TURBO EQUALISATION BASED ON DATA DIRECTED ESTIMATION FOR STANDARD HIGH FREQUENCY WAVEFORMS

by

Louis Milne van der Westhuizen

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

August 2021

SUMMARY

TURBO EQUALISATION BASED ON DATA DIRECTED ESTIMATION FOR STANDARD HIGH FREQUENCY WAVEFORMS

by

Louis Milne van der Westhuizen

Supervisor:	Prof. Warren P. du Plessis
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Electronic Engineering)
Keywords:	Equalizers, turbo equalization, data-directed estimation (DDE), MIL-STD-188-110D (Appendix D).

Conventional receivers for high-frequency (HF) waveforms using phase-shift keying (PSK) or quadrature amplitude modulation (QAM) symbol mapping contain an equaliser to equalise channel effects and a forward error correction (FEC) decoder to improve bit error rate (BER) performance at low signal-to-noise ratios (SNRs). The equaliser is an important component of an HF waveform receiver since the ionospheric channel used for skywave propagation commonly causes a signal to propagate along multiple paths which often results in inter-symbol interference (ISI) at the receiver. Turbo equalisation, which is the process of feeding back information from the decoder to the equaliser to improve symbol estimates, has been shown to improve the performance of a receiver when ISI is present in the channel. Turbo data directed estimation (TDDE) is a turbo equalisation method based on the data directed estimation (DDE) equaliser which is suited for waveforms that interleave known symbol sections, used for channel estimation, with the unknown data symbols.

In this work, the TDDE algorithm is applied to standardised MIL-STD-188-110D Appendix D (MIL-STD-188-110D App. D) HF waveforms. This standard was selected since it is still relevant and actively being maintained and updated. The 3 kHz bandwidth variants of the low data rate waveforms in MIL-STD-188-110D App. D, referred to as waveform number 1 and waveform number 2, were studied

in this work since it was expected that it would be more challenging to improve the performance of these waveforms than the less robust and higher data rate waveforms with denser constellations. The designed TDDE receiver was tested using the methods specified in MIL-STD-188-110D App. D and using equipment from a reliable supplier of HF communications equipment, Rapid Mobile (Pty) Ltd (RapidM). The tests showed promising results for the TDDE algorithm when applied to MIL-STD-188-110D App. D waveform number 1 and waveform number 2, with consistent performance improvements in the order of 2 dB when using a Poor channel as defined in MIL-STD-188-110D App. D. Permutations of the TDDE algorithm were also experimented with in this work, but none provided substantial changes in performance.

ACKNOWLEDGEMENTS

I would like to thank all of my family members, friends and colleagues who at some time during the course of this work provided moral support, motivation or technical guidance, without which this work would have been much more challenging.

I would also like to give special thanks to the following people.

Stephan Isebeck: The CEO of RapidM, for his technical guidance, for loaning the equipment without which this project would have not been possible and for RapidM's financial support with my studies.

Prof. Warren du Plessis: My supervisor, for his enthusiasm and assistance during the course of this work and especially during the compilation of this dissertation.

Machiel Posthumus: My colleague, for his valuable insights during the course of this work and especially for going through the effort of proofreading this dissertation and contributing to its accuracy and quality.

LIST OF ABBREVIATIONS

2G	the second generation of HF technology
3G	the third generation of HF technology
4G	the fourth generation of HF technology
8-PSK	eight-ary phase-shift keying
ALE	automatic link establishment
ARM	advanced RISC machines
ARQ	automatic repeat request
AWGN	additive white Gaussian noise
BER	bit error rate
bit/s	bits per second
BLOS	byond line of sight
BPSK	binary phase-shift keying
COM	communication
CSV	comma-separated values
CW	continuous wave
DDE	data directed estimation
DFE	decision feedback equaliser
DoD	Department of Defence
DRC	data rate change
DSP	digital signal processor
FEC	forward error correction
FSK	frequency-shift keying
HF	high-frequency
IP	internet protocol
ISB	independent sideband
ISI	inter-symbol interference
ITU	International Telecommunication Union
LDDE	linear data directed estimation
LE	linear equaliser
LLR	log-likelihood ratio
LOS	line of sight

LUF	lowest usable frequency
MAP	maximum a-posteriori
MIL-STD	Military Standard
MIL-STD-188-110D App. D	MIL-STD-188-110D Appendix D
MLSE	maximum-likelihood sequence estimation
MMSE	minimum mean-square error
MSE	mean square error
MUF	maximum usable frequency
N/A	not applicable
NATO	North Atlantic Treaty Organization
NDDE	nonlinear data directed estimation
PC	personal computer
PDF	probability density function
PSK	phase-shift keying
QAM	quadrature amplitude modulation
QPSK	quadrature phase-shift keying
RAP1	RapidM application protocol 1
RapidM	Rapid Mobile (Pty) Ltd
SNR	signal-to-noise ratio
SOVA	soft output Viterbi algorithm
SSB	single sideband
STANAG	NATO Standardisation Agreement
TCP	transmission control protocol
TDDE	turbo data directed estimation
TDDE-N	TDDE with an NDDE as the first pass equaliser
TDDE-L	TDDE with an LDDE as the first pass equaliser
TNDDE	turbo non-linear data directed estimation
TNDDE-N	TNDDE with an NDDE as the first pass equaliser
TNDDE-L	TNDDE with an LDDE as the first pass equaliser
UART	universal asynchronous receiver-transmitter
USA	United States of America

VHF	very high-frequency
WALE	wideband automatic link establishment
XOR	exclusive OR

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 CHAPTER OVERVIEW

The use of turbo equalisation for standard high-frequency (HF) waveforms was investigated and an overview of the motivation for the investigation as well as the results achieved are provided in this chapter. This work was motivated by the continued relevance of HF communications and the challenges with using HF ionospheric channels, specifically the multipath effect and resulting inter-symbol interference (ISI). It was hypothesised that a turbo equalisation technique based on the data directed estimation (DDE) equalisation technique would be well suited for HF ionospheric channels which do not only vary with time, and therefore need to be dynamically estimated, but are also subjected to ISI. The focus was placed on channel equalisation and the channel estimation component was not updated with the improved equalisation results. As a consequence, bit error rate (BER) performance results obtained in this work are more likely pessimistic than optimistic. Tests were performed with MIL-STD-188-110D Appendix D (MIL-STD-188-110D App. D) waveforms which are relevant to the newer generations of military HF communications. The results obtained confirm the hypothesis and show that the turbo data directed estimation (TDDE) technique is well suited even for robust standard HF waveforms in ionospheric channel conditions.

1.2 PROBLEM STATEMENT

The MIL-STD-188-110D App. D standard was revised in 2017 and the number of matured radio products available that can support the wider bandwidths of up to 48 kHz are still very limited. It is expected that these waveforms will remain relevant at least for the near future, and likely even beyond that. With the development of new generations of HF waveform and data link layer standards, it is also important to understand the limits of existing designs and address these limitations with future iterations. To accurately quantify these limitations, modern techniques, such as turbo equalisation, must be applied to existing waveforms in order to produce results that give a true representation of

what can be achieved with the existing waveforms.

1.2.1 Context of the problem

The HF band is set between 3 and 30 MHz in the electromagnetic spectrum. HF technology was the main long range communication option during much of the 20th century. With the emergence of satellite communications in the 1960s, HF technology appeared obsolete which led to a decline in financial investment and technical progress for this technology [1]. It soon became clear that satellite communications had their own limitations. For satellite communications, line of sight to the satellite is required, which makes it less suitable for certain terrain and for indoor use. Satellites are also relatively vulnerable in terms of security and require high installation and maintenance costs for both the satellite and ground stations. As a result HF communications have remained relevant due to their ability to allow communications over long distances with little available infrastructure. Modern uses of HF technology include tactical military communications, disaster relief scenarios where little to no infrastructure is available, communications at latitudes where satellite communications are not suitable and as a backup for satellite systems [1, 2]. Institutions such as the United States of America (USA) Department of Defence (DoD) and North Atlantic Treaty Organization (NATO) continue to produce standards for HF communications. The USA DoD publishes these standards as Military Standards (MIL-STDs) while NATO produces NATO Standardisation Agreements (STANAGs). As part of a new generation of HF waveforms, MIL-STD-188-110D App. D was released in 2017 and supports bandwidths of 3 kHz to 48 kHz with data rates varying from 75 bits per second (bit/s) to 240 000 bit/s [3]. This allows for these waveforms to be used in a wide range of applications with requirements ranging from limited bandwidth to high throughput and robustness.

HF communications technology makes use of skywave propagation to achieve long distance communication with relatively low power and limited infrastructure. For skywave propagation an HF transmission is directed in the general direction of the destination but at an elevated angle. The transmitted signal is gradually refracted by electrons in a layer around Earth known as the ionosphere until it is directed back towards Earth where the receiver is stationed [2]. The signal appears to reflect off the ionosphere. In some cases the signal can be refracted by the ionosphere to Earth's surface and reflected from Earth's surface back to the ionosphere multiple times, thereby completing multiple hops between Earth and the ionosphere before the receiver is reached.

The ionosphere provides a significant advantage for HF technology over other wireless communication

methods, but also introduces a number of challenges. A significant challenge is the time varying nature of the ionosphere. Ionospheric conditions vary with time at various periods ranging from the 11-year sunspot cycle and seasonal changes to changes depending on the time of day due to radiation of the Sun [1]. Due to the uneven multi-layer structure of the ionosphere it is also possible that a transmitted signal can arrive at the receiver along multiple paths with varying propagations delays. It is common for the multipath delay to be larger than the symbol period which causes ISI [4]. Finer time variations in the ionosphere also cause Doppler spread. All these effects are superimposed on other noise also present in the channel due to losses of internal components as well as noise from man made and natural external factors.

The equaliser is a function which counteracts the effects of ISI at the receiver thereby equalising the channel. Various equalisation methods have been suggested relying on techniques such as maximum-likelihood sequence estimation (MLSE) and filters constructed using minimum mean-square error (MMSE) approximations [5]. Receivers often also include forward error correction (FEC) functions to reduce the probability of bit errors. Turbo equalisation has been suggested as a technique to improve ISI cancellation by feeding information back from the FEC function to the equaliser [6]. Turbo equalisation techniques have shown great promise, not only for HF communication systems, but also for other forms of wireless communications.

1.2.2 Research gap

The effects of various turbo equalisation methods on receivers for standardised HF waveforms such as the MIL-STD-188-110B waveforms and the STANAG 4539 waveforms have been studied and have been shown to result in significant improvements [7–11]. The focus in these studies has been mainly on waveforms with data rates of 3 200 bit/s to 9 600 bit/s and which are similar to the medium to high data rate 3 kHz waveforms in MIL-STD-188-110D App. D. Some of the sources have indicated that the gains become less significant as the data rates decrease and the robustness of the waveforms increase [9].

Improvement of the performance of the lower data rate waveforms could allow for reliable communication in conditions where it was previously not feasible. It is expected that methods which allow significant improvements at the robust low data rate waveforms should allow even greater improvements at higher data rates and ultimately decrease the signal-to-noise ratio (SNR) requirement for a specified throughput at a specified bandwidth. Some MIL-STD-188-110D App. D waveforms make

use of repeats of encoded data bits. Limited information was found on how the combining of the repeated data should be handled in an iterative equalisation scheme.

Alternative forms of turbo equalisation have recently been suggested, such as expectation propagation based turbo equalisation and TDDE [12, 13]. Applying a variety of these techniques to standard waveforms such as the MIL-STD-188-110D App. D waveforms could allow for a fair comparison between the various techniques and could improve the understanding of the channel conditions in which the various waveforms can operate reliably. This can be relevant for the development of future generations of HF waveforms as it will allow for a fair comparison between existing components of the standards and possible new developments.

1.3 RESEARCH OBJECTIVE AND QUESTIONS

The primary objective is to investigate whether a significant performance improvement can be achieved by applying turbo equalisation to robust MIL-STD-188-110D App. D waveforms. It has been shown that the medium to high data rate 3 kHz waveforms in MIL-STD-188-110D App. D can be improved through the application of decision feedback and MMSE based turbo equalisers [7–11]. No information was found regarding the possible performance improvement that can be expected from iterative equalisation with robust waveforms.

The secondary objective is to determine whether the TDDE equalisation technique can provide a significant improvement over the DDE equalisation technique for ionospheric channels. The DDE equalisation technique where known probe symbols interleaved with unknown data symbols are exploited to improve channel estimation and equalisation, has been shown to be effective for time varying ionospheric channels [14]. It is expected that the TDDE technique which is based on the DDE technique should also be well suited for ionospheric channel conditions, but no information was found available on the performance comparison between the DDE equaliser and the TDDE equaliser.

A third objective is to determine a sensible way to handle encoder repeats in the context of log-likelihood ratios (LLRs) which are used as part of turbo equalisation. While it seems like a problem that should be trivial, little information has been found on how repeated encoded bits should be combined at the receiver when using soft bit weights or LLRs. Waveforms standards such as MIL-STD-188-110D App. D make use of encoder repeats to be able to achieve more robust waveforms without having to support a variety of encoders and decoders [3].

This work will aim to answer the research questions provided below.

- How significant is the performance gain when applying turbo equalisation to robust, low data rate waveforms?
- What is the impact of the first pass equalisation algorithm and the subsequent accuracy of the information fed back from the decoder on the performance of a turbo equaliser?
- How can encoder repeats be handled in a turbo equalisation scheme?
- How will various interleaver sizes affect the performance of a turbo equaliser?

1.4 APPROACH

This work aims to show that the TDDE equaliser is a suitable iterative equalisation algorithm for standardised HF waveforms, such as the waveforms defined in MIL-STD-188-110D App. D, in ionospheric conditions. To prove this hypothesis, a suitable iterative equalisation based receiver is designed by considering various options for receiver components such as the equaliser and the decoder. The receiver is designed for the 3 kHz bandwidth waveform number 1 and waveform number 2 in MIL-STD-188-110D App. D. These waveforms were selected as they are the most robust waveforms in the standard that do not make use of Walsh orthogonal modulation and are expected to prove a challenge to further improve. The waveforms that make use of Walsh orthogonal modulation (waveform number 0) were not considered as these waveforms make use of a different waveform structure in the data phase than all other waveforms in MIL-STD-188-110D App. D. A solution which is well suited for the Walsh orthogonal modulation waveforms might not be suitable for the higher data rate waveforms in MIL-STD-188-110D App. D.

Adaptive channel estimation forms an important part of equalisation in time varying ionospheric channels since the performance of an equaliser greatly depends on the accuracy of the channel response [8, 14]. The scope of this work is limited by excluding channel estimation and relying on an external source for the channel response. While this allows for a fair comparison of the various equaliser iterations, since all iterations will make use of exactly the same channel data, it does not take into account the performance improvement associated with feeding back more accurate information to the channel estimation component and is therefore expected limit the performance gains that can be achieved.

MIL-STD-188-110D App. D specifies the channel conditions and procedure that must be followed to

test the performance of HF data modems implementing the MIL-STD-188-110D App. D waveforms [3]. The Rapid Mobile (Pty) Ltd (RapidM) RS8 HF & V/UHF Channel Simulator is used to reliably simulate HF channel conditions at baseband, as specified by the International Telecommunication Union (ITU) Ionospheric Channel Simulator standard for testing HF modems [15]. Two RapidM TC5 Wideband HF & VHF Modem and ALE Controller Modules are used as the transmitter and receiver respectively to ensure reliable functionality of waveform components outside the scope of this work, such as transmitter components, receiver time and frequency synchronisation and channel estimation.

1.5 RESEARCH GOALS

The goal of this work is to demonstrate that iterative equalisation and decoding can provide significant performance gains even for very robust MIL-STD-188-110D App. D binary phase-shift keying (BPSK) waveforms in ionospheric channel conditions through standard, reliable and reproducible test procedures. This sets a clear benchmark of what can be achieved and also allows future studies to make fair comparisons to this work and build on it.

This work also aims to demonstrate the effectiveness of the TDDE equalisation algorithm when compared to the DDE equalisation algorithms, since no existing data is available to provide a one-to-one comparison. This is important as it will demonstrate whether the TDDE algorithm is worth considering and also allows the performance of the TDDE to be compared to similar algorithms such as the MMSE equaliser with a-priori information.

1.6 RESEARCH CONTRIBUTION

In this work, standardised BER performance tests are used to prove that a turbo equalisation based receiver can provide significant performance improvements for the very robust BPSK MIL-STD-188-110D App. D waveforms in ionospheric channel conditions. The designed receiver is suitable for the 3 kHz MIL-STD-188-110D App. D waveform number 1 and waveform number 2, where a greater than 2 dB performance improvement when compared to the linear data directed estimation (LDDE) algorithm was noted, and can be easily expanded to other waveforms in MIL-STD-188-110D App. D.

The TDDE algorithm, used to achieve the performance improvements, is shown to provide significant gains over the DDE algorithms after only 1 iteration. Subsequent iterations showed limited improvement in most of the tests. Multiple methods for extracting extrinsic information on the feedback

path of a turbo equalisation based receiver with encoder repeats are also considered. The various methods are shown to result in very similar BER performance. An alternative DDE based iterative equalisation algorithm, referred to as the turbo non-linear data directed estimation (TNDDE) algorithm, is suggested, but test results show that this method appears to degrade the BER performance in spite of increased computational complexity.

1.7 RESEARCH OUTPUTS

This work will lead to the submission of a journal publication with the following provisional details.

Title: Turbo equalisation based on data directed estimation for standard military high frequency waveforms

Journal: IEEE Transactions on Wireless Communications

Abstract: The effects of data directed estimation based turbo equalisation on low data rate narrowband MIL-STD-188-110D App. D waveforms in ionospheric channel conditions were investigated. The BER performance of the turbo equalisation based receiver was measured in accordance with the guidelines set out in MIL-STD-188-110D App. D for the additive white Gaussian noise (AWGN) and Poor channel conditions specified in the standard. Data directed estimation based turbo equalisation was found to provide significant performance improvements in HF channel conditions, even for the robust low data rate BPSK waveforms in MIL-STD-188-110D App. D.

1.8 OVERVIEW OF STUDY

An outline of the chapters that make up this work is provided below.

Chapter 1: This is an introductory chapter which aims to place this work into context and detail the objectives and goals of this work as well as the research contributions made by the work.

Chapter 2: This chapter provides background information on HF technology and common challenges with using the HF ionospheric channel. The standardisation of HF modem testing is also discussed.

Chapter 3: The chapter provides background information and discusses the history of military HF standards with specific focus on MIL-STD-188-110D App. D. Some of the transmitter characteristics of the MIL-STD-188-110D App. D waveforms are also discussed, which are relevant to later sections.

Chapter 4: This chapter provides background information on the various receiver components relevant to an MIL-STD-188-110D App. D receiver. For most of the receiver components, a number of implementation options are considered and discussed. The components are first considered as part of a conventional receiver, where equalisation and decoding are performed separately, before discussing components relevant to turbo equalisation.

Chapter 5: This chapter presents a design for an MIL-STD-188-110D App. D turbo equalisation receiver. A design for a conventional receiver is first discussed since this forms part of the first receiver pass of the turbo equaliser. The iterative receiver processing functions are then discussed. Some alternative designs are also considered with varying first pass equalisation algorithms, feedback extrinsic information extraction methods and turbo equalisation algorithms.

Chapter 6: This chapter discusses the procedure followed to reliably test the turbo equaliser designed in Chapter 5. The equipment used during the tests are detailed and the hardware configuration of the tests are discussed. A step by step procedure is then provided, detailing the execution of the tests.

Chapter 7: This chapter details the results achieved by executing the test procedure described in Chapter 6 for various configurations. Results achieved with only the RapidM equipment are first provided to establish confidence in the test configuration and procedure. The results for tests using the designed turbo equaliser are then provided and observations are made based on the results.

Chapter 8: This chapter aims to highlight the significance of the results shown in Chapter 7. The results achieved with AWGN channel conditions are first discussed to give confidence in the test system. The next sections provide comparisons between the results achieved with variations in the turbo equalisation based receiver in the presence of MIL-STD-188-110D App. D Poor channel conditions.

Chapter 9: This chapter provides the concluding remarks for this work and also suggests future work and research that could possibly build on the contributions made by this work.

CHAPTER 2 HIGH FREQUENCY COMMUNICATION

2.1 CHAPTER OVERVIEW

This chapter serves as a brief overview of some aspects of HF communications. Despite the emergence of satellite communications technology in the 1960s, HF technology continues to remain relevant in the 21st century since it can be used with limited infrastructure and is sometimes usable in areas where satellite communications are not suitable. The ionospheric channel, which gives HF technology some of its unique advantages, is also the cause of unwanted channel effects such as ISI and fading, all superimposed on Gaussian noise. These factors along with the fact that ionospheric channels vary widely with geographical position and time greatly complicate the testing of HF communications equipment. For this reason the ITU has standardised HF modem baseband channel simulators which can be used to test the performance of HF modems under various channel conditions.

Section 2.2 discusses the continued relevance of HF communications technology. The propagation of HF radio waves through the ionosphere is discussed in Section 2.3. Section 2.4 discusses the standardisation of HF modem test equipment.

2.2 RELEVANCE OF HIGH FREQUENCY COMMUNICATIONS TECHNOLOGY

For the first half of the 20th century HF communications was the preferred method for achieving long range connectivity. With the emergence of satellite communications technology in the 1960s HF technology had appeared obsolete and much of the focus shifted away from this technology [1, 8]. It later became clear that satellite communications had its own limitations and that there were many use cases where HF technology was a more viable alternative, or could be used as a fallback for satellite communications.

The limitations of satellite communications include the vulnerability of ground stations, weak indoor reception due to the requirement for line of sight (LOS) to the satellite and also security vulnerabilities. The initial investment costs as well as the maintenance costs for satellites are also high when compared to HF alternatives [1]. Additional motivation for the use of HF technology includes interoperability with legacy communication systems and the fact that HF communications can serve as a viable alternative when satellite communications fail or are not available [2].

The result is that HF communications technology is still actively being used and develop especially for tactical military communications, naval communications and as a communications method during disaster relief scenarios. The Hurricane Katrina disaster in 2005 is an example of an event that highlighted the importance of HF technology in the modern world [1].

2.3 HIGH FREQUENCY PROPAGATION

The HF band is the range of frequencies between 3 and 30 MHz on the electromagnetic spectrum. The three common propagation methods for electromagnetic waves in Earth's atmosphere are space wave propagation, surface wave propagation and skywave propagation as illustrated in Figure 2.1. With space wave propagation waves propagate directly from the transmitter to the receiver through free space. This method is commonly used at higher frequencies and requires LOS from the transmitter to the receiver. With surface wave propagation the electromagnetic waves propagate along Earth's surface and are able to reach receivers that are byond line of sight (BLOS). This propagation method is only suitable for lower frequencies. Skywave propagation relies on the refraction of HF waves by electrons in a layer around Earth known as the ionosphere to achieve communications over greater distances.

2.3.1 The ionosphere

The ionosphere is a layer in Earth's atmosphere where atoms are ionised by solar radiation. The ionosphere is located between 60 and 1 000 km above Earth's surface and consists of three layers, named the D, E and F layers, each characterised by different ionisation processes [4]. The F layer, which can be divided into the F1 and F2 layers, is commonly used for HF ionospheric or skywave propagation. The maximum usable frequency (MUF) as defined by the ITU is typically determined by the F layer, while the D layer determines the lowest usable frequency (LUF) [16]. The conditions in the ionosphere are strongly dependent on the radiation from the sun and can vary widely with geographical position, time of day, seasonal changes and the 11-year sunspot cycle.

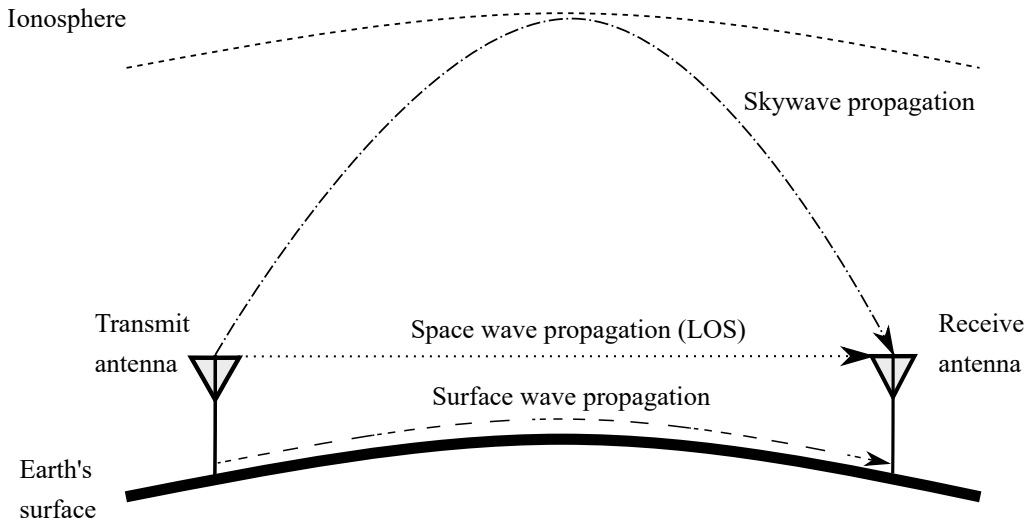

Figure 2.1. Different propagation methods for electromagnetic radio waves in Earth's atmosphere.

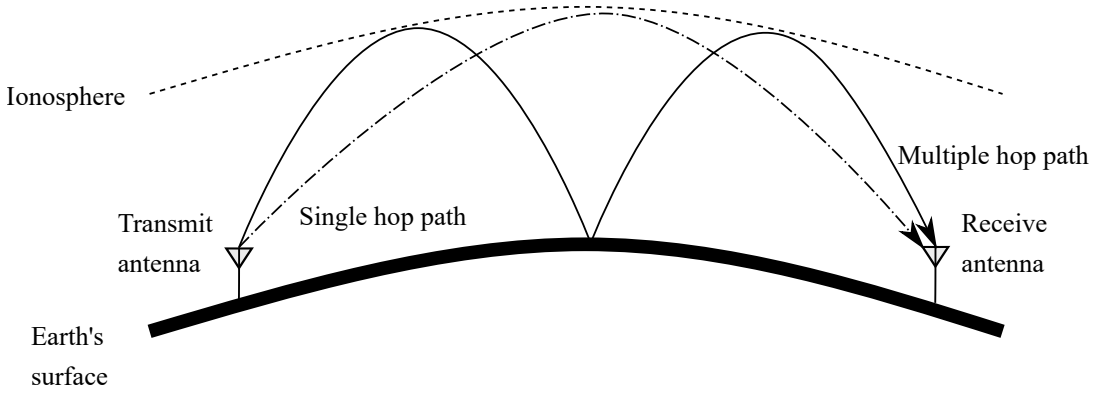

Figure 2.2. Skywave propagation of electromagnetic radio waves reaching a destination in single or multiple ionospheric hops.

2.3.2 Skywave propagation

Figure 2.2 shows how two HF communications devices can communicate using a transmit antenna and a receive antenna that are BLOS from one another. The signal is launched at an oblique angle from the the transmit antenna to the ionosphere. The transmitted signal is refracted by electrons in the the ionosphere back down to Earth's surface. In the case of the single hop path, as indicated in Figure 2.2, the transmit signal reaches the receive antenna after refracting from the ionosphere once. For the multiple hop path, as indicated in Figure 2.2, the transmitted signal is reflected back from Earth's surface to the ionosphere and again from the ionosphere to Earth's surface, thereby completing two hops before reaching the receive antenna.

HF skywave propagation provides HF technology with its main advantage over other wireless communication methods since it allows for communication over large distances with relatively low power and without significant infrastructure. However, the ionospheric channel used for HF skywave propagation is one of the most challenging wireless communication channels to communicate over and also presents some of the main challenges for HF technology.

2.3.2.1 Skip zones

Both the frequency and the incidence angle affect the ability of a transmission to pass through the ionosphere and into free space without being refracted back to earth, with higher frequency transmission and transmissions with incidence angles closer to 90° passing through the ionosphere more easily. The MUF is the maximum frequency at which ionospheric propagation is still possible with reasonable performance for a specified configuration. The critical frequency is the minimum frequency at which a transmission with a 90° incidence angle will pass through the ionosphere [4]. For a frequency above the critical frequency there is a maximum incidence angle at which a transmission will not pass through the ionosphere. The distance which a skywave transmission will travel with a single hop also decreases with an increase in the incidence angle. The result is that, with frequencies above the critical frequency, there is an area close to the transmitter where a transmission cannot reach since an increase in the incidence angle would result in the transmission passing through the ionosphere. This area is known as the skip zone.

2.3.2.2 Variations in the ionosphere

Ionospheric channel conditions vary with time as well as geographical position. Time variations can occur over periods of various lengths where the slower variations are caused by the 11-year sunspot cycle, seasonal changes and the time of day while faster changes can occur due to ionospheric motion and turbulence [17]. Due to the uncertain nature of the ionosphere it is often necessary to dynamically estimate the channel conditions at the receiver and counteract these conditions as far possible.

2.3.2.3 Multipath interference

Multipath interference occurs when the transmit signal arrives at the receiver through more than one propagation path [15]. The various paths are often associated with different propagation delays meaning that each path could arrive at the receiver at a different instance in time. This causes interference at the receiver. The multipath spread refers to the difference between the time of arrival of the first path that arrives at receiver and the time of arrival of the last path that arrives at the receiver. When the symbol period of a waveform is shorter than the multipath spread echoes of previously transmitted symbols will be superimposed upon the current transmission causing ISI [5]. Multipath interference

and fading is represented by the Watterson model and can be simulated using a Watterson channel simulator [17].

