) * h(t) \Big|_{t=kT} \quad (30)$$

Furthermore, since $h(t)$ is matched to $g(t)$ and $f(t)$ satisfies Nyquist 1st criterion, the noise samples at the output of the matched filter are uncorrelated (if the continuous time noise process $n(t)$ is uncorrelated). Therefore, equation (30) yields

$$y_k = c_k x_k + n_k, \quad (31)$$

where n_k is the noise sample at the output of the matched filter at time kT . The sequence of received noise samples $\mathbf{n}$ is Gaussian, with zero mean and variance $N_0/2$.

The flat fading process envelope is typically modeled by a Rayleigh distribution [29], given that the Line-Of-Sight (LOS) path is rarely present in terrestrial mobile wireless channels. Using Clarke's model of uniform scattering, it is possible to characterize the fading process in the frequency and time domains. In the frequency domain, Clarke's model results in a U-shaped baseband fading spectrum defined by [29]

$$S_c(f) = \frac{G}{f_d \sqrt{1 - \left(\frac{f}{f_d} \right)^2}}, \quad (32)$$

where G is a constant and f_d is the maximum Doppler spread. The maximum Doppler spread is defined as

$$f_d = \frac{v}{c} f_c, \quad (33)$$

where v is the mobile speed and c is the light speed (3×10^8 m/sec). The *fading rate* is generally defined as the ratio f_d/f_s , where f_s is the symbol rate. In the time domain, the discrete autocorrelation of the fading process $c(t)$ defined by Clarke's model is

$$R_c[k] = J_0(2\pi f_d T k), \quad (34)$$

where $J_0(\cdot)$ is the Bessel function.

3.2 Conventional Demodulation Schemes

3.2.1 Coherent Detection with Known Channel State Information

If the CSI c is known to the receiver, then the pdf of the sample y_k , conditioned on the knowledge of the transmitted symbol x_k and the channel complex gain c_k , can be expressed as [19]

$$p(y_k | x_k, c_k) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{|y_k - x_k c_k|^2}{N_0}\right), \quad (35)$$

and its associated likelihood ratio is

$$\Lambda(y_k | x_k, c_k) = \frac{p(y_k | x_k = +1, c_k)}{p(y_k | x_k = -1, c_k)} = \exp\left(\frac{4 \operatorname{Re}(y_k c_k^*)}{N_0}\right). \quad (36)$$

Therefore, the likelihood ratio associated with the *a posteriori* probability $p(x_k | y_k, c_k)$ is

$$\begin{aligned}
 \Lambda(x_k | y_k, c_k) &= \frac{p(x_k = +1 | y_k, c_k)}{p(x_k = -1 | y_k, c_k)} \\
 &= \frac{p(y_k | x_k = +1, c_k) p(x_k = +1) / p(y_k)}{p(y_k | x_k = -1, c_k) p(x_k = -1) / p(y_k)} \\
 &= \Lambda(y_k | x_k, c_k) \cdot \Lambda(x_k) \\
 &= \exp\left(\frac{4 \operatorname{Re}(y_k c_k^*)}{N_0}\right) \cdot \Lambda(x_k).
 \end{aligned} \tag{37}$$

Usually, the transmitted symbols are assumed to be equally likely a priori, and thus $\Lambda(x_k) = 1$.

The APP value expressed by equation (37) can be interpreted as the optimum coherent detection soft-output for BPSK symbols transmitted without ISI over a slow or moderate flat fading channel [30]. However, at low SNR's, it is often impractical to perfectly track the CSI using amplitude and phase tracking loops. In such conditions, deep channel fades will introduce frequent phase slips in the phase tracking loop. After a phase slip, the demodulator phase coherence will be lost, until known symbols (with known phases) are received (or until another fortunate phase slip causes the demodulator to regain its phase coherence). Coherent receivers, based on phase and amplitude tracking loops, are therefore inappropriate for most turbo-coded wireless systems, unless very slow fading is observed at all times. Non-coherent or pseudo-coherent demodulation techniques, such as differential detection and pilot symbol-assisted detection, are hence more suitable for usage in conjunction with turbo codes, since they do not require the presence of an external amplitude or phase tracking loop at the receiver side.

Note finally that equation (37) requires the knowledge of the noise power density $N_0/2$. Nevertheless, it is much easier, even at very low SNR's, to estimate the noise power than it is to estimate the channel gain. In fact, the noise power can usually be considered as constant or very slowly varying. For that reason, the noise power is assumed to be known in the remaining of this thesis.

3.2.2 Pilot Symbol-Assisted Modulation and MMSE Channel Estimation

To avoid the phase and amplitude tracking loops required by coherent detection, it is possible to use Pilot Symbol-Assisted Modulation (PSAM) to perform a pseudo-coherent detection [5][6]. With conventional non-iterative processing, the pilot symbols, whether they are time-multiplexed or code-multiplexed (in CDMA systems), are used to obtain channel estimates $\tilde{c}$ with conventional Minimum Mean Square Error (MMSE) estimation techniques. If they are time-multiplexed, the pilot symbols must be inserted to sample the fading process at a frequency larger than or equal to the Nyquist rate of the fading process, in order to provide sufficient statistics for that process.

Figure 7 – Pseudo-coherent detection using PSAM.

The CSI is estimated by extracting and interpolating the received pilot symbols $\mathbf{r}$, as shown in Figure 7. If the channel statistical characteristics are known, the interpolation filter can be designed to be optimum in the MMSE sense. This filter is obtained from the normal equations

$$\mathbf{R}_r \mathbf{h} = \mathbf{w}, \quad (38)$$

where $\mathbf{R}_r$ is the autocorrelation matrix of the sequence $\mathbf{r}$, and $\mathbf{w}$ is the cross correlation of $\mathbf{c}$ and $\mathbf{r}$. An example of a channel interpolation filter minimizing the MSE is presented in Annex B. In a practical system, the length of the observation $\mathbf{r}$, and hence of the interpolation filter $\mathbf{h}$, is kept small in order to limit the filtering lag. Furthermore, the interpolation filter can be fixed to a be low-pass filter with a cutoff frequency equal to the expected maximum fading rate, to avoid the need for the dynamic estimation of the channel autocorrelation [5].

Using the estimated channel samples obtained with this technique, in place of the true channel samples in equation (37), yields the likelihood ratio [31]

$$\Lambda(x_k | \mathbf{y}) \approx \exp\left(\frac{4\text{Re}(y_k \tilde{c}_k^*)}{N_0}\right) \cdot \Lambda(x_k), \quad (39)$$

which is of the form of equation (37) with c_k replaced by the estimated CSI. This technique usually exhibits a degradation lower than that of differential detection with respect to perfect coherent detection, at the cost of an increased overhead [31].

3.2.3 Differential Detection

Differential encoding provides a very simple way to avoid the need for coherent detection. In this case, the transmitted symbols d_k are recursively computed as $d_k = x_k d_{k-1}$, with $d_0 = 1$ or -1 . The conventional one-symbol differential detector structure is illustrated in Figure 8, where z^{-1} represents a delay of one sample.

Figure 8 – Conventional one-symbol differential detection.

If the channel state does not change significantly between consecutive samples, that is if $c_k \approx c_{k-1}$, the output of a conventional differential detector at time k can be expressed as

$$\Delta_k = y_k y_{k-1}^* \approx x_k |c_k|^2 + \underbrace{c_k (d_k n_{k-1} + d_{k-1} n_k) + n_k n_{k-1}}_{\text{noise terms}}. \quad (40)$$

The first term of the right-hand side of equation (40) is the desired term, while the second and third terms are noise components. If we assume that these noise components are independent and approximately Gaussian distributed, with zero means and variances $|c_k|^2 N_0$ and $(N_0/2)^2$ respectively, then the resulting noise term n_k' is also Gaussian, with zero mean and variance $N' = |c_k|^2 N_0 + (N_0/2)^2$. The pdf of Δ_k , given that x_k was

transmitted, is thus

$$p(\Delta_k | x_k) \approx \frac{1}{\sqrt{2\pi N'}} \exp\left(-\frac{|\Delta_k - x_k |c_k|^2|^2}{2N'}\right) \quad (41)$$

and its associated likelihood ratio is

$$\Lambda(\Delta_k | x_k) \approx \exp\left(\frac{4\operatorname{Re}(\Delta_k)|c_k|^2}{2N'}\right) \quad (42)$$

The likelihood ratio of the *a posteriori* probability is then

$$\Lambda(x_k | y) = \Lambda(x_k | \Delta_k) = \Lambda(\Delta_k | x_k) \Lambda(x_k) \approx \exp\left(\frac{4\operatorname{Re}(y_k y_{k-1}^*)|c_k|^2}{2N'}\right) \cdot \Lambda(x_k). \quad (43)$$

If E_s/N_0 is sufficiently large, then the third term of equation (40) can be neglected and N' becomes equal to $|c_k|^2 N_0$. In this case, equation (43) can be approximated by [3]

$$\Lambda(x_k | y) \approx \exp\left(\frac{4\operatorname{Re}(y_k y_{k-1}^*)}{2N_0}\right) \cdot \Lambda(x_k). \quad (44)$$

This technique is very simple and does not require knowledge of the CSI. The price to pay is a degradation in the vicinity of 3 dB for the uncoded or convolutionally coded case, and around 4 dB for the turbo-coded case [31]. The 3 dB (or more) penalty can be intuitively derived by comparing equations (37) and (44), since the noise variance is roughly twice as large in the latter.

3.3 Linear-Predictive MLSE/MAP Detection

The *a posteriori* probabilities of the transmitted symbols can also be computed using linear prediction-based detection. The linear prediction-based detection technique was first proposed in [7] and [32]. This technique was proposed in [7] for the sequence estimation of constant envelope signals transmitted over Rayleigh flat fading channels. It can be extended in a straightforward manner to the case of band limited PSK signals sampled once per symbol (without ISI) [8]. In this section, linear-predictive MLSE and MAP detections are derived for coherent and differentially encoded PSK signals.

3.3.1 Linear-Predictive MLSE

The maximum likelihood sequence is the hypothesized transmitted sequence $\hat{\mathbf{x}}$ maximizing the likelihood of the received sequence $p(\mathbf{y} | \hat{\mathbf{x}})$. This probability can be computed in the following recursive manner [7]

$$p(\mathbf{y}_1^k | \hat{\mathbf{x}}) = p(\mathbf{y}_1^{k-1} | \hat{\mathbf{x}}) p(y_k | \mathbf{y}_1^{k-1}, \hat{\mathbf{x}}) = \prod_{j=1}^k p(y_j | \mathbf{y}_1^{j-1}, \hat{\mathbf{x}}), \quad (45)$$

where $\mathbf{y}_1^j$ denotes the symbol sub-sequence $(y_1, y_2, \dots, y_{j-1}, y_j)$ of the total sequence $\mathbf{y}$. The linear-predictive approach provides a way to compute the likelihood probabilities $p(y_k | \mathbf{y}_1^{k-1}, \hat{\mathbf{x}})$ for a Rayleigh flat fading channel.

First, when the transmitted symbols have constant amplitude, it is possible to perform the following linear transformation to the received signal $\mathbf{y}$ on a conditional basis [7]:

$$y'_k(\hat{x}_k) = \hat{x}_k^* y_k, \quad (46)$$

where $\hat{x}_k$ is the hypothesized transmitted symbol and $y'_k(\hat{x}_k)$ is the transformed signal sample corresponding to this hypothesis. This transformation is sometimes referred to as *remodulation*. Since the transmitted symbols have unit amplitude, this linear transformation has a Jacobian of magnitude one. Therefore, the likelihood of the received sequence $\mathbf{y}$, given the hypothesized transmitted sequence $\hat{\mathbf{x}}$, is equivalent to the likelihood of the transformed sequence $(y'_1(\hat{x}_1), y'_2(\hat{x}_2), \dots)$, denoted $\mathbf{y}'(\hat{\mathbf{x}})$, conditioned on the same hypothesis [7],

$$p(\mathbf{y} | \hat{\mathbf{x}}) = p(\mathbf{y}'(\hat{\mathbf{x}}) | \hat{\mathbf{x}}). \quad (47)$$

The transformed sequence $\mathbf{y}'(\hat{\mathbf{x}})$ is thus statistically equivalent to $\mathbf{y}$ when conditioned on $\hat{\mathbf{x}}$ and, consequently, is a set of sufficient statistics for the MLSE of the sequence $\mathbf{x}$. Since $\mathbf{y}$ is a sequence of jointly Gaussian random variables with zero mean, $\mathbf{y}'(\hat{\mathbf{x}})$ is also of the same nature. If the hypothesized sequence $\hat{\mathbf{x}}$ corresponds to the transmitted sequence $\mathbf{x}$, the covariance matrix of $\mathbf{y}'(\hat{\mathbf{x}})$ is

$$\mathbf{R}_{\mathbf{y}'} = E[\mathbf{y}'(\hat{\mathbf{x}})\mathbf{y}'(\hat{\mathbf{x}})^H | \hat{\mathbf{x}}] \Big|_{\hat{\mathbf{x}}=\mathbf{x}} = \mathbf{R}_c + N_0 \mathbf{I}, \quad (48)$$

where $\mathbf{R}_c$ is the autocovariance matrix of the fading process $\mathbf{c}$, and $\mathbf{I}$ is the identity matrix. Note that the covariance matrix of the transformed sequence $\mathbf{y}'(\mathbf{x})$ is independent of the symbol sequence $\mathbf{x}$.

Furthermore, it is possible to produce a sequence of independent random variables

$\mathbf{y}'(\hat{\mathbf{x}})$ using another linear transformation

$$\mathbf{y}''(\hat{\mathbf{x}}) = \mathbf{P}\mathbf{y}'(\hat{\mathbf{x}}), \quad (49)$$

and choosing the transformation matrix $\mathbf{P}$ such that the transformed sequence has a diagonal covariance matrix, expressed as

$$\mathbf{R}_{\mathbf{y}''} = \mathbf{P}\mathbf{R}_{\mathbf{y}'}\mathbf{P}^H. \quad (50)$$

Here, $(.)^H$ denotes the complex conjugate transpose of a matrix or a vector. This transformation corresponds to the orthogonalization of the sequence $\mathbf{y}'(\hat{\mathbf{x}})$. This can be achieved using the innovation approach. In this case, the transformation matrix $\mathbf{P}$ is lower triangular and has the following form

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -p_1^{[1]} & 1 & 0 & \dots & 0 \\ -p_2^{[2]} & -p_1^{[2]} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ -p_{N-1}^{[N-1]} & -p_{N-2}^{[N-1]} & -p_{N-3}^{[N-1]} & \dots & 1 \end{bmatrix}, \quad (51)$$

where $p_i^{[j]}$ is the i^{th} coefficient of a j^{th} -order linear predictor for the sequence $\mathbf{y}'(\hat{\mathbf{x}})$. The resulting sequence $\mathbf{y}''(\hat{\mathbf{x}})$ therefore represents the prediction error sequence, or the innovation sequence of $\mathbf{y}'(\hat{\mathbf{x}})$, which is statistically equivalent to $\mathbf{y}'(\hat{\mathbf{x}})$ [34]. Consequently, the likelihood of the received sequence $\mathbf{y}$ conditioned on the hypothesis $\hat{\mathbf{x}}$ can be expressed as

$$p(\mathbf{y} | \hat{\mathbf{x}}) = p(\mathbf{y}'(\hat{\mathbf{x}}) | \hat{\mathbf{x}}) = p(\mathbf{y}''(\hat{\mathbf{x}}) | \hat{\mathbf{x}}) = \prod_{k=1}^N p(y_k''(\hat{\mathbf{x}}) | \hat{\mathbf{x}}), \quad (52)$$

since the elements of the sequence $\mathbf{y}''(\hat{\mathbf{x}})$ are independent. In this case, the probability $p(y_k | \mathbf{y}_1^{k-1}, \hat{\mathbf{x}})$ simply becomes

$$p(y_k | \mathbf{y}_1^{k-1}, \hat{\mathbf{x}}) = \frac{p(y_k'' | \hat{\mathbf{x}})}{p(\mathbf{y}_1^{k-1} | \hat{\mathbf{x}})} = p(y_k''(\hat{\mathbf{x}}) | \hat{\mathbf{x}}). \quad (53)$$

Since $\mathbf{y}'(\hat{\mathbf{x}})$ is a sequence of Gaussian random variables with zero mean, after the linear transformation $\mathbf{P}$, the pdf $p(\mathbf{y}_k''(\hat{\mathbf{x}}) | \hat{\mathbf{x}})$ is also Gaussian with zero mean. The variance of $y_k''(\hat{\mathbf{x}})$ is the expected squared prediction error for the $(k-1)^{\text{th}}$ -order predictor, $\sigma_p^2[k-1]$.

Thus, equation (53) yields

$$\begin{aligned} p(y_k | \mathbf{y}_1^{k-1}, \hat{\mathbf{x}}) &= \frac{1}{\sqrt{2\pi\sigma_p^2[k-1]}} \exp\left(-\frac{|y_k''(\hat{\mathbf{x}})|^2}{2\sigma_p^2[k-1]}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma_p^2[k-1]}} \exp\left(-\frac{\left|y_k'(\hat{\mathbf{x}}) - \sum_{j=1}^{k-1} p_j^{[k-1]} y_{k-j}'(\hat{\mathbf{x}})\right|^2}{2\sigma_p^2[k-1]}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma_p^2[k-1]}} \exp\left(-\frac{\left|y_k \hat{\mathbf{x}}_k^* - \sum_{j=1}^{k-1} p_j^{[k-1]} y_{k-j} \hat{\mathbf{x}}_{k-j}^*\right|^2}{2\sigma_p^2[k-1]}\right) \end{aligned} \quad (54)$$

Usually, we can assume that the channel has a finite memory of length M , that is [34]

$$E[y_k | \mathbf{y}_1^{k-1}] = E[y_k | \mathbf{y}_{k-M}^{k-1}]. \quad (55)$$

In this case, the linear predictors of order M and greater have the same coefficients. Furthermore, if a preamble of M known symbols precedes the sequence $\mathbf{x}$, then the linear predictors of order less than M can be ignored. Thus

$$p(y_k | \hat{\mathbf{x}}, \mathbf{y}_1^{k-1}) = p(y_k | \hat{\mathbf{x}}_{k-M}^k, \mathbf{y}_{k-M}^{k-1}) = \frac{1}{\sqrt{2\pi\sigma_p^2[M]}} \exp - \left(\frac{\left| y_k - \hat{x}_k \sum_{j=1}^M P_j^{[M]} y_{k-j} \hat{x}_{k-j}^* \right|^2}{2\sigma_p^2[M]} \right). \quad (56)$$

It is then possible to use the Viterbi algorithm to efficiently compute the maximum likelihood estimated sequence.

However, we are not interested in this thesis by the ML hard decisions produced by the Viterbi algorithm, but rather in the *a posteriori* probability $p(x_k = q | \mathbf{y}, \Psi)$ [8]. The conditioning on Ψ represents the knowledge of the modulation structure and the channel statistics. This notation is used to differentiate the *a posteriori* probability of the transmitted symbols computed by the demodulator, $p(x_k = q | \mathbf{y}, \Psi)$, from the *a posteriori* probability computed by a MAP filter for the code C , $p(x_k = q | \mathbf{y}, C)$.

3.3.2 Linear-Predictive Symbol-by-Symbol MAP Estimation

If we assume that the channel has a finite memory of length $M+1$, then we can represent the transmitted sequence in a trellis, where a state s_k is defined by the last M transmitted symbols $\mathbf{x}_{k-M}^{k-1}$. This representation can also be applied to differentially

encoded PSK signals, as proposed by Hoeher & Lodge in [9]. Note that in the case of a differentially encoded signal, the state s_k is defined by the last M differentially encoded transmitted symbols $\mathbf{d}_{k-M}^{k-1}$. In both cases, for a Q -ary modulation scheme, the resulting trellis has Q^M states.

