Improved receivers for digital High Frequency communications: Iterative channel estimation, equalization, and decoding (adaptive turbo equalization)

Roald Otnes

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOKTOR INGENIØR

Department of Telecommunications
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology

2002

URN:NBN:no-3359

Norwegian University of Science and Technology
Department of Telecommunications
N-7491 Trondheim
Norway

Report no. 420208

ISBN 82-471-5521-4
ISSN 0809-103X

Abstract

We address the problem of improving the throughput and the availability of digital communications in the High Frequency (HF, 3-30 MHz) band. In standardized military waveforms, the data is protected by an error-correcting code (ECC), and the code bits are shuffled by an interleaver and mapped onto a signal constellation for modulation onto a single carrier. Training sequences are multiplexed into the stream of transmitted symbols to aid the receiver in tracking the channel variations. The channel imposes severe time-varying intersymbol interference (ISI) as well as additive noise.

Conventional receivers for such a system would first perform adaptive equalization (to mitigate the ISI) and symbol demapping, deinterleave the received code bits, and finally perform decoding, where the redundancy of the ECC is used to make high-quality decisions on the transmitted data bits even when bit errors have been introduced by the channel. Such a receiver is suboptimal because the equalizer does not make use of the redundancy introduced by the ECC, and is outperformed by an iterative scheme called turbo equalization. In turbo equalization, a.k.a. iterative equalization and decoding, soft information on the code bits is fed back from the decoder to the equalizer in an iterative fashion, and by performing the equalization and decoding tasks several times the bit error rates become significantly smaller than for a conventional "single-pass" receiver.

Since we are dealing with an unknown time-varying channel, we must also perform channel estimation. We include channel estimation in the iterative loop of the turbo equalizer, using soft information fed back from the decoder as "training sequences" *between* the ordinary transmitted training sequences. Then, the receiver performs iterative channel estimation, equalization, and decoding, which can also be called adaptive turbo equalization.

We have proposed a receiver using adaptive turbo equalization, and performed simulations using the MIL-STD-188-110 waveform at 2400 bps, transmitted over an ITU-R poor channel (a commonly used channel to test HF modems). We find that the proposed receiver outperforms a conventional receiver by 2-3 dB in terms of required signal-to-noise ratio to achieve a certain bit error rate.

URN:NBN:no-3359

In this dissertation, we give an introduction to the fields of HF communications and standardized HF waveforms, channel modelling, and turbo equalization. We present an analysis of measured channel data to motivate our research in turbo equalization. We then present our research contributions to the field of turbo equalization: A low-complexity soft-in soft-out equalizer for time-varying channels, a comparative study of channel estimation algorithms using soft information as the input signal, and an investigation of adaptive turbo equalization using a technique known as EXIT charts. Finally, we present our main practical result, which is our proposed receiver and the corresponding simulation results.

URN:NBN:no-3359

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *doktor ingeniør* at the Department of Telecommunications, Norwegian University of Science and Technology (NTNU, Norges Teknisk-Naturvitenskapelige Universitet), Trondheim. My advisors have been Gunnar Stette at NTNU and Torleiv Maseng, a director of research at the Norwegian Defence Research Establishment (FFI, Forsvarets Forskningsinstitutt) at Kjeller, northeast of Oslo, and also professor at the University of Lund, Sweden.

The story started in the late spring of 1999, while I was doing my compulsory military service at FFI. I was unsuspiciously doing research on electronic warfare, until I was one day asked to come to the office of Torleiv Maseng. He wanted to discuss two things. First he said: "I have been appointed to evaluate your master's thesis; why don't you explain the main points to me." The topic was a strange concept called meteor burst communications, so no wonder he needed an explanation. After I had spent some time on that, he continued: "Have you ever considered taking a doctoral degree?" The answer was no, my intentions were to start in a "normal" job as soon as I finished my military service. But now that the idea was planted, it took about a week, including encouraging support from my girlfriend Kirsten and my parents, before I decided that I wanted to do it.

The decision was taken to go for a doctoral degree, but what should the topic be? Torleiv came up with several suggestions, but I did not really fancy any of them before he proposed I should do something related to HF communications. Being a radio amateur who had been practicing on the HF bands in my spare time while studying in Trondheim, I got really enthusiastic. I was put in contact with Vivianne Jodalen and Bjørn Jacobsen, who were working on HF communications at FFI and who could give me enough information to be able to write a project proposal.

Next up came practical issues like funding and where I should take the degree. After some turbulence, we settled down on the following solution: I should take the degree at NTNU in Trondheim, but reside at UniK – University Graduate Center (Universitetsstudiene på Kjeller) next door to FFI. UniK is a relatively small educational and research institution with bonds to the University of Oslo, NTNU, and

several of the research institutes located at Kjeller. I became part of the newly started research program FUCS (FUture Communication Systems) at UniK. In this program, the student should be employed by an industrial company, which should cover the expenses (like travelling and computer equipment) while the salary was covered by a scholarship from the Norwegian Research Council (NFR, Norges Forskningsråd). This arrangement is meant to ensure industrial interest in the research. At the time, Torleiv was in contact with a company now called Kongsberg Defence Communications (KDC), located at Billingstad in Asker, and we agreed that they should become the industrial partner for my doctoral work. So, I was employed by KDC, and the expenses were split between FFI and KDC.

In January 2000 I could finally begin the doctoral work, and started an in-depth study of HF communications. In parallel, I was browsing through publications in communications theory to find out what I could possibly do to improve the performance of HF communications. In the spring of 2001, things finally started to loosen up: I had finished most of the required courses, and had gotten the idea of applying turbo equalization to HF modems. In particular, I had come across some papers by Michael Tüchler of Munich University of Technology, proposing low-complexity solutions for turbo equalization which seemed very suitable for HF communications. In May, 2001, I met Michael at the Vehicular Technology Conference in Rhodes, Greece (a nice venue). It turned out that he was looking for applications for his ideas, and I had an application and needed knowledge on turbo equalization. So, the two of us started cooperating and coauthored several conference papers in 2001 and 2002. And we found that turbo equalization was actually very suitable for HF communications, just as I had suspected. As an exchange, I have been working for about a week in Munich and Michael has been working for about a week at Kjeller in the summer and autumn of 2002. In the summer and autumn of 2001, I was also staying at QinetiQ in Malvern, England, for a total of 4 weeks. I there obtained further knowledge on HF communications, particularly from Paul Cotterill and Kathleen Davies. In the summer of 2001, a new project which should focus on HF communications was started at FFI, lead by Vivianne Jodalen. The project will go on until December, 2004. Although I have not been employed by FFI, I have been a *de facto* member of that project.

During 2002, I have realized that the number of open questions just increases the more I study the problems. So, I decided I should rather write a dissertation containing the knowledge required until now, than wait for all questions to be answered. I have now been appointed to work at FFI for the next two years, continuing research along the lines outlined in this dissertation. An intriguing thought is: Will the number of open questions be larger or smaller after two more years?

My work style during the last three years has been nomadic: When I have not been out travelling, I have spent one day a week at KDC, one day a week at my

home office (I live in Fredrikstad, 100 km south of Oslo), and three days a week at Kjeller, sometimes at UniK and sometimes at FFI. I have been to many conferences, and most of the required courses were taken as a long-range commuter to Trondheim. Being effective in such an environment has been possible due to a laptop computer containing all my work. A heavy toll has been put on that computer: It was new in the autumn of 2000, and since then the motherboard has had to be changed three times due to malfunction.

Acknowledgments

I would like to thank the following people, without whom this dissertation would not have been what it is:

My supervisors: Torleiv Maseng for encouraging me to start this work and for supporting me along the way; Gunnar Stette for supervising a doctoral work outside his main area of interest and organizing everything including a self-defined course in HF communications.

At UniK: Pål Spilling for initialization and administration of the FUCS program; all staff for providing a good work environment; my co-students (in particular Trond Nypan and Asgeir Nysæter) for being useful discussion partners and travelling companions; Nicolai Bauer for starting a master's project related to my work.

At KDC: Tom Christensen for being so positive and organizing my employment; Tore Ulversøy for supporting efforts on future implementation of the ideas presented in this dissertation; my friends at the lunch table for interesting information from the world of hardware design; the entire company for gladly accepting me and inviting me to social events.

At FFI: Vivianne Jodalen for many useful discussions on HF communications, and for providing HF standards and DAMSON data; Bjørn Jacobsen for information on ionospheric physics; Bodil Hvesser Farsund, Terje Johnsen, and Knut Inge Hvidsten for programming support; all the people in the corridor for being kind and making good waffles.

Michael Tüchler, of course.

Odd Trandem at Sintef Telecom and Informatics in Trondheim for programming support and good inputs in the early stages of this work.

For proofreading parts of the dissertation, in addition to people mentioned elsewhere: John Nieto at Harris RF Communications, Rochester, NY, USA, Bjørn Solberg at Thales Communications, Oslo (also a consultant to FFI), Carl Fredrik Leanderson at the University of Lund, and Nils Holte at NTNU. Also Håkon Hanebrekke at NTNU, for saying yes to proofreading a chapter few days before he sadly passed away.

Last, but certainly not least, my family: My soon-to-be wife Kirsten for love and

encouragement, for giving life a meaning more important than research, and for bearing with me; Therese for coming to the world and providing light and astonishment during my process of writing; my late father Will for supporting most of my choices throughout life (such a pity he missed two things he was really looking forward to, namely the birth of his granddaughter Therese and me finishing my doctoral degree); my mother Annelie for English consultations and her husband Knut for running in the woods with me; Kirsten's mother Inger for lots of practical help and cheap rent; Kirsten's sisters Elisabeth and Astrid for baby-sitting; all other relatives and friends of me and Kirsten for providing a good framework to life.

Contents

URN:NBN:no-3359

List of Figures

List of Tables

URN:NBN:no-3359

Notation and symbols

We write column vectors as bold lowercase letters, e.g. $\mathbf{x}$, and matrices as bold uppercase letters, e.g. $\mathbf{X}$. Scalars (real or complex) are written as lowercase Greek or Arabic letters, e.g. x. $\mathbf{X}^T$ denotes the transpose of $\mathbf{X}$, $\mathbf{X}^H$ denotes the complex conjugate transpose (Hermitian transpose) of $\mathbf{X}$, and x^* denotes the complex conjugate of x. The operator Diag$[\mathbf{x}]$ applied to a length N vector produces an $N \times N$ diagonal matrix containing the elements of $\mathbf{x}$, and the operator Diag$[\mathbf{X}]$ applied to an $N \times N$ matrix produces a length N column vector containing the diagonal elements of $\mathbf{X}$.

A signal defined in continuous time is written as $x(t)$. If the signal is sampled in discrete-time, e.g. at the symbol rate $1/T_s$, we represent the dependency on time as a subscript index, e.g. $x_n = x(nT_s)$. Dependencies on several variables, e.g., a time-varying impulse response h which is a function of time and lag, are denoted as follows: When time and lag are continuous we write $h(t, \tau)$, and when time and lag are discrete (after sampling) we write $h_{n,l}$.

When statistics are used, $E\{x\}$ denotes the expectation (average value) of x and $Cov\{x, y\} = E\{xy^*\} - E\{x\}E\{y*\}$ denotes the covariance of x and y. We write the probability density function (pdf) of x as $p_x(x)$, and the conditional pdf of x given y as $p_{x|Y}(x|y = Y)$.

We usually consider a bit to be in the domain $\{+1, -1\}$ rather than in $\{0, 1\}$. The mapping between those domains is that $+1$ corresponds to 0, and -1 corresponds to 1. With this mapping, exclusive-or in the $\{0, 1\}$ domain is equivalent to multiplication in the $\{+1, -1\}$ domain, as can be verified by performing the calculations for all four combinations of two bits.

All signals are complex unless otherwise stated. We use a bandpass equivalent model for all signals and systems, except in the beginning of Chap. 4. The constellation used for a transmitted signal x_n is denoted $\mathcal{S}$, i.e. $x_n \in \mathcal{S}$. We use a bar to denote expectation, $\bar{x}_n = E\{x_n\}$, and a tilde $\tilde{x}_n$ to denote an estimated symbol output from an equalizer. $\tilde{x}_n$ can be any number in the complex plane, and a hard decision $\hat{x}_n$, denoted by a hat, is an estimate of x_n which is equal to one of the complex numbers in the constellation $\mathcal{S}$. In block diagrams we use the symbol shown in Fig. 1 to represent a hard-decision device, converting $\tilde{x}_n$ to $\hat{x}_n$. We also use a hat to denote

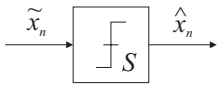


Figure 1 A hard decision device. $\hat{x}_n \in \mathcal{S}$, where $\mathcal{S}$ is a signal constellation, such that $|\tilde{x}_n - \hat{x}_n|$ is minimized.

estimates of the channel impulse response and the noise variance.

Some symbols, which are used only short-term during an argument or a derivation, are not included in the list of symbols below. Note that common letters are sometimes used in other meanings than in the list of symbols; those cases should be clear from the context.

List of symbols

β	Step size of LMS algorithm
ϵ_n	Error signal used for direct adaptation of equalizer
ζ_n	Scrambling symbols
$\boldsymbol{\theta}_n$	Estimated crosscovariance vector, used in channel estimation algorithms
κ	Oversampling rate (relative to symbol rate)
λ	Forgetting factor of channel estimation algorithms
$\mu_{n,i}$	Average value of conditional pdf of $\tilde{x}_n$ given that $x_n = s_i$
ν	Doppler shift
ν_d	Doppler spread
ξ_n	Scrambled transmitted symbols
$\mathbf{\Pi}_n$	Covariance matrix of $\mathbf{z}_n$
$\sigma^2_{i,n}$	Variance of conditional pdf of $\tilde{x}_n$ given that $x_n = s_i$
σ^2_L	Variance of Gaussian distributed LLRs, with mean $\sigma^2_L/2$
σ^2_w	Noise variance
$\sigma^2_{w,n}$	Noise variance at symbol interval n
$\hat{\sigma}^2_{w,n}$	Estimated noise variance at symbol interval n
$\hat{\sigma}^2_{e,n}$	Estimated error variance at symbol interval n
$\mathbf{\Sigma}_n$	Noise covariance matrix
τ	Delay (introduced by channel)
τ_m	Delay spread
ϕ	Phase
φ_n	Cost function for ordinary RLS algorithm

$\mathbf{\Phi}_n$	Estimated covariance matrix, used in channel estimation algorithms
χ_n	Difference between actual transmitted symbol x_n and *a priori* mean $\bar{x}_n$
ψ_n	Cost function for modified RLS algorithm
ω_c	Carrier frequency
a_m	Data bits
$\hat{a}_m$	Estimated data bits (out of decoder)
B	Bandwidth
$c_{k'}$	Code bits
c_k	Interleaved code bits
C	Constraint length of convolutional code
$\mathbb{C}$	The set of all complex numbers
$d_n,\ d_{\eta,n}$	Error signal used for soft iterative channel estimation
$d'_n,\ d'_{\eta,n}$	Error signal used for hard iterative channel estimation
$e_n,\ e_{\eta,n}$	Error signal used for channel estimation
E_h	Energy (average power gain) of channel impulse response
f_a	Information data rate
f_s	Symbol rate
$\mathbf{f}_n$	Time-varying vector of equalizer filter coefficients
$\mathbf{g}_n$	Vector of equalizer filter coefficients when a certain constraint is not fulfilled
$g(D)$	Generator polynomial for convolutional code
$h(t,\tau)$	Channel impulse response (CIR), at time t and delay τ
$h_{n,l}$	Discrete-time CIR, at symbol interval n and delay index l
$\mathbf{h}_n$	Discrete-time CIR vector, $\mathbf{h}_n = [h_{n,0}\ \cdots\ h_{n,M-1}]^T$
$\hat{\mathbf{h}}_n$	Estimated CIR vector at symbol interval n
$h_{n,l,i}$	Fractionally spaced discrete-time CIR
$\mathbf{h}_{n,i}$	Fractionally spaced discrete-time CIR vector, $\mathbf{h}_{n,i} = [h_{n,0,i}\ \cdots\ h_{n,M-1,i}]^T$
$\mathbf{H}$	Channel convolution matrix
$\mathbf{H}_n$	Time-varying channel convolution matrix, at symbol interval n
$I(L;C)$	Mutual information between LLRs and code bits
I_d	Mutual information at output of decoder, $I_d = I(L_e^D(c_{k'}); c_{k'})$
I_e	Mutual information at output of equalizer, $I_e = I(L_e^E(c_k); c_k)$
$\mathbf{I}_N$	$N \times N$ identity matrix
$\mathbf{0}_N$	$N \times N$ all-zero matrix
j	$\sqrt{-1}$