2.3.2.4 Doppler shift and Doppler spread

The ionosphere is constantly moving and changing causing variations in perceived height and ionisation density. This movement can introduce frequency shifts due to the Doppler effect, known as Doppler shifts, in signals refracted by the ionosphere [4]. Due to the varying and unstable nature of the ionosphere it is very likely that different transmit paths will be subjected to different Doppler shifts which is known as differential Doppler. This together with other changes and inconsistencies in the ionosphere can result in a spectral broadening of the received signal relative to the transmitted signal. The widening of the spectrum received at the receiver is referred to as the Doppler spread of the channel and results in independent fading of the multipath components.

2.3.2.5 Gaussian noise

In addition to the channel effects described in previous sections there are also various internal and external factors contributing to the noise in an ionospheric channel [4]. Internal noise consists of all noise caused by components internal to the communications systems and include antenna circuit losses as well as losses in the transmitter and receiver. External noise is the result of factors outside of the radio communication system and include cosmic background noise, losses due to absorption by gasses in the atmosphere and man made noise. These noise sources are often collectivity modelled as additive Gaussian noise superimposed on the other ionospheric channel effects.

2.4 STANDARDISATION OF HIGH FREQUENCY MODEM BASEBAND CHANNEL SIMULATORS

Ionospheric channel simulators which can be used to test HF modems have been standardised by the Radiocommunication sector of the ITU [15]. ITU-R F.1487 recommends a Gaussian scatter HF channel model, also referred to as the Watterson model, for simulating ionospheric conditions at baseband. This model consists of a tapped delay line with a tap for each individual multipath component. Each individual tap is subjected to amplitude and phase modulation to introduce effects such as Rayleigh fading as well as Doppler shifts. Finally Gaussian noise is added to the modulated taps. ITU-R F.1487 specifies the configuration of the Watterson model for the simulation of channels at Low, Mid-, and High latitudes and also for quiet, moderate and disturbed conditions at each latitude.

2.5 CONCLUSION

HF communications technology has not lost its relevance in the modern world and with its unique use cases it is likely to remain relevant for the foreseeable future. To continue to exploit this technology and to expand on its uses, there are various channel conditions related to the ionosphere that need to be overcome. The verification of newly developed HF communications equipment can be challenging due to the uncertain and varying nature of the ionosphere. ITU-R F.1487 standardises a number of HF channel conditions which can be used to qualify HF modem equipment. Also important for the continued improvement of HF technology is the development of new generations of waveform as well as higher layer standards which can be qualified with the recommendations of ITU-R F.1487 and which allow for HF ionospheric channels to achieve data rates and provide services that were not possible in the past.

CHAPTER 3 HIGH FREQUENCY WAVEFORM STANDARDISATION

3.1 CHAPTER OVERVIEW

HF communications technology is widely used in military applications and has been evolving to better accommodate and overcome challenges with the ionospheric channel. The first generations of HF communications technology required highly skilled operators. Newer generations make use of automatic link establishment (ALE) and are even able to select the operation bandwidth with the use of some cognitive radio concepts. Automatic repeat request (ARQ) protocols allow for reliable data delivery and can dynamically update the throughput based on channel conditions when used together with data rate change (DRC) protocols. The development of these protocols have ensured that HF communications technology is suitable for a wider variety of applications. Along with the higher layer protocols, waveforms must be developed to service these protocols and to allow the protocols in turn to provide a range of new services. MIL-STD-188-110D App. D contains a collection of traffic waveforms that are suitable for a variety of conditions. Most of the MIL-STD-188-110D App. D waveforms make use of FEC encoders, interleave the data before transmission as a defence against burst errors, make use of phase-shift keying (PSK) or quadrature amplitude modulation (QAM) symbol mapping and multiplex unknown data symbols with known symbols which can be used to train the receiver,

The history of the standardisation of military HF technology is discussed in Section 3.2. Section 3.3 provides details on the MIL-STD-188-110D App. D waveform standard.

3.2 HISTORY OF HIGH FREQUENCY COMMUNICATIONS STANDARDISATION

The first generation of HF radio technology required highly skilled operators and mostly made use of single sideband (SSB), continuous wave (CW) and frequency-shift keying (FSK) modulation

techniques [1]. Development of ALE techniques started in the late 1970s with the intention of simplifying the use of HF communication equipment by eliminating the need for highly trained operators [2].

ALE was seen as part of the second generation of HF technology (2G) and was standardised in MIL-STD-188-141A Appendix A. Examples of physical layer standards associated with 2G are documented in MIL-STD-188-110A, STANAG 4285 and STANAG 4539 while the associated data link layer is documented in STANAG 5066 [1]. The waveforms defined in MIL-STD-188-110A make use of various PSK and QAM modulation techniques and also included a half rate convolutional encoder. Data rates of 75 bit/s to 2 400 bit/s are supported by the waveforms defined in MIL-STD-188-110A [1].

The third generation of HF technology (3G) was standardised in the late 1990s and early 2000s. 3G standards aimed to shorten link establishment times, increase link establishment robustness, improve throughput and support larger networks. 3G ALE was standardised in MIL-STD-188-141B and STANAG 4538 which also included two data link protocols designed for high throughout and robustness respectively [1]. The physical layer components associated with these protocols are also defined in STANAG 4538. MIL-STD-188-110B was released in 2000 and improved on MIL-STD-188-110A by supporting data rates up to 9 600 bit/s for SSB mode operation while also including a special two independent sideband (ISB) mode which could support data rates of up to 19 200 bit/s.

The introduction of the fourth generation of HF technology (4G) started in the early 2010s. HF communication has evolved to support a wide range of payloads including email, chat, file transfer, internet protocol (IP) and even video conferencing [2]. These applications all have specific requirements varying from high throughput to low latency. One way in which 4G aims to meet this new range of requirements is through the revelation of wideband HF waveforms which allow for much higher throughput. MIL-STD-188-110C was introduced in the early 2010s and improved on MIL-STD-188-110B by defining waveforms with bandwidths up to 24 kHz and data rates of up to 120 000 bit/s. MIL-STD-188-110D was introduced in 2017 and defined waveforms with bandwidths of up to 48 kHz and data rates of up to 240 000 bit/s.

Another way in which 4G aims to meet the new range of requirements by applying cognitive radio concepts to better utilise the available bandwidth. MIL-STD-188-141D was released in 2017 and contains the specification for 4G ALE protocols and waveforms. 4G ALE is also referred to as

wideband automatic link establishment (WALE) due to its support for wideband traffic waveforms, as defined in MIL-STD-188-110D. As an example of cognitive radio functionality, WALE is able to automatically detect interference on all available channels during scanning and to negotiate a suitable bandwidth within the available bandwidth when it has selected the the most ideal channel [18].

3.3 MIL-STD-188-110D APPENDIX D

MIL-STD-188-110D App. D describes a set of HF data modem waveforms with contiguous bandwidths of 3 to 48 kHz. For each bandwidth this appendix supports a set of waveforms varying in data rate and robustness. Each of the waveforms available for the bandwidth is identified with a waveform number, which is unique within the specific bandwidth [3]. For all bandwidths waveform number 0 is the most robust waveform with the slowest data rate and makes use of Walsh modulation. Waveforms one to thirteen make use of PSK and QAM modulation schemes, which will be the focus of this work.

Figure 3.1 shows the basic waveform structure for the PSK and QAM modulated waveforms described in MIL-STD-188-110D Appendix D. The first part of the waveform that is transmitted over the air is know as the synchronisation preamble. The synchronisation preamble allows the receiver to perform synchronisation as well as frequency and time alignment. A part of the synchronisation preamble also allows the receiver to detect the interleaver length, waveform number and encoder constraint length of the transmitted waveform. This is known as the autobaud functionality.

The data phase of the waveform is transmitted after the synchronisation phase. The data phase consists of the unknown user data multiplexed with known or probe sections as shown in Figure 3.1. The first as well as the last block in the data phase will always be probe sections so that each user data section is enveloped by a preceding and succeeding probe section. The probe sections consist of poly-phase code symbols and are used by the receiver to update the local channel estimate during reception [3]. The modulation for the user data sections depend on the waveform number of the active waveform. The number of symbols in the probe sections and user data sections respectively are determined by the bandwidth as well as the waveform number of the active waveform.

3.3.1 User data section

The processing functions that user data bits will pass through before being transmitted are shown in Figure 3.2. The stream of user data bits ($\mathbf{u}$) is first passed through a forward error correction encoder to produce a stream of coded bits ($\mathbf{c}$). The stream of coded bits are then repeated (producing $\mathbf{c^r}$) and

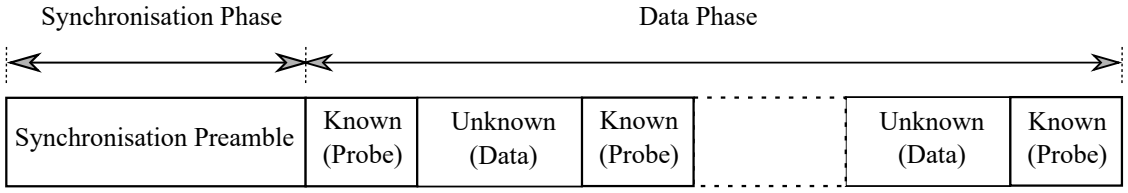

Figure 3.1. MIL-STD-188-110D Appendix D PSK/QAM waveform frame structure.

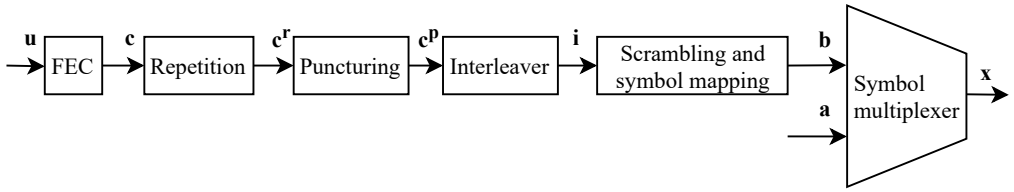

Figure 3.2. User data bit stream processing functions for MIL-STD-188-110D Appendix D PSK/QAM waveforms.

punctured (producing $\mathbf{c^p}$). The punctured bits are interleaved to produce a stream of interleaved bits (**i**). The interleaved bits are then converted to symbol numbers, scrambled and mapped to symbols in order to produce the stream of user data or unknown symbols (**b**). The user data symbols are finally multiplexed with the known probe symbols to produce the transmit symbol stream (**x**). Some of the transmit operations are described in more detail in the following sections.

3.3.1.1 Forward error correction encoding

The data received from the user are passed through a FEC encoder. MIL-STD-188-110D App. D uses a half rate convolutional encoder with a constraint length (k) of either seven or nine and full tail-biting [3]. For the constraint length seven encoder the generator polynomial is [3]

$$T_0 = x^6 + x^4 + x^3 + x + 1 \tag{3.1}$$

for the first bit (c_0) and

$$T_1 = x^6 + x^5 + x^4 + x^3 + 1 \tag{3.2}$$

for the second bit (c_1). The generator polynomials for the constraint length nine encoder are

$$T_0 = x^8 + x^6 + x^5 + x^4 + 1 \tag{3.3}$$

and

$$T_1 = x^8 + x^7 + x^6 + x^5 + x^3 + 1. \tag{3.4}$$

The half rate encoder with a constraint length of seven is shown in Figure 3.3. The encoder contains a shift register with seven elements numbered x^6 down to x^0. Each new user bit is shifted into the register

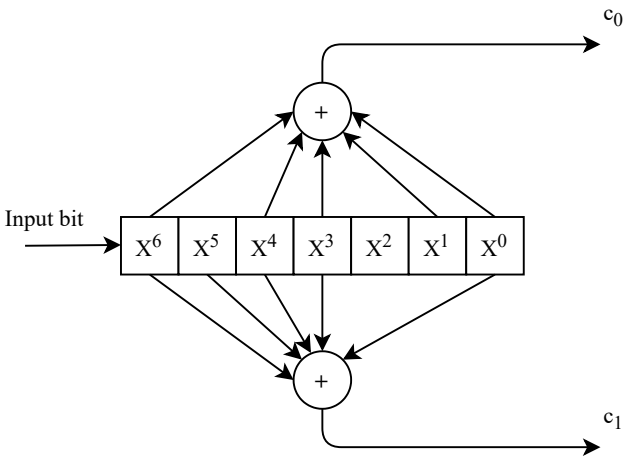

Figure 3.3. MIL-STD-188-110D Appendix D constraint length seven convolutional encoder. Taken from [3], © 2017 US DoD.

at x^6 and produces outputs c_0 and c_1 by performing a modulo-2 addition of some of the elements in the shift register.

Tail-biting is used to ensure that states of the encoder shift register before the first bit in the user data block is produced matches the states in the encoder shift register after the final user data bit has been produced. This allows the decoder at the receiver to concatenate multiple copies of the received data block in order to produce a larger data block. This allows for longer trace-back paths if a trellis based decoder is used, which increases the likelihood of convergence. The tail-biting process is described by the following steps:

1. Store the first $k-1$ user input bits.
2. Load the first $k-1$ user input bits into the encoder while discarding the output bits (c_0 and c_1) produced for each bit.
3. Proceed to load all the remaining user input bits that form part of the data block into the encoder while keeping the output bits (c_0 and c_1) produced for each bit. The first two output bits produced by the encoder is therefore produced after the kth bit has been loaded in.
4. After loading in the entire data block proceed to once again load in the first $k-1$ bits that have been stored previously into the encoder and keep the encoded bits produced.

3.3.1.2 Repetition and puncturing

Some of the waveforms in MIL-STD-188-110D App. D make use of repetition and puncturing of encoded bits in order to achieve different code rates with the same half rate encoder [3]. Repetition is relevant where a code rate lower that $1/2$ is desired, which will result in a more robust waveform with a lower data rate. When repetition is relevant encoded bits will be repeated a number of times in the output bit stream. If for example bits c_0 and c_1 are produced and a $1/4$ code rate is desired, the resulting bit sequence will be inserted into the output stream twice resulting in c_0, c_1, c_0, c_1. At the receiver all instances of c_0 corresponding to the same encoded bit will need to be combined to produce a single bit that can be input to the decoder.

Puncturing is relevant where a higher code rate than $1/2$ is desired which will result in a less robust waveform with a higher data rate. Puncturing is implemented by removing some of the bits produced by the encoder from the output bit stream based on a puncturing mask. At the receiver erasures are inserted at the positions in the received bit stream where the punctured bits would have been. These erasures indicate to the decoder that no information is available for the specific received bit.

The number of repetitions as well as the puncturing mask for the waveforms in MIL-STD-188-110D App. D vary with the waveform number as well as the bandwidth. If both repetitions and puncturing are relevant for the active waveform, encoded bits will first be repeated and then punctured. At the receiver the bits will be first de-punctured and then combined to produce an estimate of the original output.

3.3.1.3 Interleaving

After the encoded bits have been repeated and punctured, the results are loaded into a block interleaver. The purpose of the interleaver is to rearrange the encoded bits so that neighbouring bits are separated as far as possible from one another [3]. This protects the encoder from burst errors that can occur due to fading, and also improves statistical independence between the equaliser and the decoder at the receiver. For most of the waveforms in MIL-STD-188-110D App. D four interleaver length options are available which are referred to as, ranging from shortest to longest, Ultrashort, Short, Medium and Long. The exact size of each interleaver option for each waveform as well as other interleaver parameters depend on the waveform number as well as the bandwidth of the waveform.

3.3.1.4 Modulation and scrambling

The interleaved bits are converted to symbol numbers corresponding to the relevant modulation for the active waveform. Before mapping the symbol numbers to the relevant constellation points, the symbol numbers are scrambled to randomise the over the air transmission in the presence of long fixed input user data streams [3]. Waveforms that make use of BPSK, quadrature phase-shift keying (QPSK) or eight-ary phase-shift keying (8-PSK) modulation are scrambled to 8-PSK constellations by modulo-8 adding each transmit symbol number with an element in the scrambling sequence. Scrambling of QAM modulated waveforms is performed by performing the XOR operation between each bit in the current symbol and a bit in the scrambling sequence.

3.4 CONCLUSION

There has been significant progress since the first generation of HF communications technology in terms of simplifying its use and expanding its capabilities. Military organisations such as the USA DoD and NATO have continued to produce newer and improved standards for HF waveforms and higher layers of the HF technology communication stack. MIL-STD-188-110D App. D specifies a set of waveforms with varying bandwidths and data rates. Most of the waveforms in MIL-STD-188-110D App. D make use of a convolutional encoder, interleaved data, PSK or QAM symbol mapping and the multiplexing of known probe symbols with the unknown data symbols. The receivers for the MIL-STD-188-110D App. D waveforms are not standardised, but need to be designed to accommodate for all of the transmitter components and must also meet BER performance requirements specified in MIL-STD-188-110D App. D.

CHAPTER 4 PRINCIPLES OF HIGH FREQUENCY WAVEFORM RECEIVERS

4.1 CHAPTER OVERVIEW

Standard waveform receivers are typically not as explicitly defined as transmitters. To be standards compliant, a receiver must usually be able to extract the original data from a received stream that has been processed by the transmitter functions and meet a set of performance requirements. The result is that most of the functions that the receiver needs to perform are well defined, but the methods used to implement these functions are implementation choices. A standards compliant receiver for the MIL-STD-188-110D App. D PSK and QAM waveforms will require a channel equaliser to equalise the channel and meet the BER performance requirements for ionospheric channels, a symbol to bit mapping function to map received symbols to bits, a deinterleaver to invert the interleaver function and a decoder to extract the original data bits form the encoded stream. The performance of the receiver will be highly dependent on the selected channel equaliser and decoder functions.

Section 4.2 provides an overview of the MIL-STD-188-110D App. D receiver. Section 4.3 discusses the equaliser function and a few different equaliser options. The symbol to bit mapping function is discussed in Section 4.4 while Section 4.5 discusses the decoder and various decoder options. Iterative or turbo equalisation is discussed in Section 4.6.

4.2 THE MIL-STD-188-110D APPENDIX D RECEIVER

Figure 4.1 shows the functional blocks associated with the data phase of a the MIL-STD-188-110D App. D waveform receiver. After the transmit symbols stream ($\mathbf{x}$) passes through an channel with ISI a stream of symbols ($\mathbf{r}$) is received at the receiver. The received symbols are first passed through an equaliser to eliminate ISI and produce estimates of the original user data symbols ($\hat{\mathbf{b}}$). The other functional blocks in Figure 4.1 perform the reverse operations of functions in Figure 3.2.

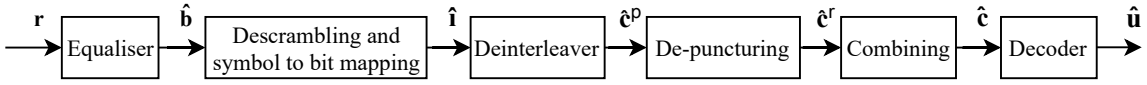

Figure 4.1. Data phase receiver functions for the MIL-STD-188-110D Appendix D PSK/QAM waveforms.

First the symbols are descrambled and mapped from symbols to bits to produce estimates of the original interleaved bits ($\hat{\mathbf{i}}$). The deinterleaver inverts the interleaver operation producing estimates of the punctured coded bits ($\hat{\mathbf{c}}^{\mathbf{p}}$). De-puncturing is performed by inserting erasures where bits have been punctured and the stream of repeated coded bit estimates ($\hat{\mathbf{c}}^{\mathbf{r}}$) is produced. The bits are then combined to produce $\hat{\mathbf{c}}$ before being decoded to produce the estimates of the original user data bits ($\hat{\mathbf{u}}$).

The following sections provide further detail on the more complex functional blocks in the receiver shown in Figure 4.1.

4.3 CHANNEL EQUALISERS

The baseband model for data communications systems is used to simplify descriptions throughout this section and the rest of this work. When a complex symbol has been transmitted over a multipath channel with ISI, the (r_k) symbol received at instance k can be defined by [19]

$$r_k = \sum_{l=0}^{L} h_l x_{k-l} + n_k \tag{4.1}$$

where h_l is the lth coefficient in the channel response, x_k is the symbol transmitted at time instance k, n_k is the additive Gaussian channel noise at time instance k and L is the channel length. The channel response ($\mathbf{h}$) can be defined by

$$\mathbf{h}^T = \begin{bmatrix} h_L & h_{L-1} & \dots & h_0 \end{bmatrix}. \tag{4.2}$$

A channel equaliser is used to recover the original signal from the received signal that has been distorted by the ISI channel.

4.3.1 Maximum likelihood sequence estimation

An MLSE based equaliser is an optimal channel equaliser which makes use of a trellis based approach to calculate the most likely sequence of transmitted symbols based on the received symbols [19]. The trellis can be solved using the Viterbi algorithm. The number of states at each stage in the trellis as well as the complexity of the MLSE scales with M^L, where M is the constellation size and L is the channel length [20]. As a result the MLSE becomes impractical for large M or L values, which is often

the case with ionospheric channels.

4.3.2 Linear equalisers

A linear equaliser (LE) attempts to eliminate the ISI by applying a linear filter at the receiver. A simple implementation of an LE is the zero-forcing equaliser. The zero-forcing equaliser derives its name from the fact that it attempts to force the ISI to zero by inverting the channel filter [20]. The disadvantage of the zero-forcing equaliser is that it does not take the AWGN (n_k in (4.1)) into account. The result is that the zero-forcing equaliser can significantly enhance the noise, especially when the channel frequency response contains spectral nulls [19]. In such a scenario the noise would have infinite gain at certain frequencies. Due to this characteristic zero-forcing equalisers are not well suited for ionospheric channels.

An alternative LE is the MMSE equaliser. The MMSE equaliser attempts to minimise the square error between the transmitted symbol at time instance k (b_k) and the estimated received symbol at instance k ($\hat{b_k}$) or [20]

$$MSE = E[b_k - \hat{b_k}]^2. \tag{4.3}$$

The result is that the MMSE equaliser takes both the ISI and the AWGN into account when calculating the optimal channel filter.

4.3.3 Data directed estimation techniques

Due to the time varying nature of HF ionospheric channels it is necessary to track and update the channel estimate during reception. Waveforms designed for HF ionospheric channels often contain known or probe sections for this purpose. DDE equalisation is a MMSE equaliser technique designed for waveforms that contain known sections. The information provided in this section is adapted from a paper by Hsu [14].

From (4.1) a stream of K received symbols can be presented as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{4.4}$$

where $\mathbf{H}$ is the channel convolutional matrix

$$\mathbf{H} = \underbrace{\begin{bmatrix} \mathbf{h}^T & & & \phi \\ & \mathbf{h}^T & & \\ & & \ddots & \\ \phi & & & \mathbf{h}^T \end{bmatrix}}_{K+L\times K}, \tag{4.5}$$

with ϕ indicating that all unspecified elements in the section of the matrix are set to 0, the received symbols ($\mathbf{r}$) are defined as

$$\mathbf{r}^T = \begin{bmatrix} r_0 & r_1 & \dots & r_{K+L-2} & r_{K+L-1} \end{bmatrix}, \tag{4.6}$$

the AWGN symbols ($\mathbf{n}$) are defined by

$$\mathbf{n}^T = \begin{bmatrix} n_0 & n_1 & \dots & n_{K+L-2} & n_{K+L-1} \end{bmatrix}, \tag{4.7}$$

and the original transmit symbols ($\mathbf{x}$) are defined by

$$\mathbf{x}^T = \begin{bmatrix} x_0 & x_1 & \dots & x_{K+L-2} & x_{K+L-1} \end{bmatrix}. \tag{4.8}$$

The known probe sections of length N can be expressed as

$$\mathbf{a}^T = \begin{bmatrix} a_0 & a_1 & \dots & a_{N-1} \end{bmatrix}, \tag{4.9}$$

while an unknown data block of length M can be expressed as

$$\mathbf{b}^T = \begin{bmatrix} b_0 & b_1 & \dots & b_{M-1} \end{bmatrix}. \tag{4.10}$$

By letting $\mathbf{x}$ in (4.4) consist of a single unknown block as well as the parts of the preceding and succeeding known probes that influence and are influenced by elements in the unknown block, the received symbols can be expressed as

$$\mathbf{r} = \begin{bmatrix} \mathbf{h}^T & & & \phi \\ & \mathbf{h}^T & & \\ & & \ddots & \\ \phi & & & \mathbf{h}^T \end{bmatrix} \begin{bmatrix} \mathbf{a_1} \\ \\ \mathbf{b} \\ \\ \mathbf{a_2} \end{bmatrix} + \mathbf{n}, \tag{4.11}$$

where the relevant section of the preceding probe is

$$\mathbf{a_1}^T = \begin{bmatrix} a_{N-L} & a_{N-L+1} & \dots & a_{N-1} \end{bmatrix}, \tag{4.12}$$

and the relevant section of the succeeding probe is

$$\mathbf{a_2}^T = \begin{bmatrix} a_0 & a_1 & \dots & a_{L-1} \end{bmatrix}. \tag{4.13}$$

By letting $\mathbf{M_1}$ be an $M+L$ row by M column matrix which contains the channel response elements that will be multiplied by elements in $\mathbf{b}$, letting $\mathbf{M_2}$ be an $M+L$ row by M column matrix containing the channel response elements that will be multiplied with elements in $\mathbf{a_1}$ and $\mathbf{M_3}$ be an $M+L$ row by M column matrix containing the channel response elements that will be multiplied with elements in $\mathbf{a_2}$, (4.11) can be rewritten as

$$\mathbf{r} = \mathbf{M_1 b} + \mathbf{M_2 a_1} + \mathbf{M_3 a_2} + \mathbf{n}, \tag{4.14}$$

where

$$\mathbf{M_1} = \underbrace{\begin{bmatrix} h_0 & & & & & \phi \\ h_1 & h_0 & & & & \\ \vdots & \vdots & \ddots & & & \\ h_L & h_{L-1} & \cdots & h_0 & & \\ & h_L & \cdots & h_1 & \ddots & \\ & & \ddots & \vdots & & h_0 \\ & & & h_L & & \vdots \\ & & & & \ddots & h_{L-1} \\ \phi & & & & & h_L \end{bmatrix}}_{M+L \times M}, \tag{4.15}$$

$$\mathbf{M_2} = \underbrace{\begin{bmatrix} h_L & h_{L-1} & \cdots & h_1 \\ & h_L & \cdots & h_2 \\ & & \ddots & \vdots \\ & & & h_L \\ \phi & & & \end{bmatrix}}_{M+L \times L}, \tag{4.16}$$

and

$$\mathbf{M_3} = \underbrace{\begin{bmatrix} & & & \phi \\ h_0 & & & \\ h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \\ h_{L-1} & h_{L-2} & \cdots & h_0 \end{bmatrix}}_{M+L\times L}, \tag{4.17}$$

This allows for the ISI due to the preceding and succeeding probes to be easily removed from the received symbols resulting in

$$\mathbf{c} = \mathbf{r} - \mathbf{M_2 a_1} - \mathbf{M_3 a_2} \tag{4.18}$$

$$= \mathbf{M_1 b} + \mathbf{n}. \tag{4.19}$$

The LDDE algorithm calculates the unknown data estimates ($\hat{\mathbf{b}}$) by using the MMSE criteria to minimise $E(|\mathbf{M_1}\hat{\mathbf{b}} - \mathbf{c}|^2)$ [14]. The mean square error (MSE) is minimised by

$$\hat{\mathbf{b}} = (\mathbf{R}^*_{\mathbf{M}})^{-1}\,\mathbf{M_1}^H\mathbf{c} \tag{4.20}$$

where $\mathbf{R}^*_{\mathbf{M}}$ is the channel correlation matrix

$$\mathbf{R}^*_{\mathbf{M}} = \mathbf{M_1}^H\mathbf{M_1}. \tag{4.21}$$

Due to its linear nature the LDDE equaliser is not well suited for channels with deep frequency selective fading. It has been shown that for the LDDE the variances of the estimated symbols in an unknown block have a bell shaped distribution where the symbols closer to the known symbols at the edges of the unknown block have the smallest variances and the symbols in the middle of the unknown block have the largest variances.

4.3.3.1 Non-linear data directed estimation technique

A nonlinear data directed estimation (NDDE) algorithm was also suggested by Hsu, which is better suited for channels with deep frequency selective fading [14]. The NDDE algorithm is mostly similar to the LDDE algorithm. The difference between the two algorithms is that after calculating the unknown symbol estimates, the NDDE only stores the estimated results for the first and last symbol in the unknown block. A hard decision is then made for each of the stored symbol results before subtracting the ISI related to these hard decisions from the unknown received symbols. The hard decisions can be made for the symbols at the edges of the block of unknown symbols since these symbols have

been shown to have much lower estimated variances than the other symbols in the unknown symbol block. This process is similar to the removal of the ISI related to the known probes in (4.18) and can also be described as treating the hard decisions related to the newly estimated symbols as part of the known symbol sections. The calculation of the unknown symbol estimates is then repeated. With the second calculation the second and second to last symbol estimates are stored and subtracted, just as the first and last symbols were during the previous step. This process continues until all of the symbol estimates have been calculated. It has been shown that the variances for symbols estimated by the NDDE algorithm have a bowl shaped distribution where symbols closer to the edges of the unknown symbol block have the largest estimated variances and the symbols in the middle of the unknown block have the smallest estimated variances. However, it must be noted that incorrect hard decisions will introduce errors and degrade performance.

4.3.4 Channel estimation

The information in this section is adapted from a paper by Hsu [14]. From (4.1) it is clear that the accuracy of the equaliser is dependent on the accuracy of the channel response. For time varying channels such as the ionospheric channel, the channel response is dynamically estimated and updated to accommodate for changes in the channel. One way of choosing a new channel response at time instance k ($\hat{\mathbf{h}}_k$) is by applying the MMSE criterion to minimise $|r_k - \hat{r}_k|^2$ where

$$\hat{r}_k = \hat{\mathbf{h}}_k^T \hat{\mathbf{x}}_k, \tag{4.22}$$

and where $\hat{\mathbf{x}}_k$ are all the known symbols and unknown symbol decisions that will affect the symbol received at time instance k.