In order to compute the *a posteriori* probabilities, a reasoning similar to that employed in Section 2.2 and Annex A for the MAP decoding of convolutional codes can be used. However, we cannot assume the independence of the received symbols $\mathbf{y}$. Therefore, the APP of the transmitted symbol x_k can be expressed as

$$p(x_k = q | \mathbf{y}, \Psi) = \sum_{(m', m): x_k = q} p(s_{k-1} = m', s_k = m | \mathbf{y}), \quad (57)$$

and thus

$$p(x_k = q | \mathbf{y}, \Psi) = \frac{\sum_{(m', m): x_k = q} p(s_{k-1} = m', s_k = m, \mathbf{y})}{p(\mathbf{y})} = \frac{\sum_{(m', m): x_k = q} p(s_{k-1} = m', s_k = m, \mathbf{y})}{\sum_{(m', m)} p(s_{k-1} = m', s_k = m, \mathbf{y})}. \quad (58)$$

Furthermore, we can decompose $p(s_{k-1} = m', s_k = m, \mathbf{y})$ as

$$\begin{aligned} p(s_{k-1} = m', s_k = m, \mathbf{y}_1^N) &= p(s_{k-1} = m', s_k = m, \mathbf{y}_1^k) p(\mathbf{y}_{k+1}^N | s_{k-1} = m', s_k = m, \mathbf{y}_1^k) \\ &= p(s_{k-1} = m', \mathbf{y}_1^{k-1}) p(s_k = m, \mathbf{y}_k | s_{k-1} = m', \mathbf{y}_1^{k-1}) p(\mathbf{y}_{k+1}^N | s_k = m, \mathbf{y}_1^k) \\ &= \alpha_{k-1}(m') \cdot \gamma_k(m', m) \cdot \beta_k(m), \end{aligned} \quad (59)$$

using a similar notation to that of Annex A. Again, $\alpha_k(m)$ and $\beta_k(m)$ can be computed recursively since

$$\begin{aligned}
\alpha_k(m) &= \sum_{m'} p(s_{k-1} = m', s_k = m, y_1^k) \\
&= \sum_{m'} p(s_{k-1} = m', y_1^{k-1}) p(s_k = m, y_k | s_{k-1} = m', y_1^{k-1}) \\
&= \sum_{m'} \alpha_{k-1}(m') \cdot \gamma_k(m', m),
\end{aligned} \tag{60}$$

and

$$\begin{aligned}
\beta_k(m) &= \sum_{m''} p(s_{k+1} = m'', y_{k+1}^N | s_k = m, y_1^k) \\
&= \sum_{m''} p(s_{k+1} = m'', y_{k+1} | s_k = m, y_1^k) p(y_{k+2}^N | s_k = m, s_{k+1} = m'', y_1^{k+1}) \\
&= \sum_{m''} \gamma_{k+1}(m, m'') \cdot \beta_{k+1}(m'').
\end{aligned} \tag{61}$$

The computation of the state transition weight $\gamma_k(m', m)$ depends on the modulation scheme, whether it is coherent or differentially encoded PSK.

3.3.3 Coherent PSK with Pilot Symbols

In the coherent PSK case, the state transition weight can be expressed as

$$\begin{aligned}
\gamma_k(m', m) &= p(s_k = m, y_k | s_{k-1} = m', y_1^{k-1}) \\
&= p(\hat{x}_k, y_k | \hat{x}_{k-M}^{k-1}, y_1^{k-1}) \\
&= p(y_k | \hat{x}_{k-M}^k, y_1^{k-1}) p(\hat{x}_k) \\
&= p(y_k | \hat{x}_{k-M}^k, y_{k-M}^{k-1}) p(\hat{x}_k),
\end{aligned} \tag{62}$$

where $p(y_k | \hat{x}_{k-M}^k, y_{k-M}^{k-1})$ can be computed as proposed by equation (56). If pilot symbols are used, as proposed in [8] by Gertsman & Lodge, the *a priori* probability $p(\hat{x}_k)$ for the pilot symbols is set to one for the pilot symbol and 0 for all the other possible symbols. This is equivalent to limit the trellis paths to only the paths including the pilot symbols.

As it was shown in [8], pilot symbols are improving significantly the performance of such a MAP demodulator in the presence of fading. Without pilot symbols, a deep fade is likely to result in a phase ambiguity, greatly reducing the performance of the MAP demodulator. The pilot symbols can eliminate these phase ambiguities. The use of pilot symbols for coherent PSK signals is therefore required if the proposed MAP demodulation structure is employed.

3.3.4 Differential PSK

If differential encoding is used, $\gamma_k(m', m)$ is expressed as

$$\begin{aligned}\gamma_k(m', m) &= p(s_k = m, y_k | s_{k-1} = m', y_1^{k-1}) \\ &= p(\hat{x}_k, y_k | \hat{\mathbf{d}}_{k-M}^{k-1}, y_1^{k-1}) \\ &= p(y_k | \hat{\mathbf{d}}_{k-M}^k, y_{k-M}^{k-1}) p(\hat{x}_k),\end{aligned}\quad (63)$$

where $\hat{\mathbf{d}}$ is the sequence of differentially encoded signal corresponding to the hypothesized sequence $\hat{\mathbf{x}}$ of transmitted symbols. Furthermore, equation (56) can be rewritten for differentially encoded symbols as

$$p(y_k | \hat{\mathbf{d}}_{k-M}^k, y_{k-M}^{k-1}) = \frac{1}{\sqrt{2\pi\sigma_p^2[M]}} \exp - \left(\frac{\left| y_k - \hat{\mathbf{d}}_k \sum_{j=1}^M p_j^{[M]} y_{k-j} \hat{\mathbf{d}}_{k-j}^* \right|^2}{2\sigma_p^2[M]} \right). \quad (64)$$

Still, in the computation of the APP $p(x_k = q | \mathbf{y}, \Psi)$, the summation over the joint probabilities $p(s_{k-1} = m', s_k = m, \mathbf{y})$ is taken over all the state transitions corresponding to

the symbol before differential encoding x_k . Note that the knowledge of the initial phase of the differentially encoded sequence $\hat{\mathbf{d}}_{k-M}^k$ is not required, since the computation of equation (56) is insensitive to the initial phase. Hence, an arbitrary initial phase can be used. Note that, for a Q -ary modulation scheme, there is Q differentially encoded sequences $\hat{\mathbf{d}}_{k-M}^k$ corresponding to the same sequence $\hat{\mathbf{x}}$. This observation can be used to reduce the number of state of the trellis to Q^{M-1} , as proposed in [35].

Note also that the use of differential encoding eliminates the risk for a phase ambiguity in the symbols before differential encoding $\mathbf{x}$, even if a “phase ambiguity” affects the differentially encoded symbols $\mathbf{d}$. Therefore, no pilot symbols are required for this MAP demodulation scheme when differential encoding is used.

3.4 Reception Diversity

When reception diversity is performed, whether it is in the form of antenna array or a RAKE receiver, the demodulation techniques presented in this chapter can still be applied. If the vector of received samples from the different diversity branches for a D^{th} order diversity is expressed as $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_D)$, where $\mathbf{y}_i$ is the vector of received samples from the i^{th} branch, then the likelihood probability of the received sequence, conditioned on the transmitted sequence $\mathbf{x}$, can be expressed as

$$\begin{aligned}
 p(\mathbf{y}|\mathbf{x}) &= p(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_D|\mathbf{x}) \\
 &= p(\mathbf{y}_1|\mathbf{x}) \cdot p(\mathbf{y}_2|\mathbf{x}) \cdot \dots \cdot p(\mathbf{y}_D|\mathbf{x}) \\
 &= \prod_{i=1}^D p(\mathbf{y}_i|\mathbf{x}),
 \end{aligned} \tag{65}$$

assuming independent samples from each diversity branch.

3.4.1 Conventional Demodulation Schemes

For the conventional (memoryless) demodulation schemes of Section 3.3, equation (65) means that the likelihood ratio for a symbol y_k (or Δ_k for a differentially encoded signal) can be written as

$$\Lambda(y_k | x_k) = \prod_{i=1}^D \Lambda(y_{i,k} | x_k), \quad (66)$$

where $y_{i,k}$ is the k^{th} received sample from the i^{th} diversity branch. Note that for the case of coherent detection with known CSI, the resulting combining scheme corresponds to the so-called Maximum Ratio Combining (MRC) [29]. For the pseudo-coherent detection scheme based on PSAM, it is thus an approximation of MRC. In the case of differential detection, the combining method suggested by (66) corresponds to adding the outputs of D one-symbol differential detectors (one for each branch).

3.4.2 Linear-Predictive Demodulation

For the linear-predictive demodulators of Section 3.3, equation (65) implies that

$$p(y_k | \hat{\mathbf{x}}_{k-M}^k, \mathbf{y}_{k-M}^{k-1}) = \prod_{i=1}^D p(y_{i,k} | \hat{\mathbf{x}}_{k-M}^k, \mathbf{y}_{i,k-M}^{k-1}), \quad (67)$$

where $\mathbf{y}_{i,k-M}^{k-1}$ is the sub-sequence of received samples from the i^{th} diversity branch, from time index $k-M$ to $k-1$. Using this substitution, the symbol-by-symbol MAP demodulation

can be carried out for an D^{th} order reception diversity scheme.

3.5 Summary

In this chapter, several soft-output demodulation schemes have been presented. In order to avoid the need for coherent detection, which is difficult in fading conditions, sub-optimal receivers for PSAM and differentially encoded signals have been presented. Furthermore, MAP linear-predictive receivers for PSAM and differentially encoded signals have been proposed. It can be shown that, if the channel complex gain is unknown, the optimal demodulator is this linear-predictive receiver [34].

For coded signals, the ideal receiver should perform joint demodulation and decoding, and in the next chapter, a novel joint iterative demodulation and decoding structure is derived for turbo-coded signals on Rayleigh flat fading channels. This new structure is based on the linear-predictive MAP demodulator for PSAM and differentially encoded signals described in Section 3.3.

Chapter 4 - Iterative Demodulation and Decoding

Ideally, the optimum receiver for a coded system should perform joint demodulation and decoding, to minimize the overall probability of symbol error. This joint demodulation and decoding process should compute the overall *a posteriori* probability $p(u_k = q | y, C, \Psi)$ for each information symbols u_k , where the conditioning on C and Ψ denotes respectively the knowledge of the code structure and the knowledge of the modulation and channel structure. Such a receiver is illustrated in Figure 9. If the computation of the overall *a posteriori* probability is realizable, then the receiver can make the optimal MAP decision regarding the information symbol u_k .

Figure 9 – Optimal receiver structure.

Because of complexity issues, demodulation and decoding are usually separated in two different tasks in the conventional receiver structure for coded systems, as illustrated in Figure 10. When the memory of the modulation and channel Ψ is not used in the demodulation process, the demodulator can compute the *a posteriori* probability for a symbol x_k based only on the corresponding observation y_k . In this case, the demodulator

has no use for the structure imposed on the transmitted sequence $\mathbf{x}$ by the channel code C , assuming that every transmitted symbols x_k have uniform *a priori* distributions. Consequently, performing disjoint demodulation and decoding does not usually cause an important loss of performance when there is no memory used in the demodulation process.

Figure 10 – Conventional soft-in/soft-out receiver structure.

If the modulation and channel memory is used in the demodulation process, then ideally the demodulator should take into account the structure imposed on the transmitted sequence $\mathbf{x}$ by the channel code C . Therefore, joint demodulation and decoding should be performed to minimize the overall probability of error, as mentioned previously. However, the presence of interleaving between the modulator and the channel encoder usually makes this joint processing highly complex. Note that this problem is similar to that of decoding parallel concatenated convolutional (turbo) codes with interleaving. This suggests that an iterative structure is also an efficient sub-optimal technique to solve the

problem of joint demodulation and decoding with interleaving.

In the conventional receiver structure illustrated in Figure 10, the information travels only in a single direction: from the demodulator to the decoder. If there is memory in the modulation and/or the channel, and if the demodulator utilizes this memory, the additional information available at the output of the channel decoder about the likelihood of a transmitted sequence can potentially improve the detection accuracy. An iterative demodulation and decoding structure would allow the demodulation to take into account the likelihood of a transmitted sequence computed by using the code structure, and thus, in some way, take into account the structure imposed by the channel code C .

In this chapter, the use of iterative demodulation and decoding as a sub-optimal implementation of a joint demodulation and decoding process is explored. While the general structures derived herein are applicable to any PAM or CPM signals transmitted over frequency-selective channels, we will focus our attention on PSK signals on flat fading channels. Therefore, the linear-predictive demodulators presented in Section 3.3 will be used in the proposed iterative demodulation and decoding structures. Note that the demodulators of Section 3.2 are not suitable for iterative processing, since these demodulators do not take advantage of the memory in the modulation and the channel. This point will be discussed later in the chapter.

The chapter is divided as follows. First, the basic structure for iterative demodulation and decoding is presented for convolutionally encoded PSK systems, since the structure is simple and easy to understand in this case. The following section is then devoted to the

extension of the iterative demodulation and decoding principle to turbo-coded PSK systems. The motivation for this extension is explained, and the new structure is derived and discussed.

4.1 Basic Structure for Convolutional Codes

The idea of iterative demodulation and decoding was initially developed for convolutionally encoded systems [8][9][36]-[44]. Several authors derived iterative demodulation and decoding structures for frequency-selective channels [37][38][42][43]. Other authors applied a similar reasoning to derive iterative demodulation and decoding structures for differentially encoded signals on AWGN channels, since the modulation memory can be used in the demodulation process [39][40][44].

In this thesis, we are concerned with time-varying flat fading channels. For this situation, two iterative demodulation and decoding schemes for convolutionally encoded systems have been proposed. Gertsman & Lodge [8][36] derive a structure based on a linear-predictive demodulator for coherent PSK signals with pilot symbols, as presented in Section 3.3.3. Another iterative demodulation and decoding scheme for flat fading channels, proposed by Hoeher & Lodge [9][41], is based on a linear-predictive demodulator for differentially encoded PSK (DPSK) signals, as presented in Section 3.3.4. In the current section, these structures are reviewed for convolutionally encoded BPSK or QPSK signals. This serves the purpose of establishing the notation and signal framework. In the next section, based on the same demodulators, the iterative demodulation and decoding structure is extended to the case of turbo-coded BPSK/QPSK

signals.

With a non-iterative receiver, the soft-output demodulator computes the ratio of the *a posteriori* probabilities $\Lambda(x_k | y, \Psi)$ for the transmitted symbols x_k based on the knowledge of the modulation/channel structure Ψ . This computation can be carried out using the soft-output demodulators presented in Chapter 3, assuming that all the transmitted symbols x_k are independent and equally likely. Obviously, this hypothesis is not valid because the symbols are produced by a convolutional encoder. However, since the output of the convolutional encoder is interleaved, the transmitted symbols in the same vicinity can be assumed independent if sufficient interleaving is applied. In general, this condition is sufficient to assume the independence of the transmitted symbols.

After deinterleaving, the decoder assumes that the soft-outputs produced by the demodulator are independent, in order justify the application of the BCJR algorithm. Here again, this hypothesis is not true, but with sufficient interleaving, it can be assumed that the soft-output demodulator decisions corresponding to coded symbols in the same vicinity are independent. Therefore, it is assumed that

$$\Lambda(x_k | y, \Psi) \approx \Lambda(x_k | y_k). \quad (68)$$

The soft-output of the demodulator is defined as the ratio of the likelihood probabilities

$$\Lambda_e^{[\Psi]}(x_k) = \Lambda(y_k | x_k) = \frac{\Lambda(x_k | y_k)}{\Lambda(x_k)} \approx \frac{\Lambda(x_k | y, \Psi)}{\Lambda(x_k)}, \quad (69)$$

which correspond to the “extrinsic” information added by the demodulation process. Equivalently, in the log-likelihood domain

$$L_e^{[\Psi]}(x_k) = L(y_k | x_k) \approx L(x_k | y, \Psi) - L(x_k). \quad (70)$$

Note that if equally likely transmitted symbols are assumed, $\Lambda(x_k) = 1$ or, equivalently, $L(x_k) = 0$. The value computed using equation (69) or equation (70) is used by the MAP decoder for the computation of the state transition metric $\gamma(m', m)$ for the convolutional code (equation (144)). It is then possible to compute the *a posteriori* probability for the information symbol u_k , $p(u_k = q | y, C)$.

If the receiver is non-iterative, the processing stops after the computation of $p(u_k = q | y, C)$. On the other hand, iterative demodulation and decoding schemes compute the *a posteriori* probabilities for the coded symbols x_k , $p(x_k = q | y, C)$, and hence the MAP decoder of Figure 10 is replaced by a MAP filter (Section 2.2.2). Furthermore, it is possible to isolate the information added by the soft-in/soft-out convolutional decoder by using the concept of extrinsic information, as presented in Section 2.3. The extrinsic information $\Lambda_e^{[C]}(x_k)$ added by the decoder regarding the coded symbols can thus be defined as the ratio of the information at the output of the MAP filter to the information at the input of the MAP filter, that is

$$\Lambda_e^{[C]}(x_k) = \frac{\Lambda(x_k | y, C)}{\Lambda_e^{[\Psi]}(x_k)}, \quad (71)$$

or equivalently in the log-likelihood domain

$$L_e^{[C]}(x_k) = L(x_k | y, C) - L_e^{[\Psi]}(x_k), \quad (72)$$

where $L_e^{[C]}(x_k)$ is the log-likelihood of the extrinsic information added by the MAP filter

for code C about the coded symbol x_k .

This improved information about the coded bits can be very helpful to improve the detection accuracy. It is therefore fed back to the demodulator, where it is used as refined *a priori* information by the linear-predictive symbol-by-symbol MAP demodulator. For the linear-predictive receiver based on coherent PSK and pilot symbols, this refined *a priori* information is used in the computation of the state transition metric $\gamma(m', m)$ defined by equation (62). When the linear-predictive demodulator based on differentially encoded PSK is used, the refined *a priori* information about the transmitted symbol x_k is taken into account in the computation of the state transition metric $\gamma(m', m)$ defined by equation (63).

Then again, the demodulator can compute an improved APP likelihood ratio $\Lambda(x_k | y, \Psi)$ for the transmitted symbol x_k , based on the refined *a priori* information. At the output of the demodulator, if it is assumed that the transmitted symbols are equally likely *a priori*, equation (69) can be rewritten as [44]

$$\Lambda_e^{[\Psi]}(x_k) \approx \frac{\Lambda(x_k | y, \Psi)}{\Lambda_e^{[C]}(x_k)}, \quad (73)$$

or using the log-likelihood

$$L_e^{[\Psi]}(x_k) \approx L(x_k | y, \Psi) - L_e^{[C]}(x_k). \quad (74)$$

This information can then be re-processed by the convolutional decoder.

Note that $\Lambda_e^{[\Psi]}(x_k)$ can be interpreted as the extrinsic information produced by the

demodulator. In fact, the problem of iterative demodulation and decoding is related to the problem of iterative decoding of serially concatenated codes with interleaving [45]. Therefore, it is possible to use the reasoning applied for the iterative decoding of serially concatenated codes to derive an iterative demodulation and decoding structure, as proposed in [46]. This interpretation is interesting, and it will be discussed further in the next chapter.

The iterative demodulation and decoding scheme presented in this section is summarized in Figure 11, where Π and Π^{-1} represent the channel interleaver and deinterleaver respectively. Note that the MAP channel decoder of Figure 10 is replaced by a MAP filter, computing the APP's of the coded bits instead of the APP's of the information bits. The refined *a priori* information produced by the MAP filter is re-interleaved and fed back to the demodulator to improve its performance for the next iteration. The improved demodulation decisions are then deinterleaved and processed again by the MAP filter. This iterative processing can be stopped when the demodulation process reaches a point where it does not benefit significantly from the refined *a priori* information. The APP's of the information bits $\mathbf{u}$ are then computed using a MAP decoder instead of a MAP filter.

Note that as for turbo decoding, the iterative processing can be stopped after a fixed number of iterations that is large enough to allow convergence of the iterative algorithm. A convergence test could also be applied to stop the iterative processing after the minimum number of iterations required for convergence, and thus to minimize the

computational complexity of the iterative scheme.

Figure 11 – Iterative demodulation and decoding structure for convolutional codes.

