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.

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.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI®

Simplified Iterative Noncoherent Demodulation and Decoding of Encoded M-DPSK Signals over Rayleigh Flat-Fading Channels

By Ronald S. Fernando, B.Eng

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

Master of Applied Science (M.A.Sc.)

**Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Ontario
September, 2001**

© Copyright

Ronald Fernando, 2001

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**385 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**385, 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.

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.

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 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.

0-612-66890-8

Canada

The undersigned hereby recommend to the faculty of Graduate Studies and Research acceptance of the thesis, "Simplified Iterative Noncoherent Demodulation and Decoding of Encoded M-DPSK Signals over Rayleigh Flat-Fading Channels", submitted by Ronald S. Fernando, B.Eng., in partil fulfillment of the requirements for the degree of Master of Applied Science (M.A.Sc.), in electrical engineering.

Chair, Department of Electronics Engineering
Thesis Co-Supervisor
Thesis Co-Supervisor

Department of Electrical Engineering

Faculty of Engineering

Carleton, University

Ottawa, Ontario

September, 2001

Abstract

After a number of seemingly unrelated ideas a common thread was identified to be at the core of this research experience. Using the ideas intrinsic to the fundamental iterative decoding algorithm for concatenated codes, we have attempted to transfer its most basic concepts to the design of a noncoherent receiver that combines performance with simplicity. Upon doing so, we conveniently arrive at a simplified joint-iterative demodulation and decoding scheme for M -DPSK signals over Rayleigh flat-fading channels. The performance bound of the proposed receiver is then demonstrated by way of simulation using “genie-aid” feedback, where the demodulator chosen is artificially driven by only correct feedback symbols .

Acknowledgements

I would like to thank everyone involved in my graduate school experience. In retrospect this was truly an amazing time and I know that the memories will stay with me for the rest of this lifetime. I am especially grateful to Prof. Jim Wight for allowing me the freedom to pursue all of my interests both academically and personally. His encouragement, counsel and willingness to share life experience will always be appreciated. Being allowed to do things on one's own terms is a rare gift. Sincerely, thank you very much. Special thanks is certainly required for Prof. Ian Marsland, my other co-supervisor, and one of the most entertaining people I have met here at Carleton. His patience, stories of travel and good humor were both motivating and made doing this research enjoyable. I would also like to thank Prof. John Knight for believing and recommending that I could actually teach an undergraduate course the first term I started this degree. Special thanks to Mohammed Houache for helping wrap up the final simulations for this work. I am indebted to my friends, in the department: Rony Amaya, Justin Fortier, Neric Fong, Michel Beshara, Vijaya Devabhaktuni, and Sundus Kubba for just being great people and dealing with, on a daily basis, that special side of me I reserve for friends. Thank you to Mike Britton for helping me out with the stochastic processes stuff. Your assistance exceeded its *expected value*. Thanks to my friends from Orléans. It has been more than a decade. Apparently we are getting old! I am extremely comforted by the fact that I know, and can say Laura Sie is a close and dear friend. The parallel our lives have run on, to this point, is uncanny. Thanks for all the support Laura. I would also like to thank the people I had to say goodbye to, during this time, for being a special part of my life. If we happen not to find our friendships again this lifetime hopefully we will find one another next lifetime. You have been and will be missed.

I will always be indebted to my mom, Maureen Fernando. Despite all the difficult times her emphasis on the need for me to get my education and dedication to making sure it happened will be appreciated forever. I would have never come so far without her.

This *research* would not have been possible without NSERC. Thank you NSERC.

Table of Contents

ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
LIST OF ACCRONYMS.....	vii
CHAPTER 1 - INTRODUCTION.....	1
1.1 Background.....	2
1.2 Research Contributions.....	7
1.3 Organization of thesis.....	8
CHAPTER 2 - COMMUNICATION SYSTEM MODEL.....	9
2.1 Transmitter.....	10
2.1.1 Convolutional Encoding.....	12
2.1.2 Notes on the Interleaver.....	13
2.2 Discrete-Time Channel Model.....	14
2.2.1 Rayleigh Frequency-Flat Fading with a Twist of White Noise.....	15
2.3 Soft/Hard-Symbol Joint Iterative Demodulator and Decoder.....	17
2.3.1. Iterative Decoding.....	19
2.3.2 Inner Decoder: Linear Prediction Based DF-DD.....	23
2.3.3 How DF-DD Simplifies the entire decoder.....	28
2.4 Reed-Solomon CODEC.....	30
2.4.1 Description of Reed-Solomon Codes over $GF(p^m)$	30
2.4.2 Encoding of a t-error correcting $RS(n,k)$ code over $GF(2^m)$	31
2.4.3 Decoding of a t-error correcting $RS(n,k)$ code over $GF(2^m)$	32

CHAPTER 3 - SIMULATION RESULTS.....	34
3.1 Convolutional Code.....	35
3.1.1 N: Smarter or Biggar.....	44
3.2 Reed Solomon Code.....	45
3.3 Convolutional Code Combined with the Reed-Solomon Code.....	47
3.4 A Final Look at the Results for the Outer Codes.....	49
3.5 Notes on the Simulation Environment.....	50
CHAPTER 4 - CONCLUSIONS AND FUTURE WORK.....	51
4.1 Summary.....	51
4.2 Simulation Results.....	52
4.3 Suggestions for Future Work.....	53
REFERENCES.....	54

List of Figures

Figure 1.1: Simplified Block Diagram of the Communication System Model.....	6
Figure 2.1: Simplified Block Diagram of the Communication System Model.....	9
Figure 2.2: Block Diagram of the Transmitter.....	10
Figure 2.3: Block diagram of the 16-state, rate 1/2 convolutional encoder, with generator $(23,35)_8$	12
Figure 2.4: Block Diagram of the Fading model.....	16
Figure 2.5: Block diagram of the iterative decoder structure.....	20
Figure 2.6: Simplified system.....	25
Figure 2.7: Linear Prediction Based Decision-Feedback Differential Detector [20].....	26
Figure 2.8: Soft/Hard-Symbol joint iterative demodulator and decoder.....	28
Figure 2.9:Block diagram for any nonbinary cyclic code for systematic encoding [34].....	32
Figure 3.1: System Model incorporating R-1/2 Convolutional Code $(23,35)_8$	35
Figure 3.2: Comparison of DF-DD demodulation to DD and “genie-aided” DF-DD demodulation BT = 0.01, Na = 1000.....	36
Figure 3.3a:Rayleigh flat fading, BT =0.01, Na = 1000.....	38
Figure 3.3b:Rayleigh flat fading, BT =0.05, Na = 1000	38
Figure 3.3c: Rayleigh flat fading, BT =0.00125, Na = 1000	38
Figure 3.4: Effect of different BT products on performance, after 4 iterations.....	39
Figure 3.5: Effect of word size on performance, BT=0.01(4 iterations).....	41
Figure 3.6: Updated, simplified system model.....	42
Figure 3.7a: Comparison of Single Differential Detection versus DF-DD based demodulation for the first iteration. BT = 0.01, Na = 1000.....	43
Figure 3.7b: Comparison of Hybrid Differential Detection versus DF-DD based demodulation after 3 itera- tions. BT = 0.01, Na = 1000.....	43
Figure 3.8: Comparison of different lengths of the DF-DD (N=2,4,5,10) for the fifth iteration of the DF-DD based receiver. BT = 0.01, Na = 1000.....	44
Figure 3.9: Differentially encoded RS(15,9) code over a Rayleigh flat-fading channel. BT = 0.01, Na = 1008.....	46
Figure 3.10: Differentially encoded RS(15,9) code over a Rayleigh flat-fading channel. BT = 0.01, Na = 1008.....	46
Figure 3.11: Combined effect of $(23,35)_8$ Convolutional code and RS(15,9), Reed-Solomon code. BT = 0.01, Na = 1008.....	48

Figure 3.12: Effect of word length on performance of the system using the (23,35)8 Convolutional code and RS(15,9), Reed-Solomon code.....48

Figure 3.13: Comparison of the rate-1/2 Convolutional Code with generator (23,35)8, RS(15,9) Reed-Solomon Code, and combination of the two after five iterations within the DFDD based receiver. BT = 0.01, Na = 1000, 1008.....49

List of Tables

Table 2.1: The Decoder Dialogue.....20

Table 2.2: Parameters for a t -error correcting RS(n,k) code over GF(p^m).....30

List of Acronyms

CODEC: Encoding and decoding
 SCCC: Serially concatenated convolutional codes
 M-PSK: M-ary phase shift keying
 M-DPSK: M-ary differentially encoded phase shift keying
 DD: Differential Detection
 MDD: Multiple Differential Detection
 BER: Bit error rate
 DF-DD: Decision Feedback Differential Detector
 AWGN: Additive white Gaussian noise
 MAP: Maximum *a posteriori*
 GM: Grey-mapping
 DE: Differential encoding
 CE: Convolutional encoding
 ISI: intersymbol interference
 APP: a posteriori probability
 ID: Inner decoder
 OD: Outer decoder
 BCH: Bose-Chauduri-Hocquenghem
 RS: Reed-Solomon
 GF: Galois Field
 BT: A measure of the rate of change of the fading process
 Na: pre-encoded framelength
 CC: Convolutional code
 DEC: Decoder

Chapter 1 - Introduction

It is apparent that since “turbo” codes were first presented in 1993 by Berrou, Glavieux, and Thitimajshima [1] the key contribution of that work was the fundamental iterative decoding algorithm as opposed to the new family of codes [2]. The concepts central to the iterative decoding algorithm have been used to design a number of concatenated encoding/decoding (CODEC) schemes and, more recently, joint demodulation and decoding of M -ary phase shift (M -PSK) signals [3, 4, 5, 6]. However, the fundamental algorithm has changed very little and much of the subsequent research has focused on the refinement [7], characterization [8], mutation [9], and explanation [10] of optimal CODECs. Thus far, the optimal and near-optimal decoding algorithms depend on constituent decoders that are state and memory intensive. In the arena of portable and low-power consumer electronics this presents a clear problem.