The channel taps can be estimated using the adaptive algorithm

$$\hat{\mathbf{h}}_k = \hat{\mathbf{h}}_{k-1} + \Delta e_k \hat{\mathbf{x}}_k^*, \tag{4.23}$$

where Δ is a constant that is larger than zero but much smaller than one and the error signal

$$e_k = r_k - \hat{r}_k. \tag{4.24}$$

To accommodate for differential Doppler phase shifts between paths, each channel weight estimated by (4.23) ($h_{k,l}$) is phase shifted by $\psi_{k,l}$ where

$$\psi_{k,l} = \psi_{k-1,l} + \Delta_1 \mathcal{I}m\left(h_{k,l}^* \hat{x}_{k-l}^* e_k\right), \tag{4.25}$$

and where Δ_1 is a constant that is larger than zero but much smaller than one and $\mathcal{I}m(x)$ is the imaginary component of complex symbol x.

4.4 SYMBOL TO BIT MAPPING

The simplest method for mapping complex symbols to bits is to select the symbol in the constellation with the shortest euclidean distance from the received symbol as the estimated received symbol. The bits associated with the estimated received symbol can then be selected as the estimated received bits. While simple, this method leads to loss of information since distance between the received symbol and the estimated symbol can provide an indication of the confidence in the decision made. This confidence level can be used to improve the performance of the decoder that will process the mapped bits.

The position of the received complex symbol is often taken into account by calculating the probability of having received each of the symbols in the constellation. If Gaussian noise is assumed, the conditional probability density function (PDF) for $\hat{b}_k$ given that complex symbol α was transmitted can be calculated by [21]

$$p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}, \tag{4.26}$$

where σ_n^2 is the noise variance. The conditional probability that transmit bit number i in symbol number k is equal to 0 given that symbol estimate $\hat{b}_k$ was received is calculated by applying Bayes' theorem and adding the probabilities for all symbols in the constellation for which bit number $i = 0$ or [22]

$$P\left(b_k^i = 0 \,\middle|\, \hat{b}_k\right) = \frac{1}{p\left(\hat{b}_k\right)} \sum_{\alpha \Rightarrow \alpha^i = 0} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right) P\left(b_k^i = 0\right) \tag{4.27}$$

where b_k^i is the ith bit of the kth symbol and $\alpha \Rightarrow \alpha^i = 0$ is the set of all symbols in the constellation for which the ith bit is equal to 0 [23]. The probability for $b_k^i = 1$ can be calculated in a similar manner.

4.4.1 Log likelihood ratio

To simplify the storage and transfer of the calculated probabilities, the probabilities are often converted to LLRs. The LLR of a bit (u) is defined as the natural logarithm of the probability that the bit is equal to 0, divided by the probability that the bit is equal to 1, or [5]

$$L(u) = \ln\left[\frac{P(u=0)}{P(u=1)}\right]. \tag{4.28}$$

From (4.28) it is clear that the sign of the LLR will indicate whether the bit is most likely a 1 or a 0, while the magnitude of the LLR will indicate the level of confidence in the bit estimation. Substituting

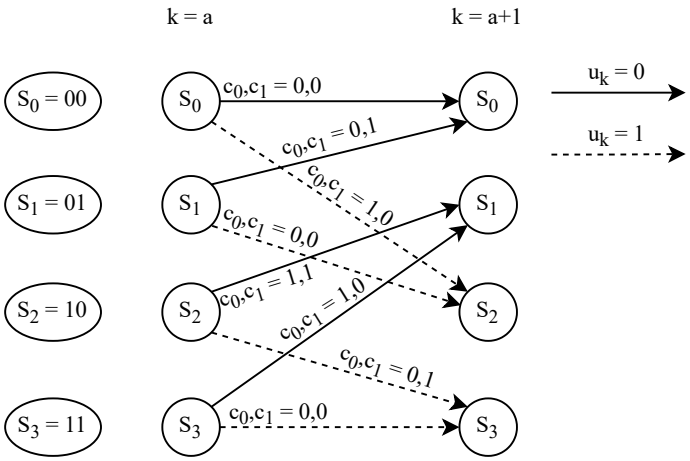

Figure 4.2. Basic example of two stages of a trellis diagram for a convolutional encoder with a constraint length of three.

(4.27) into (4.28) produces

$$L\left(b_k^i = 0 \,\middle|\, \hat{b}_k\right) = \ln\left[\frac{\frac{1}{p\left(\hat{b}_k\right)} \sum\limits_{\alpha \Rightarrow \alpha^i = 0} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right) P\left(b_k^i = 0\right)}{\frac{1}{p\left(\hat{b}_k\right)} \sum\limits_{\alpha \Rightarrow \alpha^i = 1} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right) P\left(b_k^i = 1\right)}\right] \tag{4.29}$$

$$= \ln\left[\frac{\sum\limits_{\alpha \Rightarrow \alpha^i = 0} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right)}{\sum\limits_{\alpha \Rightarrow \alpha^i = 1} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right)}\right], \tag{4.30}$$

if the a-priori probability $P\left(b_k^i = 0\right)$ can be assumed to be equal to $P\left(b_k^i = 1\right)$.

4.5 DECODERS

Trellis based algorithms are commonly used to decode convolutionally encoded bit streams [5]. An example of two stages in a trellis is presented in Figure 4.2 for illustrative purposes. As can be seen from Figure 4.2, at each stage in the trellis (k) there are a number of possible states (S_0 to S_3). For a convolutional encoder the number of states will be equal to 2^{C-1} where C is the constraint length of the encoder. Each state will therefore represent a possible set of values for the most resent $C-1$ bits in the shift register. The value of the bits in the shift register at states S_0 to S_3 are indicated in Figure 4.2.

From the transmitter perspective each new bit that is received from the user will cause a transition in states. Transitions caused when the transmitted bit (u_k) is equal to 0 are indicated with solid lines in Figure 4.2, while transitions where the transmit bit is 1 are indicated with dashed lines. The new state after each state transition is determined by shifting the transmitted bit into the shift register. Each

transition is also associated with coded bits, labelled c_0 and c_1 in Figure 4.2. In this example a half rate encoder with generator polynomials

$$T_0 = x^2 + x \tag{4.31}$$

for c_0 and

$$T_1 = x + 1 \tag{4.32}$$

for c_1 is used.

At the receiver the trellis can be solved by assigning a metric to each transition or branch. The is done by comparing the received bits to the coded bits associated with each branch and assigning a metric based on how likely it is that the specified transition occurred at the transmitter. The information over multiple stages is combine using a trellis based algorithm to determine the most likely sequence of encoder input bits. Two common examples of trellis based algorithms are the Viterbi algorithm and the maximum a-posteriori (MAP) algorithm.

4.5.1 Maximum a-posteriori probability algorithm

The information in this section is adapted from Hanzo [5]. The MAP algorithm is an optimal algorithm which calculates the probability that each transmitted bit was either a 1 or a 0 based a series of received symbols. This can be expressed in LLR form as

$$L\left(u_k \left| \hat{\mathbf{b}} \right.\right) = \ln \left[\frac{P\left(u_k = 0 \left| \hat{\mathbf{b}} \right.\right)}{P\left(u_k = 1 \left| \hat{\mathbf{b}} \right.\right)} \right] \tag{4.33}$$

$$= \ln \left[\frac{\sum\limits_{(s',s) \Rightarrow u_k = 0} P\left(s', s, \hat{\mathbf{b}}\right)}{\sum\limits_{(s',s) \Rightarrow u_k = 1} P\left(s', s, \hat{\mathbf{b}}\right)} \right] \tag{4.34}$$

where u_k is the transmitted bit, $\hat{\mathbf{b}}$ is the stream of received symbol estimates after equalisation, $P(u_k = 0|\hat{\mathbf{b}})$ is the conditional probability that $u_k = 0$ given that $\hat{\mathbf{b}}$ was received, $(s',s) \Rightarrow u_k = 1$ is the set of transitions from state s' to state s which correspond to a transmitted bit (u_k) of 1 and $P(s',s,\hat{\mathbf{b}})$ is the joint probability that the previous state was s', the next state is s and the received symbol sequence is $\hat{\mathbf{b}}$.

$P(s',s,\hat{\mathbf{b}})$ can be separated into three components

$$P\left(s', s, \hat{\mathbf{b}}\right) = \alpha_{k-1}\left(s'\right) \gamma_k\left(s', s\right) \beta_k\left(s\right) \tag{4.35}$$

where $\alpha_{k-1}(s')$, which is known as the forward metric, is the probability of having arrived at state s'

given previously received symbols, $\beta_k(s)$, known as the reverse metric, is the probability of the next state having been s given the symbols that were received after the current symbol, and $\gamma_k(s',s)$, known as the transition metric, is the probability of the transition from state s' occurring given the currently received symbol. $\alpha_{k-1}(s')$ can be calculated recursively in the forward direction using

$$\alpha_k(s) = \sum_{\forall s'} \gamma_k(s',s)\, \alpha_{k-1}(s') \tag{4.36}$$

which depends on previous values of $\alpha_{k-1}(s')$ as well as the transition metric. $\beta_k(s)$ can be calculated recursively in the reverse directions using

$$\beta_{k-1}(s') = \sum_{\forall s} \gamma_k(s',s)\, \beta_k(s) \tag{4.37}$$

which relies on future values of $\beta_k(s)$ as well as the current transition metric.

The transitions metric ($\gamma_k(s',s)$) can be further separated into two parts

$$\gamma_k(s',s) = P(c_k \,|\, \hat{b}_k)\, P(u_k) \tag{4.38}$$

where the first component is the conditional probability that the codeword (c_k) associated with the transition from state s' to s was transmitted given that $\hat{b}_k$ was received and the second component ($P(u_k)$) is the a-priori probability that the user input bit u_k was equal to the input bit associated with the transition from state s' to s. When no a-priori information is available, which is often the case, $P(u_k) = 0.5$ for $u_k = 1$ and for $u_k = 0$. The first term in (4.38) can be calculated using

$$P(c_k \,|\, \hat{b}_k) = \prod_{i=0}^{n-1} P(c_k^i \,|\, \hat{b}_k) \tag{4.39}$$

where n is the number of bits per codeword and $P(c_k^i|\hat{b}_k)$ is the conditional probability that the ith bit in codeword c_k was transmitted given that symbol estimate $\hat{b}_k$ was received. $P(c_k|\hat{b}_k)$ can be calculated with the assistance of (4.27). From (4.39) it can be noted that $P(c_k|\hat{b}_k)$ depends on multiple received symbols instead of a single received symbol. The notation $\hat{b}_k$ is still used for the symbols to indicate that the joint probability is only dependent on the received symbols associated with each code bit c_k^i and not the entire stream of received symbols.

There are two simplifications of the MAP algorithm that are commonly used due to the computational complexity of the algorithm. The first is the max-log-MAP algorithm which replaces each of the components of the MAP algorithm with their natural logarithm. The three components in (4.35)

become [5]

$$\mathrm{A}_k(s) = \ln[\alpha_k(s)], \tag{4.40}$$

$$\mathrm{B}_k(s) = \ln[\beta_k(s)] \tag{4.41}$$

and

$$\Gamma_k(s',s) = \ln\left[\gamma_k(s',s)\right]. \tag{4.42}$$

Converting the parameters to the logarithmic domain allows for the computational complexity of the algorithm to be significantly reduced. The identity

$$\ln(xy) = \ln(x) + \ln(y) \tag{4.43}$$

allows for multiplications to be replaced for addition. Additions are estimated by

$$\ln(e^x + e^y) \triangleq \max(x,y) \tag{4.44}$$

or

$$\ln(e^x + e^y + e^z) \triangleq \max[x,y,z] \tag{4.45}$$

$$= \max[\max(x,y),z] \tag{4.46}$$

which is also less computationally complex. It is important to note that (4.45) is an estimation which can become inaccurate when one of the inputs (x or y or z) are not significantly larger than the others. Due to this estimation the max-log-MAP algorithm has been shown to have less than optimal performance.

An alternative method known as the log-MAP algorithm adds a correction factor to the max operator to arrive at the exact solution by using the Jacobian algorithm. The resulting equation is

$$\ln(e^x + e^y) = \max(x,y) + \ln\left(1 + e^{-|x-y|}\right) \tag{4.47}$$

$$= \mathrm{maxj}(x,y), \tag{4.48}$$

where $\ln(1+e^{-|x-y|})$ is the correction factor and maxj is used to represent the max operation with the correction. It has been shown that the correction factor can be replaced by a lookup table with eight elements and an input of $|x-y|$ without loss of performance [24].

By considering (4.35) as well as (4.40), (4.41) and (4.42), (4.34) can be rewritten as

$$L\left(u_k \left| \hat{\mathbf{b}}\right.\right) = \ln\left[\frac{\sum\limits_{(s',s)\Rightarrow u_k=0} \alpha_{k-1}\left(s'\right)\gamma_k\left(s',s\right)\beta_k\left(s\right)}{\sum\limits_{(s',s)\Rightarrow u_k=1} \alpha_{k-1}\left(s'\right)\gamma_k\left(s',s\right)\beta_k\left(s\right)}\right] \tag{4.49}$$

$$= \ln\left[\frac{\sum\limits_{(s',s)\Rightarrow u_k=0} e^{\ln[\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)]}}{\sum\limits_{(s',s)\Rightarrow u_k=1} e^{\ln[\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)]}}\right] \tag{4.50}$$

$$= \ln\left[\frac{\sum\limits_{(s',s)\Rightarrow u_k=0} e^{\mathrm{A}_{k-1}(s')+\Gamma_k(s',s)+\mathrm{B}_k(s)}}{\sum\limits_{(s',s)\Rightarrow u_k=1} e^{\mathrm{A}_{k-1}(s')+\Gamma_k(s',s)+\mathrm{B}_k(s)}}\right] \tag{4.51}$$

$$= \max\limits_{(s',s)\Rightarrow u_k=0}\mathrm{j}\left[\mathrm{A}_{k-1}\left(s'\right)+\Gamma_k\left(s',s\right)+\mathrm{B}_k\left(s\right)\right]- \max\limits_{(s',s)\Rightarrow u_k=1}\mathrm{j}\left[\mathrm{A}_{k-1}\left(s'\right)+\Gamma_k\left(s',s\right)+\mathrm{B}_k\left(s\right)\right]. \tag{4.52}$$

By substituting the iterative forward calculation (4.36) into (4.40) the iterative forward calculation for the log-MAP algorithm becomes

$$\mathrm{A}_k\left(s\right) = \max\limits_{\forall s'}\mathrm{j}\left[\Gamma_k\left(s',s\right)+\mathrm{A}_{k-1}\left(s'\right)\right]. \tag{4.53}$$

Likewise, the iterative reverse calculation for the log-MAP algorithm can be determined by substituting (4.37) into (4.41) which gives

$$\mathrm{B}_{k-1}\left(s'\right) = \max\limits_{\forall s}\mathrm{j}\left[\Gamma_k\left(s',s\right)+\mathrm{B}_k\left(s\right)\right] \tag{4.54}$$

4.5.2 Viterbi algorithm

The Viterbi algorithm is a nearly optimal algorithm that attempts to determine the most probable path through the trellis, rather than attempting to calculate the likelihood for each bit individually [25]. For each stage in the trellis a metric is assigned to each transition based on the estimated received bits, just as it is done with the MAP algorithm. At the first stage all states are associated with the same metric unless the starting state of the trellis is known. With each new codeword received the trellis will undergo a transition and the metric for each new state will be determined from the old state as well as the transition metric. Whenever two paths meet at a state the Viterbi algorithm will save only the path with the best metric and discard the other path. Once the trellis has progressed a sufficient number of stages from stage k, the strongest surviving path can be traced back from the current stage to stage k to determine the bit received at stage k. The number of stages required are known as the convergence

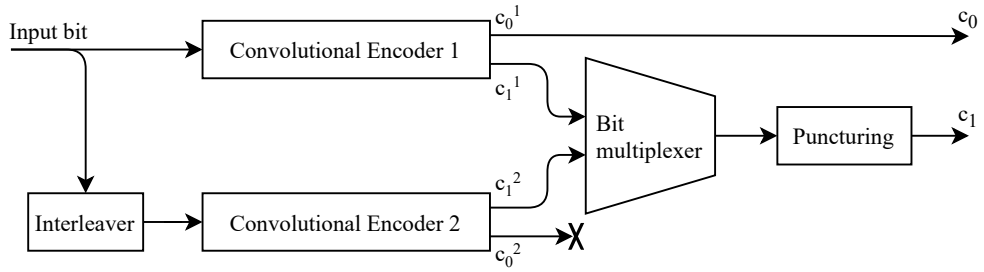

Figure 4.3. Half rate turbo encoder consisting of two half rate encoders. Adapted from [26], © 1993 IEEE.

length since the paths in the trellis are expected to converge at this point. After the optimal path is selected, all bits associated with that path are committed as the results. The Viterbi algorithm has often been preferred over the MAP algorithm due to the reduced computational complexity at the cost of slightly worse performance. The performance of the Viterbi algorithm is comparable to that of the max-log-MAP algorithm [5].

4.5.3 Turbo coding

Turbo coding is an alternative form of FEC coding which has been shown to achieve BER performance close to the Shannon limit [26]. Figure 4.3 shows a basic example of a half rate turbo encoder. Turbo encoders are often systematic meaning that one of the output bits corresponds to the input bit. Turbo encoders achieve improved performance by passing the user input bits through multiple, often identical, encoders with interleavers between the encoders. The outputs of the encoders are then multiplexed and punctured before being transmitted over the air. For the example in Figure 4.3 the two convolutional encoders are both half rate systematic encoders meaning that the first output bit of Convolutional Encoder 1 (c_0^1) corresponds to one of the input bit while the first output of Convolutional Encoder 2 (c_0^2) corresponds to an interleaved version of the input bit. This property allows for both of these bits to be recovered at the receiver even if only c_0^1 is transmitted as c_0. c_0^2 can therefore be discarded. The second output bits of each encoder (c_1^1 and c_1^2) are multiplexed before being punctured and transmitted as the second output of the turbo decoder c_1. The puncturing often alternates between the output bits from Encoder 1 and Encoder 2.

Figure 4.4 shows the turbo decoder associated with the turbo encoder in Figure 4.3. The received bits are first separated into three bit streams where one stream is associated with the systematic bit (c_0), while the other two streams are associated with second encoded bit of the two encoders (c_1^1 and c_1^2).

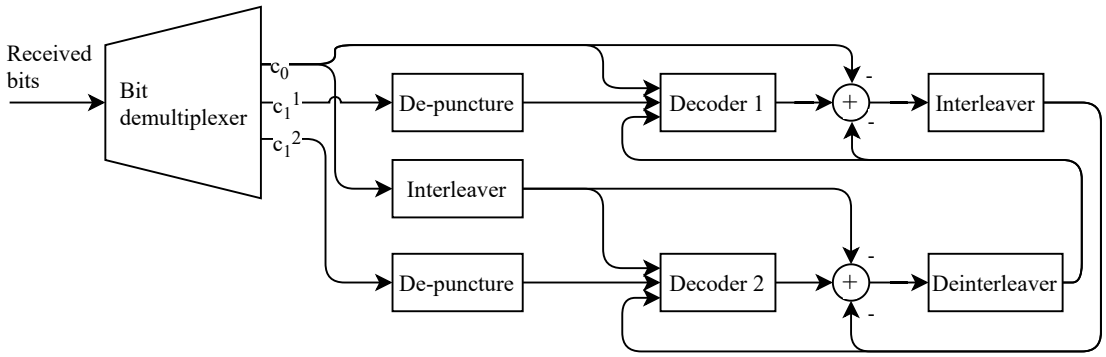

Figure 4.4. Turbo decoder block diagram for half rate turbo decoder consisting of two half rate decoders. Adapted from [26], © 1993 IEEE.

c_0 is passed directly to Decoder 1 while c_1^1 is first de-punctured to insert erasures where bits were punctured at the transmitter. After Decoder 1 has decoded a block of data the systematic bits will be subtracted from the result in LLR form before being interleaved and fed to Decoder 2. Decoder 2 will use this information from Decoder 1 as a-priori information. The a-priori information along with the interleaved version of the original systematic bits and the de-punctured bits (c_1^2) will be used to calculate the first decoder result for Decoder 2. The bits received from Decoder 1 as well as the interleaved version of c_0 are subtracted from the output in LLR form before being deinterleaved and fed back to Decoder 1 as a-priori information. This process can continue for multiple iterations. The subtraction components are necessary to ensure that only extrinsic information is exchanged between the decoders and to prevent the decoders from being fed with their own information from previous iterations.

4.6 TURBO EQUALISATION

Turbo equalisation is an iterative ISI correction method suggested by Douillard *et al.* in 1995 [27]. Turbo equalisation is based on a similar principal as turbo coding, but with extrinsic information being passed between the equaliser and the decoder instead of two decoders. As with a turbo decoder the turbo equaliser can also perform multiple iterations in an attempt to improve the results before passing the decoded bits to the user.

A simplified diagram is presented in Figure 4.5 to assist with defining some of the basic principles of a turbo equaliser. The first pass of the turbo equaliser is similar to the normal receiver processing chain without feedback. First the equaliser attempts to compensate for ISI and equalise the channel. The equalised symbols are then converted to bits before being deinterleaved and decoded. The first

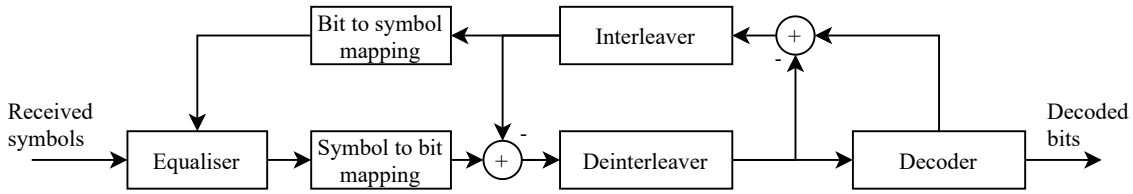

Figure 4.5. Simplified block diagram used to illustrate the basic concepts of turbo equalisation.

iteration starts by re-calculating the coded bits which can be either done by re-encoding the decoded bits or by adjusting the decoder algorithm to produce modified versions of the coded bits. The input bits to the decoder are then subtracted from the re-encoded bits in LLR form to produce the extrinsic information bits. The extrinsic bits are interleaved and mapped to symbols before being passed to the equaliser. The equaliser is now able to use the fed back symbols as a-priori information during the first iteration. The equalised symbols are converted back to bits before subtracting the fed back bits from the result. This subtraction ensures that the information received from the decoder during the previous iteration is not fed back to the decoder during the current iteration. The bits are deinterleaved and once again processed by the decoder as a new set of inputs, after which the data can be uploaded to the user, or more iterations can occur.

The difference between the feedback loops in the case of turbo coding and turbo equalisation is that the extrinsic information is used as a-priori information for the decoders with turbo coding. With turbo equalisation the extrinsic information from the decoder is used as a-priori information for the equaliser while the extrinsic information from the equaliser is used as a new set of input values to the decoder.

4.6.1 Minimum mean square error equaliser with turbo feedback

Tüchler *et al.* introduced a MMSE based turbo equaliser which compared well in performance with MAP based turbo equalisers while having significantly reduced complexity [28]. The information in this section is based on the paper by Tüchler. The MMSE turbo equaliser algorithm is derived by first defining the linear estimate $\hat{x}_k$ as

$$\hat{x}_k = \mathbf{p}_k{}^H \mathbf{r}_k + q_k, \tag{4.55}$$

where $\mathbf{r}_k$ is a column vector of the received symbols which includes N_1 causal and N_2 non-causal symbols as well as the kth symbol, and has a length of $N = N_1 + N_2 + 1$

$$\mathbf{r}_k{}^T = \begin{bmatrix} r_{k-N_2} & r_{k-N_2+1} & \dots & r_k & \dots & r_{k+N_1-1} & r_{k+N_1} \end{bmatrix}. \tag{4.56}$$

$p_{k,n}$ and q_k in (4.55) are the estimator coefficients at time instance k and

$$\mathbf{p}_k{}^T = \begin{bmatrix} p^*_{k,N_2} & p^*_{k,N_2-1} & \cdots & p^*_{k,0} & \cdots & p^*_{k,-N_1+1} & p^*_{k,-N_1} \end{bmatrix}. \tag{4.57}$$

To minimise $E(|x_k - \hat{x}_k|^2)$, $\mathbf{p}_k$ and q_k are chosen as

$$\mathbf{p}_k = \mathrm{Cov}\left(\mathbf{r}_k, \mathbf{r}_k\right)^{-1} \mathrm{Cov}\left(\mathbf{r}_k, x_k\right) \tag{4.58}$$

and

$$q_k = E\left(x_k\right) - \mathbf{p}_k{}^H E\left(\mathbf{r}_k\right). \tag{4.59}$$

As in (4.1) the received sequence can be expressed as

$$\mathbf{r}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k, \tag{4.60}$$

where $\mathbf{H}$ is once again the channel convolutional matrix defined in (4.5), the noise vector ($\mathbf{n}_k$) is

$$\mathbf{n}_k{}^T = \begin{bmatrix} n_{k-N_2} & n_{k-N_2+1} & \cdots & n_k & \cdots & n_{k+N_1-1} & n_{k+N_1} \end{bmatrix}, \tag{4.61}$$

and the transmitted symbols ($\mathbf{x}_k$) is

$$\mathbf{x}_k{}^T = \begin{bmatrix} x_{k-N_2-L+1} & x_{k-N_2-L+2} & \cdots & x_k & \cdots & x_{k+N_1-1} & x_{k+N_1} \end{bmatrix}. \tag{4.62}$$

It has further been shown that

$$E\left(\mathbf{r}_k\right) = \mathbf{H}E\left(\mathbf{x}_k\right), \tag{4.63}$$

$$\mathrm{Cov}\left(\mathbf{r}_k, x_k\right) = \mathrm{Cov}\left(x_k, x_k\right) \begin{bmatrix} 0_{1\times(N_2+L-1)} & 1 & 0_{1\times(N_1)} \end{bmatrix} \mathbf{H}^H \tag{4.64}$$

and

$$\mathrm{Cov}\left(\mathbf{r}_k, \mathbf{r}_k\right) = \sigma_n^2 \mathbf{I}_N \mathbf{H} \mathrm{Cov}\left(\mathbf{x}_k, \mathbf{x}_k\right) \mathbf{H}^H \tag{4.65}$$

where $\mathbf{I}_N$ is an N by N identity matrix and σ_n^2 is the noise variance.

$E(x_k)$ can be calculated using the mean of the transmitted symbols ($\bar{x}$)

$$\bar{x} = E\left(x_k\right) \tag{4.66}$$

$$= \sum_{\alpha_i \in S} \alpha_i P\left(x_k = \alpha_i\right) \tag{4.67}$$

while $\text{Cov}(x_k, x_k)$ can be calculated using the transmitted symbol variance (v_k)

$$v_k = \text{Cov}(x_k, x_k) \tag{4.68}$$

$$= \left[\sum_{\alpha_i \in S} |\alpha_i|^2 P(x_k = \alpha_i) \right] - |\bar{x}_k|^2, \tag{4.69}$$

where S is a set containing all symbols in the constellation. Conveniently $P(x_k = \alpha_i)$ is a function of the a-priori information and

$$P(x_k = \alpha_i) = \prod_{j=1}^{Q} P\left(c_{k,j} = s_{i,j}\right), \tag{4.70}$$

where Q is the number of bits per symbol, $c_{k,j}$ is the jth bit of the kth transmit symbol and $s_{i,j}$ is the jth bit in the bit pattern that corresponds with constellation symbol i [28].

Through further derivation $\hat{x}_k$ is rewritten as

$$\hat{x}_k = \bar{x}_k + v_k \mathbf{f}_k^H (\mathbf{r}_k - \bar{\mathbf{r}}_k), \tag{4.71}$$

where $\bar{\mathbf{r}}_k$ is the mean of the received symbols calculated from the transmit symbol mean ($\bar{x}_k$) and the channel response and $\mathbf{f}_k$ is equivalent to a set of filter coefficients which can be calculated through

$$\mathbf{f}_k = [\text{Cov}(\mathbf{r}_k, \mathbf{r}_k)]^{-1} \mathbf{s}, \tag{4.72}$$

where

$$\mathbf{s} = \mathbf{H} \begin{bmatrix} 0_{1\times(N_2-1)} & 1 & 0_{1\times(N_1)} \end{bmatrix}. \tag{4.73}$$

As previously mentioned it is important that the output symbols produced by the turbo equaliser not be directly influenced by the corresponding a-priori symbols received from the decoder. This is achieved by setting $\bar{x}_k = 0$ and as a result $v_k = 1$ when calculating $\hat{x}_k$, which results in

$$\hat{x}_k = \mathbf{K}_n \mathbf{f}_k^H (\mathbf{r}_k - \bar{\mathbf{r}}_k + \bar{x}_k \mathbf{s}), \tag{4.74}$$

where

$$\mathbf{K}_k \triangleq \left[1 + (1 - v_k) \mathbf{f}_k^H \mathbf{s}\right]^{-1}. \tag{4.75}$$

Figure 4.6 shows a block diagram of the MMSE equaliser with turbo feedback. When compared to Figure 4.5 the main difference is that the fed back information is in the form of a symbol mean and variance instead of symbol estimates. For the MMSE turbo equaliser it is also not necessary to subtract

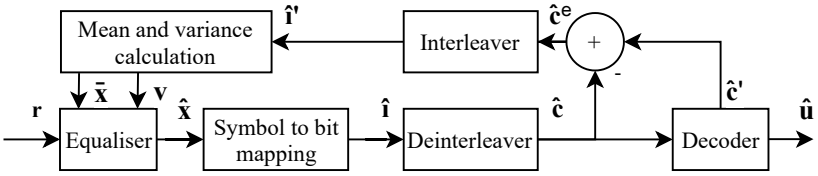

Figure 4.6. A block diagram of a minimum mean square error equaliser with turbo feedback.

the fed back bits from the newly calculated bits, since the MMSE turbo equaliser already ensures that the output bits are not directly influenced by their fed back counterparts.