It is also worth noting that, as mentioned previously, the soft-output demodulators presented in Section 3.2 are not suitable for iterative demodulation and decoding, because there is no memory used in the demodulation process. This can be easily demonstrated mathematically, since the output of the demodulator can be expressed as

$$\Lambda(x_k | y_k, \Psi) = \Lambda(y_k | x_k) \cdot \Lambda(x_k), \quad (75)$$

and the computation of $\Lambda(y_k | x_k)$ does not depend on the *a priori* probability $\Lambda(x_k)$ (see equations (37) and (39)). In these cases, the value of the extrinsic information $\Lambda_e^{[\Psi]}(x_k)$ at the output of the demodulator, computed using equation (73), is not a function of the *a*

priori probabilities. Therefore, the accuracy of these demodulation techniques is not improved by using the refined *a priori* information $\Lambda_e^{[C]}(x_k)$ produced by the decoder, and hence, they are not suitable for iterative demodulation and decoding. These non-iterative demodulation techniques are used in this thesis as benchmark references to draw comparisons with the proposed iterative demodulation and decoding schemes.

4.2 Extension to Turbo Codes

4.2.1 Motivations

Previous work on the iterative demodulation and decoding of convolutionally encoded systems on flat fading channels showed the potential of this technique. The results obtained in [8] and [9] demonstrated that the performance of iterative demodulation and decoding is close to that of perfect coherent detection (with known CSI), even for differentially encoded signals. Therefore, the iterative demodulation and decoding method appears to be a very promising technique.

Nevertheless, convolutional codes are likely to be replaced by more powerful iteratively decodable channel codes, such as turbo codes [11]. It appears thus interesting to apply the principle of iterative demodulation and decoding to turbo-coded mobile communication systems. Furthermore, iteratively decodable codes are already using an iterative decoding structure, and hence they are good candidates for iterative demodulation schemes, in particular because they are usually designed using component codes with relatively low-complexity soft-in/soft-out decoding algorithms.

On the other hand, convolutional codes are usually designed assuming that the hard-output Viterbi algorithm will be used at the decoder, and thus long constraint length (e.g. $K = 9$) codes are chosen to achieve a good error rate performance. Such codes will require much more complexity when they are decoded using a soft-in/soft-out algorithm. For instance, a typical $K = 9$ convolutional code has 256 states, while the constituent codes in a typical turbo code have 8 or 16 states.

Therefore, when used in an iterative demodulation and decoding scheme, a single convolutional code is likely to require more computational complexity than most iteratively decodable concatenated codes. In other words, although the convolutional codes used in non-iterative demodulation schemes usually require a relatively low computational complexity with respect to turbo codes, this situation might be reversed when iterative demodulation is performed, since the low-complexity Viterbi algorithm has to be replaced by a more complex soft-in/soft-out algorithm. For a convolutional code with a large constraint length, this substitution of the decoding algorithm already means an important increase in complexity. Moreover, since iterative demodulation and decoding schemes are requiring the convolutional code to be decoded multiple times, the complexity increase is even more important.

Iteratively decodable codes, such as turbo codes, are already requiring low complexity soft-in/soft-out algorithms for the iterative decoding of the component codes. Thus, using an iterative demodulation and decoding scheme with a turbo code often imply no important increase of the decoding complexity. For that reason, turbo codes

appear to be well suited to be used in conjunction with an iterative demodulation and decoding scheme.

Based on the principles presented in Section 4.1 for the iterative demodulation and decoding for convolutionally encoded PSK signals, it is possible to extend the same concept for arbitrary channel codes. In the next section, the principles of iterative demodulation and decoding are thus extended to the turbo-coded case. Once again, the iterative demodulation scheme is derived for flat fading channels, using the soft-output linear-predictive demodulators described in Section 3.3.

Note that a decoding structure for turbo-coded DPSK signals in flat fading conditions including the demodulator in the iterative process has already been proposed by Marsland & Mathiopoulos [4]. However, the iterative demodulation and decoding scheme proposed herein is different, although related, and is developed using a different approach. The differences will be discussed further at the end of this chapter.

4.2.2 Iterative Demodulation and Decoding Algorithm

The easiest way to approach the problem of iterative demodulation and decoding of turbo-coded signals is to treat the turbo code as a single, yet concatenated, channel code. It is therefore desirable to replace the MAP filter for the convolutional code by an equivalent MAP filter for the turbo code. In a conventional turbo decoder, the APP's are computed for the systematic bits only. In this case, the estimated APP's ratio computed by the turbo decoder after one iteration is defined in Section 2.3 as (equation (17))

$$\begin{aligned}
\Lambda(x_{s,k} | y, C) &= \frac{p(x_{s,k} = +1 | y_k, C)}{p(x_{s,k} = -1 | y_k, C)} \\
&= \frac{p(y_k | x_{s,k} = +1)}{p(y_k | x_{s,k} = -1)} \cdot \frac{p(x_{s,k} = +1)}{p(x_{s,k} = -1)} \cdot \frac{e_{1,k}(x_{s,k} = +1)}{e_{1,k}(x_{s,k} = -1)} \cdot \frac{e_{2,k}(x_{s,k} = +1)}{e_{2,k}(x_{s,k} = -1)} \\
&= \Lambda_e^{[\Psi]}(x_{s,k}) \cdot \Lambda_e^{[C_1]}(x_{s,k}) \cdot \Lambda_e^{[C_2]}(x_{s,k}),
\end{aligned} \tag{76}$$

where $x_{s,k} \equiv u_k$, $e_{1,k}(x_{s,k})$ and $e_{2,k}(x_{s,k})$ are the extrinsic information about systematic bits $x_{s,k}$ produced by C_1 and C_2 respectively, and equally likely bits are assumed. C_1 and C_2 refer to the constituent codes of the turbo code, while C denotes the turbo code itself, as a concatenated code. Hence, the extrinsic information about the systematic bits produced after one iteration of the turbo code is simply

$$\Lambda_e^{[C]}(x_{s,k}) = \frac{\Lambda(x_{s,k} | y, C)}{\Lambda_e^{[\Psi]}(x_{s,k})} = \Lambda_e^{[C_1]}(x_{s,k}) \cdot \Lambda_e^{[C_2]}(x_{s,k}), \tag{77}$$

or, in the log-likelihood domain,

$$L_e^{[C]}(x_{s,k}) = L(x_{s,k} | y, C) - L_e^{[\Psi]}(x_{s,k}) = L_e^{[C_1]}(x_{s,k}) + L_e^{[C_2]}(x_{s,k}). \tag{78}$$

The extrinsic information added by one iteration of the turbo code regarding the systematic bits is thus simply the sum of the extrinsic information added by the two constituent codes for these bits. This soft-in/soft-out, non-iterative, turbo decoder structure is illustrated in Figure 12 for operation in the log-likelihood domain. Π_t and Π_t^{-1} represent respectively the turbo interleaver and the turbo deinterleaver. P^{-1} represents the “unpuncturing” operation, which consists in replacing the likelihood ratios of the punctured bits by a unit-value likelihood ratio (or a null log-likelihood ratio).

Figure 12 – Soft-in/Soft-out turbo decoder for one iteration.

Note that in an iteration of the turbo code, the extrinsic information about the systematic bits x_s obtained for the 2nd constituent code is computed using the output of the first constituent decoder. This extrinsic information is therefore based on the demodulator output $L_e^{(\Psi)}(x_s)$, and on the extrinsic information produced by the 1st constituent decoder about the systematic bits in this iteration, $L_e^{(C_1)}(x_s)$. Similarly, the

extrinsic information about the systematic bits produced by the 1st constituent decoder is based on the demodulator output $L_e^{[\Psi]}(\mathbf{x}_s)$, which is enhanced by the extrinsic information produced by the 2nd constituent decoder about the systematic bits in the previous iteration, $L_e^{[C_2]}(\mathbf{x}_s)$. This information flow is required to allow each constituent code of the turbo code to use the information added by the other constituent code, as it is for a standard turbo decoder. In fact, if the structure of Figure 12 is iterated, an iterative structure equivalent to the turbo decoding structure defined in Figure 5 is obtained.

Note that the structure of Figure 12 computes the APP's for the systematic bits only. In order to conceive a MAP filter corresponding to one iteration of the turbo code, the APP's for the parity bits are required, in addition to the APP's for the systematic bits. This can be achieved by replacing the constituent MAP decoders by MAP filters, which compute the APP's for both the systematic and the parity bits. These MAP filters can be used in the structure proposed in Figure 12. The resulting structure is illustrated in Figure 13. The APP's for the parity bits produced by the MAP filters can then be expressed as

$$\Lambda(x_{1,k} | \mathbf{y}, C_1) = \Lambda_e^{[\Psi]}(x_{1,k}) \cdot \Lambda_e^{[C_1]}(x_{1,k}) \quad (79)$$

and

$$\Lambda(x_{2,k} | \mathbf{y}, C_2) = \Lambda_e^{[\Psi]}(x_{2,k}) \cdot \Lambda_e^{[C_2]}(x_{2,k}), \quad (80)$$

where $\Lambda_e^{[C_i]}(x_{i,k})$ is the extrinsic information added by the MAP filter for the code C_i about the parity bit $x_{i,k}$.

Consequently, the extrinsic information about the parity bits produced by the equivalent MAP filter for the turbo code C is

$$\Lambda_e^{[C]}(x_{1,k}) = \Lambda_e^{[C_1]}(x_{1,k}) = \frac{\Lambda(x_{1,k} | y, C_1)}{\Lambda_e^{[\Psi]}(x_{1,k})} \quad (81)$$

and

$$\Lambda_e^{[C]}(x_{2,k}) = \Lambda_e^{[C_2]}(x_{2,k}) = \frac{\Lambda(x_{2,k} | y, C_2)}{\Lambda_e^{[\Psi]}(x_{2,k})}. \quad (82)$$

Using the log-likelihood ratios, we have

$$L_e^{[C]}(x_{1,k}) = L_e^{[C_1]}(x_{1,k}) = L(x_{1,k} | y, C_1) - L_e^{[\Psi]}(x_{1,k}), \quad (83)$$

and

$$L_e^{[C]}(x_{2,k}) = L_e^{[C_2]}(x_{2,k}) = L(x_{2,k} | y, C_2) - L_e^{[\Psi]}(x_{2,k}). \quad (84)$$

In summary, the extrinsic information produced by the MAP filter corresponding to one iteration of turbo decoding, as illustrated in Figure 13, depends on the nature of the bits. For the systematic bits, the extrinsic information is defined by equation (78), while for the parity bits produced by the constituent codes C_1 and C_2 , the extrinsic information is defined by equations (83) and (84) respectively. The vector of extrinsic information produced by the MAP filter corresponding to one iteration of the turbo decoder can thus be expressed as

$$L_e^{[C]}(\mathbf{x}) = \begin{pmatrix} L_e^{[C_1]}(\mathbf{x}_s) + L_e^{[C_2]}(\mathbf{x}_s) \\ L_e^{[C_1]}(\mathbf{x}_1) \\ L_e^{[C_2]}(\mathbf{x}_2) \end{pmatrix}, \quad (85)$$

if the vector of coded bits $\mathbf{x}$ is defined as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_s \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \quad (86)$$

where $\mathbf{x}_s$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the vectors corresponding respectively to the systematic bits, the parity bits from C_1 , and the parity bits from C_2 .

As it was proposed for the iterative demodulation and decoding of convolutionally encoded signals, the extrinsic information defined by equation (85) can be fed back to the demodulator to improve the performance of the demodulation process. The resulting iterative demodulation and decoding structure for turbo codes is illustrated in Figure 14. Note that this structure is closely related to that obtained previously for convolutionally encoded signals (Figure 11). However, in the proposed structure for turbo-coded signals, the single MAP filter used for the convolutional code is replaced by the equivalent MAP filter for one iteration of the turbo code, implemented as a pair of MAP filters (one for each constituent code).

Figure 14 – Iterative demodulation and decoding structure for turbo codes.

The linear-predictive soft-output demodulator produces a first value for the demodulation extrinsic information vector $L_e^{[\Psi]}(\mathbf{x})$ based on the knowledge of the modulation/channel structure Ψ . This information is transferred to the pair of MAP filters. The extrinsic information produced by the pair of MAP filters $L_e^{[C]}(\mathbf{x})$ is then fed back to the demodulator. This extrinsic information is used by the demodulator as a refined *a priori* information. The improved information $L_e^{[\Psi]}(\mathbf{x})$ is then passed again to the pair of MAP filters.

This iterative demodulation and decoding goes on until, hopefully, the estimated APP's for the systematic bits, defined in the log-likelihood domain as

$$L(\mathbf{u} | \mathbf{y}, C, \Psi) = L(\mathbf{x}_s | \mathbf{y}, C, \Psi) = L_e^{[\Psi]}(\mathbf{x}_s) + L_e^{[C_1]}(\mathbf{x}_s) + L_e^{[C_2]}(\mathbf{x}_s), \quad (87)$$

converge to the true APP's. Here again, a convergence test could be used to minimize the amount of iteration required.

4.3 Discussion

In this chapter, we have first derived the basic structure for the iterative demodulation and decoding of convolutionally encoded PSK signals. This structure can be extended for an arbitrary channel code with a soft-in/soft-out MAP decoding algorithm. In the remaining of the chapter, the extension has been performed for the case of turbo-coded PSK signals. The main difference between the resulting iterative demodulation and decoding receiver structure for turbo-coded and that previously derived for

convolutionally encoded systems is in the definition of the extrinsic information fed back to the demodulator by the decoder. For the turbo-coded case, this information has a non-uniform definition. For the systematic bits, extrinsic information from both constituent MAP filters is available and fed back to the demodulator. For the parity bits produced by the constituent codes C_1 and C_2 , the extrinsic information is obtained from the MAP filter of the corresponding code. Using all this extrinsic information, the demodulator output is improved after each iteration of the turbo decoder.

Note that this iteration schedule is not an exclusive solution for the problem of iterative demodulation and decoding of turbo-coded signals. Other scenarios are also possible. The algorithm proposed by Marsland & Mathiopoulos in [4] is a good example of a different iteration schedule. With this iterative decoding algorithm, the demodulator output is improved after each application of a MAP filter to one of the constituent codes. In this case, only the extrinsic information produced by the applied MAP filter is used to improve the demodulator output for the DQPSK symbols corresponding to the parity bits added by the corresponding constituent code. This can be achieved using an odd-even channel interleaving scheme, ensuring that each DQPSK symbol is representing one systematic and one parity bit (for a rate-1/2 turbo code). Therefore, the demodulator output for half of the symbols is improved each time a MAP filter is applied to one of the constituent code.

Note also that the demodulator proposed in the structure of [4] is based on a Multiple Differential Detector (MDD) for flat fading channels [32]. This demodulation method is

based on a linear-predictive approach similar to the one derived in Section 3.3. However, the MDD of [4] is not implemented as a symbol-by-symbol MAP demodulator. Furthermore, the receiver structure proposed in this thesis is applicable to both differentially encoded PSK signals and to coherent PSK signals with pilot symbols.

Finally, it is important to notice that the different scheduling solutions proposed in this chapter and in [4] are a subset of a large set of possible solutions. For the iterative demodulation and decoding of convolutionally encoded signals, only one solution was possible for the scheduling of the extrinsic information exchange between the demodulator and the decoder. The set of possible solutions becomes large if more decoders are involved, as it is the case with turbo-coded systems.

Alternative scheduling scenarios to the ones proposed in this chapter and in [4] can be derived using a more general approach to the problem of joint iterative demodulation and decoding. In the next chapter, such an approach based on Bayesian graphs and Pearl's algorithm is proposed to tackle the problem of joint demodulation and decoding. This approach brings further insights in the iterative structure proposed in this chapter.

Chapter 5 - Bayesian Graphs Interpretation

In Chapter 4, an iterative structure has been presented to solve the problem of joint demodulation and decoding of convolutionally encoded signals. Using the same approach, this structure has then been extended to the case of turbo-coded signals. The approach proposed in Chapter 4 is closely related to the one used for the iterative decoding of concatenated codes. In fact, it is possible, as proposed in [46], to interpret the problem of iterative demodulation and decoding of convolutionally encoded signals as a problem of iterative decoding of serially concatenated codes [45]. In this case, the modulation/channel is the inner code of this concatenated scheme. As in the iterative decoding of turbo codes, the component decoders exchange extrinsic information about their common symbols. Using this approach, the iterative demodulation and decoding structure proposed in Section 4.1 for convolutionally encoded signals can be derived in a straightforward manner.

This approach of simply deriving an iterative demodulation and decoding technique from the techniques of [11] and [15] for the iterative decoding of concatenated codes, although intuitively appealing, is not based on a solid theoretical background. The resulting iterative decoding methods are usually justified using simulated performance rather than theoretical explanations. Since the performance are usually very good, the iterative decoding approach based on the exchange of the extrinsic information seems widely accepted. Still, the iterative decoding algorithms, proposed by [11] and [15] or their descendents, appear like *ad hoc* methods for the decoding of concatenated codes.

Using similar structures for the iterative demodulation and decoding can therefore be questioned.

An interesting development, aiming to give a stronger theoretical background to iterative decoding principles, is the application of graphical models, such as Bayesian graphs, to this problem [47]. When the Bayesian graph approach can be used, there is a known iterative algorithm to compute the *a posteriori* probabilities of the constituent random variables, called Pearl's algorithm. McEliece, MacKay & Cheng [10], and Kschischang & Frey [48], first proposed to use Bayesian graph representations and Pearl's algorithm in the area of iterative decoding of concatenated codes. Pearl's algorithm is optimal under certain conditions on the Bayesian graph. These conditions are not met by most Bayesian graphs representing concatenated codes, but Pearl's algorithm still gives very interesting results in these cases. For instance, the celebrated turbo decoding algorithm of [11] can be derived using Pearl's algorithm, from the Bayesian graph associated with a turbo code [10]. Bayesian graph theory even allows the derivation of alternative algorithms for the iterative decoding of turbo codes, giving rise to very interesting results [49]. Therefore, Bayesian graph theory provides a fruitful theoretical background to the principles of iterative decoding.

In this chapter, Bayesian graph theory is applied to the problem of joint demodulation and decoding. This work provides a different interpretation to the iterative algorithms derived in Chapter 4. In this chapter, the theory of Bayesian graphs and Pearl's algorithm is briefly presented. For a textbook treatment of these topics, the reader is referred to

[47]. Then, the theory of Bayesian graph is applied to the problem of joint demodulation and decoding for convolutionally encoded signals, and the results are used to justify the iterative approach already proposed by several authors [8][9][36]-[44]. The same theory is then applied to the problem of joint demodulation and decoding for turbo-coded signals, in order to verify the novel iterative structure proposed in Section 4.2. The results are finally discussed and analyzed.

5.1 Probabilistic Inference Using Bayesian Graphs

In this section, Bayesian graph representations are first explained for a set of random variables. Afterwards, Pearl's algorithm, allowing the iterative estimation of the *a posteriori* probabilities for the random variables represented by a Bayesian graph, is presented.

5.1.1 Bayesian Graphs (Bayesian Networks)

Let $Z = \{Z_1, Z_2, \dots, Z_N\}$ be a set of N discrete random variables. If a subset Ω of random variables, termed *evidence set*, is observed, then the *evidence* is the event E defined as

$$E = \{Z_i = o_i : Z_i \in \Omega\}, \quad (88)$$

where o_i is the value observed for the random variable Z_i . The probabilistic inference problem consists in computing the *a posteriori* probabilities $p(Z_i | E)$ for the random variables $Z_i \notin \Omega$. Note that this problem applies to a lot of different situations. Among

others, the problem of joint demodulation and decoding is a problem of probabilistic inference.