k	Time index for interleaved code bits
k'	Time index for non-interleaved code bits
l	Index for delay, measured in symbol intervals
$L(c_k)$	Log-likelihood ratio (LLR) on the code bit c_k
$L_e(c_k)$	"Extrinsic" LLR
$L_e^D(c_k)$	LLR input to equalizer
$L_e^D(c_{k'})$	LLR output from decoder
$L_e^E(c_k)$	LLR output from equalizer
$L_e^E(c_{k'})$	LLR input to decoder
m	Time index for data bits
M	Length of actual or estimated CIR, measured in symbol intervals
M'	Length of actual CIR when different from length of estimated CIR
n	Time index for transmitted and received symbols
n'	Time index for data symbols (before multiplexing with training symbols)
N	Equalizer filter length, $N = N_1 + N_2 - 1$
N_1	Number of precursor taps in equalizer filter
N_2	Number of postcursor taps in equalizer filter
N_{block}	Number of transmitted symbols per interleaver block
N_{int}	Number of code bits per interleaver block
$p(l\|c)$	Shorthand notation for conditional pdf $p_{L\|c}(L_e(c_k)\|c_k = c)$
Q	Number of bits per symbol
r_n	A quality measure on the symbol estimate $\tilde{x}_n$
R_c	Code rate
R_f	Frame pattern efficiency
R	Overall code rate, $R = R_c R_f$
$R_h(u, \tau)$	Autocorrelation function at lag u of channel tap with delay τ
$\mathcal{S}$	Signal constellation, 2^Q-ary
s_i	One symbol in $\mathcal{S}$, $i \in \{0, \ldots, 2^Q - 1\}$
$\mathbf{s}$	$(N_1 + 1)$th column of $\mathbf{H}$
$\mathbf{s}_n$	$(N_1 + 1)$th column of $\mathbf{H}_n$
$S_h(\nu, \tau)$	Doppler spectrum at Doppler frequency ν of channel tap with delay τ
t_n	Training symbols
T_m	Sampling period used when generating time-varying channel taps
T_r	Sampling period of received signal
T_s	Symbol period

v_n	*A priori* variance of transmitted symbol
$\mathbf{V}_n$	*A priori* covariance matrix of $\mathbf{x}_n$, a diagonal matrix containing *a priori* variances v_n
$w(t)$	Noise signal at time t
w_n	Noise signal, sampled at the symbol rate
$w_{n,i}$	Noise signal, sampled several times per symbol interval
$\mathbf{w}_n$	Vector of noise samples
$x(t)$	Transmitted signal
x_n	Transmitted symbols, either data symbols $y_{n'}$ or training symbols t_n
$\tilde{x}_n$	Estimate of transmitted symbol, output from equalizer
$\hat{x}_n$	Hard-decided estimate of transmitted symbol
$\bar{x}_n$	*A priori* average value (mean) of transmitted symbol
$\mathbf{x}_n$	Vector of transmitted symbols. When we discuss channel estimation $\mathbf{x}_n = [x_n \ \cdots \ x_{n-M+1}]^T$ of length M, and when we discuss equalization $\mathbf{x}_n = [x_{n+N_1} \ \cdots \ x_{n-N_2-M+1}]^T$ of length $N+M-1$
$\bar{\mathbf{x}}_n$	*A priori* average value of $\mathbf{x}_n$
$y_{n'}$	Data symbols, obtained by mapping Q consecutive c_k onto $\mathcal{S}$
$z(t)$	Received signal
z_n	Received signal, sampled at the symbol rate
$z_{n,i}$	Received signal, sampled several times per symbol interval. The index $i \in \{0, \ldots, \kappa - 1\}$ denotes the κ samples per symbol interval n
$\mathbf{z}_n$	Vector of received samples

URN:NBN:no-3359

Abbreviations

2G	Second Generation
3G	Third Generation
ACS	Automatic Channel Selection
ALE	Automatic Link Establishment
ALM	Automatic Link Maintenance
AM	Amplitude Modulation
APP	*A Posteriori* Probability
ARCS	Automatic Radio Control System
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
baud	channel symbols per second
BER	Bit Error Rate
bps	bits per second
BPSK	Binary PSK (2-PSK)
BW	Burst Waveform
CCIR	former (French) name of ITU-R
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CRC	Cyclic Redundancy Check
DAMSON	Doppler and Multipath SOunding Network
DFE	Decision Feedback Equalizer
ECC	Error-Correcting Code
EXIT chart	EXtrinsic Information Transfer chart
FER	Frame Error Rate
FIR	Finite Impulse Response
FSK	Frequency Shift Keying
GPS	Global Positioning System

HDL	High rate Data Link
HF	High Frequency (3-30 MHz)
IIR	Infinite Impulse Response
LDL	Low latency Data Link
LDPC codes	Low-Density Parity-Check codes
LLR	Log Likelihood Ratio
LMS	Linear Mean Square
LSSE	Least Sum of Squared Errors
MAP	Maximum *A posteriori* Probability
MIL-STD	Military Standard
MLSE	Maximum Likelihood Sequence Estimator
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
MUF	Maximum Usable Frequency
LE	Linear Equalizer
LSU	Link SetUp
OFDM	Orthogonal Frequency Division Multiplexing
IP	Internet Protocol
ICI	InterCarrier Interference
ISI	InterSymbol Interference
ITU-R	International Telecommunications Union – Radio communications
NATO	North Atlantic Treaty Organization
NVIS	Near Vertical Incidence Skywave
OSI	Open Systems Interconnection
QPSK	Quaternary PSK (4-PSK)
Π	Interleaver
Π^{-1}	Deinterleaver
pdf	probability density function
PSP	Per-Survivor Processing
PPP	Point to Point Protocol
PSK	Phase Shift Keying
RLS	Recursive Least Squares
RF	Radio Frequency
SISO	Soft-In/Soft-Out
SNR	Signal to Noise Ratio
SOVA	Soft Output Viterbi Algorithm

SSB	Single SideBand
STANAG	Standardization agreement (by NATO)
xDL	Collective term for HDL and LDL
US DoD	United States Department of Defence

URN:NBN:no-3359

Chapter 1

Introduction

This dissertation is about the application of turbo equalization to receivers for digital high frequency (HF) communications. In this first chapter, we give a brief introduction to the subject areas of HF communications and turbo equalization, since these are the essential subjects of this dissertation.

1.1 HF communications

The HF band is defined as the electromagnetic spectrum between 3 MHz and 30 MHz, corresponding to wavelengths between 10 m and 100 m. Radio communications using this frequency band is referred to as HF communications. The lower limit is sometimes stretched below the formal definition, down to about about 1.5 MHz, when using the term HF communications.

Detailed information on all aspects of HF communications can be found in the books by Goodman (1992), Johnson, Desourdis, Earle, Cook and Ostergaard (1997), and Hall, Barclay and Hewitt (1996).

1.1.1 The role of military HF communications

For military applications, there will always be a need for long-range communications without relying on existing infrastructure, like telephone lines or radio base stations, in the battlefield. From the advent of radio communications in the late 19th century until the 1960's, radio communications using frequencies in and below the HF band was the only alternative for such communications.

From the 1960's and onwards, satellite communications gained popularity. Transponders were launched into orbit, and would receive, amplify, and retransmit radio signals back to earth. Such systems use frequencies in the microwave band (i.e. above 300 MHz), where bandwidth is abundant compared to HF frequencies, and higher

data rates can thus be supported. However, HF systems were still in service, but some people considered them as a “last resort” in case satellite communications should become unavailable. Actually, satellite transponders are vulnerable to jamming from the earth, and a small country without its own satellites, like Norway, cannot always be guaranteed capacity on a satellite owned by another country or company.

Traditionally, HF communications has been thought of as cumbersome and it required a lot of training of the operators. Knowledge of radios, antennas, and ionospheric propagation conditions were required in order to set up and maintain a communication link. However, the 1990’s saw the introduction of several new standards for HF communications, described in Chap. 2, which ease the operation of HF systems and also improve the performance in terms of availability and data rates. For example, using an automatic radio control system the operator does not need to consider which frequency to use; that is all taken care of by the system. Because HF systems are improving and the vulnerabilities of satellite communications are subject to certain apprehensions, interest in HF communications for military purposes has revived in the last years.

We can call this history “the fall and rise of military HF communications”. In (Renfree, 2001), the same story is told from the viewpoint of the US Navy.

1.1.2 Propagation mechanisms

Two major propagation mechanisms exist for radio waves in the HF band: Groundwave, where the waves propagate along the surface of the earth, and skywave, where the waves are reflected back to earth from the ionosphere.

The propagation range of the groundwave depends on the transmission frequency (decreasing range with increasing frequency) and the conductivity of the surface of the earth. The largest conductivity is found at sea, where radio waves at 3 MHz can propagate up to about 500 km. Therefore, groundwave is particularly applicable to maritime communications.

Skywave can be used for communication distances between 50 km and around the earth, using one or more reflections from ionospheric layers and from the earth surface. Ionospheric physics is a large subject area, described e.g. in the book by Davies (1990). Details of ionospheric physics are outside the scope of this dissertation, but a rough explanation follows here:

Molecules are ionized when radiation from the sun is absorbed in the atmosphere. The ionization is concentrated in layers or regions determined by the density profile of the different types of molecules, the solar zenith angle, and the strength of the ionizing radiation. Since the ionization depends on the solar zenith angle, it is stronger at day than at night, and stronger in summer than in winter. It also varies with solar activity and the 11-year sunspot cycle.

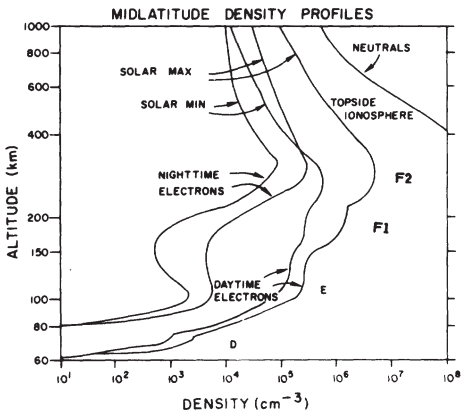


Figure 1.1 Typical electron density profile of the ionosphere, from (Jursa, 1985).

Fig. 1.1 shows typical day-time and night-time electron density profile of the ionosphere, and the altitude of the D, E, and F regions. The E and F regions refract radio waves in the HF band, and act as reflectors. The D region has lower concentration of electrons such that HF radio waves are not refracted, but the concentration is large enough to absorb energy from waves travelling through the region.

The geomagnetic field also affects the ionosphere, particularly in the auroral regions. This is discussed further in Section 1.1.3.

The received noise seen at HF frequencies consists of man-made noise, atmospheric noise (static discharges) and galactic noise. This noise may have a bursty nature (investigated in the dissertation by Giles (1995)), but a simplifying assumption is to model the received noise as bandlimited additive white Gaussian noise.

When using skywave for communications at relatively short distances, between 50 and 200 km, the angle of incidence on the ionosphere is close to zero degrees. Such a communications channel is called NVIS (near vertical incidence skywave) and has some peculiarities making it more challenging than longer paths.

1.1.3 Problems at high latitudes

The HF channel is particularly challenging at high latitudes. This is where the geomagnetic field lines are almost perpendicular to the atmosphere, leading high-energetic particles originating from the sun towards earth in the polar regions, as illustrated in Fig. 1.2. This causes the phenomenon known as aurora, which may produce spectacular visual effects. Aurora too weak to produce visual effects may still cause problems for radio communications, so-called radio aurora. Aurora increases

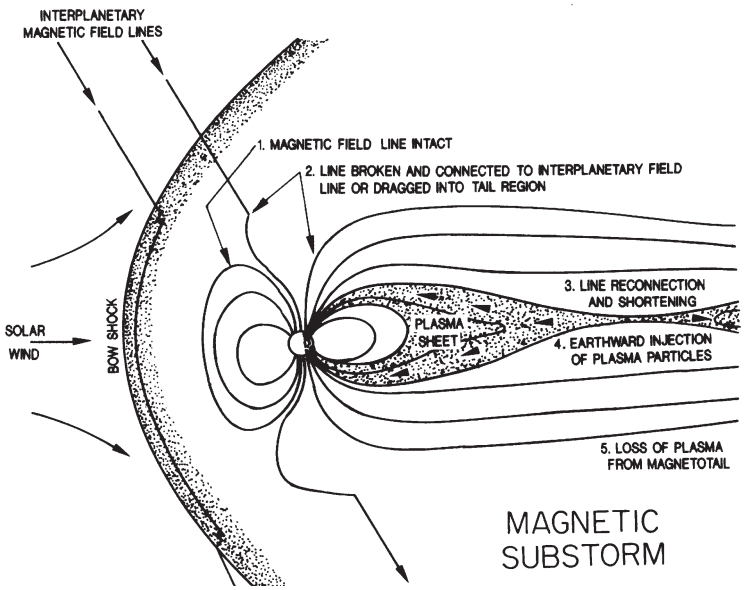


Figure 1.2 Figure describing the phenomena of aurora and geomagnetic substorms. The "plasma sheet" contains high-energetic particles originating from the sun, entering the atmosphere in the polar regions. From (Goodman, 1992), where a more detailed explanation is given.

the turbulence in the ionosphere, causing ions to move more rapidly, and the particle precipitation causes increased electron density and thus increased absorption in the D region.

Poor propagation conditions occur at all latitudes, but are more common at high latitudes due to auroral effects. The challenges are: Increased absorption leads to reduction in the received SNR, multiple propagation paths give rise to delay spread (time dispersion) in the received signal, and random movements of the individual ions and electrons cause Doppler spread (fading). In Chap. 4 we describe how these effects are modelled mathematically.

The level of geomagnetic activity, which is related to the problems described above, is given as different types of indices. Of these, we have in Chap. 3 used the K index and the Q index. Both these indices are measured locally, and are logarithmically scaled in the range 0 to 9 (where 0 corresponds to quiet and 9 to very disturbed conditions). Further details on HF propagation conditions at high latitudes can be found in (Jodalen, 1996; Cannon, Angling, Davies, Willink, Jodalen, Jacobsen, Lundborg and Bröms, 2000) and references therein.

The primary goal of this doctoral work is to increase the availability of HF com-

munications, particularly at high latitudes. In Chap. 3 we define the term availability as used here, and present an analysis of measured channel data investigating how the availability can be increased. It is found that the best approach is to improve the SNR performance (sensitivity) of the modems. We have chosen to improve the SNR performance by introducing to HF modems a receiver technology known as turbo equalization (Douillard, Jézéquel, Berrou, Picart, Didier and Glavieux, 1995; Tüchler, Koetter and Singer, 2002). Turbo equalization has been developed for systems having some features in common with HF communication systems: Mobile radio communication systems, underwater acoustic communication systems, and magnetic recording systems. An introduction to turbo equalization is given in the next section.

1.2 Turbo equalization

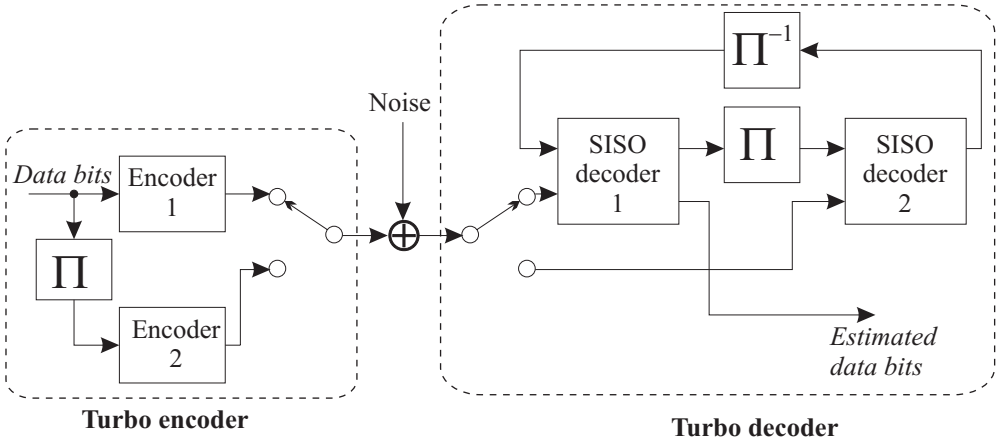
Before starting on the following historical introduction to turbo equalization, the reader should be aware of one common misunderstanding: Turbo codes are **not** a prerequisite for applying turbo equalization, which can be used regardless of what kind of error-correcting code is used.

When turbo codes were first introduced at a conference by Berrou, Glavieux and Thitimajshima (1993), the coding theory community was unbelieving and stunned. Performance close to the theoretical limits of Shannon was achieved with relatively simple code structure and decoding algorithm. Later it turned out that the revolutionary innovation was not the code itself; codes with similar structure and performance had been proposed earlier but could not be decoded with reasonable complexity. The novelty was the decoding algorithm: Iterative decoding of two simple constituent codes by exchanging soft information.

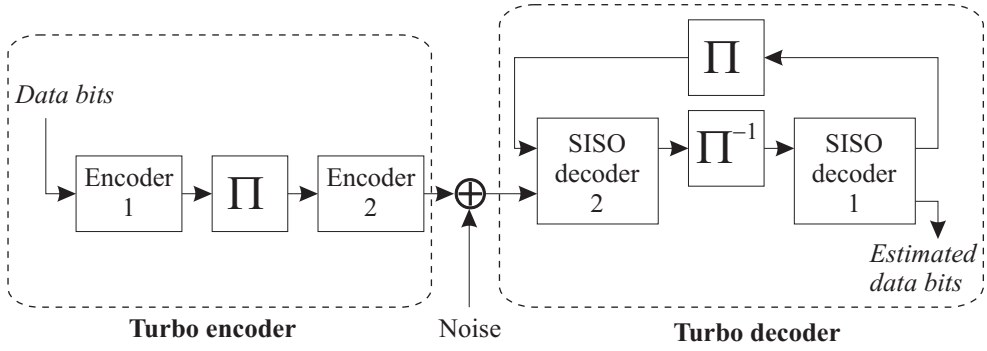
A turbo code consists of two concatenated codes, separated by an interleaver. The effective codeword length will be equal to the size (delay) of the interleaver. The concatenation can be parallel, as in the original proposal by Berrou et al., or serial, as first proposed by Benedetto and Montorsi (1996). Fig. 1.3(a) shows parallel concatenation, and Fig. 1.3(b) shows serial concatenation. The turbo decoding algorithm operates by exchanging soft (reliability) information between the decoders for the two constituent codes, and by doing a number of such iterations the performance gets close to the performance of the optimal, but often unrealizable, decoder for the overall code.