4.6.2 Data directed estimation equaliser with turbo feedback

A TDDE algorithm has also been suggest which relies on the same principle as the MMSE turbo equaliser. An important difference is that TDDE aims to minimise $E(|\mathbf{H}\hat{\mathbf{b}} - \mathbf{c}|^2)$ where $\hat{\mathbf{b}}$ is the estimations for only the unknown symbols and $\mathbf{c}$ is the received symbols with the ISI contribution of known probes subtracted as defined in (4.18) [13]. The generalised channel correlation matrix, with reference to (4.21), is redefined as

$$\mathbf{R} = {\sigma_n}^2 \mathbf{V}^{-1} + {\mathbf{M_1}}^H \mathbf{M_1}, \tag{4.76}$$

where $\mathbf{V}$ is a matrix of all zeros except for the main diagonal which contains the vector of symbol variances

$$\mathbf{v} = \begin{bmatrix} v_0 & v_1 & \dots & v_{M-2} & v_{M-1} \end{bmatrix}. \tag{4.77}$$

Through a slightly different derivation than what is used for the turbo MMSE, the unknown symbol $\hat{b}_k$ can be estimated with [13]

$$\hat{b}_k = \rho_k {\mathbf{f}_k}^H {\mathbf{M_1}}^H \left(\mathbf{c} - \mathbf{M_1}\bar{\mathbf{b}} + \bar{b}_k \mathbf{M_1}\mathbf{s}_k \right), \tag{4.78}$$

where $\bar{b}_k$ is the mean of the kth unknown symbol, $\mathbf{s}_k$ is a column vector of all zeros except for the kth element, ${\mathbf{f}_k}^H$ is the redefined filter coefficients

$${\mathbf{f}_k}^H = {\mathbf{s}_k}^H \mathbf{R}^{-1} \tag{4.79}$$

and ρ_k is defined as

$$\rho_k = \left[1 + {\sigma_n}^2 \left(1 - \frac{1}{v_k} \right) {\mathbf{f}_k}^H \mathbf{s}_k \right]^{-1}. \tag{4.80}$$

The main difference between the Turbo MMSE algorithm and the TDDE algorithm is that the TDDE

excludes received symbols related to the known probes and only calculates estimates for the unknown data symbols ($\mathbf{x}_k$). The a-priori mean and variance calculated using (4.67) and (4.69) respectively are also calculated for only the unknown symbols.

The most complex operation in (4.78) is the inversion of the matrix $\mathbf{R}$. A further simplification of the algorithm is suggested by averaging the feedback variance instead of calculating each individual variance, resulting in all the elements in $\mathbf{v}$ to be the same. The result is that $\mathbf{R}$ becomes a Toeplitz matrix which can be solved using the Levinson-Durbin algorithm, as with the channel correlation matrix in the DDE algorithm [13, 14].

4.7 CONCLUSION

A receiver suitable for MIL-STD-188-110D App. D PSK and QAM waveforms will require an equaliser, a symbol to bit mapping function, a deinterleaver and a decoder, among other functions. Not much variation is possible in the deinterleaver function. For other functions, multiple implementation options are available with varying performance capabilities. The equaliser can be implemented using almost optimal but computationally complex MLSE methods or less optimal but computationally more feasible methods such as DDE. Implementation options for the decoder are the Viterbi algorithm and MAP based algorithms. Turbo feedback receivers have also been suggested which can feed information from the decoder back to the the equaliser allowing for improved performance after multiple iterations. When designing a MIL-STD-188-110D App. D receiver, it is important to weigh the performance of these components against the computational complexity in order to arrive at a solution that will be able to meet the requirements set out in MIL-STD-188-110D App. D and also run in real time on a target platform.

CHAPTER 5 HIGH FREQUENCY WAVEFORM TURBO RECEIVER DESIGN

5.1 CHAPTER OVERVIEW

The design of a turbo receiver suitable for the 3 kHz variants of MIL-STD-188-110D App. D waveform number 1 and waveform number 2 is presented in this chapter. This design can also be expanded to other waveforms in MIL-STD-188-110D App. D, excluding waveform number 0 which makes use of Walsh orthogonal modulation. To design a turbo receiver, it is first necessary to consider the first equaliser pass which is similar to a conventional receiver that does not make use of turbo feedback. The most important design choices for the first receiver pass are the equaliser selection and the decoder selection since these components significantly impact the performance of the receiver and can also in some cases mandate the design or selection of other receiver components. The decoder selection can also simplify the generation of feedback information since some decoder algorithms can be adapted to produced updated representations of coded bits and eliminate the need for re-encoding bits before feeding them back. The main consideration for the feedback path of the turbo equaliser is the selection of the turbo equalisation algorithm which does not only impact the performance, but can also mandate the format of the feedback information and the selection of feedback components as a result.

Section 5.2 discusses the selection of waveforms for implementation. Section 5.3 details the design of a conventional HF receiver for the selected waveforms which will be used during the first receiver pass. The iterative or turbo receiver design is detailed in Section 5.4.

5.2 WAVEFORM SELECTION

This chapter focuses on the waveforms defined in MIL-STD-188-110D App. D since these waveforms are relevant and practical examples of standardised waveforms [2]. A wide variety of configuration options are also available which can be varied to test the effectiveness of the turbo equaliser with

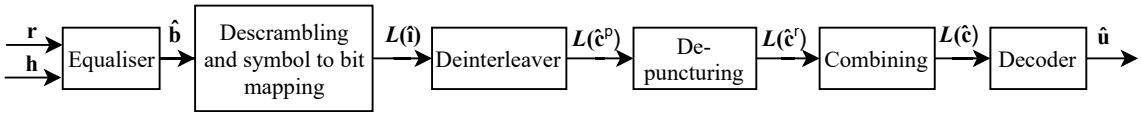

Figure 5.1. Processing functions relevant to a conventional MIL-STD-188-110D App. D waveform receiver where equalisation and decoding are handled independently.

waveforms with varying characteristics. Tests with the STANAG 4539 waveforms, which are similar to the 3 kHz MIL-STD-188-110D App. D waveforms have shown that the less robust and higher data rate waveforms tend to show greater gains when applying turbo equalisation techniques [9]. The two most robust 3 kHz waveforms, waveform number 1 and waveform number 2, were selected for this design while also considering the extensibility of the design to accommodate for other waveform numbers and bandwidths. If gains are achieved with these waveforms it is expected that greater gains could be achievable at the waveforms with higher throughput and less reliability. Waveform number 1 and waveform number 2 are very similar to one another except for the fact that waveform number 1 makes use of four encoder repeats where waveform number 2 only makes use of two repeats. This will allow for a fair comparison to study the effects that the presence of encoder repeats has on turbo equalisation.

5.3 CONVENTIONAL RECEIVER PROCESSING FUNCTIONS

Figure 5.1 shows the processing functions relevant to a conventional MIL-STD-188-110D App. D receiver where equalisation and decoding are handled independently. The data phase of the reception starts when the equaliser receives symbols as well as a channel response from the channel estimation component on the receive modem. The equaliser groups the received symbols ($\mathbf{r}$) into equaliser decision blocks or frames, each consisting of an unknown data block of symbols with one preceding known probe block as well as one succeeding known probe block of symbols. The equaliser processes one decision block at a time using the associated channel response ($\mathbf{h}$) and produces the equalised symbol estimates ($\hat{\mathbf{b}}$). The equalised symbol estimates are descrambled and mapped to LLRs producing the interleaved bit estimate LLRs ($L(\hat{\mathbf{i}})$). These bits are loaded into the deinterleaver. Bits can only be fetched from the deinterleaver and passed through the remaining processing blocks after an entire deinterleaver block has been filled.

Once enough samples have been received to fill a deinterleaver block, the receiver starts loading the bit estimates from the deinterleaver. These bit estimates represent the estimates of the coded punctured bits

($\hat{\mathbf{c}}^p$). The punctured bit estimates are first de-punctured to produce the repeated coded bit estimates ($\hat{\mathbf{c}}^r$) and then combined to produce the original coded bit estimates ($\hat{\mathbf{c}}$). After de-puncturing and combining all bit estimates, the receiver stores a copy of the block of coded bit estimates before passing the block through the decoder. The decoder produces the decoded user data bits ($\hat{\mathbf{u}}$) which are passed along to the data user.

The remainder of this section describes the design for each receiver block that forms part of the conventional receiver starting at the first block to be executed, the equaliser, and ending with the final processing block, the decoder.

5.3.1 Equaliser

5.3.1.1 Algorithm selection

Various examples of channel equalisers were described in 4.3 with varying levels of computational complexity and performance. MLSE based equalisers such as MAP equalisers and Viterbi algorithm based equalisers are expected to provide near-optimal performance. However, the computational complexity of MLSE algorithms exponentially increases with an increase in channel length and constellation density, making these options impractical especially for the higher constellation waveforms [19, 20]. The LDDE equaliser was designed for waveforms with known probe and unknown data sections but, like other LEs, is sensitive to deep frequency selective fading which is common in ionospheric channels [14]. The NDDE equaliser has been shown to closely match the performance of the decision feedback equaliser (DFE) but is expected to converge faster when using a similar adaptive channel estimation algorithm [14, 29].

The NDDE algorithm was selected for the equaliser of the conventional receiver. In addition the LDDE algorithm was also implemented to demonstrate the improvement that could be gained by making use of non-linear equaliser. This also allows for an investigation into the relevance of the first pass equaliser in the context of turbo equalisation.

5.3.1.2 Waveform parameters

The equaliser is designed for the 3 kHz bandwidth versions of waveform number 1 and waveform number 2 in MIL-STD-188-110D App. D. Table 5.1 shows the number of symbols in the known (N) and unknown (M) sections, as well as the symbol mapping for waveform number 1 and waveform number 2. MIL-STD-188-110D App. D provides further details on the structure of the poly-phase known probe sections [3]. The mapping of BPSK and 8-PSK symbols numbers to complex constellation

Table 5.1. Symbol mapping and number of known and unknown symbols for the 3 kHz versions of waveform number 1 and waveform number 2 as specified in MIL-STD-188-110D App. D [3].

Waveform number	Symbol mapping	Number of known symbols (N)	Number of unknown symbols (M)
1	BPSK	48	48
2	BPSK	48	48

Table 5.2. Mapping of 8-PSK and BPSK symbol numbers to complex symbols as specified in MIL-STD-188-110D App. D [3].

8-PSK Symbol number	Phase	BPSK Symbol number	In-phase	Quadrature
0	0	0	1	0
1	$\frac{\pi}{4}$	not applicable (N/A)	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
2	$\frac{\pi}{2}$	N/A	0	1
3	$\frac{3\pi}{4}$	N/A	$-\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
4	π	1	-1	0
5	$\frac{5\pi}{4}$	N/A	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
6	$\frac{3\pi}{2}$	N/A	0	-1
7	$\frac{7\pi}{4}$	N/A	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$

coordinates are shown in 5.2. As shown, symbol number zero in the BPSK constellation corresponds to the same coordinate as symbol number zero in the 8-PSK constellation while symbol number one in the BPSK constellation corresponds to symbol number four in the 8-PSK constellation. Symbol numbers are mapped to bits through their binary representations [3].

5.3.1.3 Equalisation procedure

The equaliser receives the symbol spaced channel response (**h**) as well as a stream of received symbols (**r**) as inputs from the channel estimation component. The channel length (L) selected by the channel

estimation component is chosen to sufficiently model ISI in the received signal. The order of the channel response will be equal to $L+1$. To process the first decision block in a transmission the equaliser will require at least one unknown data section with length M (48 symbols) and the preceding and succeeding known probe sections each with length N (48 symbols). Each subsequent decision block can be processed after having received another unknown data section and the succeeding known probe section.

The detailed procedure for executing an DDE equaliser is described in the steps below with steps 1 to 6 being relevant for both the LDDE and the NDDE equalisers and step 7 only being relevant for the NDDE equaliser.

1. Use the symbol spaced channel response to construct $\mathbf{M_1}$, $\mathbf{M_2}$ and $\mathbf{M_3}$ as shown in (4.15), (4.16) and (4.17) respectively.
2. Locally generate a symbol spaced representation of the known channel probes. This will be used to determine $\mathbf{a_1}$ and $\mathbf{a_2}$. As shown in (4.12) only the last L symbols of the probe are necessary to construct $\mathbf{a_1}$ while (4.13) shows that only the first L symbols from the probe are necessary to construct $\mathbf{a_2}$.
3. Remove the ISI associated with the known probe sections from the received symbols by multiplying $\mathbf{M_2}$ by $\mathbf{a_1}$ and $\mathbf{M_3}$ by $\mathbf{a_2}$ and subtracting the results from the received symbols to produce $\mathbf{c}$ as shown in (4.18).
4. Calculate the conjugate transpose of $\mathbf{M_1}$.
5. Use $\mathbf{M_1}$ and its conjugate transpose ($\mathbf{M_1}^H$) to calculate the channel correlation matrix $\mathbf{R}^*{}_M$ as shown in (4.21).
6. Use the channel correlation matrix, the complex conjugate of $\mathbf{M_1}$ and the received symbols without probes ($\mathbf{c}$) to estimate the M unknown data symbols that were received using (4.20).
7. Perform the following additional steps for the NDDE.

   (a) Commit only the first and the last symbol estimates as equaliser results.

   (b) If PSK symbol mapping is used, descramble the first and last symbol estimates that have been committed as results. This is necessary to allow for more accurate symbol decisions in the next step.

(c) Select the constellation symbols closest to the first and last symbol estimates as the symbol decisions.

(d) If PSK symbol mapping is used, scramble the symbol decisions made for the first and last symbols once again.

(e) Multiply each symbol decision with the channel response.

(f) Subtract the results of the previous step from **c** to produce a new set of received symbols which excludes the symbol decisions and their effects. This equivalent to treating the first and the last symbol as part of the preceding and succeeding known symbol blocks with the symbol decisions as new known symbols.

(g) Adjust $\mathbf{M_1}$ to exclude sections of the channel response that were multiplied by the first and the last symbol.

(h) Repeat steps 4 to 7g until all symbol estimates have been calculated.

5.3.2 Descrambling

The handling of scrambling and also descrambling depends on whether a PSK or a QAM symbol mapping is used. For PSK symbol mapping (BPSK, QPSK and 8-PSK) all symbols are scrambled to an 8-PSK constellation through modulo-8 addition of each symbol number with a scrambling sequence element. It is important to note that descrambling the symbols before making a symbol decision will have an impact on performance. If for example a BPSK constellation is transmitted and scrambled to an 8-PSK constellation the receiver would be less accurate if symbol decisions were made on the 8-PSK constellation instead of descrambling the symbols back to the BPSK constellation and then making the symbol decisions. Looking at the phases associated with each 8-PSK symbol in 5.2 it is noted that the symbols are conveniently spaced so that the phase of each symbol in the complex domain is equal to $\pi/4$ multiplied by the symbol number. This allows for scrambling to be performed in the complex domain by multiplying a symbol by the complex symbol associated with the corresponding symbol number in the scrambling sequence. Likewise descrambling can be performed in the complex symbol domain by multiplying a symbol by the complex conjugate of the associated scrambling symbol.

For waveforms with QAM symbol mapping scrambling does not change the symbol constellation and is implemented by performing the exclusive OR (XOR) operation on each bit in the received symbol and a bit in the scrambling sequence. Descrambling is only performed after the symbols have been converted to soft weights. The descrambler will invert the sign of the LLR if the relevant bit in the scrambling sequence was a one or leave the LLR as is if the relevant bit was a zero.

The waveforms that have been selected for implementation, waveform number 1 and waveform number 2 in MIL-STD-188-110D App. D, both make use of BPSK symbol mapping as indicated in Table 5.1. As a result de-scrambling is performed before symbol to LLR conversion by multiplying the equalised symbols by the complex conjugate of the associated scrambling symbols.

5.3.3 Symbol to bit mapping

Received symbols are mapped to LLRs based on the squared euclidean distance between the received symbol estimates and the relevant constellation points. For each symbol in the constellation (α) the PDF is calculated using (4.26). To calculate the probability that the ith bit of the kth transmitted symbol was 0, (4.27) is used to add all the probability distributions of constellation symbols for which bit $i = 0$. The probability that the ith bit of the kth transmitted symbol was 1 can be calculated in a similar manner. The LLR of each bit is then calculated using (4.28).

5.3.3.1 Noise variance calculation

Since AWGN has a zero mean, the noise variance in (4.26) is be calculated by

$$\sigma_n{}^2 = \frac{\sum_{i=0}^{r-1} (n_i - \bar{n})^2}{r} \tag{5.1}$$

$$= \frac{\sum_{i=0}^{r-1} (n_i)^2}{r}, \tag{5.2}$$

where n_i is the magnitude of the noise sample, r is the number of samples and $\bar{n}$ is the noise mean. The magnitude of the noise for each sample is estimated by calculating the euclidean distance between the estimated sample and the constellation point determined to be the symbol decision.

5.3.3.2 Simplifications

The simplification in this section is similar to simplifications performed in other works [5]. The LLR calculation can be simplified by

$$L\left(b_k{}^i\right) = \ln\left[\frac{P\left(b_k{}^i = 0\right)}{P\left(b_k{}^i = 1\right)}\right] \tag{5.3}$$

$$= \ln\left[\frac{\sum\limits_{\alpha\Rightarrow\alpha^i=0} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right)}{\sum\limits_{\alpha\Rightarrow\alpha^i=1} p\left(\hat{b}_k \,\middle|\, b_k = \alpha\right)}\right] \tag{5.4}$$

$$= \ln\left[\frac{\sum\limits_{\alpha\Rightarrow\alpha^i=0} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}}{\sum\limits_{\alpha\Rightarrow\alpha^i=1} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}}\right] \tag{5.5}$$

$$= \ln\left[\frac{\sum\limits_{\alpha\Rightarrow\alpha^i=0} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}}{\sum\limits_{\alpha\Rightarrow\alpha^i=1} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}}\right] \tag{5.6}$$

$$= \ln\left[\sum_{\alpha\Rightarrow\alpha^i=0} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right] - \ln\left[\sum_{\alpha\Rightarrow\alpha^i=1} e^{\left(\frac{-|\alpha-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right]. \tag{5.7}$$

To explain (5.5) to (5.7), firstly note that all terms will be divided by $\sqrt{2\pi\sigma_n^2}$ as is shown in (5.5) and the equation can be simplified to (5.6) by cancelling this common term in the numerator and the denominator. The identity

$$\ln\left(\frac{x}{y}\right) = \ln(x) - \ln(y) \tag{5.8}$$

is used to remove the division and to produce (5.7). A further simplification is possible by using (4.47) to remove the summations in (5.7). When the summation adds two symbols, α_1 and α_2 and if we let

$$x = \left(\frac{-|\alpha_1 - \hat{b}_k|^2}{2\sigma_n^2}\right), \tag{5.9}$$

and

$$y = \left(\frac{-|\alpha_2 - \hat{b}_k|^2}{2\sigma_n^2}\right), \tag{5.10}$$

the resulting equation will be

$$\ln\left(e^x + e^y\right) = \max(x,y) + \ln\left(1 + e^{-|x-y|}\right). \tag{5.11}$$

(5.11) can be re-applied for each necessary addition that forms part of the summation by taking the result of the previous addition as x and updating y for the next symbol in the constellation. This simplification eliminates the need to use the natural logarithm function as well as the exponent function for every symbol. The term in (5.11) that relies on an exponent and a natural logarithm function, is a correction factor and can be replaced with a lookup table.

5.3.3.3 MIL-STD-188-110D App. D waveforms

MIL-STD-188-110D App. D waveform number 1 and waveform number 2 both make use of BPSK symbol mapping. For BPSK symbol mapping (5.7) can be further simplified by noting there is only one symbol in the set $\alpha \Rightarrow \alpha^i = 0$ and one symbol in the set $\alpha \Rightarrow \alpha^i = 1$. (5.7) becomes [13]

$$L({b_k}^i) = \ln\left[\sum_{\alpha\Rightarrow\alpha^i=-1} e^{\left(\frac{-|\alpha_0-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right] - \ln\left[\sum_{\alpha\Rightarrow\alpha^i=1} e^{\left(\frac{-|\alpha_4-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right] \tag{5.12}$$

$$= \ln\left[e^{\left(\frac{-|\alpha_0-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right] - \ln\left[e^{\left(\frac{-|\alpha_4-\hat{b}_k|^2}{2\sigma_n^2}\right)}\right] \tag{5.13}$$

$$= \left[\frac{-|\alpha_0-\hat{b}_k|^2}{2\sigma_n^2}\right] - \left[\frac{-|\alpha_4-\hat{b}_k|^2}{2\sigma_n^2}\right] \tag{5.14}$$

$$= \left[\frac{-|1-\hat{b}_k|^2}{2\sigma_n^2}\right] - \left[\frac{-|-1-\hat{b}_k|^2}{2\sigma_n^2}\right] \tag{5.15}$$

$$= \left[\frac{-1+2\hat{b}_{kI}-\hat{b}_{kI}^2-\hat{b}_{kQ}^2}{2\sigma_n^2}\right] - \left[\frac{-1-2\hat{b}_{kI}-\hat{b}_{kI}^2-\hat{b}_{kQ}^2}{2\sigma_n^2}\right] \tag{5.16}$$

$$= \frac{2\hat{b}_{kI}}{\sigma_n^2}, \tag{5.17}$$

where $\hat{b}_{kI}$ and $\hat{b}_{kQ}$ are the in-phase and quadrature components of the complex symbol $\hat{b}_k$ respectively.

5.3.4 Deinterleaving

The deinterleaving function is mandated by the standards and allows little flexibility in term of design. At the transmitter interleaving is performed by loading transmit bits into an memory block of a specified size while modulo-P incrementing the load location after each load, where P is the interleaver block size [3]. The interleaver fetch is performed by loading the bits out of the interleaver in the order in which they have been stored in memory. To invert the interleaving operation, the deinterleaver loads bits into a memory block in the order they are received. When the bits are fetched form the memory block, the deinterleaver will also modulo-P increment the fetch index by the same number as it did for the interleaver load step.

Table 5.3. MIL-STD-188-110D App. D 3 kHz bandwidth waveform 1 and 2 interleaver block size and increment values for all interleaver size options.

Bandwidth (kHz)	Waveform number	Interleaver option	Equaliser frames	Number of bits (P)	Increment value
3	1	Ultrashort	4	192	25
3	1	Short	16	768	97
3	1	Medium	64	3072	385
3	1	Long	256	12288	1543
3	2	Ultrashort	4	192	25
3	2	Short	16	768	97
3	2	Medium	64	3072	385
3	2	Long	256	12288	1543

5.3.4.1 MIL-STD-188-110D App. D deinterleaver

The interleaver parameters for the implemented waveforms are shown in Table 5.3. If for example waveform number 2 is used with a long interleaver, the receiver will need to wait for 256 equalisation steps which will produce 12288 LLRs. Once all of the LLRs have been loaded into the memory block, the deinterleaver will fetch LLRs from the memory block while modulo-12288 incrementing the fetch index by 1543 after every fetch.

5.3.5 De-puncturing

De-puncturing is a simple operation performed by adding erasures to the received and deinterleaved stream of LLRs. The erasures are added according to the puncture mask and indicate that the specific LLRs were not transmitted and that no information is available for those LLRs. The erasures equally value 0 and 1 bit values and therefore do not influence the decisions made by the decoder. Neither waveform number 1 nor waveform number 2 of the 3 kHz group of MIL-STD-188-110D App. D make use of puncturing.

5.3.6 Combining of coded bits

At the transmitter encoded data bits are repeated multiple times for some of the MIL-STD-188-110D App. D waveforms to achieve lower code rates and more robust wave-

Table 5.4. Coded data bit repetitions and code rates for 3 kHz MIL-STD-188-110D App. D waveform number 1 and waveform number 2.

Bandwidth (kHz)	Waveform number	Code rate	Number of repetitions
3	1	$\frac{1}{8}$	4
3	2	$\frac{1}{4}$	2

forms using the same half-rate convolutional encoder. Table 5.4 shows the number of repetitions as well as the resulting code rate for the selected waveforms. Both waveform number 1 and waveform number 2 make use of repetitions. Waveform number 1 contains more repetitions resulting in a more robust waveform, but also a slower data rate.

5.3.6.1 Combining repeated coded bits with MAP decoding

To determine a sensible method for combining repeated bit LLRs before executing the decoder, it is important to consider how the decoder will interpret the LLRs. At the transmitter the repeated bits are interleaved and will be transmitted at different time instances and, in the case of a time varying channel, slightly different channel conditions. At the receiver the bits will be received as part of different receive symbols which will individually be converted to LLRs and deinterleaved. The bits will only be available for combining after the deinterleaver has rearranged the bits into the original order and the de-puncturing step has added erasures for the bits that were not transmitted. When using the MAP algorithm, the transition metric needs to be updated to take multiple received symbols into account by letting

$$\gamma_k\left(s',s\right) = P\left(c_k \,\middle|\, \hat{b}_k^0, \hat{b}_k^1, \ldots, \hat{b}_k^{n-1}\right) P\left(u_k\right), \tag{5.18}$$

where n is the number of repeats and $\hat{b}_k^0, \hat{b}_k^1, \ldots, \hat{b}_k^{n-1}$ is a series of n equalised symbols that contain information relevant to the kth bit and where $\hat{b}_k^0$ is the estimate for the first symbol transmitted that is relevant to bit k and $\hat{b}_k^{n-1}$ is relevant to the last symbol transmitted that is relevant to bit k.

At the transmitter the repeated bits ${c_k}^{r0}, {c_k}^{r1}, \ldots, {c_k}^{rn-1}$ are all equivalent since they all depend entirely on encoded bit c_k. These bits are interleaved and mapped to different unknown data symbols $(b_k^0, b_k^1, \ldots, b_k^{n-1})$ that are transmitted at different time instances. At the receiver these symbols are

estimated by the equaliser. Considering (4.1) the equaliser bases a symbol estimate on a series of received symbols with length $L+1$. The assumption is made, based on the interleaver increment values and reasonable worst case estimate of L, that the equalised symbol estimates ($\hat{b}_k^0$ to $\hat{b}_k^{n-1}$) each depend on a different set of received symbols. The symbol estimates ($\hat{b}_k^0$ to $\hat{b}_k^{n-1}$) are taken to be conditionally independent given that the set of received symbols related to each symbol estimate is known.

The conditional independence allows (5.18) to be rewritten as

$$\gamma_k\left(s',s\right) = P\left(c_k \,|\, \hat{b}_k^0\right) P\left(c_k \,|\, \hat{b}_k^1\right) \ldots P\left(c_k \,|\, \hat{b}_k^{n-1}\right) P\left(u_k\right). \tag{5.19}$$

By substituting (4.39) into (5.19)

$$\gamma_k\left(s',s\right) = P\left(u_k\right) \prod_{i=0}^{x-1} \left[P\left(c_k^i \,|\, \hat{b}_k^0\right) P\left(c_k^i \,|\, \hat{b}_k^1\right) \ldots P\left(c_k^i \,|\, \hat{b}_k^{n-1}\right)\right], \tag{5.20}$$

where x is the number of bits per codeword. Combining is performed by letting

$$P\left(c_k^i \,|\, \hat{b}_k\right) = \left[P\left(c_k^i \,|\, \hat{b}_k^0\right) P\left(c_k^i \,|\, \hat{b}_k^1\right) \ldots P\left(c_k^i \,|\, \hat{b}_k^{n-1}\right)\right]. \tag{5.21}$$

LLRs are used to transfer bit probabilities and the LLR of coded repeated bit $\hat{c}_k^{r0}$ is,

$$L\left(\hat{c}_k^{r0}\right) = \ln\left[\frac{P\left(\hat{c}_k^{r0}=0\right)}{P\left(\hat{c}_k^{r0}=1\right)}\right], \tag{5.22}$$

and

$$L\left(\hat{c}_k^{r0}\right) + L\left(\hat{c}_k^{r1}\right) = \ln\left[\frac{P\left(\hat{c}_k^{r0}=0\right)}{P\left(\hat{c}_k^{r0}=1\right)}\right] + \ln\left[\frac{P\left(\hat{c}_k^{r1}=0\right)}{P\left(\hat{c}_k^{r1}=1\right)}\right] \tag{5.23}$$

$$= \ln\left[\frac{P\left(\hat{c}_k^{r0}=0\right) P\left(\hat{c}_k^{r1}=0\right)}{P\left(\hat{c}_k^{r0}=1\right) P\left(\hat{c}_k^{r1}=1\right)}\right]. \tag{5.24}$$

(5.24) shows that the summation of the LLRs for all repeated coded bits that are related to the same encoder output bit at the transmitter is equivalent to calculating an LLR for the combined coded bit probabilities as shown in (5.21). The decoder can then treat the combined LLR as it would have treated the LLR for a bit that was not repeated.

5.3.7 Decoder

5.3.7.1 Decoder algorithm

The soft output Viterbi algorithm (SOVA) as well as the MAP algorithm were considered for the decoder algorithm. The SOVA is less processing intensive than the MAP but leads to a slight degradation in performance, especially when used with a turbo equaliser [5]. An additional consideration is also that the SOVA is slightly more complex to implement than the MAP. The log-MAP algorithm provides the same performance as the MAP algorithm, but with greatly reduced computational complexity. While

this algorithm is computationally still more complex than the SOVA, it does show better performance. The log-MAP algorithm has the additional property that it allows for a new estimation of the coded symbols to be produced as a by-product of the decoding algorithm [5]. This property eliminates the need for the decoded bits to be re-encoded. The log-MAP algorithm was selected due to this property and due to its optimal performance.