Suppose that, for the sake of simplicity, the evidence set Ω is $\{Z_{N-K+1}, \dots, Z_N\}$, where K is the cardinality of Ω . Using the brute force approach, the joint pdf of $\mathbf{Z}$ can be marginalized to compute the *a posteriori* probability for the random variable Z_1

$$p(Z_1 = q | E) = \sum_{\mathbf{Z}, \mathbf{Z} \in \Omega} p(q, z_2, \dots, z_{N-K}, o_{N-K+1}, \dots, o_N). \quad (89)$$

This approach is usually computationally too expensive to be used in practice. One way to reduce the complexity is to take into account partial independencies between the random variable. These partial independencies can be described graphically using a Bayesian graph (also called Bayesian network). Such a graph exploits the fact that any joint pdf for a set of random variables $\mathbf{Z}$ can be factored such that

$$p(Z_1, \dots, Z_N) = \prod_{i=1}^N p(Z_i | A(Z_i)), \quad (90)$$

where $A(Z_i)$ is a subset of $\mathbf{Z}$ called the *parents* of Z_i . The variables for which Z_i is a parent are called the *children* of Z_i . This description of the joint pdf can be illustrated graphically using a Bayesian graph [47]. With such a graph, each random variable is represented as a node, while the statistical dependence relation is represented by an arrow from the parent to the child. An example of a Bayesian graph is illustrated in Figure 15, describing the following joint pdf

$$p(Z_1, Z_2, Z_3, Z_4, Z_5) = p(Z_1) p(Z_2 | Z_1) p(Z_3 | Z_1) p(Z_4 | Z_2, Z_3) p(Z_5 | Z_3). \quad (91)$$

Finally, note that a Bayesian graph is a Directed Acyclic Graph (DAG), meaning that there are no *directed* loops in the graph. Nevertheless, a DAG can have undirected loops. For instance, the Bayesian graph of Figure 15 has an undirected loop following the path $Z_1 \rightarrow Z_2 \rightarrow Z_4 \rightarrow Z_3 \rightarrow Z_1$. Bayesian graphs with undirected loops are termed loopy or multiply-connected graphs, while Bayesian graphs without loops are called loop-free graphs, singly-connected graphs, or simply trees [49].

Figure 15 – Example of a Bayesian graph.

5.1.2 Pearl's Algorithm for Probabilistic Inference

An iterative algorithm has been designed by Pearl to solve the problem of probabilistic inference for Bayesian graphs [47]. Pearl's algorithm is optimal for loop-free graphs, i.e. the estimated APP's computed by this algorithm for a loop-free graph converge to the exact APP's after of finite number of iterations. Furthermore, this algorithm is usually requiring much less complexity than the brute force computation of

the APP's.

Pearl's algorithm is a message-passing algorithm that circulates estimated probabilities (the messages) between the connected nodes of the Bayesian graph until the APP computed for each random variable (node) converges to the exact APP. Each node in the Bayesian graph is activated according to an iterative schedule, and when activated, the node sends messages (probabilities) to its parents and children. This probability propagation process allows each node to improve its estimated APP at each iteration. For a given node Z , and its associated evidence or observation $E = e$, the APP $p(z|e)$ is computed as

$$\begin{aligned}
 p(Z = z|e) &= \frac{p(z,e)}{p(e)} \\
 &= \frac{p(z,e^+,e^-)}{p(e)} \\
 &= \frac{p(e^-|z)p(z,e^-)}{p(e)} \\
 &= \frac{p(e^-)}{p(e)} \cdot p(e^-|z)p(z|e^-)
 \end{aligned} \tag{92}$$

where e^+ and e^- refer respectively to the observations corresponding the evidence subsets $\Omega^+ = \Omega \cap A(Z)$ and $\Omega^- = \Omega \setminus A(Z)$. Thus, e^+ corresponds to the observation related to the parents of Z , while e^- corresponds to the observation related to its children. Defining

$$\lambda_z(z) = p(e^-|z) \tag{93}$$

and

$$\pi_z(z) = p(z|e^+), \quad (94)$$

gives

$$p(Z = z|e) = \alpha \lambda_z(z) \pi_z(z), \quad (95)$$

where $\alpha = p(e^+)/p(e)$ is a constant.

Figure 16 – A node Z of a Bayesian graph, with its parents V and its children W .

The Bayesian graph of Figure 16 illustrates a node Z with a set of parent nodes $V = \{V_1, \dots, V_K\}$, and a set of child nodes $W = \{W_1, \dots, W_L\}$. When the node Z is activated, it sends a message to every parent nodes $V_i \in A(Z)$, for all possible values of v_i . It also sends a message to each of its child nodes W_j , for all possible values of z . The flow of messages is illustrated in Figure 17. For the activated node Z , the probabilities $\lambda_z(z)$ and $\pi_z(z)$ are first computed from the probabilities $\pi_{v_i,z}(v_i)$ and $\lambda_{w_j,z}(z)$ received

respectively from the parents V_i and the children W_j . This is accomplished as follows [10]:

$$\lambda_Z(z) = \prod_{i=1}^L \lambda_{W_i Z}(z) \quad (96)$$

$$\pi_Z(z) = \sum_{\mathbf{v}} \left[p(z | \mathbf{v}) \prod_{i=1}^L \pi_{V_i Z}(v_i) \right], \quad (97)$$

where $\mathbf{v} = (v_1, \dots, v_K)$. The APP can then be computed as in (95).

Figure 17 – Message flow illustration for a node Z of a Bayesian graph.

Then the probabilities to send to the parents and the children are [10]

$$\lambda_{Z V_i}(q) = \sum_{\mathbf{v}: v_i = q} \left[\left(\sum_z \lambda_Z(z) p(z | \mathbf{v}) \right) \cdot \prod_{\substack{j=1 \\ j \neq i}}^K \pi_{V_j Z}(v_j) \right] \quad (98)$$

and

$$\pi_{z_{W_i}}(z) = \pi_z(z) \cdot \prod_{\substack{j=1 \\ j \neq i}}^L \lambda_{w_j z}(z). \quad (99)$$

The probability $\lambda_{z_{V_i}}(q)$ can be interpreted as a likelihood probability of the form [49]

$$\lambda_{z_{V_i}}(q) = p(e_{z_{V_i}} | V_i = q), \quad (100)$$

where $e_{z_{V_i}}$ is the evidence (observation) available to V_i through Z . Similarly, $\pi_{z_{W_i}}(z)$ can be interpreted as

$$\pi_{z_{W_i}}(z) = p(z | e_{z_{W_i}}), \quad (101)$$

where $e_{z_{W_i}}$ is the evidence provided to W_i from Z .

If the Bayesian graph is loop-free, Pearl demonstrated that after an upper-bounded number of iteration of the activation schedule, the APP estimated at each node using equation (95) converges to the exact APP. If the graph is loopy, then convergence to the exact solution is not guaranteed. Nevertheless, in digital communications problems, the final hard decision on the information bits is usually more of interest than the APP's themselves. Therefore, in this case, Pearl's algorithm needs only to allow convergence to the correct hard-decisions. Fortunately, in practice, it has been observed that Pearl's algorithm "converges with high probability to the correct decisions" [50].

5.2 Application to Convolutionally Encoded Signals

As we have seen in the previous section, Pearl's algorithm is an iterative method to estimate the *a posteriori* probabilities associated with a Bayesian graph. In this section, a Bayesian representation for the problem of joint demodulation and decoding of convolutionally encoded signals is proposed. Using Pearl's algorithm, an iterative method is then derived to estimate the APP's. The resulting iterative demodulation and decoding method is finally discussed and compared with the structure previously proposed in Chapter 4.

5.2.1 Bayesian Graph Representation

There exists several ways to represent the problem of joint demodulation and decoding for convolutionally encoded signals as a Bayesian graph. One possibility is to represent each symbol and each state transition (for both the channel and the convolutional code) as a node of the graph, as illustrated in Figure 18. In this graph, on the channel code side, the information symbols are represented as $\{u_1, \dots, u_i, \dots, u_L\}$ while the corresponding convolutional code state transitions and outputs are represented respectively as $\{\sigma_1, \dots, \sigma_i, \dots, \sigma_L\}$ and $\{x'_1, \dots, x'_i, \dots, x'_L\}$. On the modulation/channel side, the observed samples are represented by node y , the modulation/channel state transitions are represented by the nodes $\{s_1, \dots, s_k, \dots, s_K\}$ and the corresponding modulation symbols by $\{x''_1, \dots, x''_k, \dots, x''_K\}$. The symbols $\{x_1, \dots, x_i, \dots, x_L\}$ in-between the code and the modulation/channel are using an alphabet that can be shared by the code output and the

modulation (e.g. binary alphabet). The channel interleaving is usually performed at this level. Note that the arrows *do not represent signal flows*, as in a block diagram, but statistical dependencies between the random variables represented by the nodes.

Figure 18 – Detailed Bayesian graph for convolutionally encoded signals.

This network, although very descriptive, is very difficult to analyze using Pearl's algorithm, and hence should be simplified. We can make things easier by using clustering to merge some nodes of the Bayesian graph described in Figure 18 [49]. The resulting

graph has much less nodes, which eases the application of Pearl's algorithm and the analysis of the results.

Based on the graph described in Figure 18, it is therefore desirable to merge some of the nodes. It is first important to note that when the graph corresponding to the code is isolated, it can be solved separately using the BCJR algorithm. In fact, if Pearl's algorithm is applied to this graph, the resulting algorithm is functionally equivalent to the BCJR algorithm [10][51]. It is therefore possible to merge most of the nodes involved. The resulting simplified graph is illustrated in Figure 19. On the channel code side, the nodes representing the information symbols u_i have been merged in a single node u representing the information symbol sequence. Furthermore, the nodes representing the different state transitions in time σ_i and the output x'_i have been merged in a unique node called σ . This node represents thus the sequence of states in the code trellis corresponding to the sequence u .

Figure 19 – Simplified Bayesian graph for convolutionally encoded signals.

The same approach is applied on the modulation/channel side, where the sequence of state transitions s_i and the corresponding symbols x''_i are all merged in a single node called s . This node represents the state sequence in the combined modulation and channel. The resulting graph has much less nodes than the one proposed in Figure 18. Therefore, it is much easier to apply Pearl's algorithm on this graph and to analyze the results.

5.2.2 Pearl's Algorithm Messages for the Proposed Graph

In order to understand how Pearl's algorithm solves the problem of joint demodulation and decoding represented by the Bayesian graph of Figure 19, the messages that the nodes are exchanging when activated are defined first. Using equation (98), the messages sent from node s to a node x_i are defined as

$$\lambda_{sx_i}(q) = \sum_{x: x_i=q} \left[\left(\sum_s \lambda_s(s) p(s | \mathbf{x}) \right) \cdot \frac{\prod_j \pi_{x,s}(x_j)}{\pi_{x,s}(x_i)} \right], \quad (102)$$

for all the possible values q of symbol x_i . Note that the state sequence s is a deterministic function of the sequence $\mathbf{x}$. Therefore, $p(s | \mathbf{x})$ will be equal to 1 for the sequence of states $s(\mathbf{x})$ corresponding the sequence of symbols $\mathbf{x}$, and it will be 0 for all the other sequences. Thus,

$$\sum_s \lambda_s(s) p(s | \mathbf{x}) = \lambda_s(s(\mathbf{x})), \quad (103)$$

and, from equation (96) and the fact that $\mathbf{y}$ is the observation (evidence),

$$\lambda_s(s(\mathbf{x})) = \lambda_{ys}(s(\mathbf{x})) = p(\mathbf{y} | s(\mathbf{x})), \quad (104)$$

since $\mathbf{y}$ has no child. Therefore, equation (102) yields

$$\lambda_{sx_i}(q) = \frac{\sum_{\mathbf{x}: x_i=q} \left[p(\mathbf{y} | s(\mathbf{x})) \cdot \prod_j \pi_{x_j s}(x_j) \right]}{\pi_{x_i s}(q)}. \quad (105)$$

Now, from equation (99) and (97), the message from a node x_i to the node s is defined as

$$\pi_{x_i s}(q) = \pi_{x_i}(q) = \sum_{\sigma} p(x_i | \sigma) \pi_{\sigma x_i}(\sigma) = \sum_{\sigma: x_i=q} \pi_{\sigma x_i}(\sigma), \quad (106)$$

since x_i is a deterministic function of the sequence σ . We also have, using equation (99), that

$$\pi_{\sigma x_i}(\sigma) = \frac{\pi_{\sigma}(\sigma) \cdot \prod_j \lambda_{x_j \sigma}(\sigma)}{\lambda_{x_i \sigma}(\sigma)}. \quad (107)$$

Furthermore, from the definition (equation (97)), we have that

$$\pi_{\sigma}(\sigma) = \sum_{\mathbf{u}} [p(\sigma | \mathbf{u}) \pi_{u\sigma}(\mathbf{u})] = \pi_{u\sigma}(\mathbf{u}(\sigma)), \quad (108)$$

since the sequence of states σ is a deterministic function of the sequence $\mathbf{u}$. $\mathbf{u}(\sigma)$ denotes the sequence $\mathbf{u}$ producing the sequence of states σ . Since node $\mathbf{u}$ has no parent,

$$\pi_{u\sigma}(\mathbf{u}(\sigma)) = p(\mathbf{u}(\sigma)) . \quad (109)$$

Therefore, using equations (107)-(109), equation (106) yields

$$\pi_{x,s}(q) = \sum_{\sigma: x_i=q} \left[\frac{p(\mathbf{u}(\sigma)) \cdot \prod_j \lambda_{x_j, \sigma}(\sigma)}{\lambda_{x_i, \sigma}(\sigma)} \right] . \quad (110)$$

Finally, using equations (98) and (96), we have that

$$\begin{aligned} \lambda_{x_i, \sigma}(\sigma) &= \sum_{x_i} \lambda_{x_i}(x_i) p(x_i | \sigma) \\ &= \sum_{x_i} \lambda_{sx_i}(x_i) p(x_i | \sigma) \\ &= \lambda_{sx_i}(x_i(\sigma)), \end{aligned} \quad (111)$$

since x_i is a deterministic function of the state sequence σ . $x_i(\sigma)$ represents the symbol x_i associated with the sequence σ . We can now rewrite the messages $\pi_{x,s}(q)$ as

$$\pi_{x,s}(q) = \frac{\sum_{\sigma: x_i=q} \left[p(\mathbf{u}(\sigma)) \cdot \prod_j \lambda_{sx_j}(x_j(\sigma)) \right]}{\lambda_{sx_i}(q)}, \quad (112)$$

or equivalently

$$\pi_{x,s}(q) = \frac{\sum_{\mathbf{x}: x_i=q} \left[p(\mathbf{u}(\mathbf{x})) \cdot \prod_j \lambda_{sx_j}(x_j) \right]}{\lambda_{sx_i}(q)}, \quad (113)$$

since the sequence $\mathbf{x}$ is a deterministic function of the code state sequence σ .

5.2.3 Discussion

In Section 5.2.2, all the messages between the different nodes of the Bayesian graph have been derived. These messages are used by Pearl's algorithm to iteratively estimate the *a posteriori* probabilities. Lets consider more closely the messages exchanged between the node s and the nodes x_i . These messages, obtained from equations (105) and (113), are summarized in the following table.

Table 1 – Messages for the Bayesian Graph Associated with Convolutional Codes

Messages direction	Messages definition
From node s to nodes x_i	$\lambda_{sx_i}(q) = \frac{\sum_{x: x_i=q} \left[p(y s(x)) \cdot \prod_j \pi_{x_j s}(x_j) \right]}{\pi_{x_i s}(q)}$
From nodes x_i to node s	$\pi_{x_i s}(q) = \frac{\sum_{x: x_i=q} \left[p(u(x)) \cdot \prod_j \lambda_{sx_j}(x_j) \right]}{\lambda_{sx_i}(q)}$

If the messages $\pi_{x_j s}(x_j)$ from nodes x_j to node s are interpreted as estimated *a priori* probabilities for the symbols x_j , then

$$p(s(x)) \approx \prod_j p(x_j) \approx \prod_j \pi_{x_j s}(x_j), \quad (114)$$

where $s(\mathbf{x})$ refers to the sequence of states s associated with the sequence of symbols $\mathbf{x}$, and the symbols x_j 's are assumed independent. Then, the *a posteriori* probability $p(x_i = q | \mathbf{y}, \Psi)$ is

$$p(x_i = q | \mathbf{y}, \Psi) = \sum_{\mathbf{x}: x_i = q} \frac{p(s(\mathbf{x}), \mathbf{y})}{p(\mathbf{y})} \approx \alpha \sum_{\mathbf{x}: x_i = q} \left[p(\mathbf{y} | s(\mathbf{x})) \cdot \prod_j \pi_{x_j s}(x_j) \right], \quad (115)$$

where $\alpha = 1/p(\mathbf{y})$ is a constant. Therefore, the numerator of the message $\lambda_{s_i}(q)$ can be viewed as an *a posteriori* probability computed using messages $\pi_{x_j s}(x_j)$ as *a priori* probabilities.

Define the following likelihood ratios, assuming that the symbols x_i are drawn from a binary alphabet:

$$\hat{\Lambda}_e^{[\Psi]}(x_i) = \frac{\lambda_{s_i}^{(+1)}}{\lambda_{s_i}^{(-1)}}, \quad (116)$$

and

$$\hat{\Lambda}_e^{[C]}(x_i) = \frac{\pi_{x_i s}^{(+1)}}{\pi_{x_i s}^{(-1)}}. \quad (117)$$

Then, making use of (115),

$$\begin{aligned}
\hat{\Lambda}_e^{[\Psi]}(x_i) &= \frac{\lambda_{sx_i}(+1)}{\lambda_{sx_i}(-1)} \\
&= \frac{\sum_{x: x_i=+1} \left[p(\mathbf{y} | s(\mathbf{x})) \cdot \prod_j \pi_{x_j s}(x_j) \right] / \pi_{x_i s}(+1)}{\sum_{x: x_i=-1} \left[p(\mathbf{y} | s(\mathbf{x})) \cdot \prod_j \pi_{x_j s}(x_j) \right] / \pi_{x_i s}(-1)} \\
&= \frac{\hat{\Lambda}(x_i | \mathbf{y}, \Psi)}{\hat{\Lambda}_e^{[C]}(x_i)}, \tag{118}
\end{aligned}$$

where $\hat{\Lambda}(x_i | \mathbf{y}, \Psi)$ is the likelihood ratio of the *a posteriori* probabilities computed by the demodulator using $\hat{\Lambda}_e^{[C]}(x_i)$ as *a priori* information.

Similarly,

$$\begin{aligned}
p(x_i = q | \mathbf{y}, C) &= \sum_{\mathbf{x}: x_i=q} \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}, \\
&\approx \alpha \sum_{\mathbf{x}: x_i=q} \left[p(\mathbf{x}) \cdot \prod_j p(y_j | x_j) \right], \tag{119}
\end{aligned}$$

if independent symbols $\mathbf{x}$ and channel samples $\mathbf{y}$ are assumed. Note that this independence assumption on $\mathbf{y}$ is rarely true in practice, but that it must be used to justify the application of the BCJR algorithm (as pointed out in Section 4.1). Using the definition of equation (100) and the fact that $\mathbf{y}$ is the evidence (observation) available to node x_i through s ,

$$p(y_j | x_j) = \lambda_{sx_j}(x_j). \tag{120}$$

Then, since the sequence $\mathbf{x}$ correspond uniquely to a sequence $\mathbf{u}(\mathbf{x})$,

$$p(x_i = q | \mathbf{y}, C) \approx \alpha \sum_{\mathbf{x}: x_i = q} \left[p(\mathbf{u}(\mathbf{x})) \cdot \prod_j \lambda_{sx_j}(x_j) \right]. \quad (121)$$

This quantity is therefore a function of the *a priori* probability $p(\mathbf{u}(\mathbf{x}))$ and the demodulation output $\lambda_{sx_j}(x_j)$. Therefore, we can write the likelihood ratio of the messages $\pi_{xs}(q)$ as

$$\begin{aligned} \hat{\Lambda}_e^{[C]}(x_i) &= \frac{\pi_{xs}(+1)}{\pi_{xs}(-1)} \\ &= \frac{\sum_{\mathbf{x}: x_i = +1} \left[p(\mathbf{u}(\mathbf{x})) \cdot \prod_j \lambda_{sx_j}(x_j) \right] / \lambda_{sx_i}(+1)}{\sum_{\mathbf{x}: x_i = -1} \left[p(\mathbf{u}(\mathbf{x})) \cdot \prod_j \lambda_{sx_j}(x_j) \right] / \lambda_{sx_i}(-1)} \\ &= \frac{\hat{\Lambda}(x_i | \mathbf{y}, C)}{\hat{\Lambda}_e^{[\Psi]}(x_i)}, \end{aligned} \quad (122)$$

where $\hat{\Lambda}(x_i | \mathbf{y}, C)$ is the likelihood ratio of the *a posteriori* probabilities computed by the decoder using $\hat{\Lambda}_e^{[\Psi]}(x_i)$ as the demodulator output. The quantity $\hat{\Lambda}_e^{[C]}(x_i)$ can therefore be interpreted as the extrinsic information produced by the decoder using the demodulation outputs $\hat{\Lambda}_e^{[\Psi]}(x_i)$. Similarly, $\hat{\Lambda}_e^{[\Psi]}(x_i)$ can be interpreted as the demodulation “extrinsic” information obtained using $\hat{\Lambda}_e^{[C]}(x_i)$ as *a priori* information for the symbol x_i .