Sparked by the success of turbo codes/decoding, researchers started investigating the application of such an iterative algorithm, which was named “the turbo principle” by Hagenauer (1997), to other problems. One of those problems was equalization of digital signals suffering from intersymbol interference (ISI), and “turbo equalization” was first proposed in (Douillard et al., 1995).

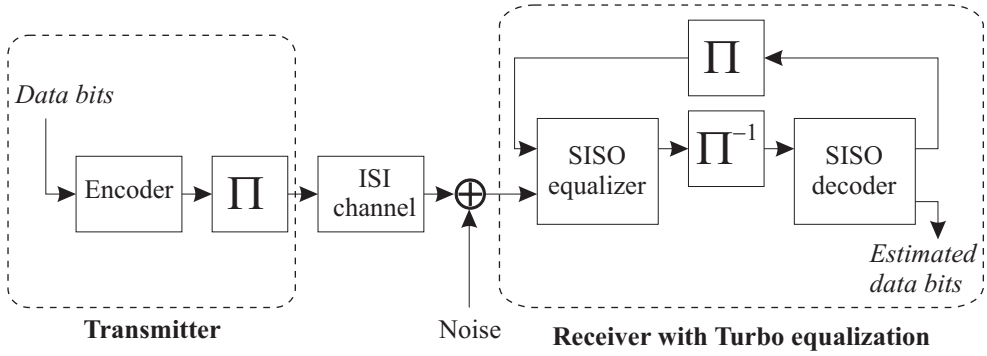
ISI occurs when the delay spread of the communications channel (e.g., caused



(a) Parallel concatenated turbo encoder and decoder.



(b) Serially concatenated turbo encoder and decoder.



(c) Conventional transmitter, and receiver with turbo equalization.

Figure 1.3 Principles of turbo codes and turbo equalization. Π denotes interleavers, and Π^{-1} denotes the corresponding deinterleavers.

by multipath) is longer than one symbol interval. This is seen as dispersion in the time domain, and as a frequency-selective (non-flat) response in the frequency domain. The role of an equalizer is to cancel as much as possible of the ISI. This can be viewed as flattening the frequency response, or as estimating the transmitted signal given the received signal and the channel impulse response (CIR). Thorough treatment of ISI and equalization can be found in several books (Proakis, 1995; Lee and Messerschmitt, 1994; Molisch, 2001; Feher, 1987; Hanzo, Webb and Keller, 2000).

Turbo equalization can be applied when the data is protected by an error-correcting code (ECC) and an interleaver shuffling the code bits before transmission over an ISI channel. Such a system can be viewed as a serially concatenated turbo code, where the inner code has been replaced with the ISI channel. An interleaver is commonly used even in systems not designed for turbo equalization, to break up bursty error patterns caused by fading or noise bursts. Most ECCs perform better when the error patterns are not bursty. Note that the ECC in a turbo equalization setup does not need to be a turbo code.

A conventional receiver for such a transmitter would first equalize the received signal, deinterleave the equalized symbols, and perform decoding to make decisions (estimates) on the transmitted data bits. In turbo equalization, a feedback loop is introduced from the decoder to the equalizer, in order to approach the performance of the optimal, unrealizable, receiver which performs joint equalization and decoding.

Turbo equalization is also known as iterative equalization and decoding. It operates by exchanging soft information between the equalizer and the decoder, as shown in Fig. 1.3(c). The receiver structure is similar to decoding of a serially concatenated turbo code.

In a turbo decoding process, the decoder for each constituent code is a soft-in/soft-out (SISO) module, with soft information at the input and output ports. For turbo equalization, a SISO equalizer is required. The optimal SISO equalizer uses the maximum *a posteriori* probability (MAP) algorithm, also known as the BCJR algorithm after the authors of the paper describing the algorithm (Bahl, Cocke, Jelinek and Raviv, 1974). Such an equalizer is trellis-based and too complex for typical HF system parameters, as discussed later in this dissertation. Research in turbo equalization has aimed at producing suboptimal SISO equalizers with reasonable complexity and good performance, and the work presented in Chap. 6 fits into that research path.

When the channel is time-varying and initially unknown, the equalizer must adapt to the changing conditions. The equalizer can be adapted directly, or a channel estimation algorithm can be used to provide an estimate of the CIR, which is then used to optimize the equalizer.

Literature on turbo equalization has to a large extent considered the CIR to be known. In this work, we have also considered fitting iterative channel estimation into the turbo equalization framework, giving a receiver with iterative channel estimation,

equalization, and decoding. We also call this approach adaptive turbo equalization.

1.3 Stated goal and major contributions of this work

This work started with a broad goal: To increase the time availability of each data rate at high latitude HF channels, or equivalently to increase the maximum data rate that can be used under any given conditions. This could be achieved by designing new waveforms, and/or by improving receiver technology.

To proceed towards this goal, a statistical study of channel conditions at high latitudes was first performed. Inspired by the results from that study, we decided to concentrate on applying turbo equalization to the reception of existing HF waveforms, i.e., the idea of designing new waveforms was discarded. Introducing new receiver technologies for existing standardized waveforms is politically easier than introducing new waveforms that would have to compete with existing standardized waveforms.

Our major contributions to research in HF communications have been:

- A new statistical analysis method to determine the availability of different waveforms/modems, and presentation of analysis results obtained using this method, in Chap. 3.

- Design of a receiver topology for HF modems using iterative channel estimation, equalization, and decoding (i.e. turbo equalization in combination with soft iterative channel estimation), in Chap. 9. We show by simulations that the proposed receiver outperforms a conventional receiver by 2-3 dB over an ITU-R poor channel.

Our major contributions to research in turbo equalization work have been:

- A comprehensive view on turbo equalization for unknown, time-varying channels, leading up to the proposed receiver for HF waveforms in Chap. 9.

- Modification of the low-complexity SISO equalization algorithm introduced by Tüchler, Singer and Koetter (2002), to account for time-varying channel conditions, in Chap. 6.

- Comparison of different channel estimation algorithms for soft iterative channel estimation in turbo equalization (i.e., improving the channel estimate over the iterations using soft information fed back from the decoder), including the introduction of a novel (but too complicated) modified RLS algorithm, in Chap. 7.

- The application of a semianalytical technique known as EXIT charts to address some interesting issues regarding adaptive turbo equalization for time-varying channels, in Chap. 8.

- Suggestions on how the proposed algorithms can be extended to fractionally spaced sampling of the received signal (the derivations and simulations have otherwise been performed using symbol-spaced sampling), at the end of Chaps. 6 and 7.

The simulations performed throughout the doctoral work have been done in Matlab, with processor-intensive algorithms implemented as mex-files using C code.

1.3.1 Published papers

During this doctoral work, the following conference papers have been published. In (Otnes and Maseng, 2001) we presented an idea of improving throughput by using nonuniform signal constellations in combination with automatic repeat request (ARQ) protocols. This idea was not pursued further and has not been included in the dissertation. In (Otnes and Jodalen, 2001; Otnes, 2001a; Otnes, 2001b), we presented statistical analyses on how the availability of HF waveforms could be increased. The papers related to turbo equalization were coauthored with Michael Tüchler. Of these, (Otnes and Tüchler, 2001; Otnes and Tüchler, 2002d; Otnes and Tüchler, 2002b; Otnes and Tüchler, 2002c; Otnes and Tuchler, 2002e) were first drafted by Roald Otnes and finalized jointly by the authors, while (Tüchler, Otnes and Schmidbauer, 2002; Otnes and Tüchler, 2003) were first drafted by Michael Tüchler and finalized jointly by the authors.

A paper based on the main points in Chaps. 6, 7, and 9 of this dissertation has been submitted to IEEE Transactions on Wireless Communications in November, 2002 (Otnes and Tüchler, 2002a).

1.4 Outline of the dissertation

This dissertation is a monograph, where Chaps. 1, 2, 4, and 5 contain material of introductory and tutorial nature, while Chaps. 3 and 6-9 are based on different conference papers and contain our major research contributions.

In Chap. 2 we give an overview of standards in military HF communications, in particular the waveform standards. In Chap. 3 we present statistical analyses using measured channel data, giving the motivation for researching turbo equalization for use in HF communications. In Chap. 4 we give an introduction to channel modelling, in particular for HF channels, and in Chap. 5 we survey the existing literature on

equalization and turbo equalization. We then delve into details on turbo equalization for time-varying channels, with low-complexity SISO equalization algorithms in Chap. 6, algorithms for soft iterative channel estimation algorithms in Chap. 7, and EXIT chart analysis in Chap. 8. Finally, we present our proposed turbo equalization-based receiver for HF waveforms in Chap. 9.

The chapters can mostly be read separately, even though there are some cross-references to details in other chapters.

Chapter 2

Standards in military HF communications

This work focuses on improving receiver technologies for existing standardized waveforms for military HF communications. This chapter gives an introduction to the hierarchy of military standards in HF communications, and a more detailed description of the ones that are relevant to this work.

Standards for military communications are being developed by NATO (North Atlantic Treaty Organization), and by US DoD (United States Department of Defence). The STANAG (Standardization Agreement) series is published by NATO, and the MIL-STD (Military Standard) series is published by US DoD. STANAGs often have MIL-STD counterparts, with only subtle differences.

The STANAGs and MIL-STDs discussed in this chapter can be found in the reference list.

2.1 The HF house

NATO has developed a reference framework to describe the relationship between the different standards for HF communications. This framework is called “the HF house”, and is shown in Fig. 2.1.

The HF house relates to the lower three layers of the OSI (Open Systems Interconnection) framework (Stallings, 1997): The physical layer (layer 1), the data link layer (layer 2), and the network layer (layer 3). At the physical layer are different waveform standards, and at layers 2 and 3 are standards for automatic repeat request, networking, link setup, and link maintenance. Interfaces to higher layers of the OSI framework are also included in the HF house.

Standards in the HF house have been named second and third generation (2G and 3G), but there is a discussion in the HF community on whether these terms are

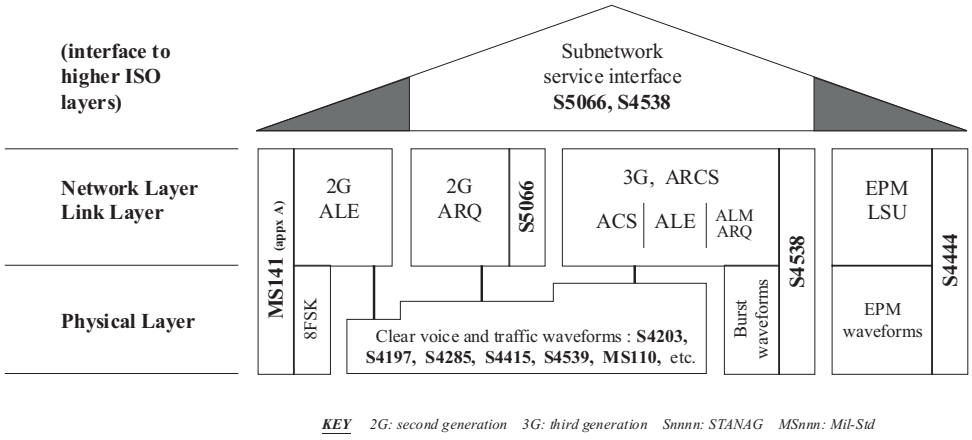


Figure 2.1 The HF house (this figure is included in most of the recent NATO STANAGs on HF communications).

misleading (Elvy, 2001; Johnson, 2002). Rather than 3G always being better than 2G, the choice of "generation" may be dependent on the application. In particular, 2G systems like STANAG 5066 were developed for maritime communications, whereas 3G systems like STANAG 4538 were designed for ground tactical communications. It is also possible to combine 2G and 3G systems.

The EPM (electronic protective measures) standard STANAG 4444 at the rightmost end of the HF house is designed to be particularly hard to detect for hostile monitoring stations, and is outside the scope of this dissertation.

2.2 Standards above the physical layer

The service provided by the physical layer is by no means free from errors. It is the task of the data link layer to provide close to error-free communications to higher layers. This is done using ARQ (Automatic repeat request) protocols (Lin, Costello and Miller, 1984), where each received packet is checked for errors using an error-detecting code (e.g. cyclic redundancy check, CRC). When a received packet contains errors, the receiver asks the transmitter for a retransmission of that packet.

Other tasks of the link layer of the HF house are to set up and maintain communication links, and to prevent different stations from transmitting at the same frequency at the same time (medium access control). In HF communications these tasks are more complex than in other systems (e.g. cellular systems), because of the additional degree of freedom of having different transmit frequencies to choose from, where the optimal frequency is dependent on link geography, time of day, and ionospheric

conditions.

In HF communications, the link layer includes some tasks not found in other data communication systems. These tasks are named ACS (automatic channel selection), ALE (automatic link establishment), and ALM (automatic link maintenance). These are all part of the broader term ARCS (automatic radio control system).

Most of the functionality provided by the HF house resides at layer 1 (the physical layer) and layer 2 (the link layer) of the OSI model. Thus, the term “network layer” on the left-hand side of Fig. 2.1 is somewhat misleading.

2.2.1 2G ALE

Second generation ALE is described in Appendix A of MIL-STD-188-141B, and was also described in the older standard MIL-STD-188-141A. All of the stations in the ALE network have a common predefined set of frequencies (this set can be different at day, night, dusk, and dawn, to accommodate for changing conditions during the day). In a 2G ALE system, there is no common time reference, and all idle stations scan all the frequencies in the set asynchronously, listening for transmissions. When one station wishes to establish a link, it starts a transmission on one of the frequencies, with a duration long enough to ensure that the idle stations go through a complete scan cycle. When the station being addressed has heard the call, it responds as soon as the calling station has stopped transmitting, and the link is established. After the link has been established, the two stations communicate using a 2G data link protocol and any traffic waveform.

Because of the asynchronous mode of scanning, a station wishing to establish a link must transmit a calling signal with a duration long enough to ensure that all other stations listen to the particular frequency used, i.e., the time it takes for all stations to scan through all frequencies. Another deficiency of 2G ALE is that FSK (frequency shift keying) waveforms with noncoherent demodulation are being used to establish the link, while the traffic waveforms use coherent PSK (phase shift keying), which is more robust to poor channel conditions than FSK (especially at low data rates). This means that even if the channel is good enough to support communications, the link may not be established because the link setup transmissions do not get through.

2.2.2 2G data link protocol

A second generation data link protocol is described in Annex C of STANAG 5066. It is a selective repeat ARQ protocol, meaning that only the packets in error are retransmitted (and not all packets after the error, as in basic go-back-N ARQ).

STANAG 5066 can use one of several traffic data waveforms, described in Sec. 2.3. The protocol includes the possibility of changing data rate, such that if the channel conditions can support a data rate higher than the one currently used, the

data rate may be increased. And similarly, if almost all packets fail because the data rate is too high, the data rate may be decreased. In this way, the data rate is adapted to changing channel conditions. Optimal algorithms (maximizing throughput) to adapt the data rate for STANAG 5066 are discussed in (Trinder and Gillespie, 2001).

An older data link protocol which is also considered to be 2G is described in Appendix B of FED-STD-1052, a United States Federal Standard.

2.2.3 3G systems

In 3G systems, layers 1 to 3 are integrated in one single standard for Automatic Radio Control Systems. This is STANAG 4538, or the equivalent Appendix C of MIL-STD-188-141B.

In 3G ALE, all stations in the net have a common time reference, e.g. from the satellite navigation system GPS (global positioning system). In this case, all stations in the net scan the frequencies in the scan set synchronously. When one of the stations wishes to establish a link, it starts transmission on one frequency. At the same instant, all idle stations are listening to the same frequency and the transmitter does not have to wait for all stations to scan through all frequencies before ceasing transmission.

Two data link protocols are defined in the 3G standards: The high rate data link protocol (HDL), and the low latency data link protocol (LDL), where LDL is more robust than HDL but gives lower throughput. A collective term for HDL and LDL is xDL. These data link protocols are based on incremental redundancy, also known as hybrid type II ARQ. Each packet of data is first sent without any error-correcting coding. If the checksum shows that there are some errors, the receiver asks the transmitter to transmit the redundancy of an ECC, and combines the two received packets in order to extract the transmitted data bits. In this manner, the effective code rate using HDL is $1/N_r$ where N_r are the number of retransmissions of a packet (including first-time transmission). Thus, the data rate is instantaneously adapted to changing channel conditions, as opposed to the more cumbersome process of changing data rate in 2G data link protocols.

It is also possible to use 2G data link protocols and waveforms together with 3G ALE.