5.3.7.2 Design considerations

Soft weights arriving at the decoder are in LLR form. Since the log-MAP algorithm makes use of bit probabilities, it is necessary to first convert the LLRs to bit probabilities by letting

$$P(c_k = 1) = 1 - P(c_k = 0), \tag{5.25}$$

and then by deriving $P(c_k = 0)$ using [5]

$$L(c_k) = \ln\left[\frac{P(c_k = 0)}{1 - P(c_k = 0)}\right] \tag{5.26}$$

$$e^{L(c_k)} = \left[\frac{P(c_k = 0)}{1 - P(c_k = 0)}\right] \tag{5.27}$$

$$e^{L(c_k)}(1 - P(c_k = 0)) = P(c_k = -1) \tag{5.28}$$

$$0 = P(c_k = -1) - e^{L(c_k)} + e^{L(c_k)}P(c_k = 0) \tag{5.29}$$

$$P(c_k = 0) = \frac{e^{L(c_k)}}{1 + e^{L(c_k)}}. \tag{5.30}$$

$P(c_k = 1)$ can then be calculated by substituting (5.30) into (5.25) which gives

$$P(c_k = 1) = 1 - P(c_k = 0) \tag{5.31}$$

$$= 1 - \frac{e^{L(c_k)}}{1 + e^{LL(c_k)}} \tag{5.32}$$

$$= \frac{1}{1 + e^{L(c_k)}}. \tag{5.33}$$

5.3.7.3 Log-MAP decoder processing steps

After an entire interleaver block has been deinterleaved and the LLRs have been de-punctured and combined, the log-MAP algorithm can be executed through the following steps.

1. For each LLR calculate and save the probability that the received bit was equal to 0 and the probability that the received bit was equal to 1 by using (5.30) and (5.33).
2. For each codeword received as part of the interleaver block, calculate the probability of having received each of the possible codeword options by applying (4.39).

3. Assume that the transition metric for each possible transition is equal to the probability of having received the codeword corresponding to that transition since $P(u_k)$ in (4.38) is always 0.5 and can be omitted.
4. Apply the natural logarithm to the codeword probabilities that are used for the calculation of the transition metrics since the log-MAP algorithm makes use of $\Gamma_k(s', s)$ as calculated by (4.42).
5. Since a full tail-biting decoder is used, set $\mathrm{A}_k(s)$ equal to $\ln(1/S)$, where S is the number of trellis states, for all states in the first trellis stage.
6. Set $\mathrm{B}_k(s)$ equal to $\ln(1/S)$, where S is the number of trellis states, for all states in the last trellis stage.
7. Apply the iterative forward algorithm (4.53) to calculate the forward metric $\mathrm{A}_k(s)$ for each state at each stage in the trellis.
8. Apply the iterative reverse algorithm (4.54) to calculate the reverse metric $\mathrm{B}_k(s)$ for each state at each stage in the trellis.
9. Use the log-MAP equation (4.52) to calculate an LLR for each decoder input bit (u_k).
10. Translate the calculated bit LLRs to bit decisions.

5.4 ITERATIVE RECEIVER PROCESSING FUNCTIONS

Figure 5.2 shows the processing functions relevant to an iterative MIL-STD-188-110D App. D receiver where the equaliser and decoder components share information to improve the accuracy of decisions made by both components. The first iteration starts after an interleaver block of data has been processed by all the conventional receiver functions. The decoder produces a set of updated coded LLRs ($\hat{\mathbf{c}}'$) which are similar to the LLRs that were input to the decoder but contain additional information gathered during the decoding process. The original input LLRs ($\hat{\mathbf{c}}$) are removed from the newly generated LLRs so that only extrinsic information about the coded LLRs ($\hat{\mathbf{c}}^{\mathbf{e}}$) is fed back to the equaliser. Repetition and puncturing is applied just as it would have been at the transmitter to produce first the repeated feedback LLRs ($\hat{\mathbf{c}}^{\mathbf{r}\prime}$) and then the punctured feedback LLRs ($\hat{\mathbf{c}}^{\mathbf{p}\prime}$) which are loaded into the interleaver. After an entire interleaver block has been filled the interleaved feedback LLRs ($\hat{\mathbf{i}}'$) are converted to scrambled symbols means ($\bar{\mathbf{b}}$) and noise variances ($\mathbf{v}$) which are fed to the iterative equaliser. The iterative equaliser produces new equalised symbols based on the fed back symbol means and variances, as well as the original received symbols and channel response that were used during the first equalisation step. The rest of the processing functions of the iterative receiver are identical to that of the conventional receiver.

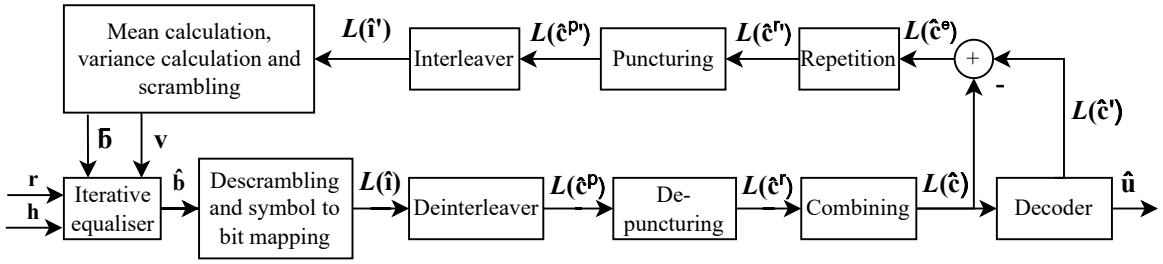

Figure 5.2. Processing functions relevant to an iterative MIL-STD-188-110D App. D waveform receiver where the equaliser and decoder components share information to improve decisions.

The remainder of this section describes the designs for each of the iterative receiver functional blocks. As with the previous section the blocks are described in the order that they are executed when running the receiver.

5.4.1 Updated coded bits

For some soft output decoders, such as the SOVA, it would be necessary to re-encode the decoded bits ($\hat{\mathbf{u}}$) to produce updated estimates of the encoded bits ($\hat{\mathbf{c}}'$). With MAP based algorithms a simple modification to the algorithm can allow for the updated coded LLRs to be directly calculated from the decoder input LLRs. Just as each transition in the trellis is associated with and input bit (u_k) it is also associated with a codeword (c_k) consisting of two coded bits (c_k^0 and c_k^1). This allows (4.52) to be easily modified by replacing the input bit with a coded bit, resulting in [5]

$$L\left(c_k^0 \,\middle|\, \hat{\mathbf{b}}\right) = \ln\left[\frac{\sum\limits_{(s',s)\Rightarrow c_k^0=0} e^{\mathrm{A}_{k-1}(s')+\Gamma_k(s',s)+\mathrm{B}_k(s)}}{\sum\limits_{(s',s)\Rightarrow c_k^0=1} e^{\mathrm{A}_{k-1}(s')+\Gamma_k(s',s)+\mathrm{B}_k(s)}}\right] \tag{5.34}$$

$$= \underset{(s',s)\Rightarrow c_k^0=0}{\mathrm{jmax}} \left[\mathrm{A}_{k-1}\left(s'\right)+\Gamma_k\left(s',s\right)+\mathrm{B}_k\left(s\right)\right] - \underset{(s',s)\Rightarrow c_k^0=1}{\mathrm{jmax}} \left[\mathrm{A}_{k-1}\left(s'\right)+\Gamma_k\left(s',s\right)+\mathrm{B}_k\left(s\right)\right], \tag{5.35}$$

where c_k^0 is the first coded bit in the kth codeword and $(s',s) \Rightarrow c_k^0 = 0$ is the set of all transitions which would result in coded bit c_k^0 being equal to zero. By substituting c_k^0 in (5.35) with c_k^1, the LLR for the other coded bit can also be calculated.

5.4.2 Extracting extrinsic log-likelihood ratios

An important concept of the turbo equalisation principle is that only extrinsic information should ever be transferred between the decoder and the equaliser [21]. Stated otherwise, when the decoder transfers LLRs to the equaliser, the information contained in each LLR must not depend on the corresponding

LLR received from the equaliser. This principle is also valid when the equaliser produces new symbols to be converted to LLRs and handled by the decoder.

The extrinsic LLRs ($L(\hat{\mathbf{c}}^{\mathbf{e}})$) can easily be extracted from the updated coded bit LLRs ($L(\hat{\mathbf{c}}')$) by subtracting the corresponding decoder input LLRs ($L(\hat{\mathbf{c}})$). This is demonstrated by letting the updated LLR correspond to bit number 0 of the kth codeword and by using (4.38) and (4.39) to expand (5.34) to [22]

$$L\left(c_k^0 \middle| \hat{\mathbf{b}}\right) = \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k \middle| \hat{b}_k\right)P(u_k)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k \middle| \hat{b}_k\right)P(u_k)}\right] \tag{5.36}$$

$$= \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\prod_{i=0}^{n} P\left(c_k^i \middle| \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\prod_{i=0}^{n} P\left(c_k^i \middle| \hat{b}_k\right)}\right] \tag{5.37}$$

$$= \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^0 \middle| \hat{b}_k\right)P\left(c_k^1 \middle| \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^0 \middle| \hat{b}_k\right)P\left(c_k^1 \middle| \hat{b}_k\right)}\right] \tag{5.38}$$

$$= \ln\left[\frac{P\left(c_k^0=0 \middle| \hat{b}_k\right)\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}{P\left(c_k^0=1 \middle| \hat{b}_k\right)\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}\right] \tag{5.39}$$

$$= \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}\right] + \ln\left[\frac{P\left(c_k^0=0 \middle| \hat{b}_k\right)}{P\left(c_k^0=1 \middle| \hat{b}_k\right)}\right] \tag{5.40}$$

$$= \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \middle| \hat{b}_k\right)}\right] + L\left(\hat{c}_k^0\right). \tag{5.41}$$

The first term in (5.41) is the extrinsic information for bit 0 of the kth codeword since it is not affected by $L(\hat{c_k}^0)$.

5.4.2.1 Extrinsic log-likelihood ratios for repeated encoded data

For the 3 kHz variants of MIL-STD-188-110D App. D waveform number 1 and waveform number 2 coded bits are repeated before being transmitted. Combining of the coded bits was discussed in

Section 5.3.6.1. Substituting (5.21) into (5.40) produces

$$L\left(c_k^0 \,|\, \hat{\mathbf{b}}\right) = \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \,|\, \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \,|\, \hat{b}_k\right)}\right] + \ln\left[\frac{P\left(c_k^0=0 \,|\, \hat{b}_k^0\right) P\left(c_k^0=0 \,|\, \hat{b}_k^1\right) \ldots P\left(c_k^0=0 \,|\, \hat{b}_k^{n-1}\right)}{P\left(c_k^0=1 \,|\, \hat{b}_k^0\right) P\left(c_k^0=1 \,|\, \hat{b}_k^1\right) \ldots P\left(c_k^0=1 \,|\, \hat{b}_k^{n-1}\right)}\right] \tag{5.42}$$

$$= \ln\left[\frac{\sum_{(s',s)\Rightarrow c_k^0=0} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \,|\, \hat{b}_k\right)}{\sum_{(s',s)\Rightarrow c_k^0=1} \alpha_{k-1}(s')\,\beta_k(s)\,P\left(c_k^1 \,|\, \hat{b}_k\right)}\right] + L\left(\hat{c}_k^{r0,0}\right) + L\left(\hat{c}_k^{r0,1}\right) + \ldots + L\left(\hat{c}_k^{r0,n-1}\right), \tag{5.43}$$

where $L(\hat{c}_k^{ri,n})$ represents the nth repeat of the ith bit in the kth codeword. In this case the extrinsic information for an LLR received from the equaliser, for example $L(\hat{c}_k^{r0,0})$, includes not only the first term in (5.43), but also the information produced by other equalisation stages relating to the same original repeated bit ($L(\hat{c}_k^{r0,1}) + \ldots + L(\hat{c}_k^{r0,n-1})$). It is suggested that the process flow shown in Figure 5.2 can be updated so that the subtraction used to extract extrinsic information is only performed after the fed back LLRs have been repeated, as shown in Figure 5.3. The result will be that the LLRs fed back to an equaliser stage will contain not only the information produced by the decoder, but also information produced by other equalisation stages during the first equalisation pass related to the same original coded bit and based on different sets of received symbols ($L(\hat{c}_k^{r0,1}) + \ldots + L(\hat{c}_k^{r0,n-1})$). This method will be referred to as equaliser extrinsic extraction since the extracted information is extrinsic with reference to the current equaliser stage. The method where extraction is performed on the encoded bits before repeating them is referred to as decoder extrinsic extraction, since the information fed back is extrinsic to the decoder input.

5.4.3 Repetition and puncturing

The puncturing and repetition steps in Figure 5.2 are very similar to the puncturing and repetition steps performed by the transmitter. Based on the number of repeats for the specific waveform, the bits in the extrinsic LLR stream ($L(\hat{\mathbf{c}}^{\mathbf{e}})$) are each repeated multiple times to produce the repeated feedback LLRs ($L(\hat{\mathbf{c}}^{\mathbf{r}'})$). Bits from the repeated feedback LLR stream are then removed based on the relevant puncturing mask to produce the punctured feedback LLRs ($L(\hat{\mathbf{c}}^{\mathbf{p}'})$). No puncturing is used for MIL-STD-188-110D App. D 3 kHz waveform number 1 or waveform number 2.

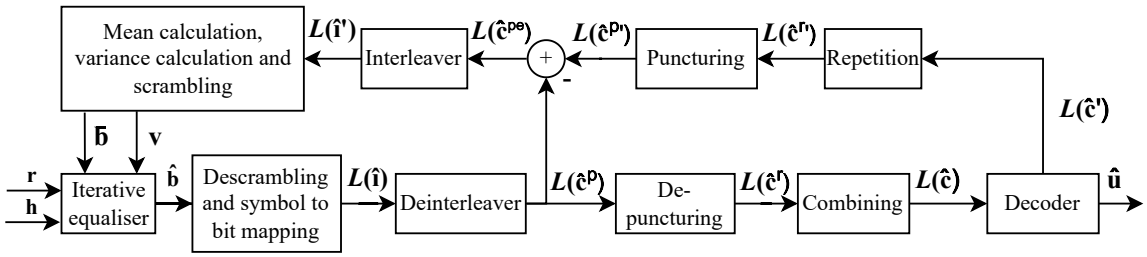

Figure 5.3. Processing functions relevant to an iterative MIL-STD-188-110D App. D waveform receiver where the equaliser and decoder components share information to improve decisions and where the subtraction has been shifted to extract the extrinsic information only after bits have been repeated. This method is referred to as equaliser extrinsic extraction.

5.4.4 Interleaving

The punctured feedback bit estimates ($L(\hat{\mathbf{c}}^{\mathbf{p}'})$) are interleaved to produce the interleaved feedback bit estimates ($L(\hat{\mathbf{i}}')$). The interleaver process is also described in Section 5.3.4. The LLRs are loaded one by one into an interleaver block of a specified size while incrementing the load index by a specified interleaver increment value after each load. This procedure is continued until the entire interleaver block has been filled. LLRs are then fetched from the interleaver block starting at the first index and incrementing the fetch index by 1 after each fetch. The interleaver increment values as well as the interleaver block sizes for the 3 kHz MIL-STD-188-110D App. D waveform number 1 and waveform number 2 waveforms are shown in Table 5.3.

5.4.5 Soft weight to symbol conversion

While most sources perform symbol to soft weight or LLR conversion in a similar manner, various methods have been suggested to convert the LLRs back to symbols or quantities that are usable by the iterative equaliser during iterations [5, 13]. When using a MAP equaliser the fed back LLRs are converted to symbol probabilities, which is a simple operation. The symbol probabilities are then used as a-priori information. Some equalisers require symbol estimates of the fed back information to perform iterations. In these cases the symbols can be calculated by either attempting to invert the symbol to soft weight conversion formula, or by calculating the symbol probabilities and calculating a mean symbol value using the probabilities [5]. The MMSE turbo equaliser and the TDDE equaliser both rely on a-priori mean and variance information [13, 28]. The TDDE was selected as the preferred iterative equaliser method, as will be explained in the following section. As a result the LLRs were used to calculate the mean ($\bar{\mathbf{x}}$) and variance ($\mathbf{v}$) of each feedback symbol by applying (4.67) and (4.69)

respectively.

5.4.6 Scrambling

Scrambling for the feedback symbols is handled in a similar manner to scrambling at the transmitter, which is described in Section 5.3.2. If PSK symbol mapping is used the feedback mean symbols are scrambled before being passed along to the iterative equaliser as $\bar{\mathbf{x}}$. Scrambling is performed on a symbol level by multiplying the mean value with the relevant scrambling symbol.

5.4.7 Iterative equalisation

Multiple examples of turbo equalisers were investigated to determine a well suited option for the MIL-STD-188-110D App. D waveforms. The MAP based turbo equaliser is an example of an optimal equaliser [5]. The complexity of the MAP algorithm increases exponentially as the channel length and the constellation size increases which makes the MAP algorithm less practical for ionospheric channel conditions where long delay spreads can be expected, and completely impractical for use with more densely populated constellation as are present in the MIL-STD-188-110D App. D waveforms. While MAP based equalisers could be feasible for the selected waveforms (waveform number 1 and waveform number 2), it is not suitable for many other waveforms in MIL-STD-188-110D App. D and is therefore not ideal.

The MMSE based turbo equalisers require much less processing power with limited performance degradation [28]. The TDDE is similar to the MMSE turbo equaliser but was designed for waveforms that contain known data as well as unknown probe sections, as are used in MIL-STD-188-110D App. D [13]. The TDDE was selected as it is an appropriate option that could likely scale well to other waveforms in MIL-STD-188-110D App. D. If the LDDE or NDDE equaliser is used for the first equaliser pass matrices $\mathbf{R}^*_M$ and $\mathbf{M_1}$ as well as vector $\mathbf{c}$ can be stored for each decision block since these variables are also used to calculate the equalised symbols during the first iteration of the iterative equaliser.

5.4.7.1 Procedure

The iterative TDDE equaliser will process M feedback means and variances at a time where M is once again the number of symbols in an unknown block. The TDDE algorithm is executed through the following steps.

1. Construct the $\mathbf{V}$ matrix of size $M \times M$ which consists of all zeros except for the main diagonal which contains the M feedback variances as defined in (4.77).

2. Invert the $\mathbf{V}$ matrix to produce $\mathbf{V}^{-1}$.
3. Update the channel correlation matrix using (4.76) where ${\sigma_n}^2$ is the noise variance calculated as described in Section 5.3.3.1 during the previous pass or iteration and where ${\mathbf{M_1}}^H\mathbf{M_1}$ is the channel correlation matrix calculated during the first pass.
4. Invert the channel correlation matrix calculated in the previous step.
5. Perform the following steps for each symbol (k).

   (a) Calculate $\mathbf{f}_k^H$ using (4.79) where $\mathbf{s}_k$ is a selection column vector containing all zeros except for the kth element which contains a one.

   (b) Calculate ρ_k using (4.80) where v_k is the feedback variance related to the symbol k.

   (c) Calculate the estimate for the current symbol using (4.78) where $\mathbf{c}$ is persisted from the first equaliser pass for this decisions block, $\bar{\mathbf{b}}$ is the M feedback means that are being processed as part of this decisions block and $\bar{b}_k$ is the feedback mean corresponding to the current symbol.

5.4.7.2 Non-linear data directed estimation based turbo equaliser

The TDDE method is based on the LDDE algorithm. Along with the LDDE algorithm the NDDE algorithm was also suggested which is shown to perform better in ionospheric conditions [14]. An iterative equalisation method based on the NDDE equalisation method is suggested. This will be referred to as the TNDDE algorithm due to it being a turbo equalisation algorithm based on the NDDE algorithm.

The difference between the TDDE and the TNDDE is that, as with the NDDE, only the first and last symbols are calculated at first for the TNDDE algorithm. The symbol decisions for the first and last symbols are then made and treated as part of the known symbol sections when calculating the next two symbols which are the second and second to last symbols. This process continues until all symbol estimates have been calculated.

The procedure for the TNDDE is similar to the TDDE procedure described in Section 5.4.7.1 and is described below.

1. For each symbol pair, where a symbol pair consists of the first and the last remaining symbols in the unknown block, perform the following steps.

(a) Construct the $\mathbf{V}$ matrix of size $M \times M$ which consists of all zeros except for the main diagonal which contains the M feedback variances as defined in (4.77).

(b) Invert the $\mathbf{V}$ matrix to produce $\mathbf{V}^{-1}$.

(c) Update the channel correlation matrix using (4.76) where ${\sigma_n}^2$ is the noise variance calculated as described in Section 5.3.3.1 during the previous pass or iteration.

(d) Invert the channel correlation matrix calculated in the previous step.

(e) Calculate $\mathbf{f}_k^H$ for the first and the last symbol using (4.79) where $\mathbf{s}_k$ is a selection column vector containing all zeros except for the kth element which contains a one.

(f) Calculate ρ_k for the first and the last symbol using (4.80) where v_k is the feedback variance related to the calculated symbol.

(g) Calculate the estimates for the current symbol pair using (4.78), where $\bar{\mathbf{b}}$ is the M feedback means that are being processed as part of this decisions block and where $\bar{b}_k$ is the feedback mean corresponding to the current symbol.

(h) If PSK symbol mapping is used, descramble the first and last symbol estimates that have been committed as results. This is necessary to allow for more accurate symbol decisions in the next step.

(i) Select the constellation symbols closest to the first and last symbol estimates as the symbol decisions.

(j) If PSK symbol mapping is used, scramble the symbol decisions made for the first and last symbols once again.

(k) Multiply the symbol decisions with the channel response.

(l) Subtract the results of the previous step from $\mathbf{c}$ to produce a new set of received symbols. This equivalent to treating the first and the last symbol as part of the preceding and succeeding known symbol blocks with the symbol decisions as new known symbols.

(m) Adjust $\mathbf{M_1}$ to exclude sections of the channel response that were multiplied by the first and the last symbol.

(n) Adjust the feedback mean ($\bar{\mathbf{b}}$) and variance ($\mathbf{v}$) column vectors by removing the first and last elements.

(o) Reduce M by 2.

5.5 CONCLUSION

Designs have been presented for conventional as well as turbo HF waveform receivers where the conventional receiver also forms part of the first pass of the turbo receiver. The designs were structured for the 3 kHz variants of MIL-STD-188-110D App. D waveform number 1 and waveform number 2, but also with other PSK and QAM waveforms in MIL-STD-188-110D App. D in mind. The NDDE algorithm was selected as the first pass equalisation algorithm due to its near optimal performance. The LDDE algorithm was also selected for comparison. The log-MAP algorithm was selected to perform decoding due to its optimal performance and the ability to produce updated coded bits as a by-product. The TDDE algorithm was selected as the turbo equalisation algorithm since it is based on the DDE algorithms which are well suited for waveforms with known probe sections and for ionospheric channels. An alternative algorithm referred to as the TNDDE algorithm was also be considered and will be used for comparison. Other design considerations include the handling of coded bits repeated at the transmitter and the extraction of extrinsic information for these bits. With a method referred as decoder extrinsic extraction, only extrinsic information from the decoder is fed back to each equaliser stage. A method referred to as equaliser extrinsic extraction feeds back not only extrinsic information from the decoder, but also extrinsic information produced by other equalisation stages when equalising symbols related to repetitions of the same bit.

CHAPTER 6 EXPERIMENTATION

6.1 CHAPTER OVERVIEW

When evaluating the BER performance of a waveform receiver, it is important to conduct tests that are easily re-producible and that can be trusted to generate enough reliable information for the results to be considered significant. MIL-STD-188-110D App. D specifies methods for testing HF modems using baseband HF channel simulators as defined in ITU-R F.1487 [3, 15]. This chapter defines test configurations and procedures that were designed using MIL-STD-188-110D App. D as a reference. Existing HF modems as well as an HF channel simulator produced by RapidM are used as part of the test configuration to contribute to the reliability of the results and to provide a reference against which the designed turbo receiver can be tested. Comparing the results achieved with the turbo receiver to the results achieved with the existing equipment can give an indication of whether the turbo receiver is functioning as expected, or whether either the turbo receiver or the bit error rate measuring tools are showing unexpected behaviour.

Section 6.2 provides a high level description of the tests that will be executed. The functional blocks that form part of the test system are discussed in Section 6.3. Section 6.4 provides information on the RapidM equipment used during the tests. Section 6.5 describes how the equipment is configured and Section 6.6 describes the procedure followed to execute the tests.

6.2 TEST DESCRIPTION

The tests executed to evaluate the performance of the designed iterative equalisation and decoding receivers were strongly influenced by the guidelines and BER performance requirements for AWGN and multipath fading channels defined in MIL-STD-188-110D App. D [3]. MIL-STD-188-110D App. D recommends that modem functionality should be tested at baseband with channel conditions simulated by a baseband channel simulator. MIL-STD-188-110D App. D specifies that BER tests should be

Table 6.1. BER performance requirements for MIL-STD-188-110D App. D waveform number 1 and waveform number 2 with the long interleaver option and on AWGN and MIL-STD-188-110D App. D Poor channels [3]. The SNRs indicated are the average SNRs, measured in the specified bandwidth, for which the waveforms are required to achieve a BER of 10^{-5} or less.

Waveform number	Channel type	SNR (dB)
1	AWGN channel	-3
1	Poor channel	3
2	AWGN channel	0
2	Poor channel	5

conducted with the long interleaver option on AWGN channels and Poor channels. The Poor channels must comply with the Mid Latitude-Distributed Channel as defined in ITU-R F.1487 [15]. The Mid Latitude-Distributed Channel consists of two independently fading paths with a multipath delay of 2 ms between the paths and a fading bandwidth of 1 Hz for each path. The paths have equal mean attenuation and no frequency shifts. The SNR requirements for MIL-STD-188-110D App. D waveform number 1 and waveform number 2 are specified in Table 6.1. Tests on AWGN channels are required to run for at least 60 minutes while tests on Poor channels must run for at least five hours [3].

6.3 FUNCTIONAL BLOCKS

Figure 6.1 provides an overview of the functional blocks relevant to the test environment. The test environment requires only limited external interfaces for the user to configure tests and for the test results to be reported. A user interface provides a convenient method through which tests can be set up and executed. The test controller is the component which runs the tests and performs some status monitoring to ensure that tests are executing as expected. The configuration handler interfaces with the various components in the test environment and exposes an interface which can be used by the user interface and the test controller to control the components and monitor status. The bit error rate tester component generates the test data, downloads the data to the transmit modem and eventually receives data from the turbo equaliser. The received data is compared to the transmit data and results are reported via the configuration handler. The turbo equaliser is the iterative receiver component being tested, as described in Chapter 5. The transmit modem receives user data from the bit error rate tester

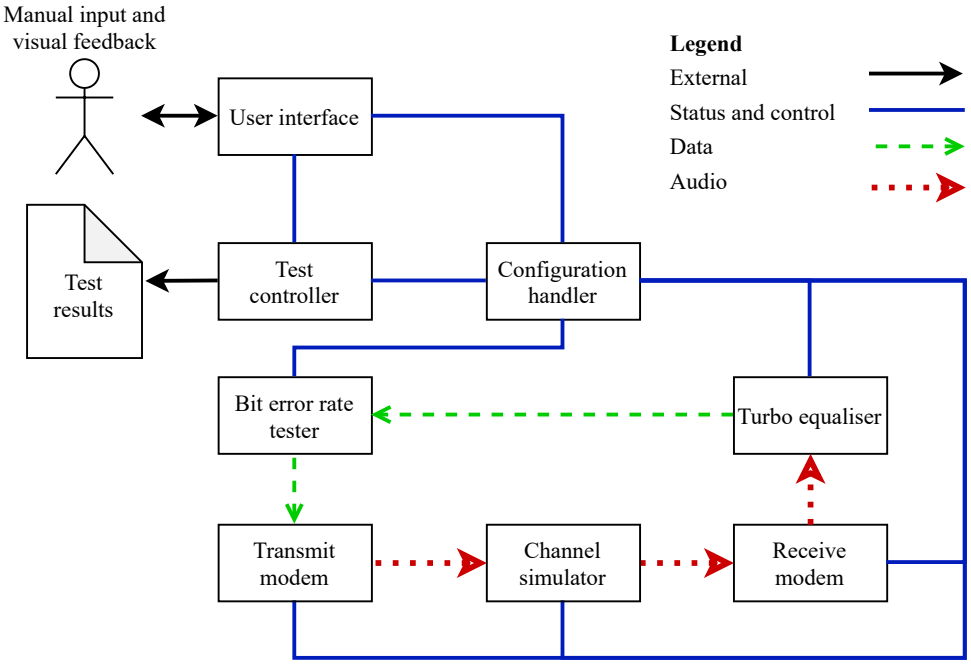

Figure 6.1. Functional block diagram for turbo equaliser BER performance test environment.

component, packs the data into the configured waveform and performs a baseband audio transmission. The channel simulator receives the transmission from the transmit modem and applies the configured channel effects before passing a the audio stream to the receive modem. The receive modem is a fully functioning modem, but also exposes an interface through which the turbo equaliser can gain access to the received symbols and the latest estimated channel response.

6.4 EQUIPMENT SPECIFICATION

Due to the specificity of the requirements laid out in MIL-STD-188-110D App. D and the complications of implementing reliable transmitter and receiver components as well as a baseband channel simulator that could execute for continuous periods of at least 5 hours, it was advantageous to make use of existing baseband modems as well as a baseband channel simulator to increase the reliability of the tests. RapidM is a South African company that specialises in the development of HF and very high-frequency (VHF) modem technology and which loaned the equipment necessary to reliably execute the tests. Further details about the equipment used are provided in the remainder of this section.