This result means that when Pearl’s algorithm is applied to the graph of Figure 19, describing the problem of joint demodulation and decoding of convolutionally encoded signals, the resulting algorithm is exactly the one described in Section 4.2 (compare

equations (118) and (122) with equations (73) and (71)). The definition of extrinsic information, for both the decoder and the demodulator, can be easily derived from the definitions of messages $\lambda_{s,x_i}(q)$ and $\pi_{x_i,s}(q)$, which are exchanged between the nodes x_i and s . This suggests that the exchange of extrinsic information is a valid way to improve the accuracy of the demodulation and decoding processes iteratively. Furthermore, the way this extrinsic information is used by the other decoder in the computation of $\lambda_{s,x_i}(q)$ or $\pi_{x_i,s}(q)$ correspond to the structure for iterative demodulation and decoding of convolutionally encoded signals described in Section 4.2, and proposed by most authors [8][9][36]-[44].

5.3 Application to Turbo-Coded Signals

In the previous section, Bayesian graph theory was applied to the problem of joint demodulation and decoding of convolutionally encoded signals. When Pearl's algorithm is applied to the graph representing the problem, the resulting iterative algorithm is equivalent to the iterative structure usually proposed. It is of interest to apply the same approach to the problem of joint demodulation and decoding of turbo-coded signals. The algorithm obtained using Bayesian graph theory could then be compared with the novel structure proposed in Section 4.2, and developed using an analogy with the case of convolutionally encoded signals. The results will be used to confirm the validity of the algorithm proposed in this thesis.

5.3.1 Bayesian Graph Representation

Based on the same principles as those applied in Section 5.2.1 for convolutional codes, it is possible to represent the problem of joint demodulation and decoding of turbo-coded signals using the Bayesian graph illustrated in Figure 20. In this graph, the information bits are represented by the set of nodes $\mathbf{u} = \{u_1, \dots, u_i, \dots, u_N\}$. The state sequences of the constituent codes C_1 and C_2 are represented respectively by the nodes σ_1 and σ_2 . The parity bits added by codes C_1 and C_2 are represented respectively by the set of nodes $\mathbf{x}_1 = \{x_{1,1}, \dots, x_{1,i}, \dots, x_{1,M1}\}$ and $\mathbf{x}_2 = \{x_{2,1}, \dots, x_{2,i}, \dots, x_{2,M2}\}$. The state sequence for the modulation/channel is represented by node s , while the observed channel outputs are represented by node y .

Figure 20 – Bayesian graph for turbo-coded signal

5.3.2 Pearl's Algorithm Messages for the Proposed Graph

As it was done previously for convolutionally encoded signals, it is possible to devise an iterative algorithm for the estimation of the APP's by applying Pearl's algorithm to the Bayesian graph of Figure 20. The derivation of the different messages sent by the activated nodes of the graph is similar to the derivation of Section 5.2.2 for

convolutionally encoded signals and is provided in Annex C. The messages sent by the nodes are summarized in Table 2 and Table 3, for bottom up messages and top down messages respectively.

Table 2 – Bottom up messages for the Bayesian graph associated with turbo codes

Messages direction	Messages definition
From node s to nodes u_i , $x_{1,i}$ and $x_{2,i}$	$\lambda_{su_i}(q) = \frac{\sum_{x:u_i=q} \left[p(y s(x)) \cdot \prod_j \pi_{u_j,s}(u_j) \cdot \prod_k \pi_{x_{1,k}s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l}s}(x_{2,l}) \right]}{\pi_{u_i,s}(q)}$ $\lambda_{sx_{1,i}}(q) = \frac{\sum_{x:x_{1,i}=q} \left[p(y s(x)) \cdot \prod_j \pi_{u_j,s}(u_j) \cdot \prod_k \pi_{x_{1,k}s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l}s}(x_{2,l}) \right]}{\pi_{x_{1,i},s}(q)}$ $\lambda_{sx_{2,i}}(q) = \frac{\sum_{x:x_{2,i}=q} \left[p(y s(x)) \cdot \prod_j \pi_{u_j,s}(u_j) \cdot \prod_k \pi_{x_{1,k}s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l}s}(x_{2,l}) \right]}{\pi_{x_{2,i},s}(q)}$
From nodes σ_1 and σ_2 to node u_i	$\lambda_{\sigma_1 u_i}(q) = \frac{\sum_{u:u_i=q} \left[\prod_j \pi_{u_j,\sigma_1}(u_j) \cdot \prod_k \lambda_{sx_{1,k}}(x_{1,k}(u)) \right]}{\pi_{u_i,\sigma_1}(q)}$ $\lambda_{\sigma_2 u_i}(q) = \frac{\sum_{u:u_i=q} \left[\prod_j \pi_{u_j,\sigma_2}(u_j) \cdot \prod_k \lambda_{sx_{2,k}}(x_{2,k}(u)) \right]}{\pi_{u_i,\sigma_2}(q)}$

Table 3 – Top down messages for the Bayesian graph associated with turbo codes

Messages direction	Messages definition
From node u_i to node s	$\pi_{u,s}(q) = p(u_i = q) \cdot \lambda_{\sigma_1 u_i}(q) \cdot \lambda_{\sigma_2 u_i}(q)$
From nodes $x_{1,i}$ and $x_{2,i}$ to node s	$\pi_{x_{1,i},s}(q) = \frac{\sum_{x_1: x_{1,i} = q} \left[\prod_j \pi_{u,\sigma_1}(u_j(x_1)) \cdot \prod_k \lambda_{sx_{1,k}}(x_{1,k}) \right]}{\lambda_{sx_{1,i}}(q)}$ $\pi_{x_{2,i},s}(q) = \frac{\sum_{x_2: x_{2,i} = q} \left[\prod_j \pi_{u,\sigma_2}(u_j(x_2)) \cdot \prod_k \lambda_{sx_{2,k}}(x_{2,k}) \right]}{\lambda_{sx_{2,i}}(q)}$
From node u_i to nodes σ_1 and σ_2	$\pi_{u,\sigma_1}(q) = p(u_i = q) \cdot \lambda_{su_i}(q) \cdot \lambda_{\sigma_2 u_i}(q)$ $\pi_{u,\sigma_2}(q) = p(u_i = q) \cdot \lambda_{su_i}(q) \cdot \lambda_{\sigma_1 u_i}(q)$

5.3.3 Discussion

In order to comprehend the iterative algorithm resulting from the application of Pearl's algorithm to the Bayesian graph of Figure 20, the messages of Table 2 and Table 3 must be interpreted. First, the messages from node s , corresponding to the state sequence of the modulation/channel Ψ , to its parents $x = \{u, x_1, x_2\}$, can be associated with the output of the demodulation process. In fact, it can be shown as in Section 5.2.3 that the numerators of messages $\lambda_{su_i}(u_i)$, $\lambda_{sx_{1,i}}(x_{1,i})$ and $\lambda_{sx_{2,i}}(x_{2,i})$ are proportional to the

a posteriori probabilities of symbols u_i , $x_{1,i}$ and $x_{2,i}$, using messages $\pi_{u,s}(u_i)$, $\pi_{x_{1,s}}(x_{1,i})$ and $\pi_{x_{2,s}}(x_{2,i})$ as estimated *a priori* probabilities. Therefore, assuming a binary random variable, the likelihood ratio of the output of the demodulation process for a symbol x_i can be defined as

$$\begin{aligned}\hat{\Lambda}_e^{[\Psi]}(x_i) &= \frac{\lambda_{sx_i}(+1)}{\lambda_{sx_i}(-1)} \\ &= \frac{\sum_{x: x_i=+1} \left[p(y|s(x)) \cdot \prod_j \pi_{x,s}(x_j) \right]}{\sum_{x: x_i=-1} \left[p(y|s(x)) \cdot \prod_j \pi_{x,s}(x_j) \right]} \bigg/ \frac{\pi_{x,s}(+1)}{\pi_{x,s}(-1)} \\ &= \hat{\Lambda}(x_i | y, \Psi) \bigg/ \frac{\pi_{x,s}(+1)}{\pi_{x,s}(-1)},\end{aligned}\tag{123}$$

where $x_i \in \mathbf{x} = \{\mathbf{u}, \mathbf{x}_1, \mathbf{x}_2\}$, and $\hat{\Lambda}(x_i | y, \Psi)$ is the ratio of the APP's for symbol x_i using the modulation/channel structure Ψ and messages $\pi_{x,s}(x_j)$ as *a priori* probabilities. These messages $\pi_{x,s}(x_j)$ are the messages sent by nodes $\mathbf{x}$ to node s . They are thus computed based on the knowledge of the code structure C . It is therefore possible to define the likelihood ratio of this estimated *a priori* probability as

$$\hat{\Lambda}_e^{[C]}(x_i) = \frac{\pi_{x,s}(+1)}{\pi_{x,s}(-1)},\tag{124}$$

and hence

$$\hat{\Lambda}_e^{[\Psi]}(x_i) = \frac{\hat{\Lambda}(x_i | y, \Psi)}{\hat{\Lambda}_e^{[C]}(x_i)}. \quad (125)$$

This estimated *a priori* information $\hat{\Lambda}_e^{[C]}(x_i)$, obtained from the messages $\pi_{x_s}(x_i)$, is computed differently depending on the nature of x_i . First, if $x_i \in \mathbf{x}_1$, the message $\pi_{x_{1,s}}(x_{1,i})$ is computed as (using Table 3)

$$\begin{aligned} \pi_{x_{1,s}}(q) &= \frac{\sum_{x_1: x_{1,i}=q} \left[\prod_j \pi_{u_j, \sigma_1}(u_j(x_1)) \cdot \prod_k \lambda_{sx_{1,k}}(x_{1,k}) \right]}{\lambda_{sx_{1,i}}(q)} \\ &= \frac{\sum_{x_1: x_{1,i}=q} \left[\left(\prod_j p(u_j(x_1)) \lambda_{\sigma_2 u_j}(u_j(x_1)) \right) \cdot \left(\prod_k \lambda_{su_k}(u_k(x_1)) \cdot \prod_l \lambda_{sx_{1,l}}(x_{1,l}) \right) \right]}{\lambda_{sx_{1,i}}(q)} \end{aligned} \quad (126)$$

This message is thus computed using the structure imposed by code C_1 . In fact, using the reasoning about equation (121), it is seen that the numerator of equation (126) is proportional to the APP probability of symbol $x_{1,i}$ computed using messages $\lambda_{su_k}(u_k)$ and $\lambda_{sx_{1,k}}(x_{1,l})$ as demodulator outputs for the coded symbols, and $p(u_j) \lambda_{\sigma_2 u_j}(u_j)$ as *a priori* information. We can therefore write

$$\begin{aligned} \hat{\Lambda}_e^{[C]}(x_{1,i}) &= \hat{\Lambda}_e^{[C_1]}(x_{1,i}) \\ &= \frac{\pi_{x_{1,i}s}(+1)}{\pi_{x_{1,i}s}(-1)} \\ &= \frac{\hat{\Lambda}(x_{1,i} | y, C_1)}{\hat{\Lambda}_e^{[\Psi]}(x_{1,i})}, \end{aligned} \quad (127)$$

where $\hat{\Lambda}(x_{1,i} | y, C_1)$ is computed using $p(u_j)\lambda_{\sigma_2 u_j}(u_j)$ as *a priori* information. Similarly, if $x_i \in \mathbf{x}_2$, we can write

$$\hat{\Lambda}_e^{[C_1]}(x_{2,i}) = \hat{\Lambda}_e^{[C_2]}(x_{2,i}) = \frac{\hat{\Lambda}(x_{2,i} | y, C_2)}{\hat{\Lambda}_e^{[\Psi]}(x_{2,i})}, \quad (128)$$

where $\hat{\Lambda}(x_{2,i} | y, C_2)$ is computed using messages $p(u_j)\lambda_{\sigma_1 u_j}(u_j)$ as *a priori* information.

Finally, if $x_i \in \mathbf{u}$, we have

$$\begin{aligned} \hat{\Lambda}_e^{[C_1]}(u_i) &= \frac{\pi_{u_i}(+1)}{\pi_{u_i}(-1)} \\ &= \frac{p(u_i = +1)}{p(u_i = -1)} \cdot \frac{\lambda_{\sigma_1 u_i}(+1)}{\lambda_{\sigma_1 u_i}(-1)} \cdot \frac{\lambda_{\sigma_2 u_i}(+1)}{\lambda_{\sigma_2 u_i}(-1)} \\ &= \frac{\lambda_{\sigma_1 u_i}(+1)}{\lambda_{\sigma_1 u_i}(-1)} \cdot \frac{\lambda_{\sigma_2 u_i}(+1)}{\lambda_{\sigma_2 u_i}(-1)}, \end{aligned} \quad (129)$$

assuming equally likely symbols.

Note that all of the estimated *a priori* information values $\Lambda_e^{[C_1]}(x_i)$ used by the demodulation process are computed using messages $\lambda_{\sigma_1 u_j}(u_j)$ and $\lambda_{\sigma_2 u_j}(u_j)$. Therefore, it is important to understand how these messages are computed. Using the result of Table 2, it is seen that the messages $\lambda_{\sigma_1 u_j}(u_j)$ are computed as follow

$$\lambda_{\sigma_1 u_i}(q) = \frac{\sum_{\mathbf{u}: u_i = q} \left[\prod_j \pi_{u_j, \sigma_1}(u_j) \cdot \prod_k \lambda_{s_{x_{1,k}}}(x_{1,k}(\mathbf{u})) \right]}{\pi_{u_i, \sigma_1}(q)}$$

$$= \frac{\sum_{\mathbf{u}: u_i = q} \left[\left(\prod_j p(u_j) \lambda_{\sigma_2 u_j}(u_j) \right) \cdot \left(\prod_k \lambda_{su_k}(u_k) \cdot \prod_l \lambda_{sx_{1,l}}(x_{1,l}(\mathbf{u})) \right) \right]}{p(u_i = q) \lambda_{\sigma_2 u_i}(q) \lambda_{su_i}(q)}. \quad (130)$$

From equation (121), it is seen that the numerator of this equation is proportional to the APP of symbol u_i , computed based on the structure of code C_1 , using $p(u_j) \lambda_{\sigma_2 u_j}(u_j)$ as a *priori* information and $\lambda_{su_k}(u_k)$ and $\lambda_{sx_{1,l}}(x_{1,l})$ as demodulator outputs for the coded symbols. This message is therefore representing the extrinsic information added when code C_1 is decoded, using as refined *a priori* information $p(u_j) \lambda_{\sigma_2 u_j}(u_j)$. Hence, we can define the likelihood ratio of this extrinsic information as

$$\begin{aligned} \hat{\Lambda}_e^{[C_1]}(u_i) &= \frac{\lambda_{\sigma_1 u_i}(+1)}{\lambda_{\sigma_1 u_i}(-1)} \\ &= \frac{p(u_i = +1 | \mathbf{y}, C_1)}{p(u_i = -1 | \mathbf{y}, C_1)} \\ &= \frac{p(u_i = +1) \cdot \lambda_{\sigma_2 u_i}(+1) \cdot \lambda_{su_i}(+1)}{p(u_i = -1) \cdot \lambda_{\sigma_2 u_i}(-1) \cdot \lambda_{su_i}(-1)} \\ &= \frac{\hat{\Lambda}(u_i | \mathbf{y}, C_1)}{\hat{\Lambda}_e^{[C_2]}(u_i) \hat{\Lambda}_e^{[\Psi]}(u_i)}, \end{aligned} \quad (131)$$

assuming equally likely symbols $\mathbf{u}$. Similarly, for code C_2 ,

$$\hat{\Lambda}_e^{[C_2]}(u_i) = \frac{\hat{\Lambda}(u_i | \mathbf{y}, C_2)}{\hat{\Lambda}_e^{[C_1]}(u_i) \hat{\Lambda}_e^{[\Psi]}(u_i)}. \quad (132)$$

This extrinsic information is therefore computed using the extrinsic information produced by the other constituent decoder as refined *a priori* information. It also means

that the refined *a priori* information $\hat{\Lambda}_e^{[C]}(\mathbf{x}_1)$ is computed using the extrinsic information $\hat{\Lambda}_e^{[C_2]}(\mathbf{u})$ as *a priori* information. Similarly, the information $\hat{\Lambda}_e^{[C]}(\mathbf{x}_2)$ is computed using the extrinsic information $\hat{\Lambda}_e^{[C_1]}(\mathbf{u})$ as *a priori* information. Finally, the refined *a priori* information $\hat{\Lambda}_e^{[C]}(\mathbf{u})$ is simply computed as

$$\hat{\Lambda}_e^{[C]}(u_i) = \hat{\Lambda}_e^{[C_1]}(u_i) \cdot \hat{\Lambda}_e^{[C_2]}(u_i), \quad (133)$$

assuming equally likely symbols. Overall, the information $\hat{\Lambda}_e^{[C]}(\mathbf{x})$, used by the demodulator as refined *a priori* information to compute $\hat{\Lambda}_e^{[\Psi]}(\mathbf{x})$, is

$$\hat{\Lambda}_e^{[C]}(\mathbf{x}) = \begin{pmatrix} \hat{\Lambda}_e^{[C_1]}(\mathbf{u}) \cdot \hat{\Lambda}_e^{[C_2]}(\mathbf{u}) \\ \hat{\Lambda}_e^{[C_1]}(\mathbf{x}_1) \\ \hat{\Lambda}_e^{[C_2]}(\mathbf{x}_2) \end{pmatrix}, \quad (134)$$

if the vector of bits $\mathbf{x}$ is defined as

$$\mathbf{x} = \begin{pmatrix} \mathbf{u} \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}. \quad (135)$$

It is now very important to note that this definition of the refined *a priori* information used by the demodulator corresponds exactly to the one proposed in Chapter 4 (see equation (85)). The definitions obtained here for the demodulation outputs (equation (125)) also correspond exactly to the definition proposed in Chapter 4 (equation (73)). Furthermore, the definition of extrinsic information from codes C_1 and C_2 for bits $\mathbf{u}$

(equations (131) and (132)) obtained here correspond to the definition of extrinsic information proposed in Chapter 2, since from (16), we can write

$$\begin{aligned}
 \Lambda_e^{[C_1]}(u_k) &= \frac{e_{1,k}(u_k = +1)}{e_{1,k}(u_k = -1)} \\
 &= \frac{\frac{p(u_k = +1 | y, C_1)}{p(u_k = -1 | y, C_1)}}{\frac{p(y_{s,k} | u_k = +1)}{p(y_{s,k} | u_k = -1)} \cdot \frac{p(u_k = +1)}{p(u_k = -1)} \cdot \frac{e_{2,k}(u_k = +1)}{e_{2,k}(u_k = -1)}} \\
 &= \frac{\Lambda(u_k | y, C_1)}{\Lambda_e^{[\Psi]}(u_k) \Lambda_e^{[C_2]}(u_k)},
 \end{aligned} \tag{136}$$

assuming equally likely symbols u_k , and similarly

$$\Lambda_e^{[C_2]}(u_k) = \frac{\Lambda(u_k | y, C_2)}{\Lambda_e^{[\Psi]}(u_k) \Lambda_e^{[C_1]}(u_k)}. \tag{137}$$

Therefore, when applied to the Bayesian graph representing the problem of joint demodulation and decoding of turbo-coded signals, *the information flow obtained with Pearl's algorithm correspond precisely to the information flow of the iterative structure proposed in Chapter 4.*

Note however that Pearl's algorithm does not impose any order in the schedule used to update the different messages. Therefore, the schedule proposed in Chapter 4 is only one solution among a large set of solutions. For instance, the demodulator does not have to update the demodulation output each time both constituent codes have been decoded. One could iterate a few times the decoders before improving the demodulation output. Another interesting possibility would be to stop including the demodulator in the iterative

process once the demodulation output is no more improved by the refined *a priori* information. This could help to reduce significantly the complexity of the overall scheme, since typically the demodulation process will be more complex than the decoding process.