2.2.4 Subnetwork interface

Layer 3 and above of a data communication system is connected to the standards in the HF house through a “subnetwork service interface”. Such an interface is defined in STANAG 5066 as well as in STANAG 4538. The client connecting the data communication system to the subnetwork interface can e.g. be an IP (Internet protocol) client or a PPP (point to point protocol) client. An implementation of an IP client for STANAG 5066 is described in (Kallgren and Smaal, 2001).

2.3 Waveform standards

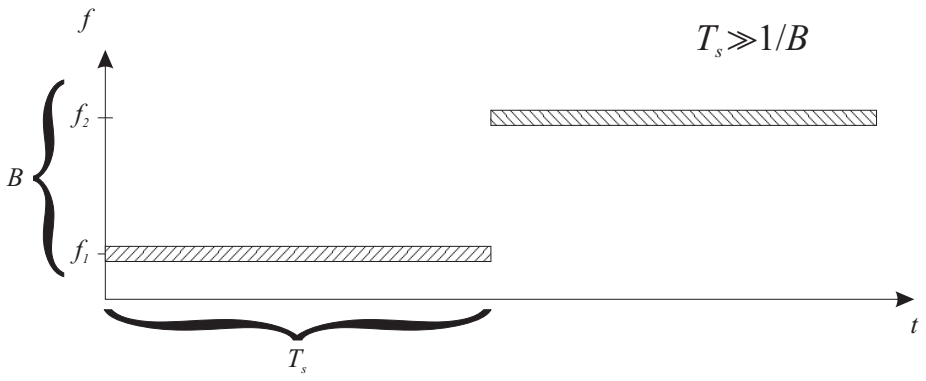
Most frequency assignments (defined by frequency management organizations) in the military HF bands have a bandwidth of 3 kHz. This is partly for historical reasons; 3 kHz was the bandwidth needed for analog voice using SSB (single sideband) modulation. Also, HF modems are often connected to the audio (voice) interface of an HF radio, which has a bandwidth of approximately 2.7 kHz. This bandwidth limitation puts constraints on the maximum available data rate for HF modems. Some efforts are being made to increase the bandwidth by using several frequency allocations (contiguous or non-contiguous) simultaneously, but this is not considered in the present work. For more information on those efforts, see (Jorgenson, Johnson and Moreland, 1999; Jorgenson, Johnson, Moreland, Bova and Jones, 2000)

In HF terminology, the word "waveform" is used to describe all the baseband signal processing at the physical layer in the transmitter, i.e., the conversion from data bits to the signal delivered to the audio interface of the radio. This encompasses e.g. pulse shaping, signal constellations, frame structure, and error correcting coding. HF radios use SSB modulation to convert the input audio signal (the "waveform") to the transmitted RF signal.

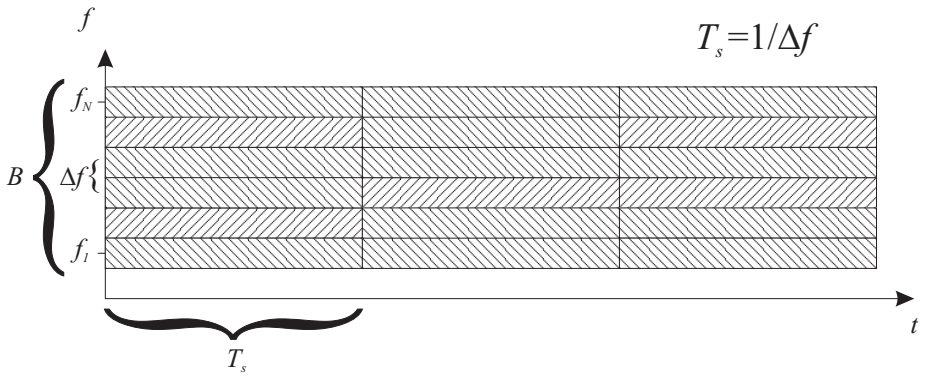
When designing a waveform, one has to take into account the limited bandwidth and the time-varying multipath nature of the radio channel. Three fundamentally different approaches are used to accomplish this: Frequency shift keying (FSK), parallel-tone waveforms, and serial-tone waveforms. These different principles are illustrated in Fig. 2.2.

Noncoherent FSK, shown in Fig. 2.2(a), means transmitting a "zero" as one frequency, called the space frequency, and a "one" as another frequency, called the mark frequency. Both frequencies reside within the bandwidth B. Higher-order FSK can be applied by using more than two frequencies. The receiver for FSK is quite simple: A narrowband filter is centered at each of the frequencies, and the filter having the strongest output signal indicates which frequency was transmitted. FSK is inefficient in terms of power as well as bandwidth, and is currently used mainly in the link setup waveforms in 2G ALE.

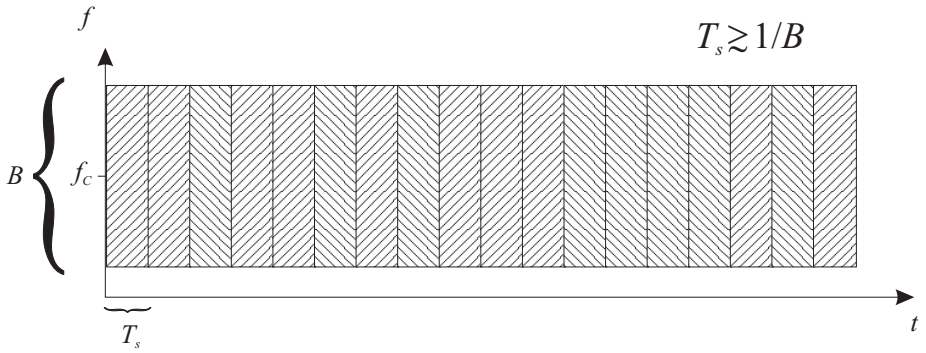
Parallel-tone waveforms, known as orthogonal frequency division multiplexing (OFDM) in the communications theory literature, are illustrated in Fig. 2.2(b). Here, the bandwidth B is shared among N different subcarriers with a frequency spacing Δf. Each subcarrier is modulated using PSK or some other signal constellation. If the symbol time T_s is chosen such that $T_s \Delta f = 1$, the signals transmitted on the different subcarriers will be orthogonal (hence the term OFDM). In this case, the demodulator and modulator can be implemented as an FFT (fast Fourier transform) and IFFT (inverse FFT), respectively. In parallel-tone waveforms, multipath is handled by inserting a guard time T_g (not shown in the figure) between the transmitted symbols.



(a) Frequency shift keying (FSK). f_1 and f_2 are the mark and space frequencies.



(b) Parallel-tone waveform. f_1 to f_N are the N carrier frequencies, and Δf is the frequency spacing.



(c) Serial-tone waveform. f_C is the carrier frequency.

Figure 2.2 The principles of different kinds of waveforms, showing the transmitted frequency spectrum as a function of time. Each hatched rectangle denotes one transmitted symbol. T_s is the symbol time, and B is the total bandwidth.

No intersymbol interference (ISI) will occur if $T_g > \tau_m$, where τ_m is the maximum delay spread. The frequency-selectivity caused by the multipath will, however, give a different received signal level at the different subcarriers. Also, the Doppler spread caused by the time-varying nature of the channel will cause intercarrier interference (ICI), unless $\Delta f \gg \nu_d$, where ν_d is the maximum Doppler spread.

Serial-tone waveforms (or single-tone waveforms), shown in Fig. 2.2(c), uses a single carrier frequency, modulated with PSK or QAM. In order to utilize the entire available bandwidth, the symbol time is chosen such that $T_s B$ is slightly larger than 1. This leads to the symbol time being smaller than typical delay spreads observed on HF channels, causing ISI. The time-varying ISI has to be handled by an adaptive equalizer, whose task is to estimate the transmitted symbols given the distorted received signal, or equivalently to flatten the frequency response of the channel. The adaptive equalization task is more complex than the receiver used in parallel-tone modems. In order to aid the equalizer in tracking the time-varying channel impulse response (CIR), training sequences which are known to the receiver are multiplexed into the transmitted symbol stream at regular intervals. A tutorial on serial-tone HF waveforms can be found in (Jorgenson and Moreland, 1999).

Parallel-tone waveforms are described in optional appendices to MIL-STD-188-110B: Appendix A describes a 16-tone waveform, and Appendix B describes a 39-tone waveform. Most other waveforms included in the HF house are serial-tone waveforms. Nieto (2001) has compared the characteristics and performance of serial-tone and parallel-tone waveforms. His conclusion is that the basic performance of the two approaches is similar, but the ECC (error-correcting code) used in the serial-tone waveform standards performs better than the ECC used in the parallel-tone waveform standards. For this reason, serial-tone waveforms have gained popularity while parallel-tone waveforms are used mainly in legacy systems. If parallel-tone waveforms had been standardized together with a better ECC, this story might well be different. An interesting point to note is that in Digital Radio Mondiale, a new digital broadcasting system using the AM bands at HF and lower frequencies (ETS, 2001), parallel-tone waveforms are used due to the simpler receiver structure, without adaptive equalization.

In this work, we concentrate on serial-tone waveforms, as these are the most commonly used waveforms for military HF communications. They use a signalling rate $f_s = 1/T_s$ of 2400 baud (channel symbols per second), modulated onto an 1800 Hz subcarrier and filtered such that the signal input to the audio interface of the radio lies in the frequency range 300-3300 Hz.

In the following sections, different serial-tone waveforms are described in detail, as a background for the design of turbo equalization-based receivers later in this dissertation.

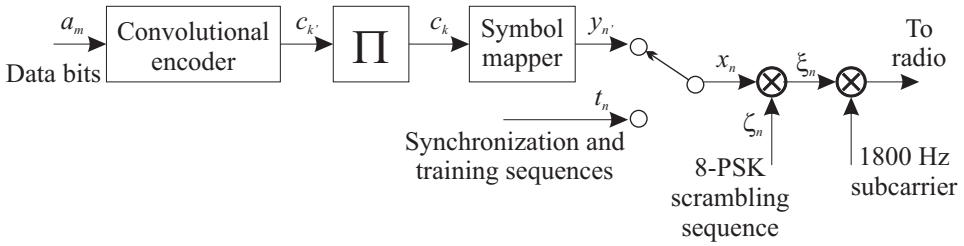


Figure 2.3 Transmitter block diagram for the medium-rate serial-tone waveforms in MIL-STD-188-110B and STANAG 4285. Some notation used for different signals throughout this work is also introduced here.

2.3.1 Medium-rate serial-tone HF waveforms

We define medium data rates as information data rates of 2400 bps and below. Two different medium-rate waveforms can be found within the HF house; one is described in MIL-STD-188-110B (this waveform was also included in the older version of the standard, MIL-STD-188-110A), and the other is described in STANAG 4285. The newer NATO waveform standard STANAG 4539, which includes all data rates in the range 75-12800 bps, points to MIL-STD-188-110B for data rates of 2400 bps and below. According to STANAG 4539, the waveforms defined in STANAG 4285 are only needed for interoperability with legacy systems. However, STANAG 4285 is still popular for broadcast applications because the acquisition time (the time needed for synchronization before starting transmitting data) is smaller than for MIL-STD-188-110B (see Sec. 2.3.1.4).

The structure of the transmitter for MIL-STD-188-110B and STANAG 4285 is shown in Fig. 2.3. In the following, we go through each block, highlighting the differences between the waveforms. For MIL-STD-188-110B, we describe the waveforms used for fixed-frequency operation. Some details are different for frequency-hopping operation, which is an option in MIL-STD-188-110B.

2.3.1.1 Error-correcting code (ECC)

The information data bits are protected by an ECC. For all the serial-tone waveform standards in the HF house, this is a strong convolutional code with constraint length 7. Its generator polynomials are $g_1(D) = D^6 + D^4 + D^3 + D + 1$ and $g_2(D) = D^6 + D^5 + D^4 + D^3 + 1$.

The encoder is shown in Fig. 2.4. The basic code rate is 1/2; lower code rates are achieved by repeating the code bits while a code rate of 2/3 is achieved through puncturing. Both standards define several information data rates, and the code rate is one of the parameters that can be changed when changing the data rate (see Table 2.1

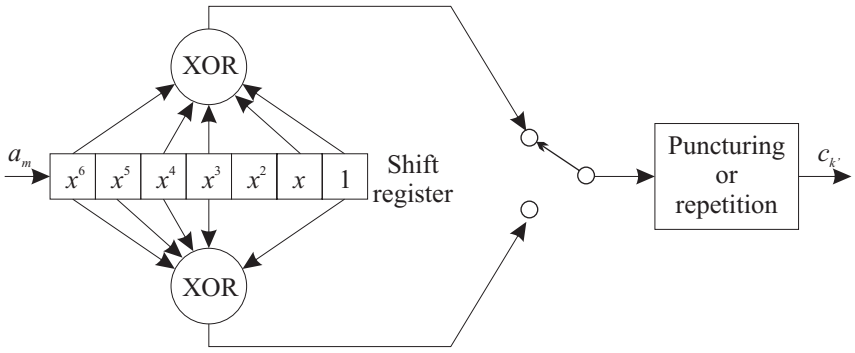


Figure 2.4 The convolutional encoder used in most serial-tone HF waveforms.

on page 24).

We denote the code rate R_c, such that for each data bit $a_m \in \{+1, -1\}$ are generated $1/R_c$ code bits $c_{k'} \in \{+1, -1\}$ (where $+1$ corresponds to a bit value of zero).

At the end of a transmitted message, the encoder is "flushed" by encoding 6 extra zeros, such that the shift register always is in the all-zero state at the beginning and end of each message. This is also known as terminating the code trellis, and causes an extra overhead which is not included when calculating the information data rate.

2.3.1.2 Interleaver

Convolutional codes perform well when bit errors are uncorrelated, but are less effective than block codes in bursty error patterns. On HF channels, such bursty error patterns are caused by fading and impulsive noise and interference, and convolutional codes would be a bad choice unless further precautions were taken.

For this reason, an interleaver (denoted by Π in the figures) is used. It shuffles the code bits, such that neighbouring code bits out of the encoder are transmitted far apart in time on the channel. The interleaver is also a prerequisite for application of turbo equalization.

Generally, a longer interleaver gives better performance, and the interleaver should at least be longer than the maximum duration of a fade or noise burst. But, a longer interleaver also gives a larger delay (latency), which is undesirable for some applications.

The interleaver in MIL-STD-188-110B is a so-called block interleaver, which shuffles one block of data at a time. This interleaver uses a complicated procedure, involving additions and modulo operations, to load and fetch the data in a matrix. Two interleaver lengths (in addition to using no interleaver at all) are defined in the standard: 0.6 s and 4.8 s. The size of the matrix depends on the interleaver length

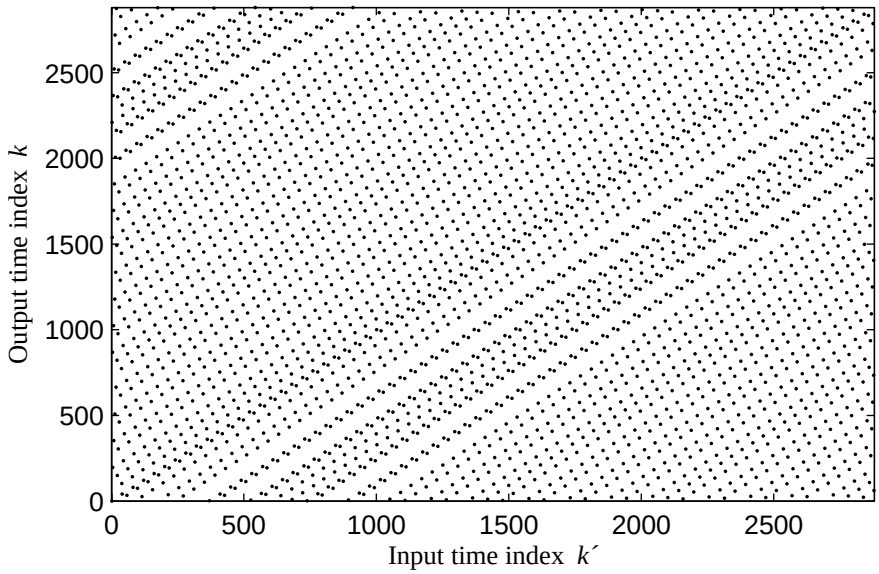


Figure 2.5 The mapping from input time indices to output time indices for one interleaver block in MIL-STD-188-110B, with an interleaver length of 0.6 s and data rate of 2400 bps. The closest points (on the densest diagonal bands) are such that input symbols with a spacing of 7 corresponds to output symbols with a spacing of 17, or vice versa.

and the data rate.

Fig. 2.5 shows one of the interleaver mappings used in MIL-STD-188-110B. We denote the output time index as $k = \pi(k')$ The deinterleaver, denoted Π^{-1}, is also a block interleaver. It performs the reverse mapping of the interleaver, written as $k' = \pi^{-1}(k)$.

The interleaver in STANAG 4285 is a so-called convolutional interleaver, which operates by sending different bits through shift registers of different lengths. The deinterleaver does the same, such that the total delay experienced by each bit from interleaver input to deinterleaver output is identical. This standard also defines two different interleaver lengths: 0.853 s and 10.24 s.