6.4.1 Baseband channel simulator

The RapidM RS8 is a baseband HF and V/UHF Channel simulator which supports HF bandwidths of 3 and 6 kHz [30]. The RS8 supports various channel configurations including pre-set configurations

Table 6.2. Specifications for the RapidM RS8 baseband channel simulator used as part of the BER performance tests.

System version	Control ARM version	Silicon serial number
V1.4.4	V4.0.1	570000174EF80101

Table 6.3. Specifications for the RapidM TC5 wideband HF modems used as part of the BER performance tests.

Device role	System version	Silicon serial number
Transmit modem	V6.7.3.5	B6AC16EF162FDCF1
Receive modem	V6.7.3.5[(a)]	8359954A085EC0BF

[(a)] For the receiver modem a slight modification to the software was made to expose the received symbols and channel response.

for the AWGN and Poor channels specified in MIL-STD-188-110D App. D. The RS8 supports two baseband audio interfaces and can be controlled and configured using the proprietary RapidM application protocol 1 (RAP1). Two processors are important for the operation of thee RS8 device. A digital signal processor (DSP) performs the main processing functions and is coupled to the system version of the unit. The so called Control advanced RISC machines (ARM) is used to offload some remote control processing functions from the DSP. Specific information about the RS8 used for testing is provided in Table 6.2.

6.4.2 High frequency modems

The RapidM TC5 is a wideband HF modem module which supports a variety of waveforms with bandwidths of up to 24 kHz, as well as ALE protocols [31]. Among the supported waveforms are the MIL-STD-188-110C Appendix D waveforms, which are identical to the 3 to 24 kHz waveform options in MIL-STD-188-110D App. D. The TC5 is configured and controlled using RAP1. User data can be downloaded to and uploaded from the TC5 module using an RS-232 universal asynchronous receiver-transmitter (UART) interface. The TC5 also supports two baseband audio channels. Specific information about the TC5 modules used for testing is provided in Table 6.3.

6.5 CONFIGURATION

Various configurations are used for the BER performance tests. The two waveforms from MIL-STD-188-110D App. D that are tested are the 3 kHz versions of waveform number 1 and waveform number 2. The preferred waveform is waveform number 2 due to its higher data rate and, as a result, shorter duration required for tests. Waveform number 2 is tested with an AWGN channel as well as a Poor channel and with the long as well as short interleaver options. Waveform number 1 is only tested with a Poor channel and with the long interleaver option. The baseband channel simulator is configured for a 3 kHz bandwidth for all tests. Three iterations of the turbo equaliser were performed during each test and the results for each iteration were recorded. The range of SNR tests points for each test were determined through experimentation and the range was selected to best demonstrate the SNR, within 0.5 dB, at which the BER decreases below 10^{-5}. Waveform number 2 tests were run for 5 hours during which 5 400 000 bits were transmitted at 300 bit/s. Waveform number 1 tests were run for 10 hours during which 5 400 000 bits were also transmitted but at 150 bit/s.

6.5.1 Hardware configuration

The hardware interfaces between the various components as well as some of the external interfaces are shown in Figure 6.2. The BER test application as well as the turbo equaliser application are run on a personal computer (PC). The user can interface with the PC to configure and execute tests and also monitor the status of active tests. After executing a test, a test result is produced as a comma-separated values (CSV) file. The PC makes use of two communication (COM) ports. One COM port is used to download transmit data to the transmit TC5 while the other is used to upload received data from the receive TC5. The PC also has a Ethernet interface (Eth.) which is connected to a layer 2 switch. The layer 2 switch is connected to remote control Ethernet interfaces on the transmit TC5, the receive TC5 and the RS8. The PC uses the Ethernet interface to establish transmission control protocol (TCP) connections to each of the devices. RAP1 is used to control the devices over the TCP connections. When the turbo equaliser is active the received audio samples as well as the channel response will also be received over the RAP1 over TCP connection between the PC and the receive TC5. Baseband audio interfaces are used to transmit data from the transmit TC5 to the RS8 and from the RS8 to the receive TC5.

6.6 TEST PROCEDURE

The steps listed below describe the procedure followed to perform a BER performance test at a single SNR point.

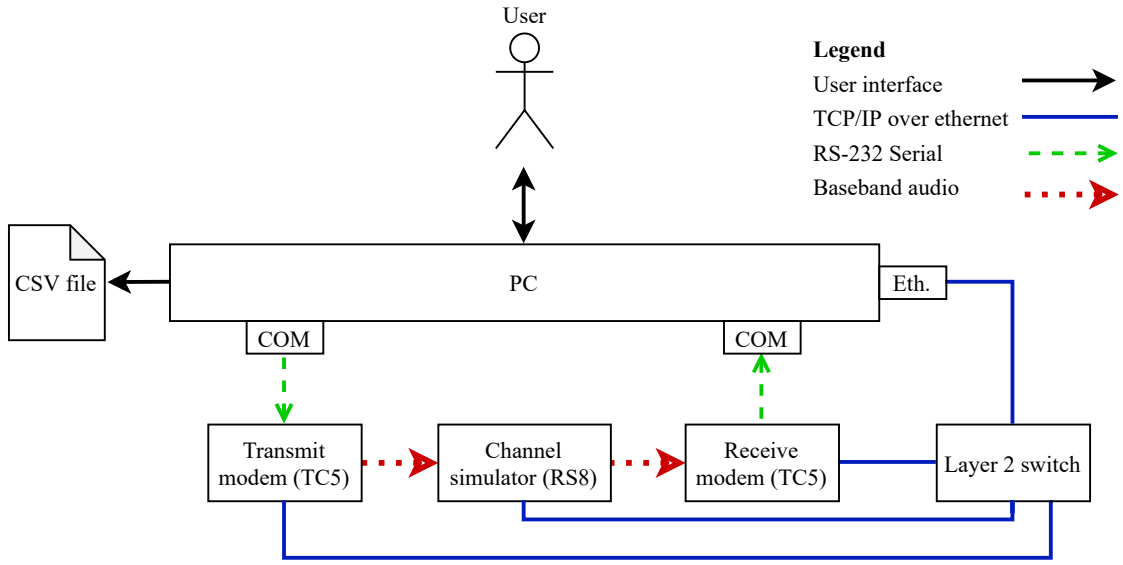

Figure 6.2. Turbo equaliser BER performance test hardware configuration and user interfaces.

1. Configure the RS8 by selecting the appropriate bandwidth and loading the appropriate channel pre-set.
2. Configure the appropriate waveform on both the transmit and the receive TC5.
3. Generate the appropriate number of randomised data bits and start downloading the data to the transmit modem.
4. While downloading data to the transmit TC5, respect flow control to prevent buffer overflows.
5. As the receive TC5 starts producing received samples and channel responses using RAP1 indications, pass the information to the turbo equaliser application running on the PC.
6. Equalise and decode the received symbols with the turbo equaliser application and pass the results to the BER test application.
7. The BER test application records the BER during the test and produces a report in the form of a CSV file after the test has executed.

6.6.1 Establishing a baseline

To gain confidence in the test system and to ensure the reliability of the test results, BER performance tests are first performed with only the RapidM modem components and with the turbo equaliser excluded. Data are downloaded to the transmit modem and uploaded from the receive modem using the RS-232 interfaces. These tests are executed for both the AWGN channel and the Poor channel and the results are compared to the turbo equaliser test results to gain some confidence as to whether the turbo equaliser is performing as expected.

6.7 CONCLUSION

The BER performance tests detailed in this chapter were designed according to the recommendations of MIL-STD-188-110D App. D and also make use of existing baseband HF channel simulators and modems to further improve the reliability of the tests and the credibility of the results. As recommended by MIL-STD-188-110D App. D, tests will be performed at baseband with a baseband channel simulator and with the focus on the Poor channel conditions specified in MIL-STD-188-110D App. D. Some tests will also be conducted with AWGN channel conditions to gain confidence in the test system and in the designed turbo receiver. All MIL-STD-188-110D App. D Poor channel tests will run for at least five hours while AWGN channel tests will run for at least one hour as mandated by the standard. The working region is defined as the SNRs where the measured BER is smaller than 10^{-5}. For each test, the points before and after the measured BER transitions into this region will be determined through experimentation. Measurements will also be made for multiple points preceding the transition into the working region to give an indication of the slope of the graph and to improve the confidence in and the understanding of the results measured.

CHAPTER 7 RESULTS

7.1 CHAPTER OVERVIEW

The results obtained during the BER performance tests described in Chapter 6 are presented in this chapter. Some baseline test results are first presented to give confidence in the test system and the results produced by the system. These tests made use of only the RapidM equipment and not the turbo receiver designed as part of this work. All subsequent results are also accompanied by baseline results recorded from the RapidM equipment during the relevant tests with identical channel conditions. This provides a reference point when comparing results from different tests since the conditions for some of the channels used, especially the Poor channel specified in MIL-STD-188-110D App. D, are subjected to random variations and will not have been exactly the same for any two tests. The waveforms tested were the 3 kHz variants of waveform number 1 and waveform number 2 of MIL-STD-188-110D App. D. Tests were performed in AWGN channel conditions as well as Poor channel conditions. Both long and short interleaver options as defined in MIL-STD-188-110D App. D were tested to determine whether the interleaver length has an impact the performance gains achieved by the turbo receiver. For each test, the SNR at which the BER of the configuration transitions below the 10^{-5} working point was determined through experimentation and measurements were made at this point as well as at a few SNRs before this point. For the first pass equalisation, both the LDDE algorithm and the NDDE algorithm were tested. Tests were also run for two alternate methods of extracting extrinsic data on the feedback path. Both the TDDE and the TNDDE iterative equalisation algorithms were tested. Unless otherwise specified, for each BER performance graph presented in this section, all traces on that graph were generated with identical channel conditions. Traces presented on different graphs can be assumed to indicate separate tests and the baseline results will need to be taken into account when performing a comparison.

Section 7.2 presents the BER test results generated during the various tests and first focuses on the

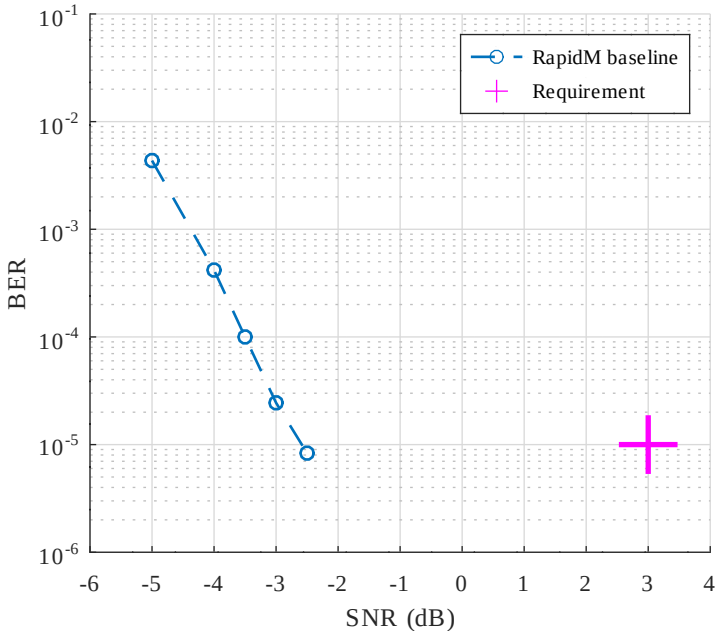

Figure 7.1. BER performance for RapidM equaliser used as baseline with configuration: waveform number: 1; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

baseline test results before presenting the results produced with various permutations of the turbo receiver.

7.2 BIT ERROR RATE TEST RESULTS

7.2.1 Baseline results

The results presented in this section were produced using only the RapidM equipment and with no turbo equalisation or other external receiver components. These results serve as a baseline to gain confidence in the test system and in the other results presented. Figure 7.1 shows the BER performance of MIL-STD-188-110D App. D waveform number 1 running on the RapidM modems with a Poor channel as defined in MIL-STD-188-110D App. D. First it is important to note that the gradient of the trace appears consistent with no significant outliers. This gives confidence that the 5 400 000 bits transmitted to produce each data point were enough to provide an accurate result. It is also important to note that the RapidM modems perform well within the requirements set out in MIL-STD-188-110D App. D for this waveform with the graph showing a BER smaller than 10^{-5} for an SNR of -2.5 dB, which is 5.5 dB below the requirement as indicated in Table 6.1.

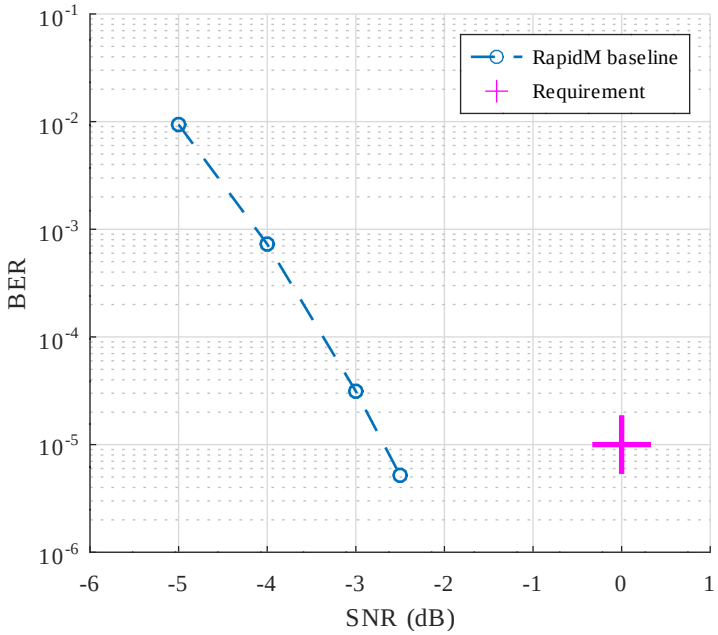

Figure 7.2. BER performance for RapidM equaliser used as baseline with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: AWGN.

Figure 7.2 and Figure 7.3 show the BER performance of the RapidM modem using MIL-STD-188-110D App. D waveform number 2 with an AWGN channel and a Poor channel as defined in MIL-STD-188-110D App. D. The graph in Figure 7.2 once again follows an smooth gradient with no significant outliers. A BER of 10^{-5} is achieved at an SNR of -2.5 dB which is 2.5 dB below the SNR requirement of 0 dB as specified in Table 6.1. From Figure 7.3 it appears that the measured BER value at an SNR of either 1 dB or 1.5 dB could be an outlier. The expectation is that the outlier is likely at an SNR of 1.5 dB since the BER at this point is 3×10^{-6} which might not always be measured accurately during a test that consists of 5 400 000 bits. Despite the outlier it is still very plausible that the RapidM modem will reach a BER smaller than 10^{-5} at an SNR of 1.5 dB, which is 3.5 dB below the requirement.

The baseline results presented in this section indicate that the RapidM modems are capable of functioning well below the SNR requirements as set out by MIL-STD-188-110D App. D and also that the test system can produce reliable results from long BER performance tests. It also appears that in most cases 5 400 000 bits are enough to provide an accurate estimate of the results. At some exception cases slight outliers were present. It was decided that these deviations were not significant enough to warrant longer tests since the duration for waveform number 2 tests are already five hours and the duration

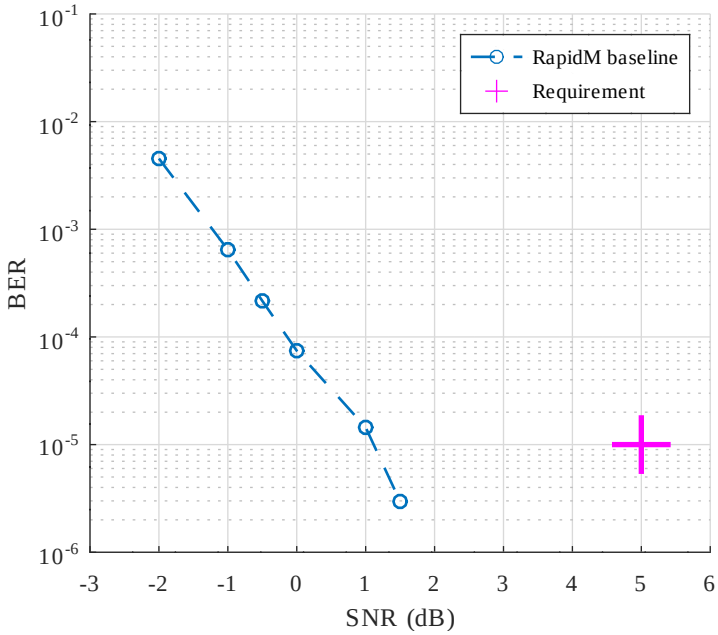

Figure 7.3. BER performance for RapidM equaliser used as baseline with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

of waveform number 1 tests are 10 hours. Testing for longer periods for all data points will not be practical for this work.

7.2.2 Linear data directed estimation with turbo feedback

The iterative receiver used to produce the results presented in this section makes use of the LDDE algorithm for first pass equalisation. On the feedback path extrinsic information is extracted before repeating the feedback LLRs as shown in Figure 5.2. As a result fed back LLRs do not include information from other first pass equalisation stages. The TDDE algorithm is used to perform the iterative equalisation stages. This combination is referred to as TDDE with an LDDE as the first pass equaliser (TDDE-L) during this work. The BER performance of the RapidM modems were also recorded during tests to provide a reference when comparing performance graphs produced by separate tests.

Figure 7.4 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 1 over a Poor channel. The LDDE algorithm reaches a BER of 10^{-5} at an SNR of -1.5 dB, which is well below the 3 dB performance requirement. The first

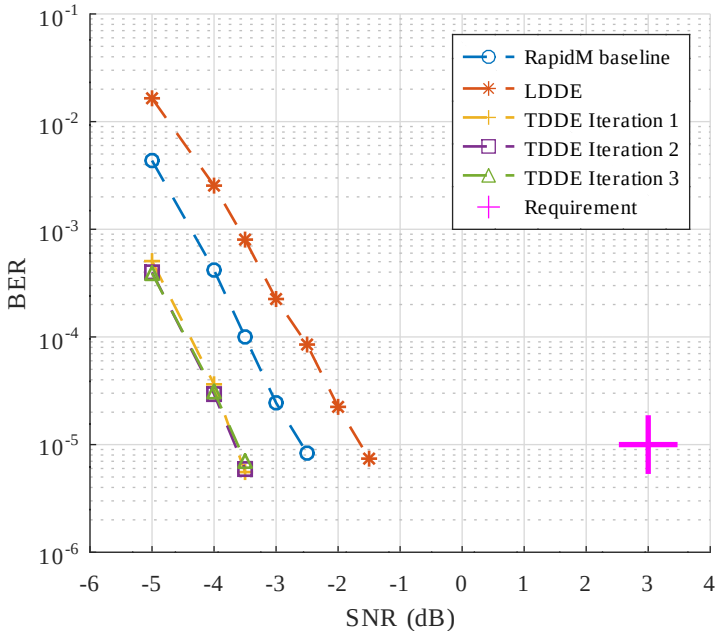

Figure 7.4. BER performance for LDDE based TDDE with configuration: waveform number: 1; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

iteration of the TDDE algorithm further improves the BER performance by reaching below a BER of 10^{-5} at an SNR of -3.5 dB. Further iterations show negligible improvements.

Figure 7.5 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over an AWGN channel. The LDDE algorithm as well as the TDDE iterations all reach a BER of 10^{-5} at an SNR of -2.5 dB, which is well below the 0 dB requirement specified in MIL-STD-188-110D App. D. The TDDE iterations show no improvement over the LDDE first pass, producing almost identical performance.

Figure 7.6 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel. The LDDE algorithm reaches a BER of 10^{-5} at an SNR of 2.5 dB, which is well below the 5 dB performance requirement. It can further be noted that at an SNR of 2 dB the BER performance of the LDDE is already close to the 10^{-5} point and at 2.5 dB the BER is well below 10^{-5}. It therefore appears that the transition point is closer to 2 dB than 2.5 dB. The first iteration of the TDDE algorithm further improves the BER performance by reaching below a BER of 10^{-5} at an SNR of 0 dB. This amounts to a performance improvement

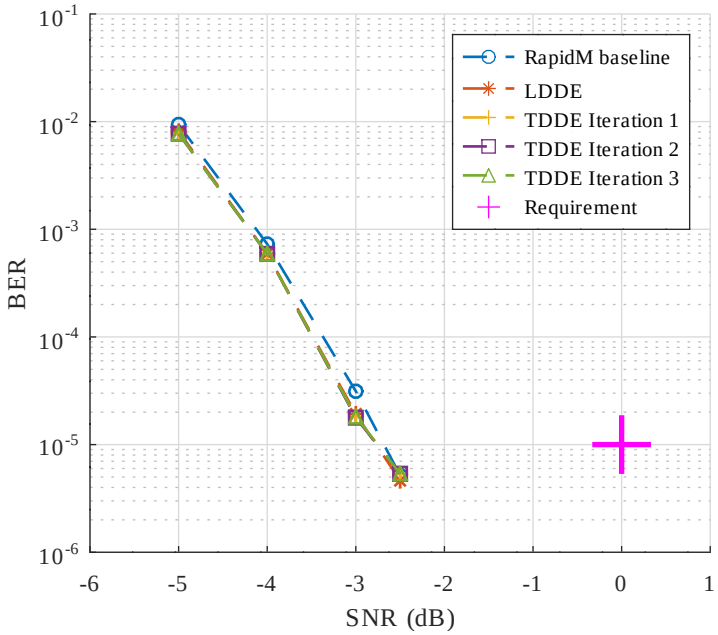

Figure 7.5. BER performance for LDDE based TDDE with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: AWGN.

of at least 2 dB. The BER at an SNR of -0.5 dB is also close to 10^{-5} indicating that the transition is likely close to -0.5 dB. With the second iteration the slight improvement in BER performance allows the TDDE algorithm to reach a BER just smaller than 10^{-5} at -0.5 dB. Subsequent iteration appear to have no significant effect. The gradients of the various traces indicate possible outliers at -0.5 dB and at 1 dB which should be taken into account when interpreting the results.

Figure 7.7 shows the result of transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel, but with the short interleaver option instead of the long interleaver option. It was noted that the results produced using the short interleaver option were much less consistent than that of the Long interleaver option. It was deemed necessary to transmit 10 800 000 bits instead of the 5 400 000 transmitted during other tests. From Figure 7.7 it is apparent that some points still lay slightly outside the expected trend, likely at SNRs of 8 dB and 10 dB. The performance requirement is not indicated since MIL-STD-188-110D App. D does not specify a performance requirement for the short interleaver options. The LDDE algorithm reaches a BER smaller than 10^{-5} at an SNR of 10.5 dB. This is significantly higher than the SNR necessary to achieve a BER below 10^{-5} with the long interleaver option, which is smaller than 2.5 dB as is indicated in Figure 7.6. With the first iteration of the TDDE a BER below 10^{-5} is achieved at 8 dB, indicating a 2.5 dB improvement over the LDDE algorithm.

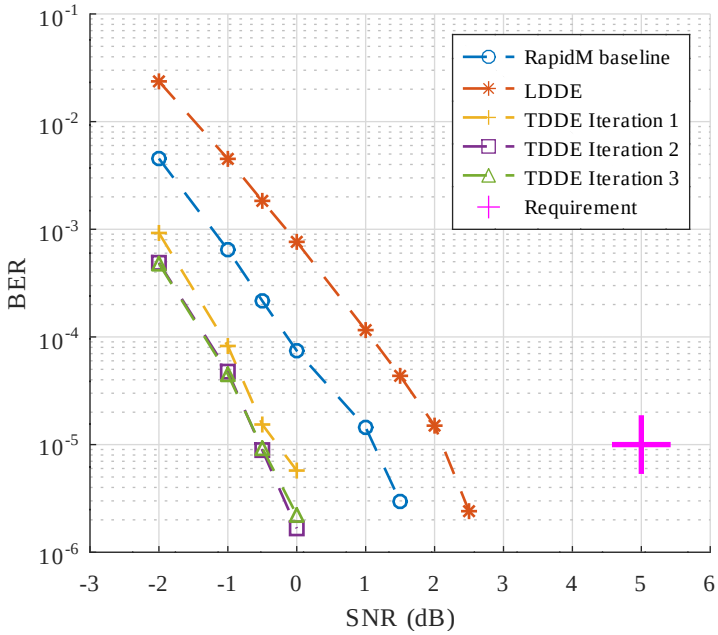

Figure 7.6. BER performance for LDDE based TDDE with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

Subsequent iterations of the TDDE result in no significant change.

7.2.3 Alternative extraction of extrinsic information

The iterative receiver used to produce the results presented in this section makes use of the TDDE-L equaliser scheme. On the feedback path extrinsic information is extracted after repeating the feedback LLRs as shown in Figure 5.3. As a result, fed back LLRs include information from other first pass equalisation stages. This method is referred to as equaliser extrinsic extraction in this work. The BER performance of the RapidM modems were also recorded during tests to provide a reference when comparing performance graphs produced by separate tests.

Figure 7.8 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 1 over a Poor channel. The LDDE algorithm reaches a BER smaller than 10^{-5} at an SNR of -1.5 dB, which is well below the 3 dB performance requirement. The first iteration of the TDDE algorithm further improves the BER performance by reaching below 10^{-5} at an SNR of -3.5 dB. Further iterations of the TDDE result in no significant change. A possible slight outlier is noted at an SNR of -3 dB.

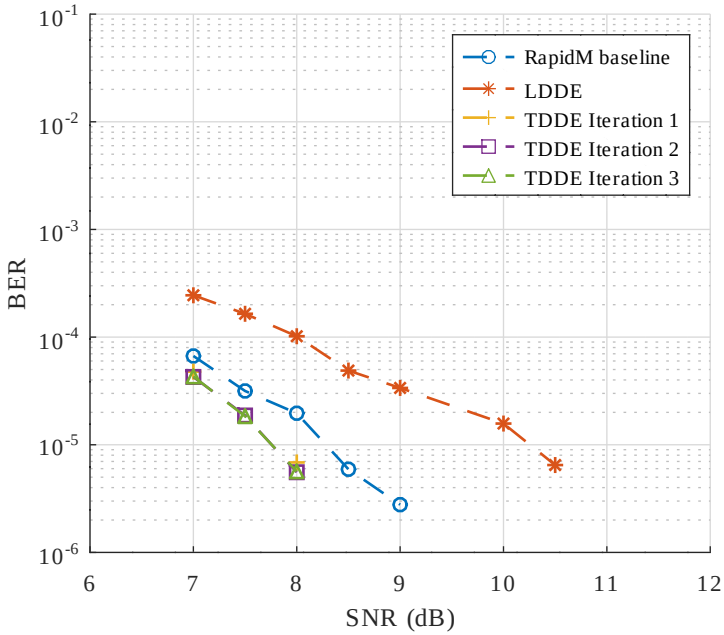

Figure 7.7. BER performance for LDDE based TDDE with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: short; channel conditions: MIL-STD-188-110D App. D Poor channel.

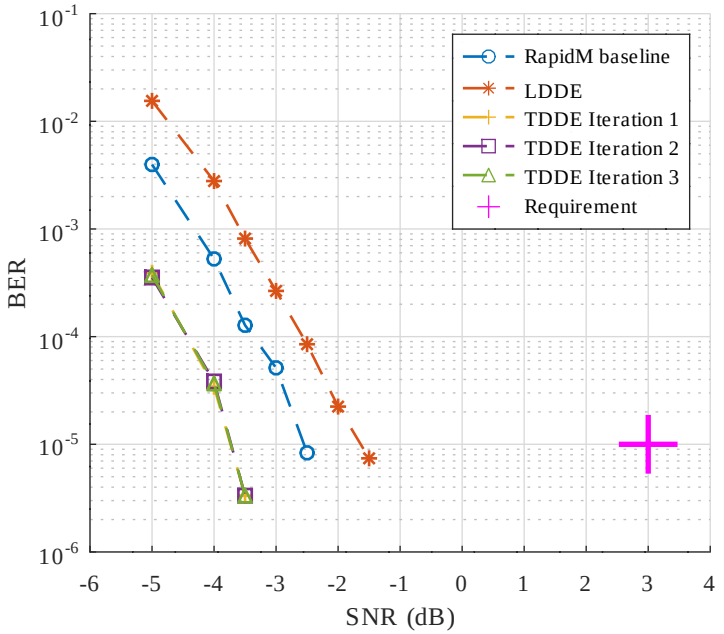

Figure 7.8. BER performance for the TDDE-L receiver using equaliser extrinsic extraction with configuration: waveform number: 1; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

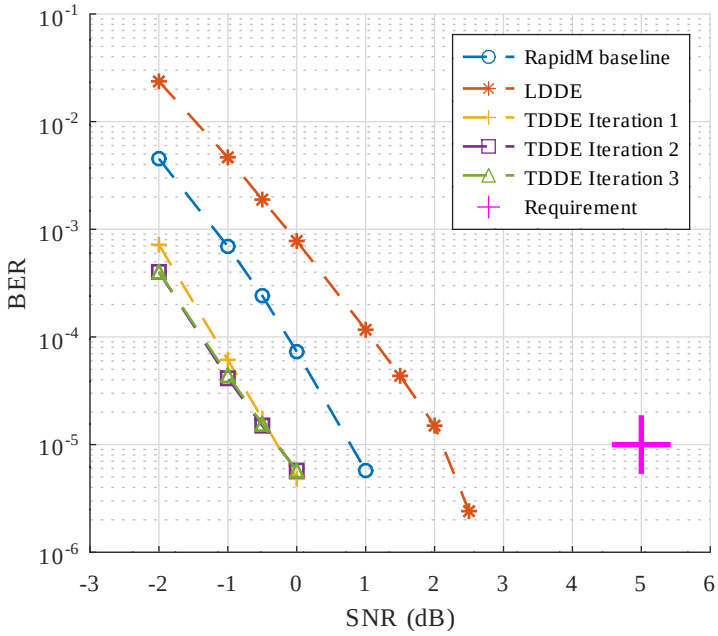

Figure 7.9. BER performance for the TDDE-L receiver using equaliser extrinsic extraction with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

Figure 7.9 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel. The LDDE algorithm reaches below a BER of 10^{-5} at an SNR of 2.5 dB, meeting the 5 dB performance requirement. The transition into the workable region appears to occur close to an SNR of 2 dB. The first iteration of the TDDE algorithm further improves the BER performance by reaching below a BER of 10^{-5} at an SNR of 0 dB. This amounts to a performance improvement of at least 2 dB. The second iteration of the TDDE algorithm appears to provide a very slight improvement. Subsequent iterations show no discernible change in BER performance.