Using the fundamental iterative decoding algorithm, for concatenated codes, we have attempted to transfer its intrinsic concepts to the design of a noncoherent receiver that combines performance with simplicity. Upon doing so, we have conveniently arrived at a simplified joint-iterative demodulation and decoding scheme for M -PSK signals over Rayleigh flat-fading channels. The improvement is a result of the identification of a number of problems, to be described shortly, and recognizing that differential encoding is a form of recursive convolutional encoding; making it apparent that this transmission scheme could be viewed as a subset of the class of serially concatenated convolutional codes (SCCC's) [11], that are decoded iteratively. Thus suggesting (to me) that the iterative decoding structure presented in [12] could be a useful starting point for design of this receiver.

1.1 Background

In a wireless communication link employing M -PSK for the transmission of information symbols, maintaining accurate phase synchronization is essential for reliable communication [13,14,15,16]. For each information symbol, the transmitter for an M -PSK system selects one of M possible phases of the carrier wave to represent that symbol. To the regret of many researchers, the transmitter and receiver can not be truly synchronized and the resulting carrier phase error (offset) between the transmitter and receiver obscures the transmitted symbols. This issue can be resolved by either coherent or noncoherent demodulation. Coherent demodulation requires that the phase error be removed during demodulation and demands an explicit scheme to achieve accurate phase synchronization. On the other hand, noncoherent demodulation removes the phase error after demodulation and the phase synchronization is implicit, thus relaxing the constraints on the analogue circuitry of the system [2, 11].

Reference to coherent demodulation may be made throughout this work in relation to noncoherent demodulation; however, the subject is a large one and will not be treated in detail in this thesis. Details pertaining to coherent demodulation strategies are readily available in communications literature and the author of this thesis politely asks the reader to look for details elsewhere on that topic, [15].

One common form of noncoherent demodulation involves the use of differential encoding of the M -PSK information symbols at the transmitter, resulting in M -ary differential phase shift keying (M -DPSK) transmitted symbols. With M -DPSK symbols, the information is encoded on the change of the carrier phase over two consecutive symbols [13]. Under the

usual assumption that the phase errors affecting two consecutive symbols do not differ significantly, the receiver can determine the transmitted data by observing the change in carrier phase over two symbol intervals. This simple approach, known as differential detection (DD), is advantageous because it greatly reduces the requirement for accurately tunable analogue circuitry in comparison to coherent demodulation, and it is considered more robust due to the fact that carrier phase acquisition and tracking is not required. However, because the phase error is implicitly estimated using only the previously received symbol, and additive noise is inevitable, the result is a substantial noncoherence penalty over an ideally coherent demodulator [13]. Much research has been done to overcome this penalty, however it all condenses down to the idea that a number of previous received symbols can be used to better estimate the phase error [17,18].

Solutions previously proposed to overcome this noncoherence penalty were based on improving the reference symbols used for differential detection. These techniques generally involve increasing the number of previously received symbols that are used to estimate the phase error [17,18]. In fact, the iterative scheme proposed have been shown to actually provide better performance compared to systems using absolutely encoded coherent demodulation [19]. The optimal techniques, such as multiple differential detection (MDD) [17] have been designed to minimize the bit-error-rate (BER) at the output of the detector.

Although very effective, the simplification gained by adopting the noncoherent demodulation scheme is, for the most part, lost within the computational complexity of the proposed algorithms. The primary source of the huge increase in complexity is a result of using a

multiple differential detector (MDD) receiver structure, which is based on the Viterbi [17] or MAP algorithm [2]. Details pertaining to the MDD receiver can be found in [2,17].

To diminish this complexity, it was logical that a new, less complex demodulator be found to replace it. The replacement was found in [20], taking the form of a linear prediction decision feedback differential detector (DF-DD). Using the linear prediction DF-DD, it was found that the overly complex, memory intensive MDD receiver structure could be replaced by a simple modified tapped delay line (the constituent DF-DD structure) with acceptable performance degradation.

So, within the scope of this work simplified techniques for the noncoherent demodulation of M-DPSK signals were investigated in the context of an iterative demodulation and decoding scheme; whereby, error corrected symbols are fed back from the decoder to the demodulator so that the received frame can be re-processed. The iterations continue until such time that the gain from doing so is diminished. Of primary interest was the performance of a system that could employ different outer encoders, over a Rayleigh frequency-flat fading channel. This channel model is commonly used to model the multipath interference environment when it is assumed that all of the multipath components, of the received signal for a single symbol, arrive at the receiver with a delay spread much less than the symbol duration [20, 14].

For the Rayleigh frequency-flat fading channel, it is assumed that the transmitted signal simultaneously follows different paths to the receiver. Each transmission path has its own time-variant attenuation and phase distortion and when these signals meet at the receiver it is assumed they combine in a destructive manner, as they are no longer synchronized in

time, such that the combined received signal is further attenuated. As a result, the SNR of the combined signal is also lower than the SNR of any of the individual transmission paths. It should be stressed that this attenuation, commonly referred to as fading, is due to phase cancellation not power dissipation of the signal through the channel. Furthermore, the time-variant nature of the phase error in wireless channels leads to an error floor, with respect to the BER, that cannot be reduced merely by increasing the SNR. A detailed analysis of this phenomenon can be found in [13,14,15,16].

When employing noncoherent demodulation phase synchronization is not required; but, for a reliable wireless communication link knowledge of the channel statistics must be known by the receiver irrespective of the type of demodulation adopted. For a strictly AWGN channel only the static carrier phase error must be estimated; however, when noncoherent demodulation is employed this is a moot point since this is implicitly done, as a result of undoing the differential encoding. In the case of flat fading the dynamic phase error must be estimated and tracked [2]. When coherent detection is used, the channel estimation is incorporated during the synchronization of the analogue received signal, whereas with noncoherent detection, having implicit phase synchronization, the channel is estimated from the received samples as a part of the digital signal processing of the signal.

Although accurate channel estimation is necessary for reliable communication over any channel, the presence of background noise and multipath fading, which, particularly in wireless communication systems, can be rather insidious, means that some form of error control scheme is required to achieve acceptable system performance [2]. Often this will be done with the use of an error correcting code, such as a convolutional code; with which

strictly controlled redundancy, as specified by the code, is inserted into the data prior to transmission.

As mentioned, iterative decoding has already proved to be useful for a variety of applications. The concept of iterative decoding is perhaps best described in the context of decoding serial concatenated convolutional codes (SCCC's) [11,12]. The SCCC's can be decoded using the structure in Figure 2.1. The key to this scheme is the passing of soft information from one decoder to the next and repeating the decoding process in an iterative fashion, allowing both decoders to make full use of the soft information provided by the decoder that processed the frame before it.

Figure 1.1: Simplified Block Diagram of the Communication System Model

1.2 Research Contributions

In this thesis we attempt to find a simple yet effective method for supporting noncoherent detection of encoded M -DPSK signals. To that end we propose using the simple DF-DD structure presented in [20] with a convolutional code for error correction. We further propose employing the fundamental iterative decoding algorithm to improve the performance of the system. Unlike previously proposed iterative systems, this approach does not require a complex soft-output decoder, for the convolutional code (such as the MAP algorithm). Instead, the simple Viterbi algorithm can be used, maintaining local receiver complexity. Additionally, we investigate the use of a Reed-Solomon code and concatenated Reed-Solomon/convolutional code as replacements for the outer code. Computer simulations are used to investigate the BER performance of the systems.

Theoretically, on flat-fading channels, it has been shown in [21,22], that turbo codes can be effective in facilitating the use of low SNR's whilst still having acceptable bit-error rates (BER's). These studies assumed the use of ideally coherent M-PSK neglecting the immense difficulty of tracking and synchronizing signals in such harsh conditions. In fact, at such low SNR's frequent phase slips and long acquisition times would degrade performance (of the phase synchronization and tracking circuitry) so much that it would be, for all practical purposes, nearly impossible to achieve such favorable results. Attempts have been made [2,3,23,24] to work around this by differentially encoding the turbo-coded signal prior to transmission; But, again, as stated above, the inner decoder, used to undo the differential encoding, is very complex and memory intensive. In this work we also present

the before mentioned receiver as a simpler method of implicitly maintaining phase synchronization at reasonably low SNR's with acceptable performance degradation.

Furthermore we suggest that turbo-codes, in this type of system, may not be the best choice for an outer code. The reason being is that these codes are costly in terms of the requirements placed on the receiver to be able to process the computationally complex decoding algorithm and to provide sufficiently large amounts of memory per frame of received symbols. Notably the MAP algorithm required to decode this family of codes is the primary problem, as it adds immense complexity to the system. Hence if our goal is to present a simplified system, then it would be wise to be simple.

1.3 Organization of the thesis

First, Chapter 2 describes the communication systems model. This includes the components of the transmitter, the channel model, and components of the receiver. Specifically the decision feedback differential detector (DF-DD) is described, and its proposed use in this work is outlined in reasonable detail.