The convolutional interleaver is not suitable for turbo equalization. This is because the operation of deinterleaving followed by interleaving will constitute a delay for each iteration, and the delay in a turbo equalization-based receiver will thus be proportional to the number of iterations. When using block interleavers, deinterleaving and interleaving can be performed instantaneously as soon as the entire interleaver block has been received. For this reason, MIL-STD-188-110B should be used rather than STANAG 4285 if one wants to use turbo equalization in the receiver.

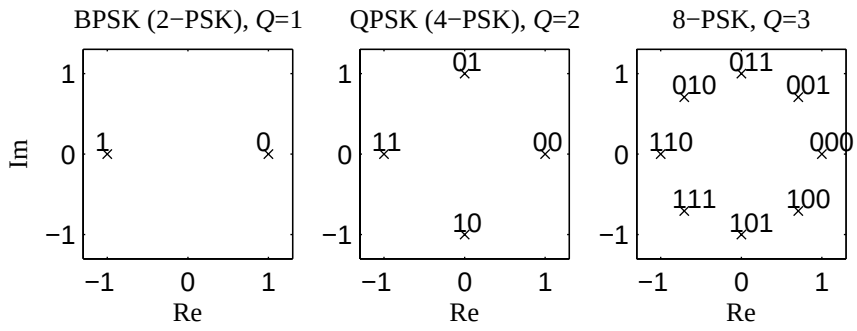


Figure 2.6 The three PSK constellations used in medium-rate serial-tone waveforms, shown in the complex plane. Each point is annotated by the corresponding values of Q consecutive c_k (writing 0 instead of +1 and 1 instead of -1).

2.3.1.3 Symbol mapping

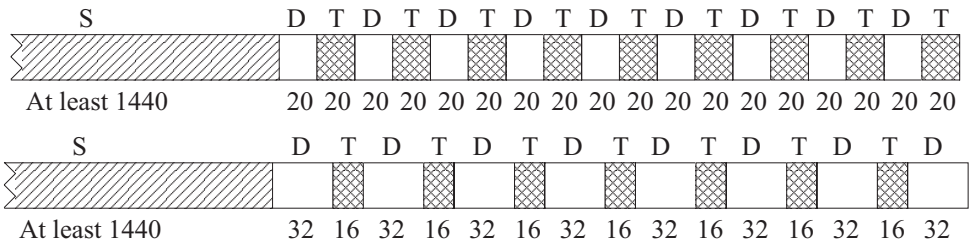
Considering the overhead imposed by the ECC and the training sequences, more than one bit must be transmitted per channel symbol in order to transfer an information data rate of 2400 bps with a channel symbol rate of 2400 baud. This is achieved by mapping Q code bits at a time onto a channel symbol from a 2^Q-PSK signal constellation, where $2^Q \in \{2, 4, 8\}$ is the number of points in the signal constellation, and Q is the number of bits per channel symbol.

The three signal constellations used in MIL-STD-188-110B and STANAG 4285 are shown in Fig. 2.6. $Q \in \{1, 2, 3\}$ consecutive interleaved code bits c_k are mapped onto a complex symbol $y_{n'} = e^{j\phi_{n'}}$. The amplitude $|y_{n'}|$ is always 1 and the information lies in the phase $\phi_{n'}$, hence the name PSK. The QPSK and 8-PSK constellations are Gray-coded, such that two neighbouring symbols in the constellation differ in only one bit position.

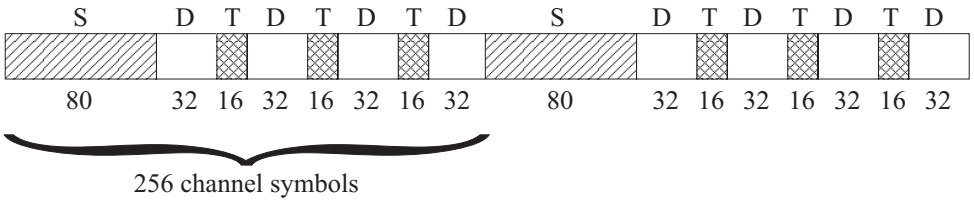
2.3.1.4 Synchronization and training sequences

Symbols t_n known to the receiver are multiplexed with the data symbols $y_{n'}$, to form the sequence x_n of transmitted symbols. The known symbols are synchronization sequences, used to aid the receiver in time and frequency synchronization, and training sequences used to aid the equalizer in tracking the time-varying channel.

Fig. 2.7 shows the frame structures used in MIL-STD-188-110B and STANAG 4285. In STANAG 4285 and the 2400 bps waveform of MIL-STD-188-110B, training sequences of length 16 are inserted with 32 data symbols in between. At 1200 bps and below in MIL-STD-188-110B, training sequences of length 20 are inserted with 20 data symbols in between. At this point in the system (before scrambling),



(a) Frame structures in MIL-STD-188-110B. A long synchronization sequence of the same length as one interleaver block is inserted at the beginning of transmission only.



(b) Frame structure in STANAG 4285. A synchronization sequence of 80 symbols is inserted every 256 symbols.

Figure 2.7 Frame structure showing the sequence of synchronization (S), data (D), and training (T) symbols for the medium-rate serial-tone waveforms.

all training symbols (denoted by "T" in Fig. 2.7) are equal to $1 + j0$, i.e., the "zero" symbol in the signal constellation. Because of the scrambling described in the next subsection, this gives a sequence of symbols transmitted on-air which is well suited for channel estimation.

In MIL-STD-188-110B, each transmitted message is started by one long synchronization sequence, and no further synchronization is transmitted before the end of the message. After the synchronization sequence, 32 out of every 48 channel symbols contains information for the 32/16 pattern. Thus, the bandwidth efficiency of this frame pattern (fraction of transmitted symbols containing information) is $R_f = 2/3$ if the message is long enough to neglect the overhead imposed by the initial synchronization sequence. Similarly, the bandwidth efficiency is $R_f = 1/2$ for the 20/20 pattern under the same assumption.

In STANAG 4285, a synchronization sequence of 80 symbols is transmitted every 256 symbols. The bandwidth efficiency of this frame pattern is $R_f = 1/2$, as 128 out of every 256 channel symbols are known to the receiver.

In MIL-STD-188-110B, the synchronization sequence contains information about which data rate and interleaver setting is used, such that the receiver can make the proper adjustments. This is called an "autobaud" capability, and is very useful when adapting the data rate to changing channel conditions, e.g., when using STANAG

5066. STANAG 4285 has no autobaud capability (the synchronization sequence is always the same), and problems will arise if the transmitter and receiver have not agreed on which data rate and interleaver setting to use (e.g., by transmitting an explicit message when changing data rate).

2.3.1.5 Scrambling sequence

The training and data symbols x_n are multiplied by a pseudo-random 8-PSK sequence ζ_n, producing the actual transmitted symbols ξ_n. The purpose of this is to randomize the transmitted data and training symbols, and to make the transmitted signal appear as 8-PSK on-air even when x_n is a BPSK or QPSK signal.

The synchronization sequences are treated differently. In STANAG 4285, they are not scrambled at all, while in MIL-STD-188-110B they are scrambled with a different sequence than the training and data symbols.

The length (period) of the scrambling sequence is 160 channel symbols for MIL-STD-188-110B, while for STANAG 4285 it is 176 (the exact number of symbols between two synchronization sequences).

2.3.1.6 Modulation and filtering

The scrambled 8-PSK data symbols ξ_n are complex numbers, represented as an in-phase and a quadrature-phase signal. These are modulated onto a subcarrier with the frequency 1800 Hz, in order to generate a real-valued signal that can be input to the audio interface of a radio.

With a complex signal at a symbol rate of 2400 baud, the minimum required bandwidth according to the Nyquist criterion would be 2400 Hz if a rectangular filter was applied. Using a realizable filter with smooth transition bands, the bandwidth used is 3000 Hz centered at 1800 Hz, i.e. the band 300-3300 Hz.

In this dissertation, we do not treat in detail what happens with the signal between the audio interface and the antenna, except noting that the SSB modulation used can be mathematically described as a frequency shift from voiceband to the transmitted frequency band.

2.3.1.7 Parameters for the medium-rate waveforms

As previously stated, different information data rates are defined in the medium-rate standards. Table 2.1 summarizes the code rate, signal constellation size, and frame pattern used for each data rate in MIL-STD-188-110B and STANAG 4285. The information data rate is given by $f_a = R_c Q R_f f_s$, where $f_s = 2400$ baud is the channel symbol rate.

Data rate f_a (bps)	110B			4285		
	R_c	Q	R_f	R_c	Q	R_f
(4800)	1	3	2/3			
(3600)				1	3	1/2
2400	1/2	3	2/3	2/3	3	1/2
1200	1/2	2	1/2	1/2	2	1/2
600	1/2	1	1/2	1/2	1	1/2
300	1/4	1	1/2	1/4	1	1/2
150	1/8	1	1/2	1/8	1	1/2
75	Robust			1/16	1	1/2

Table 2.1 Parameters used for the different data rates in MIL-STD-188-110B and STANAG 4285. R_c is the code rate, Q is the number of bits per channel symbol, and R_f is the bandwidth efficiency of the frame pattern.

Both standards also define one data rate above 2400 bps, which does not use any ECC (i.e. $R_c = 1$). These waveforms suffer from very high bit error rates because they are uncoded, and they are therefore seldom used in practice.

The 75 bps waveform in MIL-STD-188-110B is similar to the robust waveform described in Sec. 2.3.3.

2.3.2 High-rate serial-tone HF waveforms

We define high data rate HF waveforms as waveforms with an information data rate of 3200 bps or above. Such waveforms are described in Annex B of STANAG 4539 and in Appendix C of MIL-STD-188-110B. These two standards describe identical high-rate waveforms; the only difference is that the performance requirements of STANAG 4539 are stricter.

The block diagram of the transmitter is similar (but not identical) to the block diagram for medium-rate waveforms, shown in Fig. 2.3. In the following sections, we point out where the high-rate waveforms differ from the medium-rate waveforms.

2.3.2.1 Error-correcting code (ECC)

The ECC is the same convolutional code as in the medium-rate waveforms, but now punctured to an overall code rate of $R_c = 3/4$ for all data rates in the 3200-9600 bps range.

Instead of terminating the trellis by forcing the shift register to the zero state at the end of each message, “tail-biting” is used for each interleaver block. This means that the end state is forced to be the same as the start state of the trellis (see the standard for a detailed description). The benefit of this approach is that the extra overhead caused

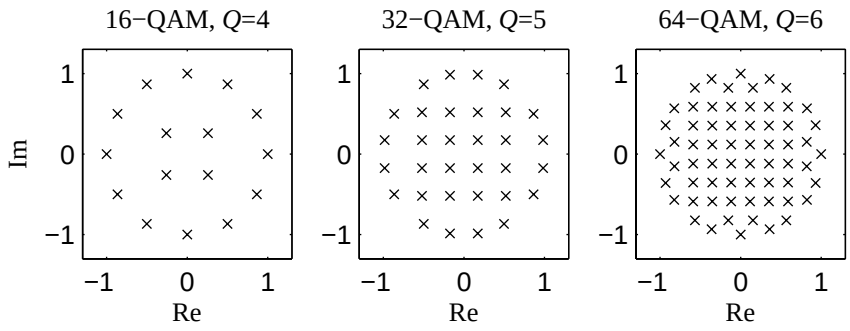


Figure 2.8 The three QAM constellations used in high-rate serial-tone waveforms for 6400 bps and above. The mapping from bits to symbols (not shown) is pseudo-Gray, i.e., symbols which are close in the constellation differ in as few bit positions as possible.

by encoding zeros is removed, and each interleaver block is encoded independently (as opposed to each message in the medium-rate waveforms).

2.3.2.2 Interleaver

The interleaver used in the high-rate waveforms is a block interleaver, with a simpler structure than the interleaver defined for the medium-rate waveforms in MIL-STD-188-110B. Six different interleaver lengths are defined: 0.12 s, 0.36 s, 1.08 s, 2.16 s, 4.32 s, and 8.64 s.

2.3.2.3 Symbol mapping

Higher-order signal constellations are needed in order to transmit data rates up to 9600 bps with a symbol rate of 2400 baud. The number of bits per channel symbol for the high-rate waveforms is $Q \in \{2, 3, 4, 5, 6\}$.

For $Q \in \{2, 3\}$ the QPSK and 8-PSK constellations in Fig. 2.6 are used, as for the medium-rate waveforms. For higher Q, it is more power efficient to use QAM than PSK constellations. The high-rate standards define 2^Q-QAM constellations with peak-to-average energy ratios smaller than square QAM constellations, for $2^Q \in \{16, 32, 64\}$. The constellations are shown in Fig. 2.8.

2.3.2.4 Synchronization and training sequences

The frame pattern used in the high-rate waveforms (shown in Fig. 2.9) is more complicated than for the medium-rate waveforms. Now, as many as 256 consecutive data

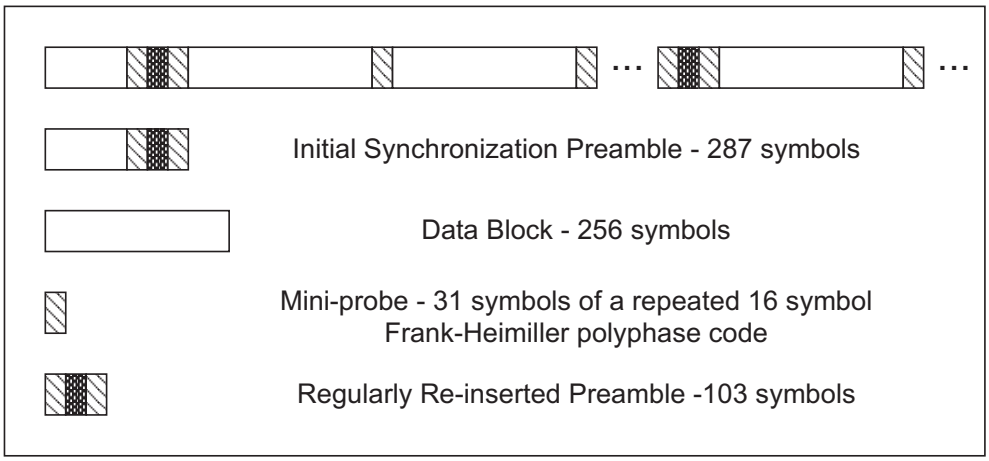


Figure 2.9 Frame structure in the high-rate serial-tone waveforms (from STANAG 4539).

symbols are transmitted between two training sequences of length 31 (called mini-probes). Every 72nd mini-probe is replaced by a so-called reinserted preamble of length 103.

The bandwidth efficiency of this frame pattern (excluding the overhead imposed by the initial synchronization preamble) can be calculated to be $R_f = 8/9$.

Regardless of which signal constellation is used for the data symbols, the known symbols of the high-rate waveforms are always 8-PSK symbols. They also contain some information about the interleaver length and data rate (i.e., not all the "known" symbols are actually known, but they can be easily detected). Thus, the high-rate waveforms have autobaud capability.

2.3.2.5 Scrambling

As in the medium-rate waveforms, the signal is scrambled before transmission. Only the data symbols are scrambled, and not the training symbols as in the medium-rate waveforms.

The scrambling procedures used in the high-rate waveforms are shown in Fig. 2.10. When using PSK constellations, the data symbols are scrambled by multiplying the data symbols $y_{n'}$ with an 8-PSK sequence $\zeta_{n'}$, as in the medium-rate waveforms. For the QAM constellations, the scrambling is rather applied to the interleaved code bits c_k, by exclusive-or (i.e. multiplication in the $\{+1, -1\}$ domain) with a BPSK scrambling sequence ${\zeta^b}_k$.

The BPSK scrambling sequence ${\zeta^b}_k$ is a pseudo-random sequence with a period of 511 bits. The 8-PSK scrambling sequence $\zeta_{n'}$ is generated by mapping 3 consec-

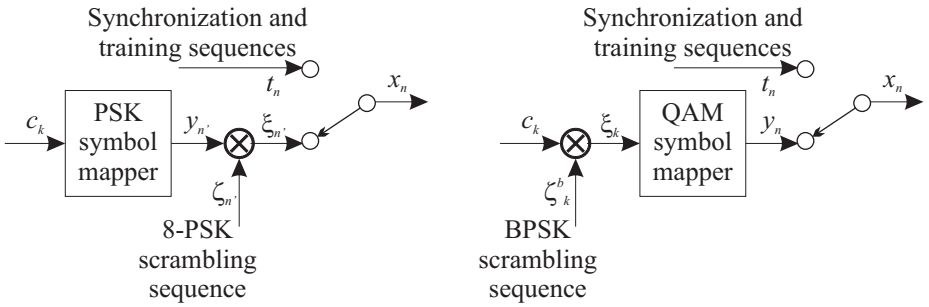


Figure 2.10 Scrambling procedure used in the high-rate waveforms for PSK constellations (left) and QAM constellations (right).

Data rate f_a (bps)	Code rate R_c	bits/symbol Q	Frame pattern eff. R_f
(12800)	1	6	8/9
9600	3/4	6	8/9
8000	3/4	5	8/9
6400	3/4	4	8/9
4800	3/4	3	8/9
3200	3/4	2	8/9

Table 2.2 Parameters used for the different data rates in the high-rate waveforms of STANAG 4539 and MIL-STD-188-110B.

utive $\zeta^b{}_k$ onto the 8-PSK constellation.