7.2.4 Non-linear data directed estimation with turbo feedback

The iterative receiver used to produce the results presented in this section makes use of the NDDE algorithm for first pass equalisation. On the feedback path extrinsic information is extracted before repeating the feedback LLRs as shown in Figure 5.2. As a result fed back LLRs do not include information from other first pass equalisation stages. The TDDE algorithm is used to perform the iterative equalisation steps. This combination is referred to as the TDDE with an NDDE as the first pass equaliser (TDDE-N) during this work. The BER performance of the RapidM modems were also

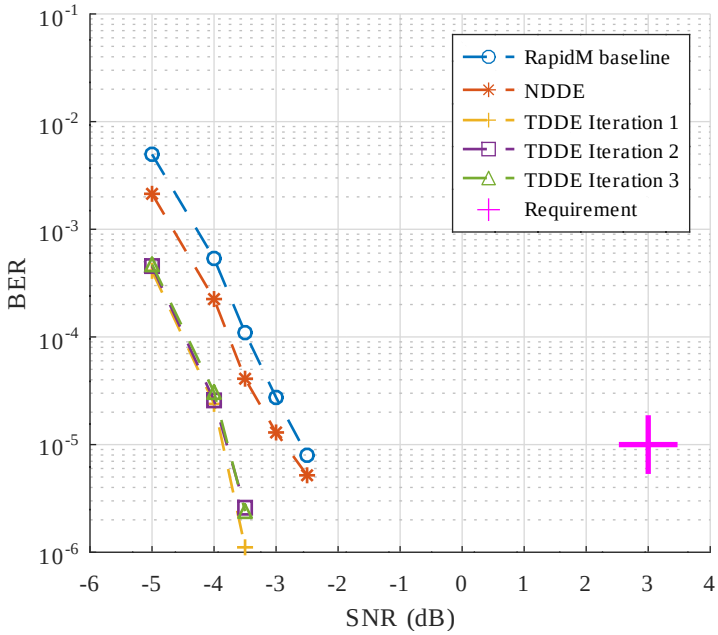

Figure 7.10. BER performance for TDDE-N receiver with configuration: waveform number: 1; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

recorded during tests to provide a reference when comparing performance graphs produced by separate tests.

Figure 7.10 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 1 over a Poor channel. The NDDE algorithm reaches a BER of 10^{-5} at an SNR of -2.5 dB, which is well below the 3 dB performance requirement. The first iteration of the TDDE algorithm further improves the BER performance by reaching below a BER of 10^{-5} at an SNR of -3.5 dB. Further iterations show no improvement and appear to perform worse at an SNR of -3.5 dB. It is important to note that the BER at an SNR of -3.5 dB is close to 10^{-6} and longer tests might be necessary to plot these points accurately. This was not deemed necessary since the workable region is specified at a BER lower than 10^{-5} for the MIL-STD-188-110D App. D waveforms and the transition point is not expected to change significantly.

Figure 7.11 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel. The NDDE algorithm reaches a BER of 10^{-5} at an SNR of 1.5 dB, meeting the 5 dB performance requirement. The transition into the

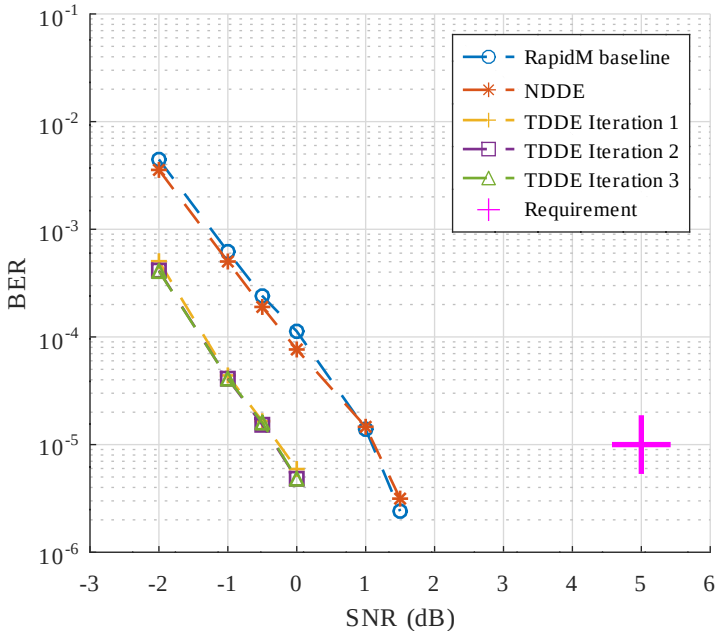

Figure 7.11. BER performance for a TDDE-N receiver with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

workable region appears to occur close to an SNR of 1 dB. The first iteration of the TDDE algorithm further improves the performance by reaching below a BER of 10^{-5} at an SNR of 0 dB. This amounts to a performance improvement of at least 1 dB. Subsequent iterations of the TDDE algorithm appear to provide a very slight improvements in BER performance.

Figure 7.12 shows the result of transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel, but with the short interleaver option instead of the long interleaver option. It was noted that the results produced using the short interleaver option were much less consistent than that of the Long interleaver option. As a result it was necessary to transmit at least 10 800 000 bits instead of the 5 400 000 transmitted during other tests. From Figure 7.12 it is apparent that some points still lay slightly outside the expected trend. The outlier is expected to be the BER measured at an SNR of 8 dB, but due to the limited number of measurements this cannot be stated with confidence. The performance requirement is not indicated since MIL-STD-188-110D App. D does not specify performance requirement for the short interleaver options. The NDDE algorithm reaches a BER smaller than 10^{-5} at an SNR of 8.5 dB. This is significantly higher than the SNR necessary to achieve a BER below 10^{-5} with the long interleaver option, which is smaller than 1.5 dB as is indicated in

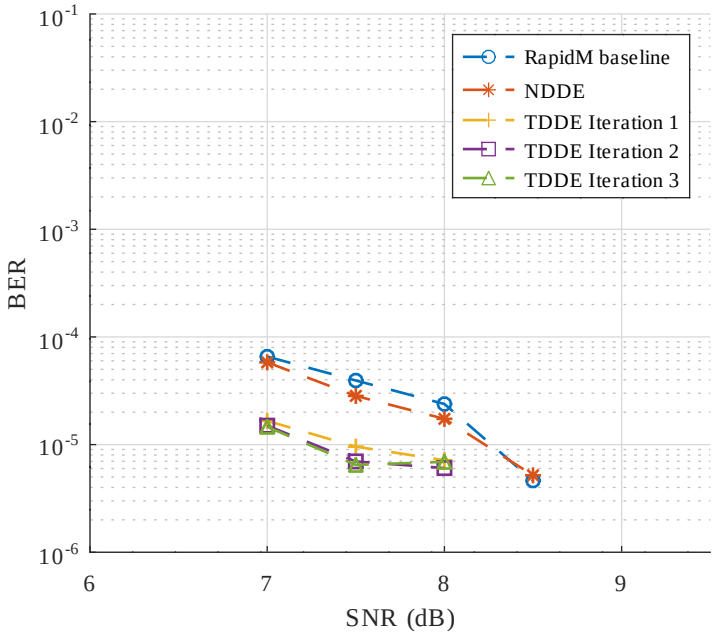

Figure 7.12. BER performance for TDDE-N receiver with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: short; channel conditions: MIL-STD-188-110D App. D Poor channel.

Figure 7.11. With the first iteration of the TDDE a BER below 10^{-5} is achieved at 7.5 dB, indicating a 1 dB improvement over the NDDE algorithm. Subsequent iterations of the TDDE appear to perform slightly better at 7.5 dB.

7.2.5 The turbo non-linear data directed estimation algorithm

Initial testing with the TNDDE algorithm showed that the algorithm is not inclined to converge after multiple iterations. The following section describes the method used to achieve convergence and possibly a more stable version of the algorithm.

7.2.5.1 Symbols in last NDDE calculation

It is noted that iterations of the TDDE algorithm relies on decoder feedback information to improve the results of the initial equalisation pass. With the TNDDE algorithm described in Section 5.4.7.2 the number of feedback symbols used as part of the calculation of new symbol estimates, decreases with each subtraction. For the calculation of the final two symbols only one feedback symbol is available since newly calculated symbols must also be independent of the corresponding fed back symbols and the corresponding symbol feedback mean for the symbol being estimated will be set to 0. It is proposed that the TNDDE algorithm might be more stable if the subtraction process does not reduce the number

Table 7.1. BER performance comparison between versions of the TNDDE algorithm with a varying number of symbols in the final TNDDE calculation at -0.5 dB.

Symbols in final subtract	SNR (dB)	Bit error rate				
		Baseline	First pass	TDDE iteration		
				1	2	3
2	-0.5	2.61×10^{-4}	2.03×10^{-3}	2.93×10^{-5}	2.78×10^{-5}	2.93×10^{-5}
4	-0.5	2.73×10^{-4}	2.03×10^{-3}	2.83×10^{-5}	2.06×10^{-5}	2.57×10^{-5}
6	-0.5	2.50×10^{-4}	1.90×10^{-3}	1.50×10^{-5}	1.46×10^{-5}	1.54×10^{-5}
8	-0.5	2.43×10^{-4}	1.99×10^{-3}	2.37×10^{-5}	1.76×10^{-5}	1.65×10^{-5}

of symbols to two for the last iteration. Four options were tested and the results are summarised in Table 7.1 where the last TNDDE calculation contains either two, four, six or eight samples. All tests were performed with an LDDE algorithm for the first pass equaliser.

From Table 7.1 all four options show very similar results, but the option with six symbols in the last iteration does appear to perform slightly better than the other options when also considering the baseline results. The six symbol option is used for all subsequent TNDDE tests, although not enough information is available to conclusively state whether this option performs better than the other options in all scenarios.

7.2.5.2 The TNDDE algorithm with linear data directed estimation as first pass

The iterative receiver used to produce the results presented in this section makes use of the LDDE algorithm for first pass equalisation. On the feedback path extrinsic information is extracted before repeating the feedback LLRs as shown in Figure 5.2. As a result fed back LLRs do not include information from other first pass equalisation stages. The TNDDE algorithm is used to perform the iterative equalisation steps. This receiver combination is referred to as the TNDDE with an LDDE as the first pass equaliser (TNDDE-L) in this work. The BER performance of the RapidM modems were also recorded during tests to provide a reference when comparing performance graphs produced by separate tests.

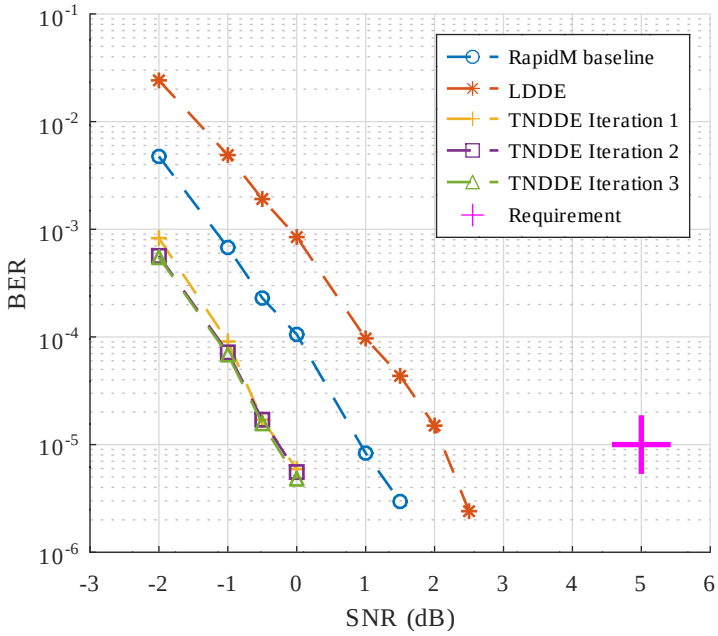

Figure 7.13. BER performance for a TNDDE-L receiver with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

Figure 7.13 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel. The LDDE algorithm reaches below a BER of 10^{-5} at an SNR of 2.5 dB, which is well below the 5 dB performance requirement. It can further be noted that at an SNR of 2 dB the BER performance of the LDDE is already close to the 10^{-5} point and at 2.5 dB the BER is well below 10^{-5}. It therefore appears that the transition point is closer to 2 dB than 2.5 dB. The first iteration of the TNDDE algorithm appears to result in a 2 dB performance improvement. Subsequent iterations do not appear to significantly affect the performance of the TNDDE algorithm.

7.2.5.3 The TNDDE algorithm with non-linear data directed estimation as first pass

The iterative receiver used to produce the results presented in this section makes use of the NDDE algorithm for first pass equalisation. On the feedback path extrinsic information is extracted before repeating the feedback LLRs as shown in Figure 5.2. As a result fed back LLRs do not include information from other first pass equalisation stages. The TNDDE algorithm is used to perform the iterative equalisation steps. This receiver combination is referred to as TNDDE with an NDDE as the first pass equaliser (TNDDE-N) in this work. The BER performance of the RapidM modems were

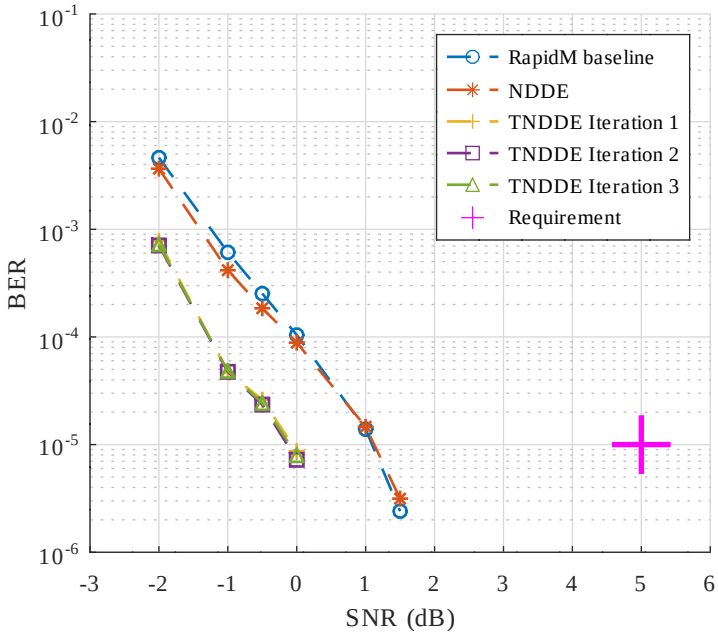

Figure 7.14. BER performance for a TNDDE-N receiver with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel.

also recorded during tests to provide a reference when comparing performance graphs produced by separate tests.

Figure 7.14 shows the BER performance for the iterative equaliser when transmitting MIL-STD-188-110D App. D waveform number 2 over a Poor channel. The NDDE algorithm reaches a BER of 10^{-5} at an SNR of 1.5 dB, meeting the 5 dB performance requirement. The transition into the workable region appears to occur close to an SNR of 1 dB. The first iteration of the TDDE algorithm further improves the performance by reaching below a BER of 10^{-5} at an SNR of 0 dB. This amounts to a performance improvement of at least 1 dB. Subsequent iterations of the TDDE algorithm do not appear to have a significant impact on BER performance. Some of the measurements do appear to be slight outliers and cause the slope of the traces to change at an SNR of -1 dB.

7.3 CONCLUSION

This chapter presented the results obtained by testing the iterative receiver designed in Chapter 5 with the procedures defined in Chapter 6. The 3 kHz variants of MIL-STD-188-110D App. D waveform number 1 and waveform number 2 were tested with long and short interleaver options and with AWGN

as well as Poor channel conditions. Some permutations of the DDE based turbo equaliser were also tested including different methods of extracting extrinsic feedback data, different first pass equaliser algorithms and an alternative iterative equalisation algorithm based on the NDDE algorithm. For most of the graphs presented in this section, the traces appear to follow a natural slope which gives confidence in the designed turbo receiver and the test system and is also an indication that a sufficient number of bits were transmitted as part of each test. The exceptions are the tests with the short interleaver options, which appeared to show widely varying results with repetitions of the tests.

CHAPTER 8 DISCUSSION

8.1 CHAPTER OVERVIEW

This chapter discusses the BER performance results and observations in more detail. Comparisons are presented to evaluate the impact of certain changes to the turbo receiver design, such as the selection of the first pass equalisation algorithm and the selection of the extrinsic data extraction method. The performance of the TDDE receiver is also compared to the TNDDE receiver. For each comparison, the variants are first considered and evaluated individually. This is to quantify the gains provided by a specific variant over the equivalent single pass equaliser. AWGN channel test results are also discussed to provide a level of confidence in the test system, since the turbo iterations are not expected to impact the BER performance in these conditions. Similar BER performance of the turbo iterations in AWGN channel conditions will be an indication that the test system and turbo receiver are working as expected.

Section 8.2 discusses the results of the AWGN channel tests. The impact of the first pass algorithm on the performance of the turbo receiver is discussed in Section 8.3. The two extrinsic data extraction methods and their performance are discussed in Section 8.4. Section 8.5 provides a performance comparison between the TDDE and TNDDE algorithms.

8.2 AWGN CHANNEL CONDITIONS

Figure 7.5 shows the results for the set of tests that were run with AWGN channel conditions. As expected, the equalisation algorithm had little effect on the BER performance on a channel only affected by AWGN. The first pass LDDE equaliser as well as the TDDE iterations showed similar performance. The fact that the RapidM modems provided very similar performance gives confidence in the iterative equalisation based receiver that was designed and implemented.

8.3 IMPACT OF THE FIRST PASS EQUALISER ALGORITHM

The selection of the algorithm used for the in the initial equalisation step has an impact on the certainty of the LLRs fed into the decoder during the first pass and the BER of the decoded data after the first decoding step. The feedback LLRs produced by the decoder will also be affected by the equaliser selection. The two equaliser options that were considered are the LDDE equaliser and the NDDE equaliser. The NDDE algorithm is computationally more complex than the LDDE algorithm, but has been shown to perform better in multi path fading channels [14]. The merit of the NDDE in the context of the iterative equaliser will depend on whether a significant performance improvement can be achieved after one TDDE iteration when compared to the case where the LDDE equaliser is used for the first pass.

8.3.1 Linear data directed estimation with turbo feedback

The results from Figure 7.4, Figure 7.6 and Figure 7.7 are summarised in Table 8.1 for ease of reference. From Table 8.1 it is clear that the TDDE iterations improve on the BER performance of the LDDE equaliser. The magnitude of the performance varies with the waveform number and interleaver selection with both of the MIL-STD-188-110D App. D waveform number 2 interleaver options showing a 2.4 dB improvement after a single iteration and with waveform number 1 showing an improvement of approximately 2 dB. Waveform number 1 and waveform number 2 are very similar in structure except for the number of encoder repeats, of which waveform number 2 has two and waveform number 1 has four. Waveform number 1 can be seen as a more robust version of waveform number 2. It is of interest to note that the more robust waveform allows for less improvement in performance with turbo iterations in this instance. It is also important to consider that the results measured with the short interleaver option were inconsistent and that longer tests might be necessary to accurately compare the short interleaver results with the other options. It is also possible that the short interleaver is not well suited for the Doppler spread of the Poor channel and that the BER is dominated by the fading characteristics instead of the channel SNR. It can further be noted that most of the tests showed no change in performance with subsequent iterations. The case that showed a performance improvement after multiple TDDE iterations was the waveform number 2 long interleaver option test. Even in this case the improvement was estimated to be only 0.3 dB. The TDDE algorithm appears to have a fast convergence and is not expected to require more than one iterations in most cases.

Table 8.1. BER performance of the TDDE-L receiver for MIL-STD-188-110D App. D waveform number 1 and waveform number 2 in Poor channel conditions. The MIL-STD-188-110D App. D required SNR indicates the SNR at which the BER of the applicable waveform must be less than 10^{-5} [3].

Waveform number	Interleaver option	MIL-STD-188-110D App. D required SNR (dB)	Estimated transition SNR (dB)			
			LDDE	TDDE iteration		
				1	2	3
1	Long	3	−1.6	−3.6	−3.6	−3.6
2	Long	5	2.2	−0.2	−0.5	−0.5
2	Short	N/A	10.3	7.9	7.8	7.8

Table 8.2. BER performance of the TDDE-N receiver for MIL-STD-188-110D App. D waveform number 1 and waveform number 2 in Poor channel conditions. The MIL-STD-188-110D App. D required SNR indicates the SNR at which the BER of the applicable waveform must be less than 10^{-5} [3].

Waveform number	Interleaver option	MIL-STD-188-110D App. D required SNR (dB)	Estimated transition SNR (dB)			
			NDDE	TDDE iteration		
				1	2	3
1	Long	3	−2.8	−3.7	−3.6	−3.6
2	Long	5	1.2	−0.2	−0.3	−0.2
2	Short	N/A	8.3	7.5	7.3	7.3

8.3.2 Non-linear data directed estimation with turbo feedback

The results from Figure 7.10, Figure 7.11 and Figure 7.12 are summarised in Table 8.2 for ease of reference. The application of the TDDE algorithm provides a clear improvement over the NDDE equaliser. For waveform number 1 the SNR at which a BER lower than 10^{-5} can be reached is decreased by 0.9 dB. For waveform number 2 the SNR at which a BER lower than 10^{-5} can be reached is decreased by 1.4 dB for the long interleaver option and 0.8 dB for the short interleaver option. The TDDE algorithm appears to converge after 1 iteration and subsequent iterations of the TDDE algorithm appear to provide limited further gains.

8.3.3 Comparison

By comparing the results in Table 8.1 to the results in Table 8.2 it is clear that the NDDE first pass provides much better BER performance than the LDDE first pass. For waveform number 1 the NDDE equaliser can reach below a BER of 10^{-5} at -2.8 dB which is a 1.2 dB improvement over the LDDE equaliser. For waveform number 2 the SNR is improved form 2.2 dB to 1.2 dB which amounts to a 1 dB decrease in required SNR.

The performance results obtained with the first and third iterations of the TDDE-L and TDDE-N receivers using waveform number 2 over a MIL-STD-188-110D App. D Poor channel are shown on the same graph in Figure 8.1. For iterations of the TDDE algorithm the results produced by the receiver with a LDDE equaliser as a first pass and the results produce with an NDDE equaliser as a first pass are very similar. Both receivers showed limited improvement with the second and third iterations, but the TDDE-L receiver did appear to be more inclined to show an improvement. Table 8.3 shows a comparison between the BERs obtained with the TDDE-L receiver and the TDDE-N receiver for SNRs of -1, -0.5 and 0. The baseline results produced with the RapidM modems are also included to provide context to the results. In all cases there are slight variations in the baseline BER results recorded during the LDDE and NDDE equaliser tests. The NDDE equaliser performs an order of magnitude better than the LDDE equaliser during the first pass. During the first iteration of the TDDE equaliser there are only slight variations in most cases between the BER achieved with the LDDE equaliser and the NDDE equaliser. The one exception is at -1 dB where the NDDE equaliser produces approximately half as many errors as the LDDE equaliser. With subsequent iterations the performance of the two equalisers are very similar with the LDDE appearing to correct most of the additional errors at the -1 dB SNR point. At an SNR of -0.5 dB the performance improvement with subsequent iterations of the TDDE-L equaliser appear to allow for the TDDE-L to perform better than the TDDE-N.

8.3.4 Conclusion

From the results presented in this section it is clear that a single iteration of the TDDE equaliser can provide a significant BER performance improvement for transmissions over Poor channels. The TDDE algorithm is also shown to converge after two iterations with the second iteration providing only a minimal improvement in the BER. When comparing the TDDE-L and the TDDE-N results there is

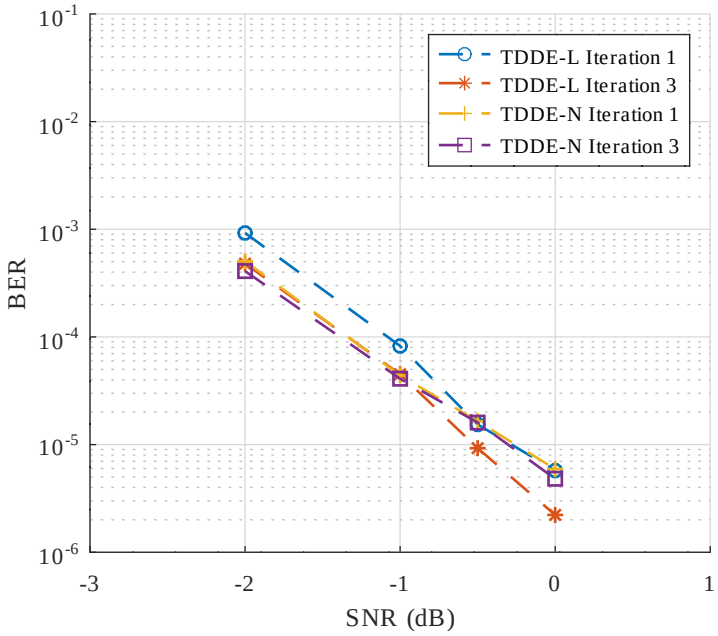

Figure 8.1. BER performance comparison between the TDDE-L receiver and the TDDE-N receiver using decoder extrinsic extraction with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel. The TDDE-L and TDDE-N traces were generated with separate BER tests and the channels were therefore not identical during the course of the tests.

no clear advantage with using the TDDE-N with the long interleaver option despite the increased computational complexity with executing an NDDE rather than an LDDE equaliser during the first pass.

For the short interleaver option the TDDE-N appears to be able to perform 0.4 dB better than the TDDE-L equaliser. It must be noted that the results achieved with the short interleaver option did vary significantly between tests. The slopes of the BER traces for the TDDE-N were also flat and showed little improvement with an increase in SNR. It is possible that the short interleaver is not well suited for the 1 Hz Doppler spread associated with the poor channel and that the BER is dominated by the fading characteristics instead of the SNR of the channel. It is expected that the short interleaver would be better suited for a channel with a higher Doppler spread and faster fading. Repeating the TDDE-L as well as the TDDE-N BER performance tests in such channel conditions could allow for a more accurate comparison between the two algorithms for the short interleaver option.

Table 8.3. BER performance comparison between the TDDE-L method and the TDDE-N method with MIL-STD-188-110D App. D waveform number 2 and a the long interleaver option in Poor channel conditions. The baseline results refer to the results obtained with the RapidM equipment and are included to allow for a more fair comparison.

First pass algorithm	SNR (dB)	Bit error rate				
		Baseline	First pass	TDDE iteration		
				1	2	3
LDDE	-1.0	6.46×10^{-4}	4.51×10^{-3}	8.24×10^{-5}	4.76×10^{-5}	4.50×10^{-5}
NDDE	-1.0	6.20×10^{-4}	5.02×10^{-4}	4.39×10^{-5}	4.09×10^{-5}	4.09×10^{-5}
LDDE	-0.5	2.16×10^{-4}	1.84×10^{-3}	1.54×10^{-5}	8.89×10^{-6}	9.26×10^{-6}
NDDE	-0.5	2.40×10^{-4}	1.90×10^{-4}	1.64×10^{-5}	1.53×10^{-5}	1.60×10^{-5}
LDDE	0.0	7.44×10^{-5}	7.64×10^{-4}	5.74×10^{-6}	1.67×10^{-6}	2.22×10^{-6}
NDDE	0.0	1.12×10^{-4}	7.65×10^{-5}	5.93×10^{-6}	4.81×10^{-6}	4.81×10^{-6}

8.4 EXTRINSIC DATA EXTRACTION ON THE FEEDBACK PATH

A comparison is presented on the results obtained with alternative methods for extracting the extrinsic LLRs on the feedback path of the turbo equaliser.

8.4.1 Decoder extrinsic extraction

The first method is referred to as decoder extrinsic extraction in this work. With this method the extrinsic LLRs are extracted immediately after the decoder has updated the coded data as shown in Figure 5.2. In this case the fed back soft weights are extrinsic relative to the decoder input, which is why it is referred to as decoder extrinsic extraction. The results for this method have already been presented and discussed in Section 8.3.1.

8.4.2 Equaliser extrinsic extraction

The second suggested method for extracting the extrinsic LLRs will be referred to as equaliser extrinsic extraction since the extrinsic information is only extracted after symbols have been repeated on the feedback path as shown in Figure 5.3. The resulting LLRs are extrinsic relevant to the equaliser outputs, which is why it is referred to as equaliser extrinsic extraction. When coded data bits are repeated before transmission the extrinsic feedback LLRs do not only contain information from the decoder, but also information form equaliser stages related to the same transmit coded bit.

Table 8.4. BER performance of the TDDE-L receiver for MIL-STD-188-110D App. D waveform number 1 and waveform number 2 with the equaliser extrinsic extraction method in Poor channel conditions. The MIL-STD-188-110D App. D required SNR indicates the SNR at which the BER of the applicable waveform must be less than 10^{-5} [3].