In Chapter 3 the simulation results are presented with related discussion and analysis.

Notably, at this point, the lower bound on performance is successfully demonstrated as described by an ideal feedback situation. Additionally, it is shown that reaching that bound is possible by way of the design presented in this work.

Finally, the results of this work and its contributions are summarized in Chapter 4 along with suggestions for future research.

Chapter 2 - Communication System Model

In this chapter the details of the communication system model adopted for this work are presented. Upon immediate inspection it is evident that the model, shown in Figure 2.1, consists of three elements: the transmitter; a transmission medium (the channel); and, finally, a receiver. The transmitter consists of an outer encoder, a symbol-by-symbol interleaver and an inner encoder. As the transmitted signal propagates through the channel it is distorted and what is observed by the receiver is different from what was transmitted. Accordingly, the purpose of the receiver is to undo the effects of both the transmitter and the insidious channel in an attempt to recover the original message sequence.

Figure 2.1: Simplified Block Diagram of the Communication System Model

The transmitter, which performs the encoding and modulation in discrete-time, is described in Section 2.1. The channel model is described in Section 2.2, with emphasis placed on the Rayleigh frequency-flat fading model. The iterative decoding process and inner decoder design, which is responsible for noncoherent demodulation and channel estimation, is discussed in Section 2.3. Finally, Section 2.4 briefly outlines the Reed-Solomon CODEC.

2.1 Transmitter

The transmitter is best described by the two functions it performs: encoding and modulation. For the proposed communication system, the encoding and modulation functions are not entirely distinct. Keeping this in mind, the result is that the transmitter can be viewed as a serially concatenated code. This is very useful as it easily lends itself to the fundamental iterative decoding algorithm. That being stated, it is still useful at this point to view the transmitter as a separate encoder and modulator, as shown in Figure 2.2. Encoding, which involves adding redundancy to the transmitted data sequence, is necessary for achieving acceptable system performance through error correction at the receiver. Once the data has been encoded, the code symbols are mapped to discrete phases used by the analogue signal generator when modulating the carrier wave. However, to facilitate noncoherent demodulation, the code symbols are represented by a change in carrier phase, not an absolute phase, so the modulated symbols are differentially encoded (DE) before transmission.

Figure 2.2: Block Diagram of the Transmitter

Encoding, in this system, is a combination of an error correcting code, represented by the outer encoder in Figure 2.2, and a symbol-by-symbol interleaver. The system was first developed using a convolutional code for the outer code, which was later replaced by a number of other codes. In sub-section 2.1.1, convolutional encoding is reviewed, while the turbo CODEC and Reed-Solomon CODEC are outlined at the end of the chapter. There were various options for the interleaver and a brief discussion is presented in sub-section 2.1.2 with respect to those options.

Modulation, in this system, takes place directly after the interleaving operation. The code symbols are mapped to the discrete phases that are used by the signal generator when modulating the carrier wave. For the sake of simplicity, the basic QPSK constellation (with Gray Mapping) was used for all of the simulations in this work. After the code symbols are mapped to points in the QPSK constellation, the modulated symbol sequence is differentially encoded. As described earlier, this means that the information is now represented by the change in carrier phase over two successive symbol intervals.

The differential encoding of the data symbol sequence could have been done before the symbols were mapped to the QPSK constellation and the end result would have been the same. In fact, this would be the better choice when implementing the system since the differential encoding could be done in the digital domain, thus simplifying the analogue circuitry in the transmitter. However, the way it is presented here better illustrates the encoding of information over a change in carrier phase, as opposed to absolutely encoding each information symbol as an absolute phase of the carrier wave.

2.1.1 Convolutional Encoding

Encoding is carried out on a fixed message word, $\underline{a}$, consisting of N_a symbols, each of k_c bits. The message word is encoded with a rate k_c/n_c convolutional code with a constraint length of L_c symbols¹. The convolutional encoder (CE) is a finite-state machine with N_s states. The encoder is initialized to the zero state prior to encoding the message word, and changes state as each message symbol is fed in. With each state transition, the convolutional encoder produces one code symbol of n_c bits. After the entire message word has been encoded an additional $L_c - 1$ zero symbols are appended to the message word to drive the encoder back to the all zero state, so the entire code word, $\underline{c}$, consists of $N_c = N_a + L_c - 1$ code symbols. Figure 2.3 shows the structure of a typical convolutional encoder.

Figure 2.3: Block diagram of the 16-state, rate 1/2 convolutional encoder with generator $(23,35)_8$.

¹ Although the focus of this thesis is on using a convolutional code for error correction, Reed Solomon and concatenated Reed-Solomon/convolutional codes are also considered.

2.1.2 Notes on the Interleavers

After the outer encoder has produced the code word, the symbols of the code word are rearranged using a pseudo-random interleaver that operates on a symbol-by-symbol basis, such that each code symbol remains intact and only the symbol order is changed. The interleaver has the dual purpose of being a necessary element for the joint iterative demodulation and decoding scheme, as well as providing protection against burst errors resulting from deep fades. In similar concatenated coding schemes the choice of the interleaver mapping, $IL[*]$, has been found to be an important parameter of the design [25,26,27] that can have an impact on the bit error rate. Most likely, that is also true for this work; however, it is not central to the design problem. Accordingly, no effort was made in this work to find an optimal interleaver mapping. It was assumed that a pseudo-random mapping with a uniform distribution would be satisfactory.

With symbol-by-symbol interleaving described above, each symbol remains intact. This differs from the more common bit-by-bit interleaving used by other researchers of this topic [11]. However symbol interleaving has the advantage of reduced complexity at the receiver, as the interleaver output remains a word of symbols denoted by $\underline{b}$. These symbols are referred to as the data symbols, to distinguish them from the message and code symbols. Furthermore, it is still unclear if there is actually a performance gain to be had by using bit-by-bit interleaving. And had it would more complexity the thought of using it was quickly abandoned.

2.2 Discrete-Time Channel Model

The impulse response of a wireless channel is dependent on a number of measurable parameters. Over the years a number of models have been developed to characterize the insidious effects of the channel upon the transmitted signal. The models are quite important in the design of transmitters and reception schemes that are meant to counter-act the effects of the channel such that a reliable communication link can be ensured. The models of interest for this work were the ones that could provide the parameters for the common additive white Gaussian noise (AWGN) process and the Rayleigh frequency-flat fading process.

For mobile wireless communication, designing for the effect of multipath interference requires special attention. In a multipath environment, replicas of the transmitted signal simultaneously traverse different paths from the transmitter to the receiver. The characteristics of the channel response are a combination of all the copies of the received signal, each of which has been altered by a different phase shift and attenuation. If we assume that all of the copies of the received signal arrive at nearly the same time the channel can be described with a *frequency-flat fading* model. This simplification also means that no intersymbol interference (ISI) has occurred; meaning that received copies of sequential symbols do not arrive such that they overlap in time. Equivalently, this is the same as saying that the signal bandwidth is less than the coherence bandwidth. Or perhaps, it is more descriptive to say that the frequency response of the channel is essentially constant (flat) over the frequency band of the transmitted signal.

In this work the envelope of the fading process was modelled as being Rayleigh distributed, given that the line-of-sight path is rarely present in mobile wireless channels. The result is the Rayleigh Frequency-Flat Fading model; which is simulated via Jakes' model [28]. Central to this, is the idea that the channel impulse response can be modelled as an impulse with time-variant gain and time-variant phase delay. The fading process is modelled as a stationary random process, correlated over time. The rate of change of this process is considered to be either fast or slow depending upon the relative speed between the transmitter and the receiver [29, 30].

2.2.1 Rayleigh Frequency-Flat Fading with Additive White Noise

Given that the fading process can be modelled as a time-variant gain and time-variant phase delay it is clear that the combined effect can be achieved with a time-variant complex gain. This is not an epiphany, but it does allow us to simplify things as can be seen in Figure 2.4, on the following page. We can write an equation for the received symbols as follows:

$$r_n = f_n v_n + w_n, \quad 1.$$

where $\{v_n\}$ are the transmitted symbols, $\{f_n\}$ is the complex discrete-time fading process, and $\{w_n\}$ is the complex discrete-time additive white gaussian noise process. In this work the land-mobile fading model was used to for the fading process, which could be simulated with Jakes' model. As such it delivered a fading process that has a complex Gaussian distribution, with zero mean and autocorrelation function of

$$\phi_m^{(F)} \equiv \frac{1}{2} E[f_n^* f_{n-m}] = \frac{1}{2} J_0(2\pi B_d m T), \quad 2.$$

Where $J_0()$ is the zeroth-order Bessel function of the first kind, B is the Doppler spread of the fading process, as T is the symbol duration. The BT product provides a measure of the rate at which the fading process changes over time. A high BT product implies a rapidly changing channel.

Figure 2.4: Block Diagram of the Fading model

The noise process is identical to that specified by a complex additive white gaussian noise process, with zero mean and autocorrelation function specified by

$$\phi_m^{(w)} \equiv \frac{1}{2} E[w_{n-m}^* w_n] = \frac{N_o}{2} \delta_m. \quad 3.$$

Where N_o is the single-sided noise power spectral density, and δ_m is the Kronecker delta function. The relationship between the transmitted symbols and received symbols, as stated above, is used to guide the design of the inner decoder presented in the following section.