2.3.2.6 Parameters for the high-rate waveforms

Table 2.2 summarizes the parameters for each data rate in the high-rate waveforms (compare to Table 2.1 for the medium-rate waveforms). 12800 bps can be achieved if the ECC is omitted, but for all the other data rates (3200-9600 bps), the only parameter that is changed is the signal constellation (i.e. the number of bits per symbol Q).

2.3.3 Robust low-rate HF waveform

A very robust low-rate waveform with an information data rate of 75 bps is described in STANAG 4415. The same waveform is also defined for 75 bps in MIL-STD-188-110B, but the performance requirements are much stricter in STANAG 4415 (i.e., the two standards describe the same transmitted waveforms, but STANAG 4415 requires the receiver to perform well under more severe channel conditions). The performance requirements of STANAG 4415 were based on channel measurements

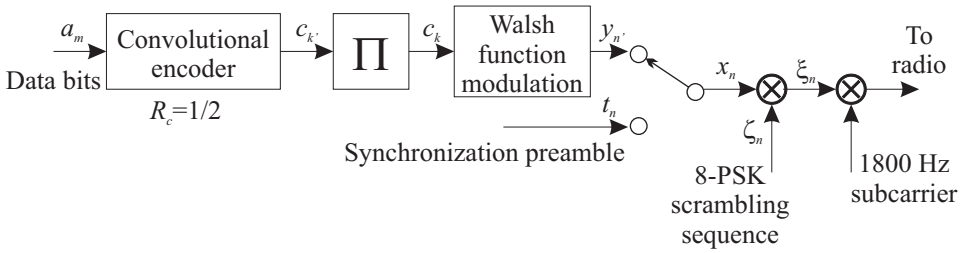


Figure 2.11 Transmitter block diagram for the robust 75 bps waveforms in STANAG 4415 and MIL-STD-188-110B.

at high latitudes under the DAMSON program (see Sec. 3.3.2). Therefore a modem which is compliant to STANAG 4415 is a good choice under severe conditions at high latitudes, if one needs high probability of being able to communicate and can cope with the low data rate.

Fig. 2.11 shows the block diagram of the transmitter for the robust 75 bps waveforms. By comparing with the medium-rate waveforms in Fig. 2.3, we see that several blocks are the same: The convolutional encoder (operating at $R_c = 1/2$, i.e. no puncturing or interleaving), the block interleaver, and the 8-PSK scrambling. The differences are that there are no known symbols except for an initial synchronization preamble, and that the code bits are modulated by orthogonal Walsh functions.

The orthogonal Walsh modulation is illustrated in Fig. 2.12. For every 2 code bits are generated 32 BPSK channel symbols, called one Walsh symbol. This means that the number of code bits per channel symbol is $Q = 2/32 = 1/16$. Because there are no training symbols, the frame pattern efficiency is $R_f = 1$, and we can verify that the information data rate is $f_a = R_c Q R_f f_s = 75$ bps.

2.3.4 Burst waveforms

The 3G ARCS standards, STANAG 4538 and MIL-STD-188-141B, define so-called "burst waveforms". These waveforms are designed for transmitting just one short burst of data at a time, rather than one long message. They are closely related to the link setup (LSU) protocols used in 3G ALE, and to the ARQ protocols LDL and HDL. The incremental redundancy feature of the ARQ protocols is mirrored in the definition of the burst waveforms.

There are a total of 6 burst waveforms, numbered BW0 to BW5. BW0 and BW5 are used for link setup, BW1 and BW4 are used for acknowledgment messages in the xDL protocols, and BW2 and BW3 are used for transmitting data packets in HDL and LDL, respectively. In the following sections, we give a summary description of BW2 and BW3. For a detailed description, the reader should consult the standard.

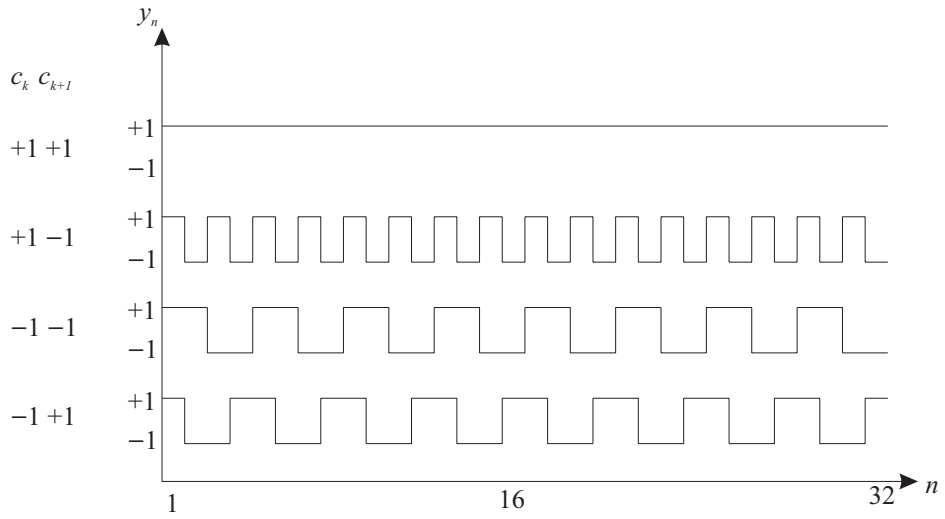


Figure 2.12 Orthogonal Walsh modulation used in the robust 75 bps waveforms. Two consecutive interleaved code bits c_k and c_{k+1} are mapped onto one out of four Walsh functions with a length of 32 bits, called one Walsh symbol. A different set of Walsh functions (not shown) is used for the last Walsh symbol in each interleaver block, for synchronization purposes.

2.3.4.1 BW2, the high-rate traffic burst waveform

When using BW2, the data to be transmitted is split into packets of length 1881 bits. A CRC checksum consisting of 32 bits is appended to the end of each packet, such that the receiver can determine whether the data packet was received correctly.

One transmitted burst contains $N_p \in \{3, 6, 12, 24\}$ data packets. Each data packet (including CRC) is protected by a convolutional ECC with constraint length 8, and the code trellis is terminated to the zero state by encoding 7 zeros at the end of the data packet. The basic code rate of the ECC is 1/4, and the generator polynomials are

$$\begin{aligned} g_1(D) &= D^7 + D^4 + D^3 + D^2 + 1, \\ g_2(D) &= D^7 + D^5 + D^4 + D^3 + D^2 + 1, \\ g_3(D) &= D^7 + D^6 + D^3 + D^1 + 1, \\ g_4(D) &= D^7 + D^6 + D^5 + D^3 + D^2 + D^1 + 1. \end{aligned}$$

The first time a data packet is transmitted, only the output data from the rate-1 convolutional code defined by $g_1(D)$ is transmitted. If any of the N_p packets in a burst contains errors (detected using the CRC), those packets are retransmitted in the

next burst using the rate-1 convolutional code defined by $g_2(D)$, and so on. The receiver uses information from all the received versions of the packet (encoded with different generator polynomials) when decoding the data. In this way, the effective code rate varies between 1 and 1/4, depending on the number of retransmissions. After the fourth transmission of a packet, it is possible to continue cycling through the four generator polynomials for further retransmissions, and the effective code rate can be even lower than 1/4 (1/5, 1/6, ...).

BW2 does not use interleaving, and therefore it is not suitable to apply turbo equalization to the receiver for BW2. We do however note that in a recent proposal (Chamberlain, 2002) for enhancements to HDL, giving an extended version called HDL+, one of the proposed enhancements is to include interleaving.

Similar to the MIL-STD-188-110B 2400 bps waveform, BW2 uses an 8-PSK signal constellation, a frame pattern of 16 training symbols for every 32 data symbols, and multiplication by an 8-PSK scrambling sequence.

The burst duration, i.e., the time it takes to transmit N_p data packets á 1881 data bits, is $(640 + N_p * 400)$ ms. Thus, the maximum effective data rate for BW2 is 4409 bps, when $N_p = 24$ and the channel conditions are so good that no retransmissions are needed.

2.3.4.2 BW3, the robust traffic burst waveform

BW3 uses orthogonal Walsh functions as in the robust 75 bps waveforms. The number of bits per channel symbol is $Q = 1/4$ (generating 16 BPSK channel symbols for every 4 code bits), rather than 1/16 as in STANAG 4415. The Walsh functions are also scrambled to appear as 8-PSK on-air.

The data is encoded using the same rate 1/2 constraint length 7 code as in most HF serial-tone waveforms (see Fig. 2.4). Incremental redundancy is applied as in BW2, but the burst is not divided into several independent data packets. For the first-time transmission, the output from the upper shift register is used, and for the second-time transmission the output from the lower shift register is used. This means that the effective code rate is 1 or 1/2, depending on whether the data is retransmitted or not (or even lower if the data is retransmitted more than once by cycling through the two generator polynomials).

For BW3, the number of data bits in a burst is $25 + 8N_b$, where the burst size is determined by $N_b \in \{64, 128, 256, 512\}$. The burst duration is $(373.33 + 13.33N_b)$ ms. Thus, the maximum effective data rate is 572.5 bps, when $N_b = 512$ and no retransmissions are needed.

Standard[a]	Data rates (bps)	Code rate	Interleaver	Frame pattern[b]	Signal constellations[c]
MIL-STD-188-110A/B	2400	$1/2$	0.6s / 4.8s block	32 data / 16 known	8-PSK
	150 - 1200	$1/8$ - $1/2$		20 data / 20 known	BPSK, QPSK
STANAG 4285	75 - 2400	$1/16$ - $2/3$	0.852s / 10.24s convolutional	32 data / 16 known	BPSK, QPSK, 8-PSK
STANAG 4539	3200 - 9600	$3/4$	0.12s - 8.64s block	256 data / 31 known	QPSK, 8-PSK, 16-QAM, 32-QAM, 64-QAM
STANAG 4538 BW2	767 - 4409[d]	$1/4$ - $1/1$	None	32 data / 16 known	8-PSK
STANAG 4538 BW3	219 - 573[d]	$1/2$ - $1/1$[e]	0.96s - 6.93s block	Walsh functions	BPSK
STANAG 4415	75	1/2[e]	0.6s / 4.8s block	Walsh functions	BPSK

[a] STANAG4539 refers to MIL-STD-188-110B for data rates of 75-2400 bps and STANAG4415 for robust 75 bps. MIL-STD-188-110B is similar to STANAG 4539 for data rates above 2400 bps, and similar to the obsolete MIL-STD-188-110A for 75-2400 bps. MIL-STD-188-110A/B is similar to STANAG 4415 for 75 bps.

[b] The *major* pattern of known/unknown symbols. A longer sequence of known symbols is used at the beginning of transmission, and at regular intervals throughout, for convergence of the synchronization and channel estimation tasks. Robust low-rate waveforms do not use this kind of pattern; they rather use orthogonal modulation through Walsh functions.

[c] BPSK and QPSK constellations are scrambled to appear as 8-PSK on-air.

[d] Burst waveforms using code combining to adapt the data rate to changing conditions. Reduction in effective data rate due to packet overhead is included in data rate calculation.

[e] Spreading to 2400 baud through Walsh functions is not included in calculation of code rate.

Table 2.3 Standardized serial-tone HF waveforms (non-frequency hopping versions).

2.4 Waveform overview, and applicability of turbo equalization

In Sec. 2 of (Otnes and Tüchler, 2002c), we investigated which of the serial-tone HF waveforms were suitable for application of turbo equalization in the receiver. For this study, the key features of the serial-tone waveform standards were summarized as shown in Table 2.3.

As discussed in Sec. 2.3.1.2, a convolutional interleaver causes the delay to increase with the number of iterations in a turbo equalization-based receiver. Therefore, STANAG 4285 is not suitable for turbo equalization. Neither is the high-rate data link (HDL) traffic data waveform "BW2" in STANAG 4538, since it does not use interleaving at all. Waveforms with an interleaver suitable for Turbo equalization are, thus: MIL-STD-188-110A/B, STANAG 4539, STANAG 4415, and the low-latency data link (LDL) traffic data waveform "BW3" in STANAG 4538.

Receiver structures for waveforms using orthogonal Walsh modulation are different than for the other serial-tone waveforms. Iterative (turbo) processing can also be applied in receivers for these waveforms, but this has not been studied in this doctoral work.

Thus, the turbo equalization receiver framework described in this dissertation can be applied to waveforms using block interleaving and not using Walsh functions. This means that we are focusing on the waveforms described in MIL-STD-188-110B and STANAG 4539, for data rates in the range 150-9600 bps.

Chapter 3

Channel data analysis: Motivation for researching turbo equalization

This chapter is based on three conference papers written in 2001, presented at MILCOM (Otnes and Jodalen, 2001), Nordic shortwave conference (Otnes, 2001a), and NATO RTO IST Panel Symposium on Military Communications (Otnes, 2001b). The three papers present different types of results produced from analyses of the same sets of channel data.

3.1 Introduction

During disturbed ionospheric conditions, HF channels may exhibit characteristics that are very challenging for data transmission: Excessive Doppler spread and/or delay spread, as well as reduction in the received signal level. This occurs most frequently on high-latitude, equatorial, and NVIS (Near Vertical Incidence Skywave) channels.

Many recent efforts in the evolution of HF communications have been in increasing data rates on relatively benign channels, and waveforms with data rates of up to 9600 bps in 3 kHz bandwidths have been standardized (described in Chap. 2). At high latitude channels, however, even the medium data rate waveforms (2400 bps and below) suffer from low availability due to the degradations mentioned above, and one must often resort to 300 bps or the robust 75 bps waveform defined in STANAG 4415.

The ever-increasing need for data transfer in tactical and strategic communications calls for increasing the availability of any data rate, or equivalently increasing the maximum data rate that can be used under the conditions at any given time. The ultimate goal of the work presented in this dissertation is to increase the availability of each data rate, with focus on 1200 and 2400 bps. This can be achieved by designing new waveforms, or by developing improved detection schemes for currently

standardized waveforms.

In this chapter, we investigate which parameter is most important: SNR (signal-to-noise ratio), delay spread, or Doppler spread. In order to increase availability, a new waveform/detection scheme must have looser requirements to one or more of these parameters, maybe even at the cost of stricter requirements to other parameters.

We also present and compare two approaches to increase the availability compared to currently standardized serial-tone waveforms and modems.

We have used channel data obtained during the DAMSON (Doppler And Multipath SOunding Network) measurement campaign (Cannon et al., 2000). We have investigated two high latitude paths, one 200 km NVIS (Near Vertical Incidence Skywave) path and one 2000 km path, at different seasons and different geomagnetic activity. The analysis is broken down into different seasons and times of day.

3.2 Two different approaches for increasing availability and throughput

When a waveform cannot be used, it is because of one or more of the following reasons: (1) The Doppler spread is too large, (2) the delay (multipath) spread is too large, or (3) the SNR is too low. As will be shown in this chapter, it is more common that only one of (1) and (2) is occurring than that the Doppler spread and multipath spread are large simultaneously.

In this section, we present two different approaches that were considered in order to achieve the goal of increasing the availability of different data rates. One approach is to improve the delay/Doppler performance of the waveform, while the other approach is to improve the SNR performance.

3.2.1 Improving the delay/Doppler performance

The standardized serial-tone waveforms have training sequences (probes) of known symbols inserted regularly in order to estimate the time-varying channel impulse response. The length of the probe determines the maximum delay spread that can be tolerated, and the interval between the probes determines the maximum tolerable Doppler spread (Jorgenson and Moreland, 1999).

With this in mind, we envisioned a system that selects from a pool of waveforms at the same data rate. The Doppler spread and delay spread of the channel are estimated by some means which we do not discuss further here, e.g. during ALE link setup. This information is used to choose the optimal waveform.

The basic waveform at each data rate is one of the currently standardized serial-tone medium data rate waveforms: MIL-STD-188-110B, STANAG 4285, or the burst waveform BW2 used for HDL in the third generation ARCS standard. All of these

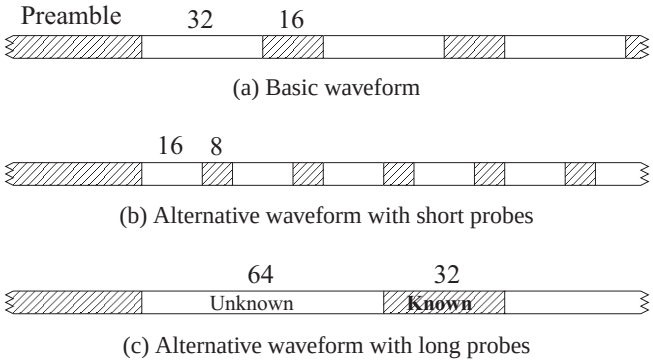


Figure 3.1 Standardized serial-tone HF waveforms and alternatives with different probe patterns.

waveforms have a frame structure similar to Fig. 3.1(a), with a preamble followed by a pattern of 32 unknown and 16 known symbols.

Our idea here is to improve the delay/Doppler performance by choosing among three different frame patterns at each data rate. If the length of unknown as well as known symbols is halved, as shown in Fig. 3.1(b), the maximum tolerable Doppler spread will be approximately doubled, while the maximum tolerable multipath spread will be approximately halved. The opposite can be achieved by doubling the intervals, as shown in Fig. 3.1(c).