Since the nodes of the Bayesian graph can be activated concurrently, parallel processing could also be used. In this case, the demodulation and the decoding processes could be applied iteratively in parallel. At each iteration, the demodulator and the decoders are using the information updated during the previous iteration. This could reduce the inherent bias toward one of the constituent decoder, generally observed when one of them is activated first. This concurrent processing also leads to an attractive hardware implementation, reducing the delay required to decode one block. This approach has already been applied successfully to the decoding of turbo codes [49].

Several other scheduling scenarios are possible. Nevertheless, even though the convergence Pearl's algorithm does not depend on the iteration schedule for loop-free graphs, it does for loopy graphs such as the one of Figure 20 [51]. Therefore, although several schedules are possible, it does not mean that all these schedules perform equivalently. Furthermore, the loopy nature of the Bayesian graph of Figure 20 prevents convergence to be guaranteed for any iteration schedule. Nevertheless, for a broad class of loopy graphs, Pearl's algorithm converges to the correct decision with a high probability [50].

5.4 Summary

In this chapter, Bayesian graph theory has been applied to the problem of joint demodulation and decoding. For Bayesian graphs, the application of Pearl's algorithm yields an iterative algorithm for the calculation of the *a posteriori* probabilities. Therefore, an iterative algorithm can be derived for the problem of joint demodulation and decoding using Pearl's algorithm.

First, this approach has been applied to convolutionally encoded signals. In this case, Pearl's iterative algorithm corresponds exactly to the iterative demodulation and decoding structure already proposed by several authors [8][9][36]-[44], and presented in Section 4.1. Therefore, Bayesian graph theory provides an interesting theoretical background for the previously proposed iterative schemes.

Then, the Bayesian graph theory has been applied to turbo-coded signals. It was shown that for the graph describing the problem of joint demodulation and decoding of turbo-coded signals, the messages defined by Pearl's algorithm correspond precisely to the information exchanged in the iterative structure derived in Chapter 4. This result confirms the definitions for the information exchange used in the iterative demodulation and decoding scheme proposed in this thesis.

Nevertheless, the iteration schedule proposed in Chapter 4 appears as only one possible solution, from a large set of such solutions, and convergence is not guaranteed. It is therefore important to evaluate the performance of the iterative demodulation and decoding scheme proposed in Chapter 4 for turbo-coded signals. This is done in the next chapter, where simulations are provided for Rayleigh flat fading channels.

Chapter 6 - Simulation Results

So far, the work presented in this thesis has been analytical. In Chapter 3, different forms of soft-output demodulation schemes were derived, for the purpose of comparing the performance of several demodulation and decoding techniques. This comparison is performed throughout this chapter, for both the cases of convolutional and turbo coding. In chapters 4 and 5, iterative demodulation and decoding algorithms for convolutional codes were reviewed, and new forms of such algorithms were derived for turbo codes.

In this chapter, the performance of the algorithms of Chapter 4 and 5 is evaluated through simulations, for transmission over flat fading channels. First, the performance of iterative demodulation and decoding for convolutionally encoded signals is briefly illustrated. The simulation results allow a comparison of the iterative demodulation and decoding on flat fading channels for pilot symbol-assisted PSK signals [8], and differentially encoded PSK signals [9]. Next, the iterative demodulation and decoding scheme for turbo-coded signals on similar channels is examined. Several scenarios are simulated, in order to evaluate the performance under different conditions. These results are finally used to discuss the validity of the proposed schemes, as well as their strengths and limitations.

Most of the results in this chapter are presented in terms of Packet Error Rate (PER), instead of Bit Error Rate (BER). This choice is justified by the increasing popularity of packet-based wireless data communications. For data applications, corrupted packets are usually discarded by an ARQ scheme, and hence PER is a more realistic performance

measure than BER.

6.1 Convolutional Codes

The iterative techniques of [8] and [9], for convolutionally encoded PSK systems, are first simulated and compared with the non-iterative detection schemes of Section 3.2. Coherent detection with known CSI (Section 3.2.1) is simulated, and its performance is used strictly as a benchmark for other detection schemes, since it is an unrealistic detection scheme. Furthermore, pseudo-coherent detection using pilot symbols and MMSE filtering to estimate the channel (Section 3.2.2), and soft-output one-symbol differential detection (Section 3.2.3), are simulated as non-iterative alternatives for the detection of PSAM and differentially encoded signals respectively. The iterative techniques are simulated with the symbol-by-symbol MAP detection algorithms for PSAM and differentially encoded signals described in Section 3.3, and the iterative structure of Section 4.1. The goal of this section is to validate the simulation framework, and to establish a comparison reference for the iterative demodulation and decoding of turbo-coded signals.

These simulations use QPSK signals, with a rate- $\frac{1}{2}$ $K = 7$ convolutional code, a block length of 2048 information bits (including flush bits) and a 64x64 bit interleaver after the convolutional encoder. The fading process is generated using Clarke's model for Rayleigh fading with a maximum Doppler spread f_d of $0.01f_s$, where f_s is the symbol rate. For the PSAM cases, one pilot symbol is inserted for each 17 transmitted symbols. For the non-iterative channel estimation based on PSAM, a FIR filter spanning 35 pilot

symbols is used. It is optimized in the MMSE sense for each signal-to-noise ratio simulated. An example for this interpolation filter is given in Annex B. For the linear prediction-based detection algorithms, a 3rd-order predictor is used. Therefore, 64-state and 16-state symbol-by-symbol MAP demodulators are used for PSAM and differentially encoded signals respectively.

Figure 21 - BER for iterative demodulation and decoding of convolutionally encoded signals

BER's are compared in Figure 21, while PER's are compared in Figure 22. The results for the iterative techniques are presented for 4 iterations. Note also that the 0.26 dB overhead due to the power in the pilot symbols is taken into account for PSAM-based detection techniques. In these figures, we can see that:

- When compared with the BER obtained with perfect coherent detection assuming known CSI, there is a degradation of approximately 0.5 dB at a BER of 10^{-5} for

Figure 22 – PER for the iterative demodulation and decoding of convolutionally encoded signals

iterative differential detection. A similar degradation was observed in [9].

- When PER's are compared, there is no degradation for the iterative differential detection and decoding technique with respect to perfect coherent detection.
- For the iterative technique based on PSAM, the degradation is about 1 dB with respect to perfect coherent detection, for both the BER's and the PER's.
- The iterative detection schemes outperform their respective non-iterative counterparts. This improvement is particularly important for differentially encoded signals. In this case, the improvement over conventional differential detection is in the order of 3 dB for the BER's, and 3.5 dB for the PER's.

6.2 Turbo Codes

For a turbo code, the system is the same as the previously described convolutionally encoded system, except that the convolutional code is replaced by a rate- $\frac{1}{2}$ turbo code. This turbo code is formed using conventional rate- $\frac{1}{2}$ $K=5$ recursive systematic convolutional codes, and the so-called “spread” interleaver with a minimum distance of 25 bits. The complete turbo coding scheme is defined in Annex D. The iterative demodulation and decoding schemes are based on the structure proposed in Chapter 4 (Figure 14). The iteration schedule includes the demodulator in each iteration of the turbo decoder. The other simulation parameters are summarized in Table 4.

Table 4 – Simulation Parameters

Parameter	Value
Modulation	QPSK
Fading	Rayleigh (Clarke’s model)
Maximum Doppler spread	$f_d = 0.01f_s$
Turbo code	Rate- $\frac{1}{2}$ $K = 5$
Block size	2048 bits
Concatenation interleaver	Spread (S-Random) with $S = 25$
Channel interleaver	64 x 64 matrix interleaver

Note that a convergence test is used for the iterative demodulation and decoding schemes. The test simply consists in comparing the extrinsic information at the output of the constituent decoders after each iteration. If the hard decisions are equivalent for both decoders, it is assumed that the algorithm has converged to the optimum solution and the iterative process is stopped. A maximum of 8 iterations are performed. This speeds up the simulations significantly, since only a few iterations are usually required (typically between 3 and 5). Furthermore, no noticeable degradation is observed with respect to the full 8 iterations when the convergence test is used. For the cases where demodulation is not included in the iteration loop, 8 iterations are always performed in the turbo decoder.

The simulation results for the parameters described in Table 4 are shown in Figure 23 (BER) and Figure 24 (PER). From these results, a number of points can be noted:

- The iterative techniques based on differential encoding and PSAM outperform their conventional non-iterative counterparts (differential detection and MMSE channel estimation).
- When BER's are compared, for the iterative differential detection, the degradation with respect to coherent detection with known CSI is about 2dB at a BER of 10^{-5} , which seems to be consistent with the results obtained in [4].
- For the iterative technique using PSAM, the degradation is only about 1.25 dB with respect to ideal coherent detection.
- When PER's are compared, the iterative detection scheme based on differential encoding and PSAM show a degradation smaller than 2 dB with respect to ideal

Figure 23 - BER for the iterative demodulation and decoding of turbo-coded signals

coherent detection.

- For the iterative technique using PSAM, the degradation is only slightly larger than 1 dB.

Note that the results obtained here for the iterative demodulation and decoding scheme based on PSAM show the same trend as the results obtained in Section 6.1 for convolutionally encoded signals. The degradation observed with respect to ideal coherent detection is of the same order than what we observed previously for convolutionally

Figure 24 - PER for the iterative demodulation and decoding of turbo-coded signals

encoded signals. However, the iterative demodulation and decoding scheme based on differential encoding shows a larger degradation with respect to ideal coherent detection when a rate- $\frac{1}{2}$ turbo code is used instead of a rate- $\frac{1}{2}$ convolutional code. Remember that in Section 6.1, for the same channel conditions, no noticeable degradation with respect to ideal coherent detection is observed for the iterative demodulation scheme based on differential encoding (when PER's are compared).

This performance difference suggests that the linear-predictive demodulation scheme

based on differential encoding performs better when an iterative scheme using a convolutional code is used than when a turbo code is used. In the former case, the demodulation output, after a few iterations, allows the convolutional decoder to perform as well as if an ideal coherent detector was used (when PER's are compared). This does not appear to be the case for turbo-coded signals, since a degradation of almost 2 dB is observed with respect to ideal coherent detection.

Therefore, the extrinsic information for the coded bits produced by a turbo decoder does not improve as much the demodulator performance as the extrinsic information produced by a convolutional decoder. This might be caused by the fact that the extrinsic information produced by the turbo decoder is multi-variable, since extrinsic information is produced for the systematic bits, as well as for the parity bits of both constituent codes (see equation (85)). For a convolutional code, the quality of extrinsic information available at the output of the MAP filter for each coded bit is comparable. On the other hand, turbo decoders produce more extrinsic information for the systematic bits than for the parity bits, since both constituent decoders produce extrinsic information about the systematic bits. This affects the ability of the extrinsic information produced by the turbo decoder about the coded bits to improve the accuracy of the detection process.

One way to verify this hypothesis consists in comparing the ability of the extrinsic information for parity bits and for systematic bits to improve the demodulation accuracy. For instance, the performance of the iterative demodulation and decoding scheme can be compared for two cases:

- 1) only the extrinsic information about the systematic bits is fed back to the demodulator;
- 2) only the extrinsic information about the parity bits is fed back.

This comparison is presented in Figure 25 for the iterative demodulation and decoding scheme based on differential encoding, and the simulation conditions described in Table 4.

Figure 25 - PER with and without information about parity and systematic extrinsic information fed back to the demodulator

Note that for a rate- $\frac{1}{2}$ code, there is a same amount of systematic and parity bits. Therefore, if the effect of the extrinsic information produced for these categories of bits is equivalently affecting the accuracy of the demodulation process, no performance difference should be noted. However, a performance difference can be noted in Figure 25, indicating that the extrinsic information produced by the turbo decoder about the systematic bits produces more improvement in the detection accuracy than the extrinsic information about the parity bits.

Hence, this difference is probably responsible in part for the difference observed between the convolutionally encoded and the turbo-coded signals, when iterative demodulation and decoding based on differential encoding is performed. Another way to verify this assertion would consist in augmenting the proportion of systematic bits in the coded bits, i.e. to increase the coding rate of the turbo code. This verification is performed in the following section.

6.2.1 Coding Rate

In this section, the performance of the iterative demodulation and decoding schemes is evaluated for turbo codes with higher coding rates (greater than $1/2$). The performance, in terms of PER's, is illustrated in Figure 26 and Figure 27, respectively for a rate- $2/3$ and a rate- $3/4$ turbo code. The simulation conditions are similar to those defined in Table 4, except that the channel interleavers are 64×48 and 62×44 matrix interleavers, for the rate- $2/3$ and the rate- $3/4$ code respectively. In these figures, the following points can be noted:

- For a rate- $2/3$ turbo code, the degradation with respect to ideal coherent detection is less than 1 dB for the iterative demodulation and decoding scheme based on PSAM, and in the order of 1.5 dB for the one based on differential encoding.
- For a rate- $3/4$ turbo code, the degradation is only a few tenths of a dB for the PSAM-based scheme, and 1 dB for the scheme based on differential encoding.

Therefore, these results show an improvement in the performance of both types of iterative demodulation and decoding schemes as the rate of the turbo code is increased. These results tend to confirm the hypothesis proposed earlier about the effect of the non-uniformity of the extrinsic information on the performance of the demodulator. For a turbo code with a higher rate, the proportion of systematic bits is higher. For instance, 75% of the bits produced by a rate- $3/4$ turbo code are systematic bits, and hence, only 25% of the bits are parity bits, with less extrinsic information available. These results will be discussed further at the end of this chapter.

Figure 26 - BER for the iterative demodulation and decoding of a rate-2/3 turbo-coded signal

Figure 27 - PER for the iterative demodulation and decoding of a rate-3/4 turbo-coded signal

6.2.2 Convergence

The number of iterations required by the iterative detection schemes to converge to the correct decisions should be estimated, in order to deal with computational complexity issues. The convergence test previously described can be used to estimate the number of iterations required for convergence. Note that the iterative detection scheme might have converged before the test detects it. Therefore, the average number of iterations measured by the convergence test should be viewed as an upper bound to the average number of iterations required for convergence. This average number of iterations, measured by the convergence test, is illustrated in Figure 28 for different SNR's, based on the simulation parameters of Table 4. The corresponding PER's are indicated on the curves. From this figure, we can note the following observations:

- For a given SNR, the iterative detection scheme based on differential encoding requires less iterations to converge, on average, than the iterative detection scheme based on PSAM.
- This observation also holds true for a given PER. For instance, for a PER of 10^{-3} , the iterative demodulation and decoding scheme based on differential encoding requires on average about 3.5 iterations, while the iterative scheme based on PSAM requires about 4.5 iterations.

These results suggest that the iterative detection scheme based on differential encoding converges more rapidly than the one based on PSAM, for a given probability of convergence on a packet (equal to $1-\text{PER}$). Moreover, for a given SNR, the iterative

Figure 28 – Number of iterations required for convergence

detection scheme based on differential encoding converges faster, on average, than the one based on PSAM, even though it has a higher probability of non-convergence on a packet (higher PER).

6.2.3 Iteration Schedules

The results of Chapter 5 suggest that several iteration schedules are possible. The schedule proposed in Chapter 4 is only one possible solution. It is therefore of interest to evaluate the performance of different iteration schedules. Particularly, it is desirable to reduce the recurrence of the demodulation process to a minimum, since it is typically the process requiring the largest computational complexity. One interesting solution consists in including the demodulator only in the first L_d iterations, and then to iterate only on the turbo code for the remaining $L-L_d$ iterations, with L being the total number of iterations. The performance, in terms of PER's, is illustrated for different values of L_d in Figure 29 and Figure 30, respectively for the iterative demodulation and decoding scheme based on PSAM and for the one based on differential encoding.

In these figures, a number of points are apparent:

- For the iterative demodulation and decoding scheme based on PSAM (Figure 29), most of the improvement over the non-iterative demodulation scheme ($L_d = 1$) is obtained if the demodulator is included only twice in the iterative scheme ($L_d = 2$).
- For the scheme based on differential encoding (Figure 30), the demodulator has to be included more than 2 times in the iteration loop to achieve most of the gain.

The case with $L_d = 4$ seems to be a good compromise.

Figure 29 - PER for different iteration schedules for the iterative demodulation and decoding based on PSAM

These results tend to show that for the iterative demodulation and decoding based on PSAM, most of the improvement due to iterative demodulation is achieved if the extrinsic information produced by the code is fed back only once. On the other hand, the demodulation scheme based for differentially encoded signals requires more iterations to achieve most of the potential improvement.

Figure 30 - PER for different iteration schedules for the iterative demodulation and decoding based on differential encoding

6.2.4 Diversity Reception

As it was pointed out in Section 3.4, it is possible to use linear-predictive receivers in conjunction with diversity reception, whether it is antenna diversity or RAKE reception. The results for a 2-branch ($D = 2$) reception diversity scheme are presented in Figure 31.

Figure 31 - PER for the iterative demodulation and decoding schemes with a $D = 2$ diversity reception

For the two branches, independent fading processes are observed, with a similar fading spectrum and an equal mean power. This is simulated using two similar channels, with different realizations, for the same transmitted signal. For the conventional non-iterative demodulation schemes, the diversity combining is carried out as described in Section 3.4.1. Note that in Figure 31, E_b represents the *total* power received for one bit. On this figure, the following points can be noted:

- As expected, the performance of every receiver is improved by the use of reception diversity.
- The iterative demodulation and decoding schemes based on differential encoding or PSAM still outperform their non-iterative counterparts (one-symbol differential detection and MMSE estimation).
- The iterative demodulation and decoding scheme based on PSAM shows a degradation slightly inferior to 1.5 dB with respect to ideal coherent detection and Maximum Ratio Combining (MRC).
- The iterative scheme based on differential encoding shows a degradation smaller than 2 dB with respect to ideal MRC.

These results demonstrate the ability of the iterative demodulation and decoding schemes to perform well in conjunction with diversity reception.

6.2.5 Effect of the Fading Rate

The effect of the fading rate on the performance of the proposed iterative demodulation and decoding schemes is examined in this section. First, when the fading rate is decreased ($< 1\%$), it is observed that the iterative demodulation and decoding schemes close the gap with respect to ideal coherent detection, in terms of PER performance. This observation is illustrated in Figure 32 for the simulation parameters described in Table 4, with the exception of the fading rate, which is reduced to 0.125% of the symbol rate. For the PSAM scheme, the pilot symbols are still included once for every 17 transmitted symbols.

In Figure 32, it can be noted that both iterative demodulation and decoding schemes perform within a few tenths of a dB of the performance of ideal coherent detection. This denotes a significant improvement in the case of the iterative scheme based on differential encoding.

On the other hand, for PSAM signals, the performance of the pseudo-coherent detection based on MMSE channel estimation is very close to that of the ideal coherent detection. Hence, there is no real advantage in using the iterative demodulation and decoding scheme for PSAM signals in slow fading conditions. Note however that the interpolation filter, for the MMSE channel estimation, spans over 241 pilot symbols. This is obviously not desirable in a practical system, since it would require an additional delay at the receiver. Therefore, a slightly larger degradation with respect to ideal coherent detection should be expected in a practical system based on MMSE channel estimation.