2.3 Soft/Hard-Symbol Joint Iterative Demodulator and Decoder

In the previous section it was established that the transmitted symbols are different from the symbol sequence received from the channel. It is then clear that the possibility of error is unavoidable. Thus, our receiver must be designed to determine the most likely transmitted message; in effect minimizing the probability of incorrectly identifying the transmitted message. To accomplish this extraordinary task the transmission scheme has to be designed such that it can be assumed, when designing the receiver, that the received sequence, despite a hostile channel, will arrive with enough information to allow the receiver to extract the intended message from the received symbol sequence. So it is assumed that the transmitter has carefully added enough redundant information to allow the receiver to recover the message, using minimal information about the channel. The criteria, used to quantify the results for analysis, was the minimization of the bit-error-rate (BER) at a given signal-to-noise ratio (SNR).

To absolutely minimize the probability of a bit error, the decoder should use the *maximum a posteriori* probability (MAP) decision rule. That is, given that the entire received frame of encoded symbols has been observed, the decoder should choose the most likely value for the message bit, as a consequence of the decoding process, of the k^{th} bit within the n^{th} message symbol. So for any message bit in the sequence, if the probability that this bit is equal to zero exceeds the probability that it may be one, after observing the entire sequence, then the decoder will choose zero as the value of the particular bit, or choose one if the opposite is true. That is, choose $a_n^{(k)} = 0$ if

$$Pr\{a_n^{(k)} = 0|r\} > Pr\{a_n^{(k)} = 1|r\} \quad 4.$$

and choose $a_n^{(k)}=1$ otherwise. In the relation $Pr\{a_n^{(k)} = i / \mathbf{r}\}$ is the *a posteriori probability* (APP) of the k^{th} bit of the n^{th} message symbol. To implement this seemingly simple decision rule, a decoder must calculate the APP for each message bit. The total number of computations to calculate the APP's is on the order of 256 computations per symbol per frame. Given the requirement for high data rates the total number of computations can become very large very quickly.

It is generally accepted that the exact computation of all the APP's is not practical for interleaved systems. Thus, the decoding process has been separated into two stages: an inner decoder and an outer decoder; each of which is designed to undo their respective encoding process taking into account whatever channel information can be gathered or assumed *a priori*. The division between the inner decoder and outer decoder is the interleaver [and deinterleaver], since interleaving reduces the statistical dependence of sequential code symbols from the outer encoder. The idea, which is inherently sub-optimal, recovers any performance loss by iteratively decoding the received frame. However, the designs proposed in [2][19] rely on algorithms for the inner decoder that require huge amounts of data to be passed from the inner decoder to the outer decoder and back to the inner decoder, via interleavers. In this thesis the decoding scheme requires amounts of data that is far less than that required by previously proposed schemes.

The trade-off that motivated this work is the one between complexity and performance. Although optimal and near-optimal decoding strategies have been presented in the past; the criteria for measuring optimality has been based on maximizing the probability that correct decisions are made. In this thesis, a novel soft/hard-symbol joint iterative demodu-

lator and decoder is presented that greatly reduces the amount of memory and computations required to provide an effective error control scheme at level of complexity that can be considered realizable [practical] in a lower power mobile device.

2.3.1 Iterative Decoding

Before presenting the proposed simplified decoder, it is beneficial to examine the common and more complex soft-in/soft-out iterative decoding structure, shown in Figure 2.5. This structure was the starting point for this work and only by understanding its intricacies could the goal of developing a simplified decoder be realized. What makes this decoder, and other iterative decoding structures, so complex are the processes involved that make iteratively decoding a frame of data better than simply decoding the frame once. That being stated, we see, in Figure 2.5, that this structure consists of an inner decoder (ID) and outer decoder (OD) connected in a loop separated by interleavers. The role of the inner decoder is to undo the differential encoding (used for the inner encoder) and attempt to compensate for the insidious effects of the channel that lead to fading. Similarly, the outer decoder is present to undo encoding of the outer encoder, compensating for the effects of AWGN imposed on the transmitted signal. The interleaver, at the encoder, was used to randomize the code word, produced by the outer encoder, thus the interleavers within the decoder are present to undo and redo that same randomization at their respective places in the loop.

“But why the loop?” This being a very good question, it deserves to be answered. The loop allows the ID and OD to share information about the received symbol frame. After each decoder has had a chance to decode the frame the following decoder can use infor-

mation about the frame from the previous decoder. So if the frame is decoded only once the OD can use information extracted by the ID, but the ID does not get a chance to use any information gained, of the frame, by the OD. Thus, closing the loop give the ID a chance on subsequent iterations a chance to consider more information than just the received symbol frame.

Figure 2.5 Block diagram of the iterative decoder structure

The nature of the information being discussed will be described in further detail shortly. For the moment let us consider an interesting abstraction of the said process [iterative decoding] written by Marsland and presented to Mathiopoulos before the turn of the last century, in defense of his ideas on the subject. The abstraction, in Table 2.1 below, is an imaginary discussion between the ID and the OD.

TABLE 2.1. The Decoder Dialogue

ID:	Upon careful examination of the received samples, I have concluded that the data symbols are $b_{(1)}$.
OD:	But you have neglected the convolutional code in your examination. I believe the data symbols are $b_{(2)}$.
ID:	That's nice, but you are too far removed from the channel to fully appreciate what is going on. However, incorporating some of your suggestions, I find that $b_{(3)}$ is quite likely.
OD:	That's better, but how about $b_{(4)}$?

Although this discussion is not the Melian dialogue, it does provide an intuitive perspective into what is going on. The witty banter typically continues for some predetermined number of iterations, based on some expected BER and diminishing rate of return on having the process continue indefinitely.

With that basic understanding we can now turn our attention to the mathematical details of the process. Using the received frame of M-DPSK symbols, the inner decoder attempts to compute the a posteriori data symbol probabilities $Pr\{b_n = b / \mathbf{r}_o^N\}$ for all hypothetical [message symbols] $b \in M$ and $n \in \{1, 2, \dots, N\}$ [each received symbol]. These APP's reflect the likelihood of each of the M possible different values for b_n being transmitted, given that the symbols $\mathbf{r}_o^N$ are received. To calculate the APP's, there exists specific algorithms, depending on a number of different factors; However, most are derived from the popular BCJR algorithm. These algorithms take into consideration the relationship between the hypothetical message symbols, $\mathbf{b}$, and the received symbol sequence $\mathbf{r}_o^N$, including encoding, modulation, phase error and SNR. For simplicity, the algorithms all assume that the data symbols are statistically independent, meaning that the *a priori* probability distributions $Pr\{b_n = b\}$ are not necessarily identical for each transmitted symbol. Because the data symbols are actually the output of a deterministic encoder, this assumption is not quite true. Although interleaving the output of the outer encoder does reduce the statistical dependence between the sequential data symbols, the inner decoder can only compute estimates of the APP's.

This is the “*information*” referred to above and is known as extrinsic information. It is then deinterleaved and passed to the outer decoder. Using the inverse of the mapping used by the interleaver within the encoder, the deinterleaver keeps the M extrinsic information quantities associated with each data symbol together as a single unit, changing only the order of the units. The deinterleaved extrinsic information is $P_{n,c}^{(Sc)}$ and represents the likelihood of the different values for each code symbol to have been transmitted based on the received samples. It is information gained about each code symbol solely from the observation of the channel, as extracted by the inner decoder. The constraints on the sequence of code symbols introduced by the outer decoder have not been exploited in determining the extrinsic information. These constraints are taken into consideration by the outer decoder, that will produce estimates of the a posteriori message symbol probabilities $P_{n,a}^{(Oa)} = Pr\{\alpha_n = a / \mathcal{L}_o^N\}$.

As noted, separating the decoding process into these two stages is not ideal and leads to larger performance losses. By assuming that the data symbols are independent within the inner decoder, considerable information, about the frame, is ignored. To compensate, the outer decoder computes estimates of the a posteriori code symbol probabilities $Pr\{c_n = c / \mathcal{L}_o^N\}$. After the extrinsic from the inner decoder has been removed by division, the resulting quantities are interleaved and passed back to the inner decoder for the next iteration. The inner decoder repeats its operation, using the same received samples $\mathcal{L}_o^N$, but uses the interleaved quantities from the outer decoder to replace the a prior probabilities $Pr\{b_n = b\}$. That is the inner decoder no longer assumes that the data symbols were ever statistically independent, thus allowing it to calculate more accurate APP's during this and subsequent

iterations. The result being that the inner decoder can be chosen, or designed, independent of any constraints imposed by the outer encoder by using iterative decoding to compensate.

Clearly, at this point, the inner decoder produces a new set of APP's, passing the updated extrinsic information back to the outer decoder. This process is repeated several times in an iterative fashion, hopefully resulting in more accurate APP's after each iteration. After some fixed number of iterations, decoding terminates, and the a posteriori probabilities of the message bits, $Pr\{\alpha_n = a / \mathcal{L}_o^N\}$, computed by the OD, are passed to a decision device from which a final result is delivered. If this the details of this sub-section were not intuitively obvious the author kindly refers the reader to [11].

2.3.2 Inner Decoder: Linear Prediction Based DF-DD

There have been a number of inner decoders designed to fit the iterative decoding structure outlined in the previous sub-sections. However, these decoders, based on either MDD [17, 18] or NSD [31], are both memory intensive and computationally expensive. This unpleasant result has occurred from attempting to minimize BER without considering the trade-off between performance and complexity. It was stated earlier in this thesis that a goal for this work is to develop a scheme that would be simple yet very effective in an iterative decoding process. In doing so, an inner decoder was found in work by Schober and Gerstacker [20], that was able to provide the right performance at a reasonable level of complexity. This decoder, designed to decode differential encoding by creating a better estimate of the previous received symbol, is based on a linear prediction model. Whereby, $(N-1)$ previous symbols and an interim decision for the previously decoded data symbol

are used to compute a better estimate for the previously received symbol used for the differential detection of the upcoming data symbol.