The waveforms should be equal in all other respects than the probe patterns (if possible), and the total overall bandwidth efficiency of the frame pattern is the same. We have not considered the practical aspects of changing waveforms, which probably is simplest if automatic recognition of the different probe patterns can be implemented. The synchronization sequences in the probes as well as in the preamble may have to be different in order to estimate channel impulse responses of different lengths.

3.2.2 Improving the SNR performance

A lot of effort has been put into the development of serial-tone HF modems, and it may seem strange to hope that the SNR performance at a given data rate could be improved further. However, the concepts of turbo codes and turbo equalization have not yet been implemented in standard modems.

Turbo codes were included in proposals for the high data rate HF waveforms in STANAG 4539 (Kotlowski, Brakemeier, Mouy, Coutulleau, Vila and Mérel, 1998), but those proposals did not win through in the contest. It is known that turbo codes are very effective on AWGN (additive white Gaussian noise) channels, while achiev-

ing optimal performance on fading channels is more cumbersome (Hall and Wilson, 1998; Marsland and Mathiopoulos, 1998; Yuan, Feng and Vucetic, 2002). In this work we have not investigated the application of turbo codes to HF communications, and hence we do not give any indication here on how many dBs can be gained by using turbo codes rather than a single convolutional code on a fading HF channel. Turbo codes are not included in the standardized HF waveforms, and in order to use them in an HF modem one must either transmit a proprietary (non-standard) waveform or try to standardize new waveforms.

Another approach to improve the SNR performance, without making any changes to the standards, is to introduce turbo equalization in the receiver. This is the main topic of this dissertation, and in our first studies on applying turbo equalization to HF modems, we found that a receiver with turbo equalization has an SNR performance about 1-3 dB better than a conventional receiver based on a DFE.

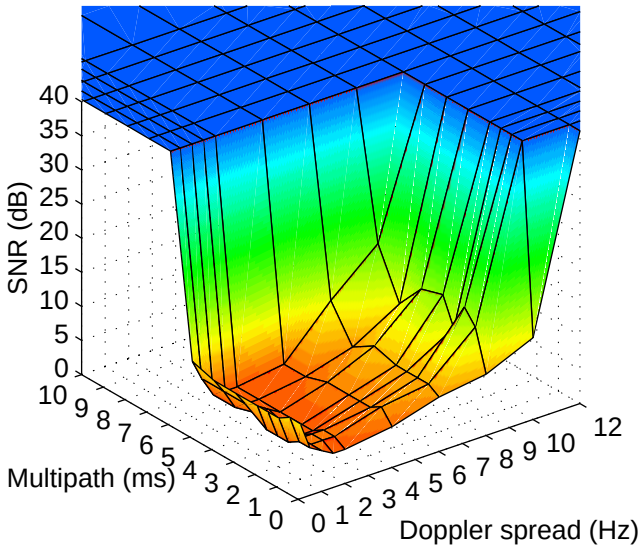
3.3 Analysis of availability

3.3.1 Approximation to waveform performance

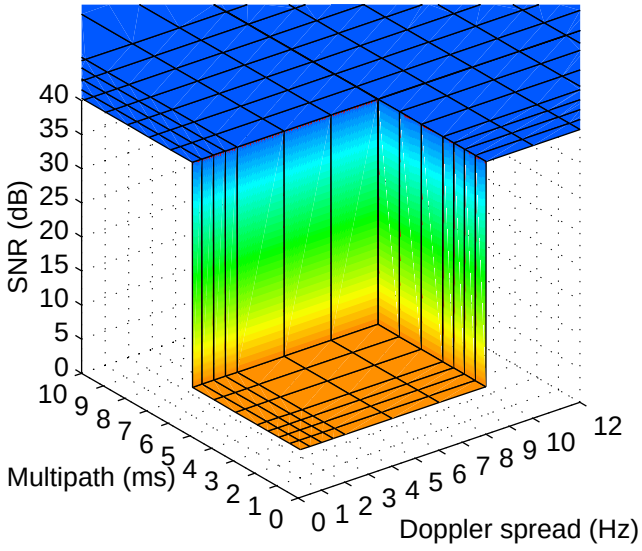
Performance surfaces (Arthur and Maundrell, 1997; ITU, 2000) is a convenient way to present the performance of HF modems. For each combination of multipath and Doppler spread, the required SNR to reach a BER (Bit Error Rate) of 10^{-3} is plotted. Fig. 3.2(a) (ITU, 2000) shows a typical performance surface for the STANAG 4285 1200 bps waveform, using the long interleaver setting. Performance surfaces typically show one region where the SNR requirement is roughly constant, one region where reliable communication cannot be achieved for any SNR, and a transition between these two regions. The transition is normally abrupt in the multipath direction, and soft in the Doppler direction. The increase in required SNR at Doppler spreads close to zero is due to the fading being slow compared to the interleaver length.

The performance surface in Fig. 3.2(a) has been obtained using a particular modem and interleaver setting; similar but not identical performance surfaces for the same data rate are found in (Arthur and Maundrell, 1997) and in STANAG 4539. We choose to approximate the performance surfaces as shown in Fig. 3.2(b), in order to simplify the analysis. This approximation should be insignificant compared to the fact that the performance surfaces are generated using an ideal two-path Watterson channel (see Secs. 4.3.1 and 4.4), while real-world scattering functions can be more complicated. The approximation gets worse at lower data rates, where the performance surfaces have even softer transitions in the Doppler direction.

Table 3.1 shows approximate requirements to SNR, Doppler spread and delay spread for different data rates, deduced from performance surfaces in (ITU, 2000) and STANAG 4539. The waveform is assumed to be available for communications



(a) Real performance surface.



(b) Approximated performance surface.

Figure 3.2 Typical performance surface for 1200 bps 8-PSK serial tone modem (ITU, 2000), and approximated performance surface of the type used in the present analysis.

Data rate	SNR	Doppler spread ν_d	Delay spread τ_m
2400 bps	> 14 dB	< 4 Hz	< 5 ms
1200 bps	> 7 dB	< 8 Hz	< 5 ms
600 bps	> 3 dB	< 12 Hz	< 5 ms
300 bps	> 0 dB	< 16 Hz	< 5 ms
150 bps	> –3 dB	< 10 Hz	< 5 ms
75 bps	> –7 dB	< 40 Hz	< 16 ms

Table 3.1 Approximate required SNR, Doppler spread and multipath spread when using current standard waveforms/modems.

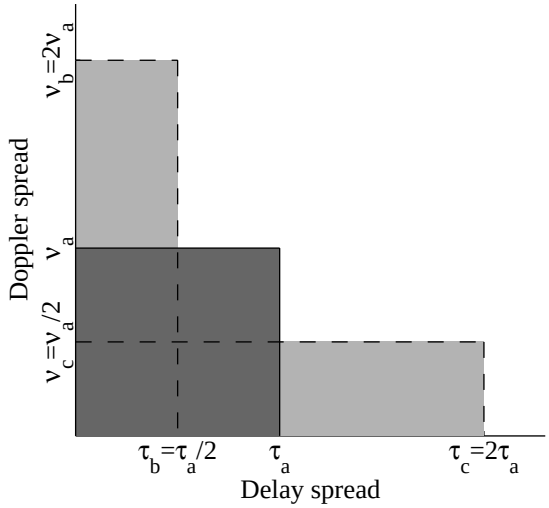


Figure 3.3 Combinations of delay/Doppler spread tolerated using the standard frame pattern (dark gray), and increased performance by introducing the alternative frame patterns (b) and (c) (light gray). The rectangles are an approximation to real-world performance, and the values of τ_a and ν_a depend on the data rate.

when the SNR, Doppler spread, and delay spread of the channel are within these limits.

When considering the approach of choosing among three different frame patterns, we approximate the combinations of Doppler and delay spreads that the different waveforms can tolerate as shown in Fig. 3.3. Here we see that by being able to choose among three different frame patterns, a larger area of the "delay/Doppler space" is accounted for.

3.3.2 DAMSON data used in the analysis

The DAMSON measurement program, described in (Cannon et al., 2000) and references therein, is a collaboration between QinetiQ (former Defence Evaluation and Research Agency, DERA) in the UK, Communication Research Center (CRC) in Canada, the Swedish Defence Research Agency (FOI), and the Norwegian Defence Research Establishment (FFI).

The DAMSON network has been gathering measurements of SNR, Doppler spread, and delay spread on high latitude HF channels since 1995. The most common setup has been two transmitters at Harstad and Isfjord, and receivers at Kiruna and Tuentangen. In each 10-minute interval, the network is cycling through 10 frequencies in the range 2.8-21.9 MHz, probing the channel with different waveforms at each frequency. A Barker-13 pulse sequence is used to measure the delay-Doppler characteristics of the channel, and the calculated scattering function has been used to estimate the composite delay spread and Doppler spread for each measurement.

In this study we have used the same data as in (Jodalen, Bergsvik, Cannon and Arthur, 2001), which are summarized in Table 3.2. Note that the sunspot numbers were relatively low in this period. We have concentrated on the short (190 km) NVIS path Harstad-Kiruna (H-K) and the long (2019 km) path Isfjord-Tuentangen (I-T). These two paths are shown on the map in Fig. 3.4. The auroral oval will always cover part of the long path (varying extent), while the NVIS path sometimes is completely within and sometimes completely outside the auroral oval.

The most important particularities of the data sets used in the analysis can be summarized as follows:

- Delay spreads are mainly below 5 ms, except at H-K in the winter period where delay spreads up to 10 ms are common.
- Doppler spreads are mainly below 10 Hz, but some data points with Doppler spreads up to 50 Hz are seen at H-K in the summer and in the disturbed period, particularly at the highest frequencies.

Season	Days[a]	# days[a]		Average	Smoothed
		I-T	H-K	K-index	sunspot number
Winter	021297-170198	25	24	0-1[b]	39-44
Summer	110698-080798	26	28	2-3	60-65
Disturbed	260898-030998	9	9	5	65-70

[a] Some days are missing and some days are incomplete within each period
[b] Five moderately disturbed days (K=3-4).

Table 3.2 DAMSON data used in this chapter.

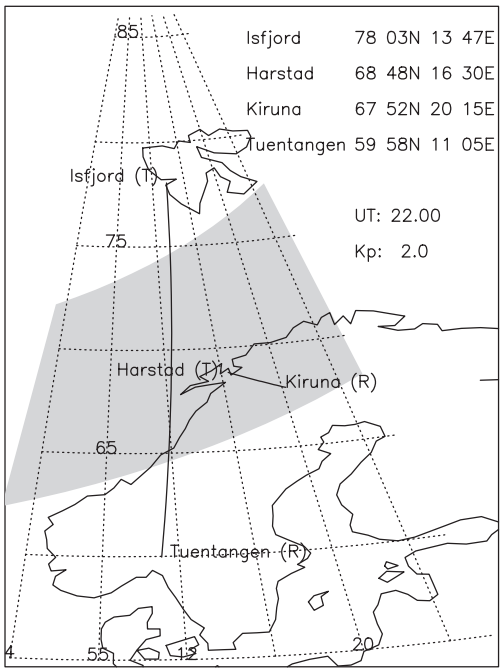


Figure 3.4 Map showing the positions of the two paths analyzed in this chapter, and the position of the auroral oval for moderately disturbed conditions.

- At both paths the SNR values are lower in the disturbed period than in the other periods.

Fig. 3.5 shows the simultaneous distribution of Doppler spread and delay spread for the data sets. All DAMSON frequencies are included, and data points with SNR below 7 dB (threshold SNR for 1200 bps waveform) have been deleted from the data sets before generating the histograms. The log-scaled color code shows the number of received probes within the Doppler/delay spread limits of each square. These plots show that when the SNR is large enough to support 1200 bps or higher, Doppler spread and multipath spread rarely are large simultaneously. Similar conclusions can be drawn from graphs in (Angling and Cannon, 1996).

Inscribed in the histograms are rectangles which illustrate the availability of the three 1200 bps waveforms with different frame patterns, described earlier. There is also a hyperbola, illustrating the availability of a hypothetic 1200 bps system which can choose from an infinite number of waveforms with different probe patterns; the restriction on all these waveforms is that the spreading factor (product of delay spread and Doppler spread) must be less than 5 ms $\cdot$ 8 Hz = 0.04.

The high Doppler spreads seen at the Harstad-Kiruna path, particularly in the

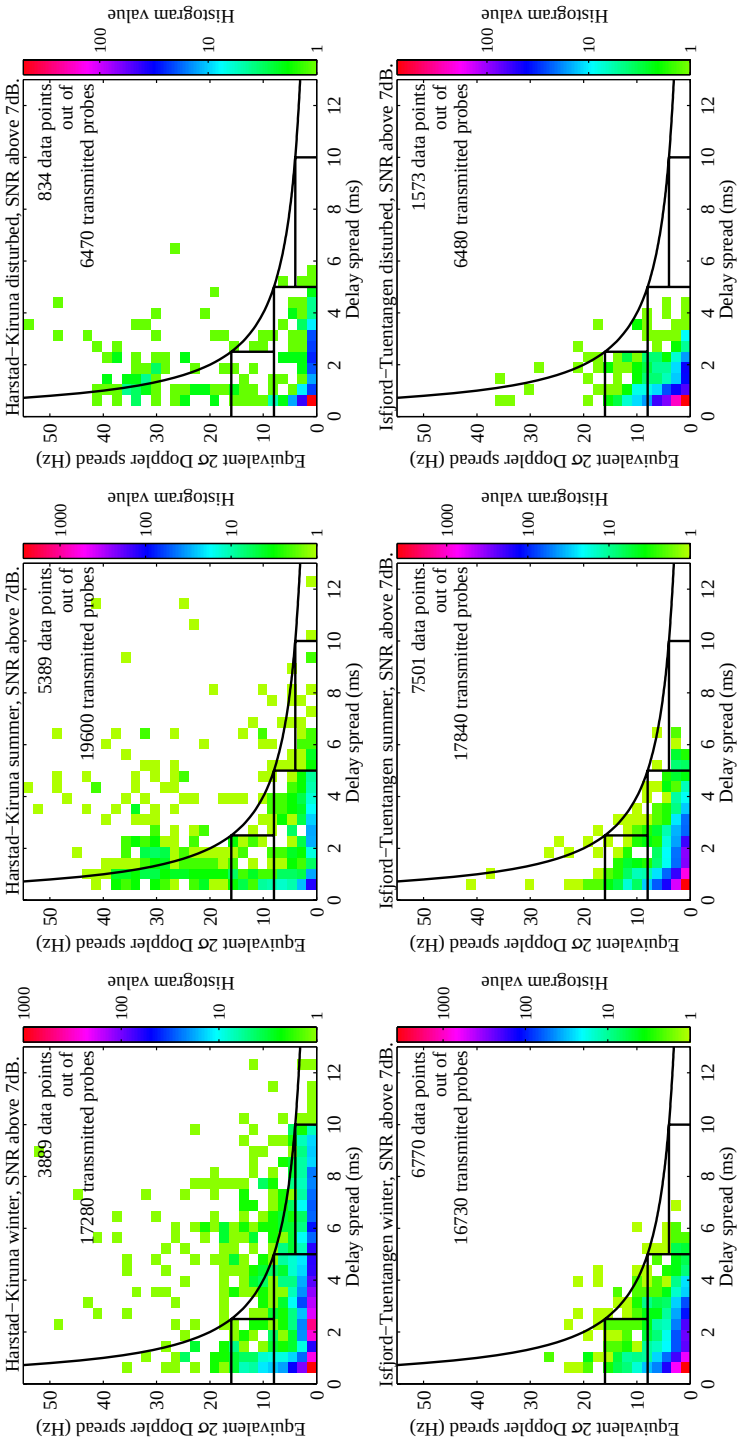


Figure 3.5 Two-dimensional histograms, showing the simultaneous distribution of delay spread and Doppler spread (all 10 DAMSON frequencies are included). The rectangles illustrate the availability of 1200 bps waveforms with three different frame patterns, and the hyperbola is a curve of constant spreading factor.

summer, mainly stem from frequencies above the MUF (maximum usable frequency) (Willink, Davies, Angling, Jodalen and Lundborg, 1999), and high multipath spreads may be due to off-great-circle paths (Warrington, Jackson, Stocker, Jones and Lundborg, 2000). Contrary to what might be expected, the data entitled ”disturbed” does not seem to have the largest occurrence of high multipath and Doppler spreads. This is partly because the ”disturbed” data set is smaller than the other data sets, giving a poorer estimate of the probability of rare events, and partly because auroral absorption may suppress extra (longer) paths and above-the-MUF modes.

3.3.3 Calculating the availability

We have assumed an ALE system which is able to select the best frequency at any time, selected from the 10 frequencies used by DAMSON in the range 2.8-21.9 MHz (this set of frequencies is of course suboptimal, but will always include a frequency close to the optimal working frequency). In each 10-minute interval, one DAMSON probe has been sent at each of the 10 frequencies. The receivers have been listening to Isfjord and Harstad in alternate 10-minute intervals, so each path has measurements in three 10-minute intervals per hour.