Figure 32 - PER for the iterative demodulation and decoding schemes
 $(f_d = 0.00125f_s)$

The results for a faster case of fading are illustrated in Figure 33. In this case, the fading rate is increased to 5% of the symbol rate. Note that in order to sample the fading process at a rate higher than the Nyquist rate, one pilot symbol is inserted for each 9 transmitted symbols (instead of 17 previously). The interpolation filter, for the MMSE

Figure 33 - PER for the iterative demodulation and decoding schemes ($f_d = 0.05f_s$)

channel estimation scheme, spans over 15 pilot symbols. In Figure 33, it can be noted that all the receivers suffer from a greater performance degradation with respect to ideal coherent detection. This is particularly true for the MMSE channel estimation scheme.

However, it is important to remember that the performance of ideal coherent detection is not attainable in practice, and represents only a reference curve. Above all, it

is important to compare the performance of the iterative schemes with their non-iterative counterparts. Hence, for PSAM signals, the advantage of the iterative demodulation and decoding scheme, over MMSE channel estimation, augments as the fading rate is increased.

6.3 Discussion

In this chapter, the performance of the proposed iterative demodulation and decoding scheme for turbo-coded PSK signals in flat fading channels has been evaluated through simulations. It is shown that the iterative demodulation and decoding schemes outperform their non-iterative conventional counterparts in all the simulated scenarios. In particular, these results demonstrate the very good performance of the iterative scheme based on PSAM. For most of the simulated scenarios, the case with iterative processing based on PSAM is within 1 dB of the performance of ideal coherent detection. This performance is consistent with the results observed in [8], or with the results obtained in Section 6.1, for convolutionally encoded signals.

On the other hand, when compared with the results obtained for convolutionally encoded signals in [9] or in Section 6.1, the iterative demodulation and decoding scheme based on differential encoding for turbo-coded signals does not seem to perform as well. While for convolutionally encoded signals, the iterative scheme of [9] approach the performance of ideal coherent detection, a performance degradation is observed with respect to ideal coherent detection for turbo-coded signals. This is due in part to the non-uniformity of the extrinsic information produced by the turbo decoder. In this chapter, it

is shown that the extrinsic information produced about the systematic bits, when fed back to the demodulator, produces a more important performance improvement than the extrinsic information about the parity bits. This result is confirmed by the performance observed for higher rate turbo codes, for which the ratio of systematic bits is higher. As the rate of the turbo code is increased, the performance of the iterative demodulation and decoding scheme approaches that of ideal coherent detection.

Since the parity bits are decoded by only one of the constituent decoder, the *a posteriori* probabilities produced by the turbo decoder about these bits are less reliable, and hence, the corresponding extrinsic information is less reliable. In the modulation process, it is likely that the channel interleaver produces groups of consecutive symbols with a high concentration of parity bits. In a non-iterative receiver, this would have no effect. However, in an iterative receiver, this concentration of parity bits in consecutive symbols reduces the ability of the extrinsic information produced by the turbo code to improve the accuracy of the detection process. Hence, a performance improvement could be achievable by jointly designing both the concatenation and the channel interleavers to allow a uniform repartition of the parity bits in the transmitted symbols. For the iterative demodulation and decoding based on PSAM, this effect is mitigated by the presence of frequent known symbols, which greatly help the iterative scheme to converge to the optimal decision.

It is also worth noting that, in absolute terms, the iterative demodulation and decoding schemes for turbo-coded signals outperform their counterparts for convolutionally

encoded signals. This result is interesting, since the iterative schemes for convolutionally encoded signals imply an important increase in decoding complexity with respect to conventional non-iterative schemes, whereas no decoding complexity increase is required for turbo-codes, since they are already using soft-in/soft-out decoders in an iterative manner. For convolutional codes, the Viterbi algorithm must be replaced by a more complex BJCR (or sub-optimal equivalent) decoding algorithm.

Furthermore, the iterative receiver for turbo-coded signal, simulated in Section 6.2, bears a computational complexity of the same order of magnitude as that of the convolutionally encoded signals, simulated in Section 6.1. Considering that most of the performance improvement can be achieved when the demodulator is included in the first few iterations of the iterative scheme for turbo-coded signals, the demodulation complexity is similar to that of the iterative scheme for convolutionally encoded signals. Moreover, the decoding complexity is similar, since it is $\mathbf{O}(2^{K-1})$ for a constraint length K convolutional code decoded using a max-log MAP algorithm [52]. Therefore, 8 iterations of 2 x 16-state ($K = 5$) codes, is equivalent, in terms of complexity, to 4 iterations of a 64-state ($K = 7$) code. It is thus interesting to note that, for the same computational complexity, the iterative demodulation and decoding schemes for turbo-coded signals outperform the iterative schemes for convolutionally encoded signals of [8] and [9].

Chapter 7 - Conclusions

7.1 Summary

The problem of joint demodulation and decoding of turbo-coded signals on flat fading channels was investigated. Based on the iterative schemes of [8] and [9], the concept of iterative demodulation and decoding has been extended to turbo-coded signals. The proposed iterative structure can be used for Rayleigh fading channels, with a linear-predictive demodulator based on either PSAM or differential encoding.

The theory of Bayesian graphs has also been applied to the problem of iterative demodulation and decoding. For such graphs, Pearl's algorithm provides an iterative scheme for the computation of the *a posteriori* probabilities. It has been shown that when Pearl's algorithm is applied to the Bayesian graph representing the problem of joint demodulation and decoding of convolutionally encoded signals, the resulting algorithm corresponds precisely to the iterative demodulation and decoding scheme proposed previously in [8] and [9].

Furthermore, when a similar approach is applied to the case of turbo-coded signals, Pearl's algorithm yields an iterative demodulation and decoding structure over which a large set of possible iteration schedules can be used. One of these possible iteration schedules corresponds to the novel structure derived in this thesis, when the concept of extrinsic information is used in an *ad hoc* fashion.

These results give an alternative insight to the problem of iterative demodulation and

decoding. For loop-free Bayesian graphs, Pearl's algorithm is optimal, in the sense that it computes the exact *a posteriori* probabilities. Since the graphs representing the problem of joint demodulation and decoding are loopy, the resulting iterative algorithm is sub-optimal. Nevertheless, Bayesian graph theory brings an appealing theoretical background to the concept of iterative demodulation and decoding. It demonstrates that the iterative demodulation and decoding principle, based on the exchange of extrinsic information, follows from a sub-optimal application of Pearl's algorithm. Hence, Bayesian graph theory provides a theoretical justification of this latter iterative approach.

7.2 Simulation Results

The performance of the iterative demodulation and decoding scheme for turbo-coded PSK signals on Rayleigh flat fading channels has been evaluated through simulations. The results show that iterative demodulation and decoding is feasible for turbo-coded signals. Particularly, the following results have been observed:

1. The iterative demodulation and decoding scheme for turbo-coded PSAM signals outperforms its non-iterative counterpart (MMSE channel estimation) for the simulated scenarios.
2. When PER's are compared, the performance of this scheme is within only 1 dB of the performance of ideal coherent detection, for most of the simulated scenarios.
3. The iterative demodulation and decoding scheme for differentially encoded signals outperforms its non-iterative counterpart (differential detection).
4. As the rate of the turbo code is increased, the performance of the proposed iterative

demodulation and decoding schemes approaches that of the ideal coherent detection.

5. Both iterative demodulation and decoding schemes converge within a small number of iterations, typically between 3 and 5.
6. It is possible to reduce the computational complexity of the iterative demodulation and decoding schemes by reducing the number of iterations performed by the demodulator. Particularly, for the iterative demodulation scheme based on PSAM, the demodulator could be included in the iteration loop for the first few iterations only, without sacrificing performance.
7. When reception diversity is used, the iterative demodulation and decoding schemes still outperform their non-iterative counterparts.
8. For slower fading conditions ($< 1\%$), both iterative demodulation and decoding schemes are within a few tenths of a dB of the performance of coherent detection.
9. Except for the slower fading case ($< 1\%$), the iterative demodulation and decoding scheme based on PSAM outperforms the one based on differential encoding.
10. For the same computational complexity, the iterative demodulation and decoding schemes for turbo-coded signals outperform the iterative schemes for convolutionally encoded signals of [8] and [9].

7.3 Contributions

This thesis covers the following contributions:

1. The concept of iterative demodulation and decoding of PSK signals on Rayleigh flat fading channels has been applied to turbo-coded signals, and an iterative structure has

been derived for this scenario.

2. Bayesian graph theory has been applied to the problem of joint demodulation and decoding of convolutionally encoded signals. The similitude of the resulting algorithm with previously proposed iterative demodulation and decoding schemes has been pointed out.
3. The theory of Bayesian graphs has also been applied to the problem of joint demodulation and decoding of turbo-coded signals. It was shown that the iteration schedule proposed in this thesis corresponds to one instance of Pearl's algorithm, demonstrating the variety of possible solutions.
4. Simulations were performed to evaluate the performance of the iterative demodulation and decoding scheme derived in this thesis for turbo-coded signals. The results show the effectiveness of this approach.

7.4 Suggestions for Future Work

The results presented in this thesis raise a number of interesting topics for future research:

1. The results obtained in this thesis for turbo codes are a first step towards the application of the iterative demodulation and decoding concept to any concatenated codes. Serially concatenated codes [45], as used in Qualcomm's HDR packet-based mobile data communications system [53], would constitute good candidates in this case.
2. The results presented here could also be helpful to extend the iterative multiuser

detection method of [54] to turbo-coded signals.

3. The linear-predictive demodulators presented in this thesis can be extended to any linearly modulated signal, and to frequency selective fading channels, as demonstrated in [34]. Therefore, the iterative demodulation and decoding schemes presented in this thesis could be extended to these situations.
4. It would also be interesting to investigate the possibility of jointly designing the concatenation and channel interleavers, as suggested in Section 6.3.

Annex A – BCJR Algorithm

Let y_i^j be the symbol sub-sequence $(y_i, y_{i+1}, \dots, y_{j-1}, y_j)$ of the total sequence y . The length of the total sequence y is N samples. Using this notation, the joint probability $p(s_{k-1} = m', s_k = m, y_1^N)$ can be expressed as

$$\begin{aligned}
 p(s_{k-1} = m', s_k = m, y_1^N) &= p(s_{k-1} = m', s_k = m, y_1^k) p(y_{k+1}^N | s_{k-1} = m', s_k = m, y_1^k) \\
 &= p(s_{k-1} = m', s_k = m, y_1^k) p(y_{k+1}^N | s_k = m) \\
 &= p(s_{k-1} = m', y_1^{k-1}) p(s_k = m, y_k | s_{k-1} = m', y_1^{k-1}) p(y_{k+1}^N | s_k = m) \\
 &= p(s_{k-1} = m', y_1^{k-1}) p(s_k = m, y_k | s_{k-1} = m') p(y_{k+1}^N | s_k = m),
 \end{aligned} \tag{138}$$

since, for a Markov source, events after time k are independent of y_1^k and s_{k-1} if s_k is known. If we define the following functions

$$\alpha_k(m) = p(s_k = m, y_1^k)$$

$$\beta_k(m) = p(y_{k+1}^N | s_k = m)$$

$$\text{and } \gamma_k(m', m) = p(s_k = m, y_k | s_{k-1} = m'),$$

then the joint probability can be rewritten as

$$p(s_{k-1} = m', s_k = m, y_1^k) = \alpha_{k-1}(m') \cdot \gamma_k(m', m) \cdot \beta_k(m). \tag{139}$$

The quantities $\alpha_k(m)$ and $\beta_k(m)$ can be computed recursively since

$$\begin{aligned}
\alpha_k(m) &= \sum_{m'} p(s_{k-1} = m', s_k = m, y_1^k) \\
&= \sum_{m'} p(s_{k-1} = m', y_1^{k-1}) p(s_k = m, y_k | s_{k-1} = m', y_1^{k-1}) \\
&= \sum_{m'} \alpha_{k-1}(m') \cdot \gamma_k(m', m),
\end{aligned} \tag{140}$$

and

$$\begin{aligned}
\beta_k(m) &= \sum_{m''} p(s_{k+1} = m'', y_{k+1}^N | s_k = m) \\
&= \sum_{m''} p(s_{k+1} = m'', y_{k+1} | s_k = m) p(y_{k+2}^N | s_k = m, s_{k+1} = m'', y_{k+1}) \\
&= \sum_{m''} \gamma_{k+1}(m, m'') \cdot \beta_{k+1}(m'').
\end{aligned} \tag{141}$$

Usually, the initial state s_0 and final state s_N are known. For instance, flush bits are frequently added at the end of the message to encode in order to bring the encoder back to the all-zero state. In this case, the boundary conditions for the recursive equations (140) and (141) are

$$\alpha_0(m_i) = \begin{cases} 1 & \text{if } m_i = m_{\text{initial}} \\ 0 & \text{otherwise} \end{cases} \tag{142}$$

and

$$\beta_N(m_i) = \begin{cases} 1 & \text{if } m_i = m_{\text{final}} \\ 0 & \text{otherwise} \end{cases}, \tag{143}$$

where the known initial and final states are respectively m_{initial} and m_{final} . If the initial or the final state is unknown, then the boundary probabilities are assumed to be uniformly distributed. We also have that

$$\begin{aligned}\gamma_k(m', m) &= p(s_k = m \mid s_{k-1} = m') p(y_k \mid s_{k-1} = m', s_k = m) \\ &= p(u_k(m', m)) p(y_k \mid x_k(m', m)),\end{aligned}\tag{144}$$

where $u_k(m', m)$ is the message symbol and $x_k(m', m)$ is the output symbol corresponding to the state transition (m', m) at time k , since a unique message symbol and a unique output symbol is associated with a state transition. The probability $p(u_k)$ is the *a priori* probability of the message symbol u_k , while the probability $p(y_k \mid x_k)$ can be estimated based on the knowledge of the channel.

The BCJR algorithm consists in the following sequence of computations:

- 1) Equation (144) is computed first for each time index k , using the channel observations $\mathbf{y}$ and the *a priori* probabilities of the message symbols $\mathbf{u}$.
- 2) Afterwards, the quantities $\alpha_k(m)$ and $\beta_k(m)$ are computed recursively using (140) and (141).
- 3) It is then possible to compute the joint probability $p(s_{k-1} = m', s_k = m, \mathbf{y})$ for all time indexes k and possible state transitions (m', m) using equation (139).
- 4) Finally, the true APP of a message symbol u_k can be obtained from equation (3).

Annex B – Channel Interpolation Filter

An optimal interpolation filter minimizing the MSE, spanning over 35 pilot symbols, is illustrated in Figure 34 for a fading rate of 1% of the symbol rate and $E_s/N_0 = 10\text{dB}$. The pilot symbols are inserted once every 16 data symbols. The frequency response of this filter is compared to the fading spectrum in Figure 35. Note that both show a normalized bandwidth of approximately 0.01. An example of the amplitude and phase tracking ability of this filter is illustrated in Figure 36 for $E_s/N_0 = 10\text{dB}$. Note that the filter can estimate with a fairly accurate precision both the amplitude and the gain of the fading process.

Figure 34 – Impulse response of an interpolation filter for $f_d = 1\%$ and $E_s/N_0 = 10\text{dB}$.

Figure 35 – Frequency response of an interpolation filter for $f_d = 1\%$ and $E_s/N_0 = 10\text{dB}$.

Figure 36 – Channel estimation for $f_d = 1\%$ and $E_s/N_0 = 10\text{dB}$.

Annex C – Pearl’s Algorithm for Joint Demodulation and Decoding of Turbo-Coded Signals

In this Annex, the messages obtained when Pearl’s algorithm is applied to the Bayesian graph of Figure 20 are derived. First, let us define the messages sent from the node s , representing the channel/modulation state sequence, to the nodes in the sets $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{u}$.

From s to $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{u}$:

From equation (98), the message $\lambda_{sx_{1,i}}(q)$ from node s to a node $x_{1,i}$ is defined as

$$\lambda_{sx_{1,i}}(q) = \frac{\sum_{x: x_{1,i}=q} \left[\left(\sum_s \lambda_s(s) p(s | \mathbf{x}) \right) \cdot \left(\prod_j \pi_{u_j s}(u_j) \cdot \prod_k \pi_{x_{1,k} s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l} s}(x_{2,l}) \right) \right]}{\pi_{x_{1,i} s}(q)}, \quad (145)$$

where $\mathbf{x}$ is the set of nodes $\{\mathbf{u}, \mathbf{x}_1, \mathbf{x}_2\}$. Since s is a deterministic function of $\mathbf{x}$, and y is childless,

$$\sum_s \lambda_s(s) p(s | \mathbf{x}) = \lambda_s(s(\mathbf{x})) = p(y | s(\mathbf{x})), \quad (146)$$

where $s(\mathbf{x})$ is the sequence of states s corresponding to the sequence of symbols $\mathbf{x}$.

Therefore,

$$\lambda_{sx_{1,i}}(q) = \frac{\sum_{x: x_{1,i}=q} \left[p(y | s(\mathbf{x})) \cdot \prod_j \pi_{u_j s}(u_j) \cdot \prod_k \pi_{x_{1,k} s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l} s}(x_{2,l}) \right]}{\pi_{x_{1,i} s}(q)}, \quad (147)$$

and similarly for messages from node s to nodes $x_{2,i}$ and u_i ,

$$\lambda_{sx_{2,i}}(q) = \frac{\sum_{x_{2,i}=q} \left[p(y|s(x)) \cdot \prod_j \pi_{u_j s}(u_j) \cdot \prod_k \pi_{x_{1,k} s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l} s}(x_{2,l}) \right]}{\pi_{x_{2,i} s}(q)}, \quad (148)$$

and

$$\lambda_{su_i}(q) = \frac{\sum_{x_{u_i}=q} \left[p(y|s(x)) \cdot \prod_j \pi_{u_j s}(u_j) \cdot \prod_k \pi_{x_{1,k} s}(x_{1,k}) \cdot \prod_l \pi_{x_{2,l} s}(x_{2,l}) \right]}{\pi_{u_i s}(q)}. \quad (149)$$

From x_1 and x_2 to σ_1 and σ_2 :

The message from a node $x_{1,i}$ to node σ_1 is defined as

$$\begin{aligned} \lambda_{x_{1,i}\sigma_1}(\sigma_1) &= \sum_{x_{1,i}} \lambda_{x_{1,i}}(x_{1,i}) p(x_{1,i} | \sigma_1) \\ &= \sum_{x_{1,i}} \lambda_{sx_{1,i}}(x_{1,i}) p(x_{1,i} | \sigma_1) \\ &= \lambda_{sx_{1,i}}(x_{1,i}(\sigma_1)), \end{aligned} \quad (150)$$

since σ_1 is the only parent of $x_{1,i}$, and $x_{1,i}$ is a deterministic function of the state sequence σ_1 for the constituent code C_1 . The variable $x_{1,i}(\sigma_1)$ refers to the symbol $x_{1,i}$ corresponding to the state sequence σ_1 . By analogy, the message from a node $x_{2,i}$ to node σ_2 is defined as

$$\lambda_{x_{2,i}\sigma_2}(\sigma_2) = \lambda_{sx_{2,i}}(x_{2,i}(\sigma_2)). \quad (151)$$

From σ_1 and σ_2 to u :

The message for node σ_1 to a node u_i is defined as

$$\lambda_{\sigma_1 u_i}(q) = \frac{\sum_{u: u_i=q} \left[\left(\sum_{\sigma_1} \lambda_{\sigma_1}(\sigma_1) p(\sigma_1 | u) \right) \cdot \prod_j \pi_{u, \sigma_1}(u_j) \right]}{\pi_{u, \sigma_1}(q)}. \quad (152)$$

Since the state sequence σ_1 is a deterministic function of the sequence u , we can write

$$\sum_{\sigma_1} \lambda_{\sigma_1}(\sigma_1) p(\sigma_1 | u) = \lambda_{\sigma_1}(\sigma_1(u)) = \prod_k \lambda_{x_{1,k} \sigma_1}(\sigma_1(u)), \quad (153)$$

where $\sigma_1(u)$ refers to the state sequence σ_1 corresponding to the information bits sequence u . Using this last equation and equation (150), we can thus rewrite

$$\lambda_{\sigma_1 u_i}(q) = \frac{\sum_{u: u_i=q} \left[\prod_j \pi_{u, \sigma_1}(u_j) \cdot \prod_k \lambda_{x_{1,k} \sigma_1}(x_{1,k}(u)) \right]}{\pi_{u, \sigma_1}(q)}. \quad (154)$$