In the work presented by Schober and Gerstacker [20], the decision-feedback differential detector (DF-DD) was observed to provide a significant improvement as compared to conventional differential detection for both Rayleigh frequency-flat fading and Ricean frequency-flat fading channels. It was also proposed that this prediction based DF-DD would be particularly well suited for a mobile wireless link as the predictor coefficients could be updated using the powerful recursive-least mean squares (RLS) algorithm, as the number of coefficients required is relatively small. In addition, the adaptation could start blind (meaning that no training sequence would be required) and no *a priori* knowledge of the channel statistics is required. In this work, the DF-DD was simulated without implementing the RLS algorithm and the channel statistics were assumed to be known as those features were not relevant to the purpose of the work. But what made this decoder so attractive was its simplicity and the simplifications it induced on the rest of the system.

For the sake of simplifying the discussion, it is best that the linear prediction DF-DD is presented within the following simplified system model, shown in Figure 2.6. In the following sub-section the DF-DD is presented within the iterative decoding scheme, including the adaptations and simplifications to both the DF-DD and the overall system. In this system model it is clear that only differential encoding and the complex fading channel are relevant to the discussion. Accordingly, the input and output of the system are assumed to be *M*-PSK symbols.

Figure 2.6: Simplified system

In reference to Figure 2.6, it is *clear* that the differentially encoded symbols are given by

$$b_n = a_n b_{n-1} \quad 5.$$

Even *more clear* is that the received symbols can be given by

$$r_n = a_n e^{j\phi_c} f_n b_{n-1} + w_n \quad 6.$$

And what is *plainly obvious* is that an estimate of the decoded symbol can be given by

$$\hat{a}_n = \argmax_{a_n} \left| - \{ Re \{ \tilde{a}_n r_{n-1} r_n^* \} \} \right| \quad 7.$$

What may *not be so clear* is that the estimate of the data symbol is dependent on the previously received symbol and the current symbol, as given in the above equation. Yet, it was evident to Schober and Gerstacker (among others), that it was not necessary to use a MDD structure make the estimate of the reference received symbol more reliable. Thus by developing a linear prediction model to mimic the MDD structure they found it would be possible to improve the overall performance, as compared to conventional differential detection, without the excessive number of computations and requirement for memory. In Figure 2.7, shown at the top of the next page, is the result of their idea. The solution is simple and very effective.

Figure 2.7: Linear Prediction Based Decision-Feedback Differential Detector [20].

From the above diagram, we see that the improved reference symbol is a weighted summation of $(N-1)$ previously received symbols. This summation is given in the following

$$\hat{r}_{n-1} = \sum_{v=1}^{N-1} r_{n-v} p_v \prod_{u=1}^{v-1} \hat{a}_{n-u} \quad 8.$$

Let us closely examine the intricacies of this design, especially those concerning the coefficients p_v^a and why the new estimated reference symbol is better. To begin with, in the case of optimum coherent detection (CD), the reference symbol $f_n b_{n-1}$ must be known; and in conventional DD, r_{n-1} is used to estimate $f_n b_{n-1}$ [11]. Since r_{n-1} is noisy, the noncoherent system will experience an inevitable performance degradation compared to the CD system. Using the linear prediction DF-DD system a more reliable estimate of reference sym-

bol, $f_n b_{n-1}$, is computed using the last $N-1$ observed symbols. These symbols are combined to form an estimate of r_{n-1} as defined by equation (9). The coefficients in that equation are obtained by minimizing the mean-square error (MSE) between $f_n b_{n-1}$ and r_{n-1} . The exact derivation can be found in the original work, [20].

However, an interesting result of solving for the optimum coefficients, p_v , is that the estimate of the reference symbol is dependent on the last $N-2$ estimates of the transmitted data symbols a_{n-v} , $1 < v < N-2$. In other words, the estimate of the reference symbol, $r_{e,n-1}$, requires that previously decoded M -PSK symbols be fed back to adjust the phase reference of the previous $N-2$ received symbols.

In the results presented in the next chapter, we also consider an artificial system where the actual transmitted symbols are fed back, instead of the interim decisions based on the output of the DF-DD. This system is referred to as “genie-aided” DF-DD.

Also from Figure 2.7 it is evident that the decision variable for the prediction based DF-DD is

$$g_n = r_n \hat{r}_{n-1}^* \quad 9.$$

In the remainder of this work we refer to g_n as the soft output of the DF-DD as the sequence $\mathbf{IL}^{-1}[\mathbf{g}]$ is used as the input the outer decoder. The feedback symbol, α_n , is an interim decision on the estimated data symbol α_n ; and is given by which M -PSK constellation symbol g_n is closest to on the complex plane. Recall that for M -PSK the complex plane is divided into M sectors that correspond to the M possible values for a_n . Therefore, the interim estimate α_n is given by observing which sector g_n falls within. In the next sub-

section we see that replacements for α_n can be supplied from the OD, so that this decoder can be incorporated into the iterative scheme described in sub-section 2.3.1.

As a final detail, it should be noted that if $N=2$, the prediction based DF-DD reduces to conventional DD, with $p_1=1.0$, on a flat-fading channel. Also, in the original work it was demonstrated that a DF-DD with $N=5$ is sufficient for practical use and any greater values of N do not provide enough added performance to justify the rise in complexity. Thus, in this work a DF-DD with $N=5$ was used in all the simulations. And, as previously mentioned it was assumed that the channel statistics were known *a priori*, such that the predictor coefficients could be calculated beforehand.

2.3.3 How DF-DD Simplifies the entire decoder

Figure 2.8: Soft/Hard-Symbol joint iterative demodulator and decoder

In Figure 2.8, the new soft/hard-symbol joint iterative demodulation and decoding structure is displayed in all of its glory. Upon first inspection it appears to be similar to the original iterative decoding structure presented in section 2.2; but there are substantial differences that make this structure far more simple than the structure presented previously.

The first significant difference is that instead of providing probabilities the OD delivers *hard* M-PSK symbols that are used to replace the interim decisions of the inner decoder. Thus the memory requirements for the interleaver in the feedback path are far less than that of the original decoding structure. After interleaving, the absolute M-PSK symbols are used as the interim hard decisions, α_n , within the linear prediction DF-DD. The idea driving this is that these hard symbols should be more accurate than the interim hard decisions computed on the first iteration by the DF-DD itself or the interleaved interim decisions of previous iterations by the outer decoder.

Second, the inner decoder (DF-DD) can now pass only the deinterleaved sequence $\mathbf{g}$, to the OD. Similar to the interleaver, the memory requirements for the deinterleaver have been greatly reduced; furthermore, the requirement to calculate the APP's within the inner decoder has been removed, greatly reducing computational expense. Thus, the new joint iterative demodulation and decoding structure is simple. During the first iteration, for processing a received frame, the soft-symbol sequence $\mathbf{g}$ is used to provide the hard decisions, α_n , used as the interim hard decisions for the DF-DD. The soft-symbol sequence $\mathbf{g}$, is then deinterleaved and used in the OD to calculate branch metrics required to decode the received code word. After which, the OD provides a new updated version of the estimated code word that is interleaved and used as the interim hard decisions within the linear prediction DF-DD. The updated, error corrected, sequence should be more reliable, thus reducing the effects of error propagation through the DF-DD, inducing it to increase its performance.

2.4 Reed-Solomon CODEC

In this final section of Chapter 2, Reed-Solomon (RS) encoding and decoding is briefly reviewed. Reed-Solomon codes are a class of nonbinary BCH codes, and in fact, the most practically important subclass of BCH codes. They have recently been proposed in the IEEE 802.16 specification for error control in wide area LAN applications. In this section the RS codes are only briefly introduced; and for a detailed description of these codes, their algebraic properties, Galois-Field (GF) arithmetic and decoding algorithms, the reader is referred to [30,34,35,36].

2.4.1 Description of Reed-Solomon Codes over $GF(p^m)$

Starting with any prime number p and any integer m , there exists a number of possible codes with symbols that originate in $GF(p^m)$. A (n, k) Reed-Solomon code, denoted here by $RS(n, k)$, is a cyclic code generated by a polynomial of degree $n - k$ with coefficients from $GF(p^m)$, which is a factor of $X^n - 1$. After the Galois field has been chosen the rest of the codes parameters can be determined by selecting the error correcting capability of the code, t . Thus, a t -error correcting $RS(n, k)$ code over $GF(p^m)$ has parameters specified in Table 2.2.

TABLE 2.2 Parameters for a t -error correcting $RS(n, k)$ code over $GF(p^m)$

Block Length:	$n = p^m - 1$
Number of parity symbols:	$n - k = 2t$
Minimum Distance:	$d_{min} = 2t + 1$

From these parameters it is observed that the length of the code (the number of code symbols), is one less than the size of the code alphabet and the minimum distance is one greater than the number of parity-check symbols.