Waveform number	Interleaver option	MIL-STD-188-110D App. D required SNR (dB)	Estimated transition SNR (dB)			
			LDDE	TDDE iteration		
				1	2	3
1	Long	3	−1.6	−3.6	−3.6	−3.6
2	Long	5	2.2	−0.2	−0.2	−0.2

The results from Figure 7.8 and Figure 7.9 are summarised in Table 8.4 for ease of reference. From Table 8.4 it is clear that the TDDE iterations improve on the BER performance of the LDDE equaliser when using the equaliser extrinsic extraction method. The magnitude of the performance improvement varies with the waveform number with MIL-STD-188-110D App. D waveform number 2 showing a 2.4 dB improvement after a single iteration and with waveform number 1 showing an improvement of 2 dB. The results appear to indicate that the more robust waveform allows for less improvement in performance. No further performance improvements are noted for subsequent iterations of the TDDE. It appears that the TDDE algorithm has a fast convergence and will not require more than one iterations in most cases when used with the equaliser extrinsic extraction method.

8.4.3 Comparison

The performance results obtained with the first and third iterations of the decoder extrinsic extraction TDDE-L and the equaliser extrinsic extraction TDDE-L receivers using waveform number 1 over a MIL-STD-188-110D App. D Poor channel are shown on the same graph in Figure 8.2. Table 8.5 shows a comparison between the BERs obtained with the TDDE-L with decoder extrinsic extraction and the TDDE-L with equaliser extrinsic extraction for MIL-STD-188-110D App. D waveform number 1. The results show that the equaliser extrinsic extraction method always performed slightly better than the decoder extrinsic extraction method after one iteration, even when the baseline results for the equaliser extrinsic extraction method were worse. The decoder extrinsic extraction method appears to show a greater improvement with multiple iterations.

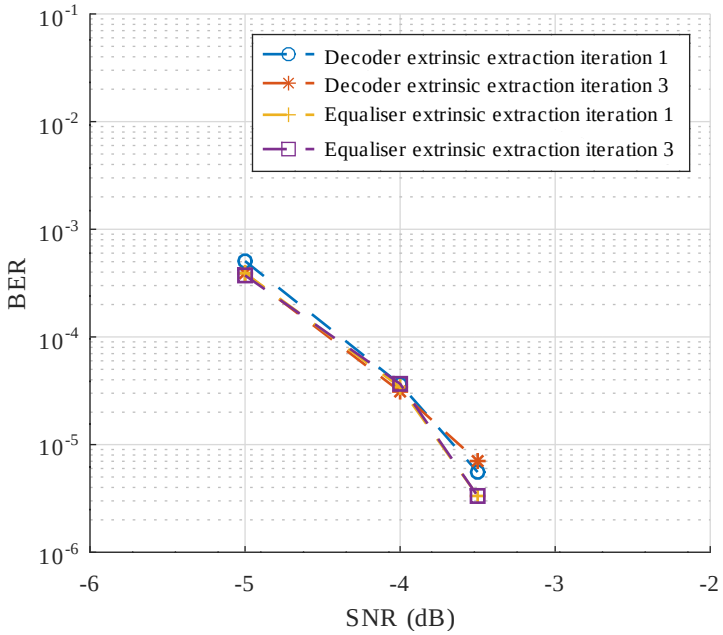

Figure 8.2. BER performance comparison between the TDDE-L receiver using decoder extrinsic extraction and the same receiver using equaliser extrinsic extraction with configuration: waveform number: 1; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel. The decoder extrinsic extraction and equaliser extrinsic extraction traces were generated with separate BER tests and the channels were therefore not identical during the course of the tests.

The performance results obtained with the first and third iterations of the decoder extrinsic extraction TDDE-L and the equaliser extrinsic extraction TDDE-L receivers using waveform number 2 over a MIL-STD-188-110D App. D Poor channel are shown on the same graph in Figure 8.3. Table 8.6 shows a comparison between the equaliser extrinsic extraction method and the decoder extrinsic extraction method with MIL-STD-188-110D App. D waveform number 2. The equaliser extrinsic extraction method performs better after one iteration except for an SNR of -0.5 dB where the decoder extrinsic extraction method has a better baseline performance. The second iteration always results in a performance improvement for the decoder extrinsic extraction method. For the equaliser extrinsic extraction method the second iteration has a less significant impact and even slightly degrades the performance for 0 dB.

Table 8.5. BER performance comparison between the TDDE-L equaliser with decoder extrinsic extraction and the same equaliser with equaliser extrinsic extraction using MIL-STD-188-110D App. D waveform number 1 in Poor channel conditions. The baseline results refer to the results obtained with the RapidM equipment and are included to allow for a more fair comparison.

Extrinsic LLR extraction	SNR (dB)	Bit error rate				
		Baseline	First pass	TDDE iteration		
				1	2	3
Decoder	-5.0	4.34×10^{-3}	1.65×10^{-2}	5.06×10^{-4}	3.98×10^{-4}	3.91×10^{-4}
Equaliser	-5.0	3.40×10^{-3}	1.55×10^{-2}	3.91×10^{-4}	3.54×10^{-4}	3.72×10^{-4}
Decoder	-4.0	4.18×10^{-4}	2.54×10^{-3}	3.63×10^{-5}	2.94×10^{-5}	3.11×10^{-5}
Equaliser	-4.0	5.27×10^{-4}	2.79×10^{-3}	3.43×10^{-5}	3.83×10^{-5}	3.67×10^{-5}
Decoder	-3.5	1.00×10^{-4}	8.02×10^{-4}	5.56×10^{-6}	5.93×10^{-6}	7.04×10^{-6}
Equaliser	-3.5	1.27×10^{-4}	8.15×10^{-4}	3.33×10^{-6}	3.33×10^{-6}	3.33×10^{-6}

Table 8.6. BER performance comparison between the TDDE-L equaliser with decoder extrinsic extraction and the same equaliser with equaliser extrinsic extraction using MIL-STD-188-110D App. D waveform number 2 in Poor channel conditions. The baseline results refer to the results obtained with the RapidM equipment and are included to allow for a more fair comparison.

Extrinsic LLR extraction	SNR (dB)	Bit error rate				
		Baseline	First pass	TDDE iteration		
				1	2	3
Decoder	-1.0	6.46×10^{-4}	4.51×10^{-3}	8.24×10^{-5}	4.76×10^{-5}	4.50×10^{-5}
Equaliser	-1.0	6.93×10^{-4}	4.66×10^{-3}	6.13×10^{-5}	4.13×10^{-5}	4.41×10^{-5}
Decoder	-0.5	2.16×10^{-4}	1.84×10^{-3}	1.54×10^{-5}	8.89×10^{-6}	9.26×10^{-6}
Equaliser	-0.5	2.41×10^{-4}	1.89×10^{-3}	1.74×10^{-5}	1.50×10^{-5}	1.56×10^{-5}
Decoder	0.0	7.44×10^{-5}	7.64×10^{-4}	5.74×10^{-6}	1.67×10^{-6}	2.22×10^{-6}
Equaliser	0.0	7.30×10^{-5}	7.78×10^{-4}	4.81×10^{-6}	5.74×10^{-6}	5.74×10^{-6}

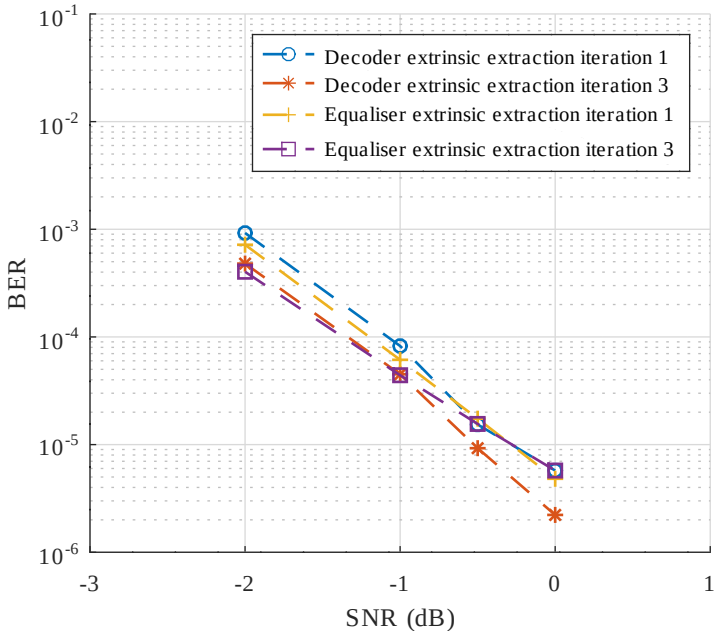

Figure 8.3. BER performance comparison between the TDDE-L receiver using decoder extrinsic extraction and the same receiver using equaliser extrinsic extraction with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel. The decoder extrinsic extraction and equaliser extrinsic extraction traces were generated with separate BER tests and the channels were therefore not identical during the course of the tests.

8.4.4 Conclusion

The decoder extrinsic extraction and equaliser extrinsic extraction methods allow for very similar BER performance. The equaliser extrinsic extraction method appears to perform better after one iteration while subsequent iterations can result in a more significant improvement for the decoder extrinsic extraction method. The equaliser extrinsic extraction method appears to perform slightly better in the case of MIL-STD-188-110D App. D waveform number 1 where coded data bits are repeated four times The decoder extrinsic extraction method appears to perform slightly better with waveform number 2 where the second iteration provides a slight improvement in performance. Despite the advantages and disadvantages of each method, the difference in performance is minimal and it might be sensible to select the same method for all waveforms to simplify implementation.

8.5 THE TURBO NON-LINEAR DATA DIRECTED ESTIMATION ALGORITHM

The TNDDE algorithm is an adaptation of the TDDE algorithm which relies on a similar mechanism as the NDDE algorithm, as described in Section 5.4.7.2. BER performance results obtained with the

application of the TNDDE algorithm are discussed in this section.

8.5.1 Linear data directed estimation for first pass equalisation

The TNDDE algorithm was first tested with an LDDE algorithm for the first pass. This algorithm is also referred to as TNDDE-L. From Figure 7.13 it can be noted that the first iteration achieves a BER of 10^{-5} at an SNR of approximately -0.2 dB which is a 2.4 dB improvement over the first pass equaliser which reaches below a BER of 10^{-5} at 2.2 dB. The improvement from subsequent equaliser iterations appear to be less than 0.1 dB.

8.5.2 Non-linear data directed estimation for first pass equalisation

The TNDDE algorithm was also tested with an NDDE algorithm for the first pass. This algorithm is referred to as TNDDE-N. Figure 7.14 shows that the TNDDE algorithm was able to reach below a BER of 10^{-5} at an SNR of approximately -0.1 dB which is a 1.3 dB improvement over the first pass equaliser which reaches below a BER of 10^{-5} at 1.2 dB. The improvement with subsequent iterations of the TNDDE algorithm does not appear to exceed 0.1 dB.

8.5.3 Comparison

The performance results obtained with the first and third iterations of the TDDE-L and TNDDE-L receivers using waveform number 2 over a MIL-STD-188-110D App. D Poor channel are shown on the same graph in Figure 8.4. The performance results obtained with the first and third iterations of the TDDE-N and TNDDE-N receivers using waveform number 2 over a MIL-STD-188-110D App. D Poor channel are also shown on the same graph in Figure 8.5. Table 8.7 shows a comparison between the BERs obtained with the TDDE-L algorithm, the TDDE-N algorithm the TNDDE-L algorithm and the TNDDE-N algorithm for SNRs of -1, -0.5 and 0. The baseline results produced with the RapidM modems are also included to allow for a fair comparison. In all cases there are slight variations in the baseline BER results recorded during the tests. During the first pass the TNDDE-N and TDDE-N methods perform an order of magnitude better than the other algorithms which make use of LDDE equalisers for the first pass. The TNDDE methods appear to perform consistently worse than their TDDE counterparts at all SNRs measured, despite the increased computational complexity.

8.5.4 Conclusion

The TNDDE-L and TNDDE-N methods do not appear to be a viable alternative for the TDDE methods due to the lack of performance improvement despite higher computational complexity. One possible

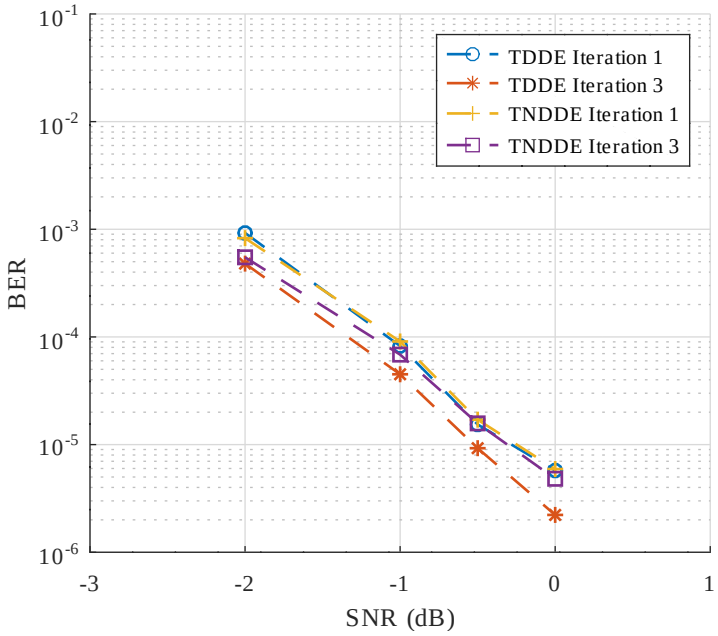

Figure 8.4. BER performance comparison between the TDDE-L receiver and the TNDDE-L receiver using decoder extrinsic extraction with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel. The TDDE-L and TNDDE-L traces were generated with separate BER tests and the channels were therefore not identical during the course of the tests.

reason is that with the subtraction process used in the TNDDE calculations, information fed back from the decoder is also removed. The information from the decoder is expected to be more accurate than the decisions made by the equaliser, which could mean that information lost during the subtractions are more significant than the noise removed.

8.6 CONCLUSION

The AWGN channel test results showed very similar performance for the turbo receiver and the RapidM HF modem receiver, which is an indication that both the turbo equaliser and the test system were functioning as expected. When testing the turbo receiver over MIL-STD-188-110D App. D Poor channel conditions, a 2 dB BER performance improvement was observed for the TDDE over the LDDE after only one iteration, while the TDDE showed a 1 dB improvement over the NDDE algorithm. The TDDE algorithm produced similar performance results after one iteration when used with the LDDE compared to the NDDE algorithm as a first pass, despite the 1 dB performance gain of the NDDE over the LDDE during the first pass. The suggested methods for extracting extrinsic data on the

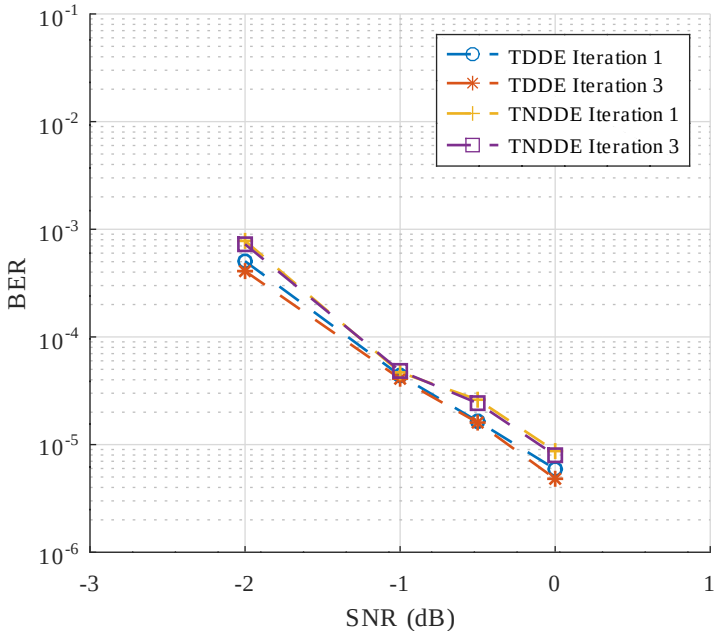

Figure 8.5. BER performance comparison between the TDDE-N receiver and the TNDDE-N receiver using decoder extrinsic extraction with configuration: waveform number: 2; bandwidth: 3 kHz; interleaver option: long; channel conditions: MIL-STD-188-110D App. D Poor channel. The TDDE-N and TNDDE-N traces were generated with separate BER tests and the channels were therefore not identical during the course of the tests.

feedback path showed very similar performance. The suggested TNDDE algorithm did not appear to provide a performance improvement over the TDDE algorithm and in some cases appeared to degrade the performance rather than improve it. A possible explanation is that the subtraction method used in the TNDDE algorithm, which is intended to remove noise, also removes information fed back from the decoder, which could have been more significant than the noise removed.

Table 8.7. BER performance comparison between the TDDE-L algorithm, the TDDE-N algorithm, the TNDDE-L algorithm and the TNDDE-N algorithm in Poor channel conditions. The baseline results refer to the results obtained with the RapidM equipment and are included to allow for a more fair comparison.

Equaliser algorithm	**SNR (dB)**	**Bit error rate**				
		Baseline	**First pass**	**TDDE iteration**		
				1	**2**	**3**
TDDE-L	-1.0	6.46×10^{-4}	4.51×10^{-3}	8.24×10^{-5}	4.76×10^{-5}	4.50×10^{-5}
TNDDE-L	-1.0	6.77×10^{-4}	4.89×10^{-3}	9.07×10^{-5}	7.19×10^{-5}	6.85×10^{-5}
TDDE-N	-1.0	6.20×10^{-4}	5.02×10^{-4}	4.39×10^{-5}	4.09×10^{-5}	4.09×10^{-5}
TNDDE-N	-1.0	6.11×10^{-4}	4.16×10^{-4}	4.69×10^{-5}	4.74×10^{-5}	4.83×10^{-5}
TDDE-L	-0.5	2.16×10^{-4}	1.84×10^{-3}	1.54×10^{-5}	8.89×10^{-6}	9.26×10^{-6}
TNDDE-L	-0.5	2.30×10^{-4}	1.91×10^{-3}	1.70×10^{-5}	1.70×10^{-5}	1.57×10^{-5}
TDDE-N	-0.5	2.40×10^{-4}	1.90×10^{-4}	1.65×10^{-5}	1.53×10^{-5}	1.60×10^{-5}
TNDDE-N	-0.5	2.53×10^{-4}	1.85×10^{-4}	2.61×10^{-5}	2.35×10^{-5}	2.43×10^{-5}
TDDE-L	0.0	7.44×10^{-5}	7.64×10^{-4}	5.74×10^{-6}	1.67×10^{-6}	2.22×10^{-6}
TNDDE-L	0.0	1.06×10^{-4}	8.46×10^{-4}	5.93×10^{-6}	5.56×10^{-6}	4.81×10^{-6}
TDDE-N	0.0	1.12×10^{-4}	7.65×10^{-5}	5.93×10^{-6}	4.81×10^{-6}	4.81×10^{-6}
TNDDE-N	0.0	1.04×10^{-4}	8.85×10^{-5}	8.70×10^{-6}	7.22×10^{-6}	7.96×10^{-6}

CHAPTER 9 CONCLUSION

9.1 GENERAL DISCUSSION

In this work it was shown that the TDDE algorithm can be applied to a MIL-STD-188-110D App. D receiver to achieve improved performance for the robust MIL-STD-188-110D App. D waveform options. It was shown that the TDDE algorithm provides a gain in the order of 2 dB for MIL-STD-188-110D App. D waveform number 1 and MIL-STD-188-110D App. D waveform number 2 over the LDDE algorithm in MIL-STD-188-110D App. D Poor channel conditions. There also appeared to be more gain with less robust waveforms which could mean that the TDDE method will be even more effective for waveform with denser constellations. Various permutations of the TDDE algorithm were tested, including the TDDE-N algorithm and the TNDDE algorithm. The permutations did not show a significant improvement in performance despite their increased computational complexity.

This work presents some suggestions on how encoder repeats can be handled. An equation for handling encoded bit combining for the inputs of a log-MAP decoder is derived. A method for removing only the effects of the relevant equalisation stage is also suggested. With this method the extrinsic information includes not only the decoder information, but also the information from other equalisation stages that are relevant to the same repeated bit. This alternative method of extrinsic data extraction did appear to show a slight improvement when used with MIL-STD-188-110D App. D waveform number 1 which contains four encoder repeats. The improvements shown were not significant enough to strongly motivate the one extrinsic data extraction method over the other.

The following findings were made with reference to the research questions stated in Chapter 1.

- The application of turbo equalisation can result in a significant BER performance improvement

for transmissions over ionospheric channels even when using robust, low data rate BPSK waveforms with encoder repeats. With the TDDE algorithm the performance improvement over the LDDE algorithm exceeded 2 dB, while the performance improvement over the NDDE algorithm exceeded 1 dB.

- Despite the improved performance of the NDDE algorithm over the LDDE algorithm, the performance of the turbo equalisation receivers based on these first pass equaliser algorithms were very similar after one iteration. Due to the difference in computational complexity the LDDE is recommended as the first pass algorithm for most cases.
- Repeated coded bits can be combined at the receiver before performing log-MAP decoding to supply the decoder with more accurate soft inputs and more accurate results. With the equaliser extrinsic extraction method information from other equaliser stages relevant to the same repeated bit can also be included in the feedback information provided to each iterative equaliser stage.
- The gains achievable with the turbo equaliser appeared to be similar with the shorter interleaver options, but the results were inconsistent. It is possible that the short interleaver option is not well suited for the Doppler spread associated with the Poor channel and that the measured BER is dominated by the fading characteristics of the channel. If this is the case, a more accurate comparison can be made by repeating the BER tests for a channel with a higher Doppler spread.

9.2 FUTURE WORK

The MIL-STD-188-110D App. D waveforms form part of new generations of HF communications technology and are expected to be used for a growing number of applications in the near future and beyond. This work contributes to understanding the capabilities of the MIL-STD-188-110D App. D waveform design and can also contribute to decisions that will be made during the design of future waveforms that might replace MIL-STD-188-110D App. D.

The TDDE algorithm can be applied to other MIL-STD-188-110D App. D waveforms to verify whether the same or larger gains can be achieved with waveforms with denser constellations and larger bandwidths. If similar or greater gains can be achieved for these waveforms, it could allow for higher data rate waveforms to be used for specified channel conditions and as a result increase the throughput of MIL-STD-188-110D App. D for a specified channel. The computational complexity of the TDDE algorithm will become an important consideration when expanding to the higher data rate waveforms. It will become important to compare the performance as well as the computational complexity of the TDDE algorithm to other iterative equalisation algorithms in order to determine

the optimal design for the MIL-STD-188-110D App. D waveforms. This can assist with determining what the limitations of the current standard are and when it might be time to develop more suitable set waveforms for future use cases.

Channel estimation is a very important component of equalisation in time varying channels since the equaliser decisions are strongly dependent on the estimated channel response and since the channel response will change over time to accommodate variations in the channel conditions. A more comprehensive study could include channel estimation and could determine the true impact of iterative equalisation when also taking the improvement of the channel estimates into account. It is expected that feeding back the more accurate equalised symbols produced by iterations of the TDDE equaliser to the channel estimator will allow the channel estimator to estimate the channel conditions more accurately, which could further improve the BER performance of the TDDE equaliser.

The HF ionospheric channel varies greatly with time and geographical location. In this work, a channel simulator simulating disturbed channel conditions at mid latitude was used to simulate an ionospheric channel. The channel effects were introduced at baseband in accordance with the modem tests specified in MIL-STD-188-110D App. D. While the Poor channel, as it is defined in MIL-STD-188-110D App. D, which was tested in this work does provide an accurate estimation of the ionospheric conditions at mid latitude, other conditions will also need to be tested to further determine the viability of the TDDE algorithm for those conditions. Channel conditions at high latitudes are particularly challenging and a study of how the TDDE algorithm performs in these conditions could provide a significant contribution. As previously mentioned, the short interleaver option tested during this work does not appear to be well suited for the Poor channel conditions and a more accurate study of the impact of the interleaver length on the TDDE equaliser can possibly be done by performing tests on a channel with a higher Doppler spread. There is also a further opportunity to perform tests using HF modems connected to HF radios transmitting over actual ionospheric channels, instead of baseband simulations.

REFERENCES

[1] M. Uysal and M. Heidarpour, "Cooperative communication techniques for future-generation HF radios," *IEEE Communications Magazine*, vol. 50, no. 10, pp. 56–63, Oct. 2012.

[2] J. Wang, G. Ding, and H. Wang, "HF communications: Past, present, and future," *China Communications*, vol. 15, no. 9, pp. 1–9, Sep. 2018.

[3] *Department of Defence Interface Standard - Interoperability and Performance Standards for Data Modems*, US Dept of Defense Std. MIL-STD-188-110D, 2017.

[4] L. Barclay, *Propagation of Radiowaves*, 2nd ed., ser. Electromagnetic Waves Series. London, UK: Institution of Engineering and Technology, 2003.

[5] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Wireless Channels*, 1st ed. West Sussex, UK: John Wiley and Sons, 2002.

[6] C. Laot, A. Glavieux, and J. Labat, "Turbo equalization: Adaptive equalization and channel decoding jointly optimized," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, pp. 1744–1752, Sep. 2001.

[7] R. Otnes and N. Bauer, "Evaluation of turbo equalization for the high-rate HF waveforms of STANAG 4539," in *Proc. Ninth International Conference on HF Radio Systems and Techniques*, Bath, UK, Jun. 2003, pp. 114–119.

[8] R. Otnes, "Improved receivers for digital high frequency communications: Iterative channel estimation, equalization, and decoding (adaptive turbo equalization)," Ph.D. dissertation, Department of Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway, 2002.

[9] J. W. Nieto, "Investigating the benefits of iterative equalization and decoding of STANAG 4539 HF waveforms," in *Proc. The Institution of Engineering and Technology 11th International Conference on Ionospheric radio Systems and Techniques (IRST 2009)*, Edinburgh, U.K., Apr. 2009, pp. 1–5.

[10] M. A. Elgenedy, E. Sourour, and M. Fikri, "Iterative bi-directional kalman-DFE equalizer for the high data rate HF waveforms in the HF channel," in *Proc. 1st International Conference on Communications, Signal Processing and Their Applications, ICCSPA 2013*, Sharjah, U.A.E., Feb. 2013, pp. 1–6.

[11] M. A. Elgenedy, E. Sourour, and M. Nafie, "Iterative MMSE-DFE equalizer for the high data rates HF waveforms in the HF channel," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2013, pp. 1243–1247.

[12] I. Santos, J. J. Murillo-Fuentes, R. Boloix-Tortosa, E. Arias-De-Reyna, and P. M. Olmos, "Expectation propagation as turbo equalizer in ISI channels," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 360–370, Jan. 2017.

[13] Z. Ma and S. Du, "Data directed estimation based turbo equalization in HF communication," in *Proc. IEEE 17th International Conference on Computational Science and Engineering, CSE 2014*, Chengdu, China, Dec. 2015, pp. 884–889.

[14] F. Hsu, "Data directed estimation techniques for single-tone HF modems," in *Proc. IEEE Military Communications Conference, MILCOM 1985*, Boston, USA, Oct. 1985, pp. 271–280.

[15] *Testing of HF Modems with Bandwidths of up to about 12 kHz using Ionospheric Channel Simulators*, ITU-R Std. F.1487-0 (05/2000), 2000.

[16] *Definitions of maximum and minimum transmission frequencies*, ITU-R Std. P.373-9 (09/2013), 2013.

[17] E. E. Johnson, "The walnut street model of ionospheric HF radio propagation," New Mexico State University, Tech. Rep., 1997.

[18] *Department of Defence Interface Standard - Interoperability and Performance Standards for Medium and High Frequency Radio Systems*, US Dept of Defense Std. MIL-STD-188-141D, 2017.

[19] B. P. Lathi and Z. Ding, *Modern Digital and Analog Communication Systems*, international 4th ed. New York, NY: Oxford University Press, 2010.

[20] J. G. Proakis and M. Salehi, *Communication systems engineering*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[21] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo equalization: An iterative equalization and decoding technique for coded data transmission," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, Jan. 2004.

[22] M. Tuchler and A. C. Singer, "Turbo equalization: An overview," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 920–952, Feb. 2011.

[23] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," in *Proc. IEEE International Conference on Communications, ICC 2002*, New York, USA, Apr. 2002, pp. 664–668.

[24] P. Robertson, E. Villebrun, and P. Hoeher, "Comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE International Conference on Communications, ICC '95*, Seattle, WA, Jun. 1995, pp. 1009–1013.

[25] G. Bauch and V. Franz, "A comparison of soft in/soft out algorithms for turbo detection," in *Proc. International Conference on Telecommunications*, Porto Caras, Greece, Jun. 1998, pp. 259–263.

[26] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and encoding: Turbo-codes (1)," in *Proc. IEEE International Conference on Communications, ICC '93*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[27] C. Douillard, M. Jézéquel, C. B. A. Picart, and P. Didier, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–512, Sep. 1995.

[28] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, Aug. 2002.

[29] B. A. Bjerke, J. G. Proakis, M. K. Lee, and Z. Zvonar, "Comparison of decision feedback equalization and data directed estimation techniques for the GSM system," in *Proc. 6th International Conference on Universal Personal Communications, ICUPC 97*, San Diego, CA, Oct. 1997, pp. 84–88.

[30] *RS8 HF & V/UHF Channel Simulator - 3, 6 kHz Datasheet*, RS8_EN_02E, Rapid Mobile (Pty) Ltd, Pretoria, RSA, 2019.

[31] *TC5 Wideband HF Modem - 24 kHz Datasheet*, TC5_HFWB_EN_01G, Rapid Mobile (Pty) Ltd, Pretoria, RSA, 2020.