Similarly, a message from σ_2 to u_i can be defined as

$$\lambda_{\sigma_2 u_i}(q) = \frac{\sum_{u: u_i=q} \left[\prod_j \pi_{u, \sigma_2}(u_j) \cdot \prod_k \lambda_{x_{2,k} \sigma_2}(x_{2,k}(u)) \right]}{\pi_{u, \sigma_2}(q)}. \quad (155)$$

From u to σ_1 and σ_2 :

Using equation (99), the message from u_i to σ_1 can be defined as

$$\pi_{u, \sigma_1}(q) = \pi_{u_i}(q) \cdot \lambda_{s u_i}(q) \cdot \lambda_{\sigma_2 u_i}(q). \quad (156)$$

Since u_i has no parent,

$$\pi_{u_i}(q) = p(u_i = q), \quad (157)$$

and we can write

$$\pi_{u_i, \sigma_1}(q) = p(u_i = q) \cdot \lambda_{su_i}(q) \cdot \lambda_{\sigma_2 u_i}(q). \quad (158)$$

Similarly, the message from u_i to σ_2 can be defined as

$$\pi_{u_i, \sigma_2}(q) = p(u_i = q) \cdot \lambda_{su_i}(q) \cdot \lambda_{\sigma_1 u_i}(q). \quad (159)$$

From u to s :

The message from a node u_i to node s , using equations (99) and (157), can be written as

$$\pi_{u_i, s}(q) = \pi_{u_i}(q) \cdot \lambda_{\sigma_1 u_i}(q) \cdot \lambda_{\sigma_2 u_i}(q) = p(u_i = q) \cdot \lambda_{\sigma_1 u_i}(q) \cdot \lambda_{\sigma_2 u_i}(q). \quad (160)$$

From σ_1 and σ_2 to x_1 and x_2 :

The message from node σ_1 to a node $x_{1,i}$ can be defined as

$$\pi_{\sigma_1, x_{1,i}}(\sigma_1) = \frac{\pi_{\sigma_1}(\sigma_1) \cdot \prod_k \lambda_{x_{1,k} \sigma_1}(\sigma_1)}{\lambda_{x_{1,i} \sigma_1}(\sigma_1)}, \quad (161)$$

with

$$\pi_{\sigma_1}(\sigma_1) = \sum_u \left[p(\sigma_1 | u) \cdot \prod_j \pi_{u_j, \sigma_1}(u_j) \right] = \prod_j \pi_{u_j, \sigma_1}(u_j(\sigma_1)), \quad (162)$$

since the sequence of states σ_1 is a deterministic function of u , where $u_j(\sigma_1)$ refers to the information bit u_j corresponding to the sequence σ_1 . We can therefore rewrite

$$\pi_{\sigma_1, x_{1,i}}(\sigma_1) = \frac{\prod_j \pi_{u, \sigma_1}(u_j(\sigma_1)) \cdot \prod_k \lambda_{x_{1,k} \sigma_1}(\sigma_1)}{\lambda_{x_{1,i} \sigma_1}(\sigma_1)}. \quad (163)$$

By analogy, the message sent from node σ_2 to a node $x_{2,i}$ can be defined as

$$\pi_{\sigma_2, x_{2,i}}(\sigma_2) = \frac{\prod_j \pi_{u, \sigma_2}(u_j(\sigma_2)) \cdot \prod_k \lambda_{x_{2,k} \sigma_2}(\sigma_2)}{\lambda_{x_{2,i} \sigma_2}(\sigma_2)}. \quad (164)$$

From x_1 and x_2 to s :

Since node s is the only child of a node $x_{1,i}$, the message from $x_{1,i}$ to s is defined as

$$\pi_{x_{1,i}, s}(q) = \pi_{x_{1,i}}(q) = \sum_{\sigma_1} p(x_{1,i} = q | \sigma_1) \pi_{\sigma_1, x_{1,i}}(\sigma_1). \quad (165)$$

Since σ_1 is a deterministic function of the sequence x_1 , and $x_{1,i} = q$ can be satisfied by more than one sequence σ_1 , we have that

$$\pi_{x_{1,i}, s}(q) = \sum_{\sigma_1: x_{1,i} = q} \pi_{\sigma_1, x_{1,i}}(\sigma_1). \quad (166)$$

Therefore, substituting equation (163) in (166) yields

$$\pi_{x_{1,i}, s}(q) = \sum_{\sigma_1: x_{1,i} = q} \left[\frac{\prod_j \pi_{u, \sigma_1}(u_j(\sigma_1)) \cdot \prod_k \lambda_{x_{1,k} \sigma_1}(\sigma_1)}{\lambda_{x_{1,i} \sigma_1}(\sigma_1)} \right]. \quad (167)$$

Furthermore, using equation (150), we have that

$$\pi_{x_{1,i}s}(q) = \frac{\sum_{\sigma_1: x_{1,i}=q} \left[\prod_j \pi_{u, \sigma_1}(u_j(\sigma_1)) \cdot \prod_k \lambda_{sx_{1,k}}(x_{1,k}(\sigma_1)) \right]}{\lambda_{sx_{1,i}}(q)}. \quad (168)$$

Finally, since the sequence of states σ_1 can be uniquely associated with a sequence x_1 , we can equivalently write

$$\pi_{x_{1,i}s}(q) = \frac{\sum_{x_1: x_{1,i}=q} \left[\prod_j \pi_{u, \sigma_1}(u_j(x_1)) \cdot \prod_k \lambda_{sx_{1,k}}(x_{1,k}) \right]}{\lambda_{sx_{1,i}}(q)}. \quad (169)$$

Similarly, the message from $x_{2,i}$ to s is defined as

$$\pi_{x_{2,i}s}(q) = \frac{\sum_{x_2: x_{2,i}=q} \left[\prod_j \pi_{u, \sigma_2}(u_j(x_2)) \cdot \prod_k \lambda_{sx_{2,k}}(x_{2,k}) \right]}{\lambda_{sx_{2,i}}(q)}. \quad (170)$$

Annex D – Turbo Coding Scheme Definition

In this annex, the turbo coding scheme used in this thesis is defined. The constituent encoders and the interleaving strategy are detailed.

The turbo code simulated in this thesis is constituted by two $K=5$ recursive systematic convolutional codes with feedback and feedforward polynomials $(23)_8$ and $(35)_8$ respectively. A block diagram of one of these constituent codes is illustrated in Figure 37. The input block is terminated with $K-1$ flush bits, so the final state of the trellis associated with the first constituent code is known, and usually selected to be the all-zero state. The second constituent code trellis is not terminated. This trellis termination approach is widespread, although there exists better schemes, particularly for small blocks [55][56].

Figure 37 – Recursive systematic convolutional code with polynomials $(23)_8$ and $(35)_8$.

The concatenation interleaver is a *spread* (or *S-random*) interleaver [57]. The interleaving sequence of a spread interleaver is selected pseudo-randomly with the following constraint: each bits separated by a distance of S bits or less in the original bitstream must be separated by a distance of S or greater in the interleaved bitstream. The parameter S is sometimes referred as the *spreading factor* of the interleaver. For an interleaver of length N , the interleaving indices are selected randomly from a set of N integers numbered from 0 to $N-1$. If the integer selected randomly is within S of one of the last S selected integers, then this integer is discarded and a new integer is selected. If not, then the integer is added to the interleaving sequence and removed from the selection set.

Note that the interleaver design is crucial in the performance of turbo codes. This is particularly true at high SNR's, where the BER curve usually begins to bend. A better performance is usually observed when some form of randomness is present in the interleaver than when a highly structured interleaver, such as a classical block (or matrix) interleaver is used. This latter is characterized by a weaker dispersion [49]. However, interleaver selected randomly can show small spreading factors. Therefore, spread interleavers offer a good compromise between structured and random interleavers. Note also that even better interleaving schemes can as well be applied when very low error rates are required [58].

The performance of the simulated turbo code on an AWGN channel is now briefly

illustrated. The performance of a turbo-coded QPSK system is compared to that of a similar system using the well-known rate- $\frac{1}{2}$ $K = 9$ convolutional code [19], and to the performance of an uncoded QPSK system. In the simulations presented in this annex, a spread interleaver with a spreading factor of $S = 25$ is used. The results for an AWGN channel are presented in Figure 38 and Figure 39, in terms of Bit Error Rate (BER) and Packet Error Rate (PER), for 1, 2 and 8 decoding iterations of the turbo code.

First, if the turbo code results for a unique iteration are observed, it can be noted that the convolutional code largely outperforms the turbo code, both in terms of BER and PER. It is worth noting that the results after a single iteration correspond to the performance of a non-iterative decoder. Therefore, if the turbo code is decoded using the conventional serial decoding of concatenated codes, then it would not be a very interesting code. However, after a second iteration, the turbo code performance is improved considerably. Further iterations will improve even more the performance of the turbo code, although the first few iterations are responsible for most of the improvement. After 8 iterations, the turbo code outperforms the convolutional code by approximately 2.2 dB, for a BER of 10^{-5} or a PER of 10^{-2} .

Figure 38 – BER performance on an AWGN channel, $N = 4096$.

Figure 39 – PER performance on an AWGN channel, $N = 4096$.

References

- [1] E.K. Hall, S.G. Wilson, "Design and Analysis of Turbo Codes on Rayleigh Fading Channels", *IEEE JSAC*, Vol. 16, No. 2, pp. 160-174, Feb. 1998.
- [2] M.C. Valenti, B.D. Woerner, "Performance of Turbo Codes in Interleaved Flat Fading Channels with Estimated Channel State Information", *VTC 1998*, 1998.
- [3] I.D. Marsland, P.T. Mathiopoulos, "Differential Detection of Turbo Codes for Rayleigh fast-Fading Channels", *IEEE Communications Letters*, Vol. 2 No. 2, pp. 42-44, Feb. 1998.
- [4] I.D. Marsland, P.T. Mathiopoulos, "Multiple Differential Detection of Parallel Concatenated Convolutional (Turbo) Codes in Correlated Fast Rayleigh Fading", *IEEE Journal on Selected Areas in Comm.*, Vol. 16, No. 2, pp. 265-275, Feb. 1998.
- [5] M.L. Moher, J.H. Lodge, "TCMP – A Modulation and Coding Strategy for Rician Fading Channels", *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 9, pp. 1347-1355, Dec. 1989.
- [6] J.K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels", *IEEE Trans. on Vehicular Technology*, Vol. 40, No. 4, pp. 686-693, Nov. 1991.
- [7] J.H. Lodge, M.L. Moher, "Maximum Likelihood Sequence Estimation of CPM Signals Transmitted Over Rayleigh Flat-Fading Channels", *IEEE Trans. on Communications*, Vol. 38, No. 6, pp. 787-794, June 1990.
- [8] M.J. Gertsman, J.H. Lodge, "Symbol-by-Symbol MAP Demodulation of CPM and PSK Signals on Rayleigh Flat-Fading Channels", *IEEE Trans. on Communications*, Vol. 45, No. 7, pp. 788-799, July 1997.
- [9] P. Hoeher, J.H. Lodge, "Turbo DPSK: Iterative Differential PSK Demodulation and Channel Decoding", *IEEE Trans. on Communications*, Vol. 47, No. 6, pp. 837-843,

June 1999.

- [10] R.J. McEliece, D.J.C. MacKay, J.-F. Cheng, "Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm", *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 2, pp. 140-152, Feb. 1998.
- [11] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes", *IEEE ICC 1993*, pp.1064-1070, May 1993.
- [12] C.E. Shannon, "Communication in the Presence of Noise", *Proc. of the I.R.E.*, Vol. 37, pp. 10-21, January 1949.
- [13] G.D. Forney, *Concatenated Codes*, MIT Press, Cambridge, 1966.
- [14] S.B. Wicker, V.K. Bhargava, *Reed-Solomon codes and their applications*, New York, IEEE, 1994.
- [15] J.H. Lodge, R. Young, P. Hoeher, J. Hagenauer, "Separable MAP Filters for the Decoding of Product and Concatenated Codes", *IEEE ICC 1993*, pp.1740-1745, May 1993.
- [16] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. on Information Theory*, Vol. 20, pp. 284-287, March 1974.
- [17] J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", *IEEE Trans. on Information Theory*, Vol. 42, No. 2, pp. 429-445, March 1996.
- [18] J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis", *IEEE Trans. on Information Theory*, Vol. 24, pp. 76-80, Jan. 1978.
- [19] J.G. Proakis, *Digital Communications – 3rd Edition*, New York: McGraw-Hill, 1995.

-
- [20] G.D. Forney, "The Viterbi Algorithm", *Proc. of the IEEE*, Vol. 61, No. 3, pp. 268-277, March 1973.
- [21] P. Robertson, E. Villebrun, P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", *IEEE ICC* 1995, pp. 1009-1013, 1995.
- [22] G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi", *Annales des Télécommunications*, Vol. 42, No. 1-2, pp. 31-38, Jan. 1987.
- [23] J. Hagenauer, P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *IEEE Globecom* 1989, pp. 47.11-47.17, Nov. 1989.
- [24] M. Moher, "Decoding via Cross-Entropy Minimization", *IEEE Globecom* 1993, pp. 809-813, 1993.
- [25] K. Gracie, S. Crozier, A. Hunt, "Performance of a Low-Complexity Turbo Decoder with a Simple Early Stopping Criterion Implemented on a SHARC Processor", *6th International Mobile Satellite Conference*, Ottawa, pp. 281-286, June 1999.
- [26] J. Hagenauer, "Soft-in/Soft-out: The Benefits of Using Soft Values in All Stages of Digital Receivers", *3rd International Workshop on DSP Techniques Applied to Space Communications*, Noordwijk, The Netherlands, Sept. 1992.
- [27] P. Hoeher, "Advances in Soft-Output Decoding", *IEEE Globecom* 1993, Houston, Texas, pp. 193-197, Nov. 1993.
- [28] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communications Systems", *IEEE Communications Magazine*, pp. 90-109, July 1997.
- [29] T.S. Rappaport, *Wireless Communications: Principles and Practice*, New Jersey: Prentice Hall, 1996.
- [30] J.K. Cavers, "On the Validity of the Slow and Moderate Fading Models for Matched Filter Detection of Rayleigh Fading Signals", *Canadian Journal of*

Electrical and Computer Engineering, Vol. 17, No. 4, pp. 183-189, 1992.

- [31] M.C. Valenti, B.D. Woerner, "Refined Channel Estimation for Coherent Detection of Turbo Codes over Flat-Fading Channels", *IEE Electronics Letters*, Vol. 34, No. 17, pp. 1648-1649, Aug. 1998.
- [32] D. Makrakis, P.T. Mathiopoulos, "Optimal Decoding in Fading Channels: A Combined Envelope, Multiple Differential and Coherent Approach", *Globecom* 1989, pp. 1551-1557, Nov. 1989.
- [33] E.K. Hall, S.G. Wilson, "Turbo Codes for Noncoherent Channels", *Globecom* 1997, 1997.
- [34] X. Yu, S. Pasupathy, "Innovation-Based MLSE for Rayleigh Fading Channels", *IEEE Trans. on Communications*, Vol. 43, pp. 1534-1544, Feb./Mar./April 1995.
- [35] L. Davis, I. Collings, "DPSK vs. pilot-aided PSK MAP equalization for fast-fading channels" *Information, Decision and Control IDC* 1999, pp. 315-320, 1999.
- [36] M.J. Gertsman, J.H. Lodge, "Symbol-By-Symbol MAP Demodulation of CPM Signals On Rayleigh Flat Fading Channels", *7th International Conference on Wireless Communications*, Calgary, Vol. 1/2, pp. 219-238, July 1995.
- [37] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo Equalization", *European Transactions on Telecommunications*, Vol. 6, No.5, pp.507-511, Sept.-Oct. 1995.
- [38] G. Bauch, H. Khorram, J.Hagenauer, "Iterative Equalization and Decoding in Mobile Communications Systems", *European Personal Mobile Communications Conference (EPMCC)* 1997, pp. 307-312, Sept.-Oct. 1997.
- [39] M. Peleg, S. Shamai, "Iterative Decoding of Coded and Interleaved Noncoherent Multiple Symbol Detected DPSK", *Elect. Letters*, Vol. 33, No. 12, pp. 1018-1020, June 1997.

-
- [40] I.D. Marsland, P.T. Mathiopoulos, "Iterative Noncoherent Detection of Convolutionally Encoded Signals", *ICT* 1998, pp. 115-120, 1998.
 - [41] P. Hoeher, J. Lodge, "Iterative Decoding/Demodulation of Coded DPSK Systems", *Globecom* 1998, pp. 598-603, Nov. 1998.
 - [42] L. Davis, I. Collings, P. Hoeher, "Joint MAP Equalization and Channel Estimation for Frequency-Selective Fast-Fading Channels", in *Proc. 7th Communications Theory Mini-Conf.* in conjunction with *Globecom* 1998, pp. 53-58. Nov. 1998.
 - [43] G. Bauch, V. Franz, "Iterative Equalization and Decoding for the GSM System", *VTC* 1998, pp. 2262-2266, 1998.
 - [44] I.D. Marsland, P.T. Mathiopoulos, "On the Performance of Iterative Noncoherent Detection of Coded M-PSK Signals", *IEEE Trans. on Comm.*, Vol. 48, No. 4, pp. 588-596, April 2000.
 - [45] S. Benedetto, G. Montorsi, D. Divsalar, F. Pollara, "Iterative Decoding of Serially Concatenated Codes with Interleavers and Comparison with Turbo Codes", *Globecom* 1997, Vol. 2/3, pp. 654-658, 1997.
 - [46] K.R. Narayanan, G.L. Stuber, "A Serial Concatenation Approach to Iterative Demodulation and Decoding", *IEEE Trans. on Comm.*, Vol. 47, No. 7, July 1999.
 - [47] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco : Morgan Kaufmann, 1988, revised 2nd printing.
 - [48] F.R. Kschischang, B.J. Frey, "Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models", *IEEE Journal on Selected Areas in Comm.*, Vol. 16, No. 2, Feb. 1998.
 - [49] C. Heegard, S.B. Wicker, *Turbo Coding*, Boston : Kluwer Academic Publishers, 1999.
 - [50] G.B. Horn, R.J. McEliece, "Belief Propagation in Loopy Bayesian Networks:

-
- Experimental Results”, *IEEE International Symposium on Information Theory* 1997, p. 232, June-July 1997.
- [51] P. Meshkat, J.D. Villasenor, “Generalized Versions of Turbo Decoding in the Framework of Bayesian Networks and Pearl’s Belief Propagation Algorithm”, *ICC* 1998, Vol. 1/3, pp. 121-125, 1998.
- [52] P. Robertson, E. Villebrun, P. Hoeher, “A comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain”, *ICC* 1995, Vol. 2/3, pp. 1009-1013, 1995.
- [53] P. Sender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, A. Viterbi, “CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users”, *IEEE Communications Magazine*, Vol. 38, No. 7, pp. 70-77, July 2000.
- [54] M. Moher, “An Iterative Multiuser Decoder for Near-Capacity Communications”, *IEEE Trans. on Communications*, Vol. 46, No. 7, July 1998.
- [55] P. Guinand, J. Lodge, “Trellis Termination for Turbo Encoders”, *17th Biennial Symposium on Communications*, Kingston, pp. 389-292, May 1994.
- [56] S. Crozier, P. Guinand, J. Lodge, A. Hunt, “Construction and Performance of New Tail-Biting Turbo Codes”, *6th International Workshop on Digital Signal Processing for Space Applications*, Noordwijk, The Netherlands, paper 1.3, Sept. 1998.
- [57] D. Divsalar and F. Pollara, “Multiple Turbo Codes for Deep-Space Communications”, JPL, *TDA Progress Report 42-121*, May 1995.
- [58] S. Crozier, J. Lodge, P. Guinand, A. Hunt, “Performance of Turbo-Codes with Relative Prime and Golden Interleaving Strategies”, *6th International Mobile Satellite Conference*, Ottawa, pp. 268-275, June 1999.