More specifically, for this work we are concerned with $GF(2^m)$. The reason for this is that binary information is easily mapped to coefficients within Galois Fields with two as the prime number defining the field. Further, by setting α to be the primitive element in $GF(2^m)$ the generator polynomial of a t -error correcting $RS(n,k)$ code can be defined by the following equation

$$g(X) = g_0 + g_1X + g_2X^2 + \dots + g_{2t-1}X^{2t-1} + X^{2t} \quad 10.$$

Clearly (of course!), $g(X)$ has roots $\{\alpha, \alpha^2, \dots, \alpha^{2t}\}$, all from $GF(2^m)$ and the code generated by $g(X)$ is a cyclic code which consists of those polynomials of degree $n-1$ or less with coefficients from $GF(2^m)$ that are multiples of $g(X)$.

2.4.2 Encoding of a t -error correcting $RS(n,k)$ code over $GF(2^m)$

For a particular code the message to be encoded is a word of k q -tuples, where q is the number of bits required to represent 2^m-1 in binary arithmetic. Thus, the message word $a(X)$ can be represented as a polynomial of degree $k-1$ with coefficients from $GF(2^m)$. To conveniently decode this class of codes it is common to adjust the code word so that it takes on a systematic form. This is done by dividing the message polynomial $X^{2t}a(X)$ by the generator polynomial $g(X)$, resulting in $2t$ parity check symbols that are collectively the coefficients of a remainder polynomial $b(X)$. A block diagram of the encoding algo-

rithm is presented in Figure 2.11. Note that all additions and multiplications are done using $GF(2^m)$ arithmetic.

Figure 2.9: Block diagram for any nonbinary cyclic code for systematic encoding [34]

2.4.3 Decoding of a t -error correcting $RS(n,k)$ code over $GF(2^m)$

Decoding of a $RS(n,k)$ code is very simple as it follows four steps. These steps are: firstly, the syndrome calculation; secondly, finding the error-location polynomial, $\sigma(X)$; Third, calculation of the error location numbers of the error pattern, $e(X)$; And finally, determining the error pattern coefficients, the error values.

Upon receiving a corrupt transmitted code polynomial, $r(X)$, $2t$ syndromes $\{S_1, \dots, S_{2t}\}$ are calculated to establish the presence of detectable errors. The syndromes are calculated by evaluating $r(X)$ at $\{\alpha, \alpha^2, \dots, \alpha^{2t}\}$ inclusive, such that: $S_1 = r(\alpha)$, $S_2 = r(\alpha^2)$, ..., $S_{2t} = r(\alpha^{2t})$. If all the syndromes happen to be zero, then no detectable error can found by this code; mean-

ing that if an error did occur, the error pattern was such that it changed the original transmitted codewords into another legitimate codewords.

If any of the syndromes are non-zero then a detectable error has been found. The next step is to find the error-location polynomial, $\sigma(X)$, assuming that the errors can be corrected. The most common way to find $\sigma(X)$ is to use Berlekamp's iterative algorithm (not to be confused with the iterative algorithm used throughout the rest of this work). This algorithm is described in any reliable error-control textbook and the reader is referred to them for the details of the Berlekamp algorithm (see, e.g., [34]).

Once the error-location polynomial has been found, the error locations are easily found by calculating the roots of $\sigma(X)$. The reciprocals of the roots are the error-location numbers of the error pattern. In other words, the error-location numbers specify the non-zero coefficients of the error polynomial $e(X)$. Then by using the already calculated syndromes we can easily find the error values (coefficients) of $e(X)$, by first creating $Z(X)$

$$Z(X) = 1 + (S_1 + \sigma_1)X + \dots + (S_v + \sigma_1 S_{v-1} + \dots + \sigma_v)X^v \quad 11.$$

Using this polynomial the error value at location $\beta_l = \alpha^{jl}$ is calculated by

$$e_{jl} = \frac{Z(\beta_l^{-1})}{\prod_{i=1, i \neq l}^v (1 + \beta_i \beta_l^{-1})} \quad 12.$$

Detailed examples of this process can be obtained in [34] among other texts on the subject. And, decoding is completed by subtracting the error polynomial from the received word (i.e. $r(X) - e(X)$).

Chapter 3 - Simulation Results

In this chapter, the performance of the proposed simplified joint-iterative demodulation and decoding algorithm is evaluated through computer simulation. The simulation results characterize the performance of the noncoherent receiver on Rayleigh flat-fading channels for differentially encoded M -PSK signals.

To begin, the performance of single differential detection for differentially encoded symbols that have been convolutionally encoded is briefly illustrated to establish a frame of reference for the proceeding simulations. Next, the iterative demodulation and decoding scheme for the same convolutional code, a Reed-Solomon code, a concatenated Reed-Solomon/convolutional code is examined. Different scenarios and slight modifications to the system are simulated to evaluate the performance under conditions that vary according to channel severity. The results are used to discuss the validity of the proposed scheme, suggest possible changes to it, and illustrate its strengths and limitations.

The simulation results start with those for the convolutionally encoded data, in Section 3.1. In what follows, within Sections 3.2 and 3.3, the effect of a Reed-Solomon code used as and/or in part of the outer encoding scheme is presented. In Section 3.4 we compare the results for all the outer encoding schemes done in this work.

3.1 Convolutional codes

The purpose of this section is to validate the simulation framework, and to establish a comparison reference for the iterative demodulation and decoding of the other outer encoding schemes to be tested. These simulations use differentially encoded QPSK signals, with a rate-1/2 $K=4$ convolutional code with generator $(23,35)_8$, a block length that varies from 1000 to 10000 information bits, and a non-uniform pseudo-random interleaver. A block diagram of this system is shown below in Figure 3.1.

Figure 3.1: System Model incorporating R-1/2 Convolutional Code $(23,35)_8$

The fading process is generated using Jakes' model to simulate Rayleigh frequency flat-fading with a maximum BT of 0.05, where, as described in Chapter 2, BT is a measure of the rate at which the fading process changes in time. The demodulation is first done by using single differential detection followed by the deinterleaver and a soft decision Viterbi decoder for the convolutional code without iterating to establish a benchmark to compare the performance of the system when using the decision-feedback differential detector (DF-DD) upon iteratively processing the received frames. After establishing the benchmark, set by the single differential detector, the DF-DD then takes the place of the inner decoder (demodulator) and the loop is closed. Upon closing the loop, simulation results for iterative demodulation and decoding are presented, for Rayleigh flat-fading channels.

Figure 3.2: Comparison of DF-DD demodulation to DD and “genie-aided” DF-DD demodulation. $BT = 0.01$, $N_a = 1000$

Figure 3.2 shows the performance of the DF-DD ($N=5$) based receiver with only the straight-through performance of frames (no iteration), at a fading rate of $BT=0.01$, with a frame size of $N_a=1000$. As a benchmark, the performance of a single differential detector based receiver is also shown. Upon inspection, it is evident that the DF-DD receiver actually performs worse than the single differential detector. However, if we re-simulate the same system with only correct feedback symbols (genie aided) the DF-DD outperforms the single differential detection scheme. *Thus by iteratively processing the frames we would hope to get closer to an ideal feedback situation, where the feedback symbols are more or less all correct.*

Before moving on, it is instructive to elaborate on the above paragraph. The curve obtained for the DF-DD system after the first iteration, without genie-aided feedback was an unexpected result. In [20] the DF-DD outperformed single differential detection; how-

ever, in [20] only differentially encode QPSK modulation was used without any type of error control coding. Thus, the results presented there were for much higher SNR's. At the low SNR's desired for this work the output of the DF-DD appears to be very error prone. We suggest that the discrepancy is due to the hard decision feedback the DF-DD is based on and its effects are pronounced when error control is added. It is important to stress that the output used to make the interim decisions, within the DF-DD, is not reliable since no error control is used to aid in their calculation.

To illustrate the error propagation, we consider a *genie-aided* version of the same system; whereby only correct decisions are artificially fed back to the DF-DD. Again, referring to Figure 3.2, we can see that this has a profound effect and the performance is much better.

So the question at this point was: *How could the system be changed so that the DF-DD would approach its genie-aided performance bound when used in conjunction with error control, so that it could be used at lower SNR's than in [20]*? The simple answer to this question is to provide the DF-DD with feedback symbols that have been subjected to error control; As we hope these will be more reliable than the direct output of the DF-DD, the error propagation should be reduced. Thus iterative processing seemed to be the obvious choice. And the loop was closed to examine the results of this idea.

In the results shown in Figure 3.3a, we see that by iteratively decoding we improve the performance of a DF-DD based system subjected to error control. The results suggest that there is a threshold at which this method begins to work. That threshold, for slowly fading channels appears to be at an SNR of approximately 5dB, and the performance bound is reached within 2dB of that threshold.

Figure 3.3a: Rayleigh flat fading, $BT = 0.01$, $N_a = 1000$

Figure 3.3b: Rayleigh flat fading, $BT = 0.05$, $N_a = 1000$

Figure 3.3c: Rayleigh flat fading, $BT = 0.00125$, $N_a = 1000$

Figure 3.4: Effect of different BT products on performance, after 4 iterations.

In the above figure the effect of an increasingly BT product presents itself in the form of a degradation in system performance. As the Doppler Spread of the channel increases, as quantified by the BT product, we see that the systems performance after 4 iterations decrease. What this means in real terms, is that the faster the channel fading rate the worse the performance of this system will be. This is not a great epiphany and was expected.

The reason this happens is due to the reliance of the DF-DD on interim hard decisions on the differentially detected QPSK symbols. During the first iteration the DF-DD creates its own feedback [hard] symbols based on a soft symbols generated within the DF-DD, which are what is used to drive the soft input Viterbi decoder. However this entire work is based on the fact that on subsequent iterations the feedback [hard] symbols come from re-encoding the output of the outer decoder.

As an explanation for the performance degradation, it is proposed here that there is a problem with this idea at faster fading rates. The first, with respect to the convolutional code, is that errors from decoding the convolutional code, upon re-encoding, will cause error propagation through to a number of symbols following the error, in addition to the fact that error from this type of code happen in groups of 4 to 5 bits at a time. Thus, a number of symbols will be incorrect and propagate errors on subsequent iterations.

Also, from observing Figure 3.3, we may conclude that as the channel becomes more severe, in terms of faster fading, joint-iterative demodulation and decoding is not required for this particular system; In fact, it would be more reasonable to interpret these results as a performance bound specifying that this system is limited to slowly fading channel rates.

The linear prediction model, initially used to design the DF-DD, is a poor idea under fast fading conditions for the following reason: If the channel is changing quickly, assuming that the reference symbol can be better estimated by using more previous symbols is a bad assumption, because previous symbols, farther in the past, will have less of an effect on the current reference symbol than the symbols, in the immediate past. And what may actually happen is that the symbols in the distant past will actually adversely effect the updated reference symbol leading to the propagation of more errors. Thus in such situations the DF-DD based receiver is probably not the best option and its performance will actually suffer from iterative processing as seen in Figure 3.3b.

Figure 3.5: Effect of word size on performance, $BT=0.01$ (4 iterations).

Also of interest to this work is the effect of changing the word length (or frame length). In Figure 3.9 the results of that inquisition is shown. It is clear that, at a BT of 0.01, the a word length on the order of 1000 message bits is required for acceptable performance.

The explanation for this is closely linked to the fading rate of the channel. In an channel that is considered to be fast fading deeper fades are short lived (on average) but occur frequently. On the other side of the spectrum, in slowly fading channels deeper fades are far less frequent but when they do occur they last for long periods of time. As such, transmitted frames of shorter message-word lengths are more susceptible to being completely lost in a slowly fading channels if the fade is severe, than frames derived from longer message word lengths. As an example, to illustrate this idea, if we allow the typical fading duration to be approximately 100 symbols. This would mean that half a frame or more would be lost as the frame length is 200 symbols, and decoding that frame may not work so well. If the frame length is say 1000, then only 10% of the symbols would be lost in a deep fade;

and after demodulation and deinterleaving it is still probable that the code could fix these errors and the message word would be error free.

Also in this section, we openly wonder about the effect of using single differential detection in place of the decision-feedback differential detector for the first iteration the received frame is processed. The motivation for wanting to do this comes from observing in the previous simulation results that the single differential detection system outperforms the DF-DD based system on its first attempt at demodulation and decoding of the received frame. The revised system to be considered is presented in the block diagram below.

Figure 3.6: Updated, simplified system model

Despite this, we can see from Figure 3.7 that using single differential detection for the first iteration of processing the frame does help lower BER faster, but not necessarily farther. Using single differential detection the BER converges to its lowest value after 3 iterations while the system incorporating the DF-DD, for the first iteration, requires 4 iterations to reach its lowest value. In comparison with results presented previously in this chapter the final BER is not lower, after this change, than it would be without it. However, the benefit comes from the fact single differential detection is less computationally expensive than is the decision-feedback differential detector and one less iteration is needed.

Figure 3.7a: Comparison of Single Differential Detection versus DFDD based demodulation for the first iteration.
 $BT=0.01$, $N_a = 1000$.

Figure 3.7b: Comparison of Hybrid Differential Detection versus DFDD based demodulation after three iterations.
 $BT=0.01$, $N_a = 1000$.

3.1.1 N: Smarter or Bigger

Before moving along, we examine the effect of changing the size (length) of the decision feedback differential detector. Thus far the results presented have been for a DF-DD of length N equal to 5. Here we examine the actual performance after five iterations (using the rate-1/2 convolutional code for the outer code), for values of N ranging from 2 to 10. As well the BT product is fixed at 0.01 and the frame size is 1000.

Figure 3.8: Comparison of different lengths of the DF-DD ($N=2,4,5,10$) for the fifth iteration of the DF-DD based receiver. $BT = 0.01$, $N_a = 1000$.

As hoped, from the figure above, we see that increases the length of the DF-DD the performance of the fifth iteration gets better. Yes, it is true that as the length increases the performance is also better. However, as mentioned previously a DF-DD with a length over five become difficult to handle with respect to adjusting their tap weights to the channel statis-

tics. And the performance from going from five to ten does not appear large enough to justify the added complexity. One may also be lead to believe that only the last five symbols really effect the performance when using the DF-DD, which would be consistent with the findings of [20].

3.2 Reed Solomon Code

In this section we examine the results of using a RS(15,9) Reed-Solomon code over GF(16), as the outer code. Effectively the code has a rate of about 1/2. This was chosen deliberately so that we could directly compare it with the rate-1/2 convolution code that was examined earlier.

In Figure 3.9 and 3.10 it can be seen that the results for this code through the Rayleigh flat-fading channel are not very good. Initially the first iteration of the system using the DFDD outperforms the system using only single differential detection. This may be happening due to the hard decisions on the received symbols, from the DFDD or straight differential detector, that are required before the Reed-Solomon code can be undone and the information retrieved. In which case this would be consistent with the original finding presented in [13]. The step of making the hard decisions may limit the effect of the propagation of errors felt by the soft input decoder system, for the convolutional code. Please note that the performance of the soft input decoder is better despite its initial penalty.

What is disappointing, is that there is no significant performance gain upon closing the loop and iterating. It simply did not happen. Possibly because the distribution of errors in the received frames did not lend itself the error correcting capability of this family of codes, which requires the bit errors to be closely grouped within a few symbols of the

codewords in each frame. Certainly increasing the size of the message word did not remedy this situation either as Figure 3.10 illustrates when compared to Figure 3.9

Figure 3.9: Differentially encoded RS(15,9) code over a Rayleigh flat-fading channel. $BT = 0.01$, $N_a = 1008$

Figure 3.10: Differentially encoded RS(15,9) code over a Rayleigh flat-fading channel. $BT = 0.01$, $N_a = 10080$

3.3 Convolutional Code Combined with the Reed-Solomon Code

From the concluding remarks of the previous section, one may question what would happen if there was a way to adjust the distribution of errors with Reed-Solomon codewords. This could be done by concatenating the output of the Reed-Solomon encoder with another code, say a convolutional code. Thus for the simulations presented in this section a Reed-Solomon encoder was combined with the convolutional encoder of Section 3.1, such that the Reed-Solomon codewords are convolutionally encoded. This effectively decreased the rate of the outer code to about a quarter. But what happened was interesting.

Despite paying a SNR penalty for a decrease in the rate of the concatenated code, there is benefit to the iterative processing gain of doing so, as compared to a Reed-Solomon code alone. In Figure 3.11 it can be seen that the first iteration of the DFDD system does not do as well as the single differential detection system. But on the second and subsequent iterations the DFDD system more than makes up for it.

However, this improvement is limited to larger message word lengths as can be extrapolated from Figure 3.11, where we readily see the effect different word lengths have on this scheme. Furthermore, these results are not that good when compared to a convolutional code alone, as this code greatly increases the complexity while halving the transmission rate, all while providing worse performance.

Improvements to this system would likely be found in the use of higher density convolutional codes and/or higher density Reed-Solomon codes. The higher density constituent codes would raise the effective rate of the outer encoding scheme thus compensating for

the penalty incurred by the unreasonable overhead resulting from the approximate quarter rate in these simulations.

Figure 3.11: Combined effect of $(23,35)_8$ Convolutional code and RS(15,9), Reed-Solomon code. $BT = 0.01$, $N_a = 1008$.

Figure 3.12: Effect of word length on performance of the system using the $(23,35)_8$ Convolutional code and RS(15,9), Reed-Solomon code.

3.4 A Final look at the Results for the Outer Codes

Figure 3.13: Comparison of the rate-1/2 Convolutional Code with generator $(23,35)_8$, RS(15,9) Reed-Solomon Code, and combination of the two after five iterations within the DFDD based receiver. $BT = 0.01$, $N_a = 1000, 1008$.

In the above figure we see the final results for the different outer codes compared with one another. Upon first glance it would seem that the convolutional code provides the best performance. However, one must remember that the combination of the Reed-Solomon code and convolutional code have an effective rate of approximately a quarter, while the other two codes have an effective rate a half (more-or-less). This is important to note because the combination code (RS + CC) will inherently suffer due to the larger amount of overhead required to be transmitted per information symbol. So the result for this code is actually quite good and it shows that the convolutional code does re-distribute the errors in the received frame so that the Reed-Solomon decoder can better deal with them (i.e. the errors exist within the necessary error correcting capability of the RS(15,9) code).

3.5 Notes on the Simulation Environment

There are a number of factors that effect the validity, or rather statistical certainty, of the simulation results. All those factors collectively imply that the simulations actually simulate the wireless environment in its entirety. This means that sufficiently large sets of data are collected and processed so that the statistics drawn from that data accurately represents the systems being analysed. In this work the statistic of primary interest, as was seen throughout this chapter, was the bit-error-rate (BER); and as such the simulations were done so that the curves generated for the BER's were statistically accurate.

This accuracy was achieved by counting a minimum number of independent bit errors while keeping in mind the state of the channel. In this work, in order to ensure that the bit errors were truly independent we instead counted a minimum number of incorrect frames. Thus whether the frame had one bit error or was completely wrong we marked the frame completely wrong and any bit errors within that frame were not considered independent. Effectively we transferred our specification on the minimum number of bit errors to be equal to that of the number of frame errors we were counting.

Thus, if we wanted to simulate something that had a BER on the order of 10^{-6} given the pre-encoded frame length was 1000 bits and 100 independent bit errors represents a sufficiently large sample of data from which to extract statistically certain statistics; then, approximately 100000 frames of data should to be transmitted collect enough information to be processed. Further, as a precaution simulations are repeated with different seeds to initialize the pseudo-random number generator within the simulation environment. Therefore, in total about 200000 to 300000 frames of data should be simulated.

Chapter 4 - Conclusions and Future Work

4.1 Summary

In this work the problem of designing a reliable, yet simple, noncoherent receiver for flat-fading channels was investigated. The solution that was found was based on the iterative decoding schemes found in [2, 12, 19] and the simple DF-DD structure found in [20].

Thus by adopting the ideas intrinsic to the fundamental iterative decoding algorithm, for serially concatenated codes, in this work, we were able to successfully transfer its most basic concepts to the design of a noncoherent receiver that was shown to combine performance with simplicity. Upon doing so, we conveniently arrived at a simplified noncoherent iterative demodulation and decoding scheme for M -PSK signals over Rayleigh flat-fading channels. The performance bound of the proposed receiver was also demonstrated by way of simulation using genie-aided feedback.

Furthermore, the design proposed is far less complex than other solutions to the design problem. The requirements for large amounts of memory and computations power have been greatly relaxed as the feedback required in the system eventually depends only on hard decisions on the received M -PSK symbols.

Additionally it was also shown that the system proposed could actually be further simplified by not using the demodulator proposed in [20] for the first iteration of processing the received frame; and that the first iteration could be done by simply using single differential detection to start the noncoherent iterative demodulation and decoding scheme.

4.2 Simulation Results

The performance of the proposed receiver was evaluated through computer simulations, using a number of different error control coding schemes. The results show that it is possible feasible to design a simple noncoherent receiver using computationally inexpensive error control and hard decision feedback. Particularly, the following results have been observed:

- The noncoherent iterative demodulation and decoding scheme outperforms its non iterative counterpart for the simulated scenarios.
- The performance of the system is limited by genie-aided feedback to the decision-feedback differential detector (DF-DD).
- The performance of the iterative receiver proposed does approach the ideal performance limit in simulation.
- It is possible to further reduce the complexity of the iterative scheme by using single differential detection for the first iteration of the iterative process.
- By using single differential detection the number of iterations required to reach the performance bound is also reduced.
- The performance of the receiver, irrespective of the error control method employed, is suited for slower fading conditions ($<1\%$).
- For faster fading rates ($>5\%$) the DF-DD linear prediction model is inherently flawed.
- For a little added computational complexity convolutionally encoding Reed-Solomon code words outperforms schemes that only employ Reed-Solomon error control.

- Increasing the length of the DF-DD beyond $N=5$ added little performance to the receiver.
- A message word length of approximately 1000 symbols appeared sufficient to achieve an acceptable level of performance.
- Beyond a word length of 1000 symbols did not add significant improvements to the complexity for the error control schemes employed.

4.3 Suggestions for Future Work

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

- In Chapter 3 it was observed that convolutionally encoding Reed-Solomon code words improved system performance over using a Reed-Solomon code alone. However, the system had a penalty incurred due to the low density of concatenating the two codes used in this work. It would be interesting to see the results of changing the rate of the codes to acquire a higher effective code rate.
- Also in this work we ignored the effect of the interleaver mapping. It may be instructive to learn the consequences of changing the interleaver mapping from the pseudo-random mapping to say something like an S-random mapping [24].
- M -PSK, specifically QPSK, was used as the modulation scheme. Changing this to some other modulation scheme, say M -Ary QAM, could provide further insight into this very interesting field.
- Finally, adapting this work to different environments (e.g. Ricean fading) could be done.

References

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo Codes," in *Proceedings of the IEEE International Conference on Communications (ICC'93)*, Geneva, May 1993, pp. 1064-1070.
- [2] I.D. Marsland, "Iterative Noncoherent Detection of Differentially Encoded M-PSK", Ph. D. Dissertation, The University of British Columbia, June 1999.
- [3] M. Peleg, S. Shamai (Shitz), "Iterative decoding of coded and interleaved noncoherent multiple symbol detected DQPSK," *IEE Electronic Letters*, vol. 33, no. 12, pp. 1018-1020.
- [4] M. Peleg, S. Shamai (Shitz), and S. Galan, "On iterative decoding for coded noncoherent MPSK communications over block-noncoherent AWGN channel," *International Conference on Telecommunications (ICT'98)*, pp. 109-114.
- [5] J. Lodge and M. Gertzman, "Joint detection and decoding by turbo processing for fading channel communications," *Proceedings of the IEEE Global Conference on Communications (GLOBECOM'98)*, Sydney, Nov. 1998.
- [6] I.D. Marsland, P.T. Mathiopoulos, and S. Kallel, "Noncoherent turbo-equalization for frequency selective Rayleigh fast fading channels," *Proceedings of the International Symposium on Turbo Codes*, pp. 196-199.
- [7] A.S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes", *IEE Electronics Letters*, vol. 30, no. 25, pp.2107-2108, Dec. 1994.
- [8] P. Robertson, "Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes," in *Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC'94)*, San Diego, Sept. 1994, pp. 1298-1303.
- [9] S. Benedetto and G. Montorsi, "Design guidelines of parallel concatenated convolutional codes," *Proceedings of the IEEE Global Conference on Communications (GLOBECOM'95)*, Singapore, Nov. 1995, pp. 2273-2277.
- [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] I.D. Marsland, P.T. Mathiopoulos, "On the Performance of Iterative Noncoherent Detection of Coded M-PSK Signal", *IEEE Transactions on Communications*, vol. 48, no. 4, pp. 588-596, April 2000.
- [12] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, no. 13, pp. 1186-1188, June 1996.
- [13] J.G. Proakis, *Digital Communications* - 3rd Edition, New York: McGraw-Hill, 1995.
- [14] T.S. Rappaport, *Wireless Communications: Principles and Practice*, New Jersey: Prentice Hall, 1996.

- [15] Meyr, Moeneclaey, and Fechtel, *Digital Communication Receivers Synchronization, Channel Estimation, and Signal Processing*, Wiley Interscience, 1998.
- [16] Ziemer and Peterson, *Digital Communications and Spread Spectrum Systems*, MacMillan, 1985.
- [17] D. Makarakis, P.T. Mathiopoulos, "Optimal decoding in Fading Channel: A combined envelope, multiple differential and coherent approach," *Proceedings of IEEE Global Conference on Communications (Globecom '89)*, Dallas, Nov. 1989, pp1551-1557.
- [18] D.Divsalar, M.K. Simon, "Multiple-symbol differential detection of MPSK," *IEEE Transactions on Communications*, vol. 38, pp. 300-308, Marcg 1990.
- [19] C. Dubuc, "Joint Iterative Demodulation and Decoding for Turbo-Coded PSK Signals on Flat Fading Channels", *Master of Engineering Dissertation*, Carleton University, July 2000.
- [20] R. Schober, W.H. Gerstacker, "Decision-Feedback Differential Detection Based on Linear Prediction for MDPSK Signal Transmitted over Ricean Fading Channels", *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 391-402, March 2000.
- [21] E.K. Hall, S. G. Wilson, "Design and performance analysis of turbo codes on Rayleigh fading channels," *Proceedings of the Conference on Information Sciences and Systems (CISS'96)*, Princeton, New Jersey, Mar. 1996, pp. 43-48.
- [22] 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 Communications*, vol. 16, no. 2, pp. 265-275, Feb. 1998.
- [23] M.C. Valenti, B.D. Woerner, "Refined Channel Estimation for Coherent Detection of Turbo Codes over Flat-Fading Channels", *IEE Electronis Letters*, vol 34, no. 17, pp. 1648-1649, Aug. 1998.
- [24] E.K. Hall, S.G. Wilson, "Turbo Codes for Noncoherent Channels", *Globecom '97*, 1997.
- [25] D. Divsalar, F. Pollara, "Multiple Turbo Codes for Deep-Space Communications", *JPL, TDA Progress Report*, pp. 42-121, May 1995.
- [26] 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
- [27] C. Heefard, S.B. Wicker, *Turbo Coding*, Boston: Kluwer Academic Publishers, 1999.
- [28] W.C. Jakes, *Microwave Mobile Communications*, Wiley, New York, 1974.
- [29] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communications Systems", *IEEE Communications Magazine*, pp. 90-109, July 1997.
- [30] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, New Jersey, Jan. 1995.

- [31] G. Colavolpe, G. Ferrar, and R. Raheli, "Noncoherent iterative (turbo) decoding," *IEEE Transactions on Communications*, to be published.
- [32] C.E. Shannon, "Communication in the Presence of Noise", *Proc. of the I.R.E.*, vol. 37, pp. 10-21, Jan. 1949.
- [33] M. Peleg, S. Galan, and S. Shamai (Shitz), "On iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel," *CC Pub. #249*, Dept. of Electrical Engineering, Technion, Haifa, Isreal, June 1998.
- [34] Lin, Costello, *Error Control Coding: Fundamental and Applications*, Prentice Hall, New Jersey, 1983.
- [35] Lidl, Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, New York, August 1994.
- [36] S.B. Wicker, V.K. Bhargava, *Reed-Solomon codes and their applications*, New York, IEEE, 1994