The availability is calculated as

$$\text{Modem availability} = \frac{N_C}{N} \tag{3.1}$$

where N is the total number of 10-minute intervals in which measurements were performed on the path in the measurement period (including those where no DAMSON probe was received because there was no propagation, see also (Jodalen et al., 2001)), and N_C is the number of 10-minute intervals where communication is possible at least at one of the DAMSON frequencies for a given set of requirements. I.e., we say that a data rate is “available” when ν_d, τ_m, and SNR are within the limits shown in Table 3.1 (and extended as shown in Fig. 3.3 for the case of choosing among three different frame patterns) at *any one* of the 10 DAMSON frequencies.

As shown in (Jodalen et al., 2001), the availability when using all 10 frequencies can normally be achieved by using the 4-6 best frequencies.

The analyses for each of the six data sets have been broken down into four different times of day to account for the diurnal variation of the ionosphere. The SNR measured by DAMSON has been used directly, not compensating for the fact that other antennas and power levels may be used on an operative HF link. One can compensate for this on the plots with the SNR requirement on the abscissa, by shifting the SNR scale by the difference in SNR between an operational system and the DAMSON system.

In the following, we will present different analysis results generated using the method described above.

3.4 Analysis results and discussion

3.4.1 Availability as a function of the waveform requirements

In this section, we present the results reported in (Otnes, 2001a).

In order to assess how important each of the requirements (to SNR, delay spread, and Doppler spread) is for the overall modem availability, we have varied one requirement at a time while keeping the two other requirements constant at values called ”default requirements”. The default requirements to SNR, delay spread and Doppler spread have been taken from Table 3.1, and in all the plots included in this section the default requirements correspond to 1200 bps.

The results for each of the six data sets are shown in Figs. 3.6-3.11. The total number of 10-minute intervals N used for calculating each availability curve is included in the legend below each plot.

3.4.1.1 The Isfjord-Tuentangen path

The analysis results for the Isfjord-Tuentangen path are shown in Figs. 3.6-3.8. On this long path, we note the following:

- The availability is lowest at night (1-7 UTC) at all seasons.
- In the winter and summer periods investigated, an SNR requirement of 10-15 dB will give an availability above 80% at all times of day.
- In the disturbed period, the availability is decreased at all times of day, but can be increased somewhat by loosening the SNR requirement. The gradient of availability vs SNR requirement in the 5-10 dB SNR range is about -1% per dB.
- A delay spread requirement of 1-2 ms is sufficient at all the seasons investigated here.
- A Doppler spread requirement of 4 Hz is sufficient at all seasons, except at night in the disturbed period. At night in the disturbed period, the Doppler spread requirement needs to be as large as 8 Hz to achieve an availability of 64%, while a Doppler spread requirement of 4 Hz corresponds to an availability of 58%.

3.4.1.2 The Harstad-Kiruna path

The analysis results for the Harstad-Kiruna path are shown in Figs. 3.9-3.11. On this short NVIS path, we note the following:

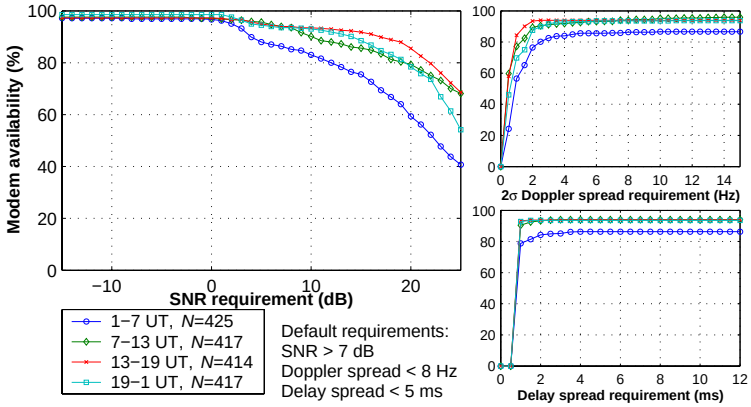


Figure 3.6 Availability analysis, Isfjord-Tuentangen in the winter.

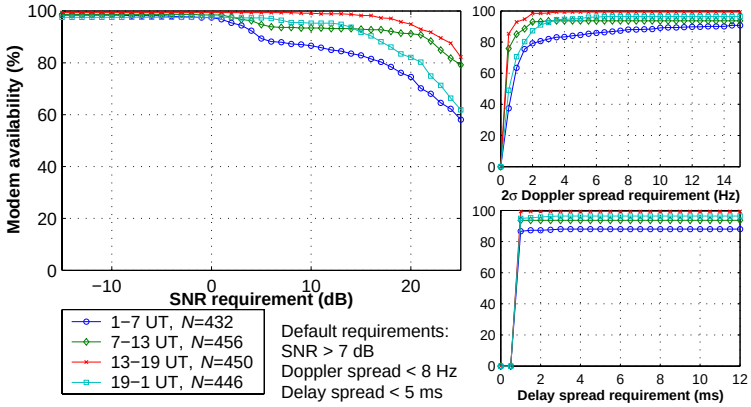


Figure 3.7 Availability analysis, Isfjord-Tuentangen in the summer.

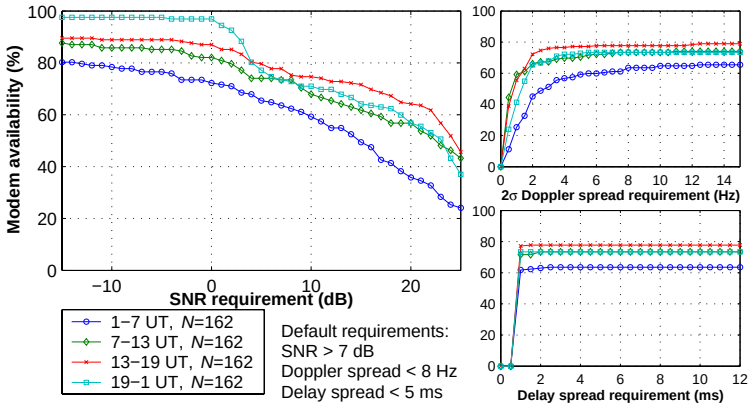


Figure 3.8 Availability analysis, Isfjord-Tuentangen in the disturbed period.

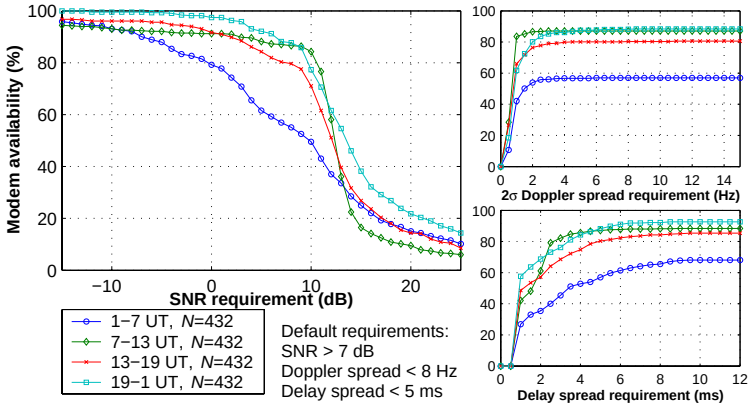


Figure 3.9 Availability analysis, Harstad-Kiruna in the winter.

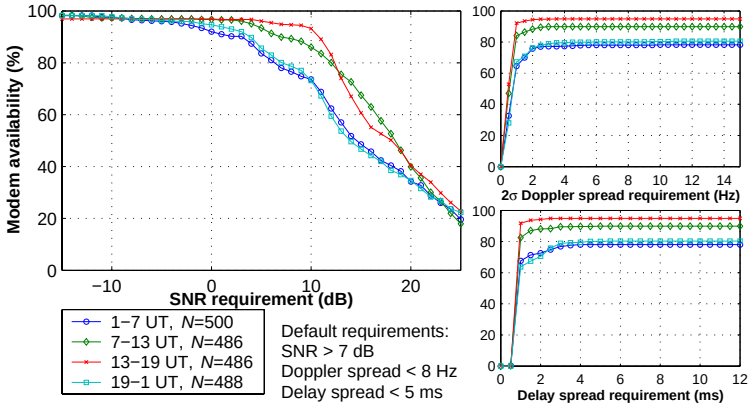


Figure 3.10 Availability analysis, Harstad-Kiruna in the summer.

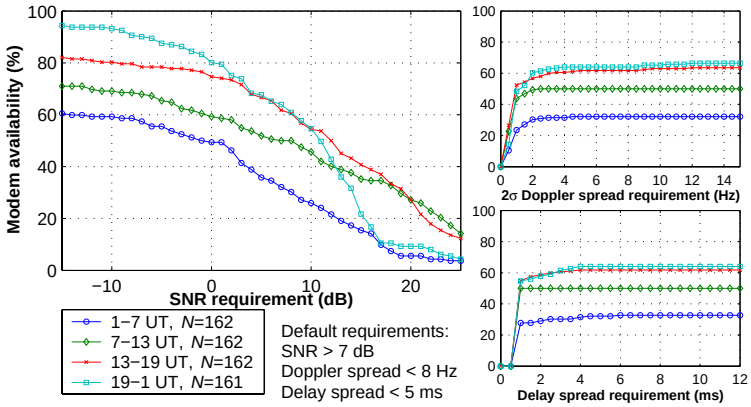


Figure 3.11 Availability analysis, Harstad-Kiruna in the disturbed period.

- In the winter, except for at night, the SNR is quite predictable: Decreasing the SNR requirement from 15dB to 10dB will increase the availability by 40-85%[1]. If the system happens to be operating in this SNR region (e.g. 2400 bps and antennas and power levels equal to the DAMSON system), every dB of decreased SNR requirement is very important.

- In the winter, the SNR is lower at night than at other times of day. This is probably because, under quiet conditions, auroral absorption is most common between 1 and 7 UTC. Here, the SNR requirement needs to be as low as 0 dB for an availability of 80%; for larger SNR requirements the gradient is about -3.5% per dB.

- In the summer, the SNR requirement for a given availability is about 5 dB lower at 19-7 UTC than at 7-19 UTC. At all times of day, the gradient is about -4% per dB when the availability is below 80%.

- In the disturbed period, all the curves of availability vs SNR requirement have gentle slopes, indicating large variations in the SNR. Also, the availabilities are lower than in the other periods. Here, gradients of -1 to -6% per dB are seen.

- In the summer and the disturbed period, delay spread requirements of 2-4 ms are sufficient.

- In the winter, large delay spreads occur frequently. In the night, a delay spread requirement of 9 ms is needed to achieve maximum availability, while a delay spread requirement of about 6 ms is needed at other times of day.

- A Doppler spread requirement of 2-4 Hz is sufficient in all the periods investigated here.

3.4.1.3 The effect of geomagnetic activity

The disturbed period analyzed above was a period of generally high geomagnetic activity. We also performed an additional analysis of the effect of geomagnetic activity, using the data from the winter period.

As noted in Table 3.2, the winter period consists mostly of quiet days and also of a few disturbed days. In order to investigate the effect of increased geomagnetic activity, we performed the availability analysis on the Harstad-Kiruna path using only 6-hour intervals with mean Q-index above 1.3. The results are shown in Fig. 3.12.

At 1-7 UTC and 7-13 UTC, the mean Q-index is above 1.3 only two days in the winter period, and these curves are therefore based on only 36 ten-minute intervals

[1]When we say that the availability has increased e.g. by 40%, we mean that it has increased e.g. from 30% to 70%. We also use this definition when calculating gradients.

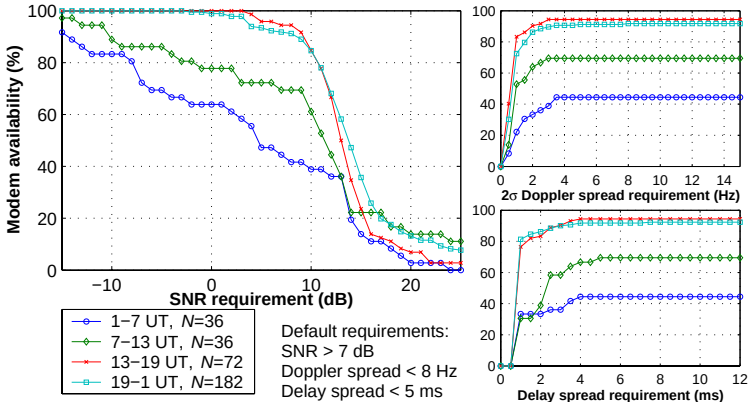


Figure 3.12 Analysis results for 6-hour periods with mean Q-index above 1.3, Harstad-Kiruna in the winter.

(two 6-hour periods), so they only give a coarse indication of the effect of increased geomagnetic activity. By comparing with Fig. 3.9, we see that the availability at night has decreased further, and at 7-13 UTC low SNRs are more common at increased Q-indices. This can be explained by auroral absorption stretching into the morning sector at increased geomagnetic activity (Goodman, 1992, p. 136). In Fig. 3.11 we see that there is frequent morning absorption also in the disturbed period, where the mean Q-index is well above 1.3 in all but a few 6-hour periods.

Also note that on the disturbed days investigated in Fig. 3.12, a delay spread requirement of 4 ms is sufficient (compared to the 9 ms of Fig. 3.9). This may be because the longer signal paths with excessive delay have not been observed because they suffer from stronger absorption than the shorter signal paths.

At Isfjord-Tuentangen, we do not see any significant differences when investigating the days with high Q-indices only.

The Q-indices used above are local values measured in Kiruna (4 measurements per hour), and have been downloaded from the Swedish Institute of Space Physics (IRF) at http://www.irf.se.

Path	Season	Number of waveforms	Availability of 1200 bps Time of day (UTC)				Availability of 2400 bps Time of day (UTC)			
			1-7	7-13	13-19	19-1	1-7	7-13	13-19	19-1
Harstad-Kiruna	Winter	1	**56,9**	87,0	**80,3**	**88,2**	**27,5**	**21,8**	**29,9**	**39,3**
		3	**68,1**	88,7	**84,7**	**92,4**	**34,5**	**25,9**	**34,5**	**46,1**
		∞	68,1	88,9	85,2	92,6	35,2	26,2	35,6	48,1
	Summer	1	78,0	89,9	94,9	80,1	51,0	72,6	66,5	47,3
		3	78,4	89,9	94,9	80,5	52,8	72,6	67,3	49,0
		∞	78,6	89,9	94,9	80,7	53,4	72,6	67,7	49,8
	Disturbed	1	32,1	50,0	61,7	**64,0**	17,3	37,7	40,7	31,7
		3	32,7	50,0	63,6	**66,5**	17,3	37,7	42,0	32,3
		∞	33,3	50,0	66,0	67,7	17,3	37,7	42,6	32,3
Isfjord-Tuentangen	Winter	1	86,4	94,0	94,0	93,5	74,8	84,7	92,0	**88,0**
		3	86,6	95,2	94,0	93,8	76,5	86,1	92,3	**90,2**
		∞	86,6	96,2	94,0	93,8	76,9	87,1	92,3	90,4
	Summer	1	**88,0**	93,6	99,3	96,4	**78,0**	92,8	98,7	**91,0**
		3	**90,7**	93,6	99,3	96,4	**83,3**	93,2	98,7	**93,7**
		∞	90,7	93,6	99,3	96,4	86,1	93,2	98,7	93,7
	Disturbed	1	63,6	73,5	77,8	73,5	**46,9**	**59,9**	70,4	65,4
		3	65,4	74,1	79,0	73,5	**52,5**	**63,0**	72,2	66,7
		∞	65,4	74,1	79,0	73,5	53,1	64,2	72,8	66,7

Table 3.3 Availability (%) of 1200 and 2400 bps modems, using 1, 3, or ∞ waveforms. It is assumed that the system can choose among all DAMSON frequencies.

3.4.2 Increase in availability when using different frame patterns

In this section, we present the results reported in (Otnes and Jodalen, 2001).

Table 3.3 shows the availability (calculated as described in Sec. 3.3) of the current standard using one waveform per data rate, a system which can choose from three waveforms at each data rate, and a hypothetic system which can choose from an infinite number of waveforms (illustrated by the hyperbolas in Fig. 3.5). The cases where the availability percentage increases by more than 2 by going from one to three waveforms are set in **bold**. Observe that the difference is always small between 3 and an infinite number of waveforms, i.e., there is little to gain from implementing more than 3 waveforms per data rate.

The reader may note that the availabilities given in Table 3.3 differ slightly from those in (Jodalen et al., 2001) even though the same data sets have been used. This is because (Jodalen et al., 2001) uses the exact performance surfaces given in (Arthur and Maundrell, 1997), while we use an approximation to the performance surfaces given in (ITU, 2000).

The data set where there is most to gain from changing the probe pattern, is the NVIS path Harstad-Kiruna in winter, illustrated in Fig. 3.13. The reason is that large multipath spreads are common in this data set. Note that the availability of 1200 bps can be increased the most at night, when it is worst. At 2400 bps the availability is in the 20-40% range at all hours, and implementing 3 waveforms increases the availability percentage noticeably at all hours.

On the long path Isfjord-Tuentangen in the disturbed period, the availability of 2400 bps benefits more than 1200 bps from using 3 waveforms instead of 1. This is because Doppler spreads in the 4-8 Hz range are common in this data set.

We have also investigated the availability if the system is forced to operate at a particular frequency, as shown in Fig. 3.14. In this case, (on Harstad-Kiruna in winter) the availability of 1200 bps benefits the most from using different probe patterns, while the availability of 2400 bps will mainly be restricted by the SNR requirement.

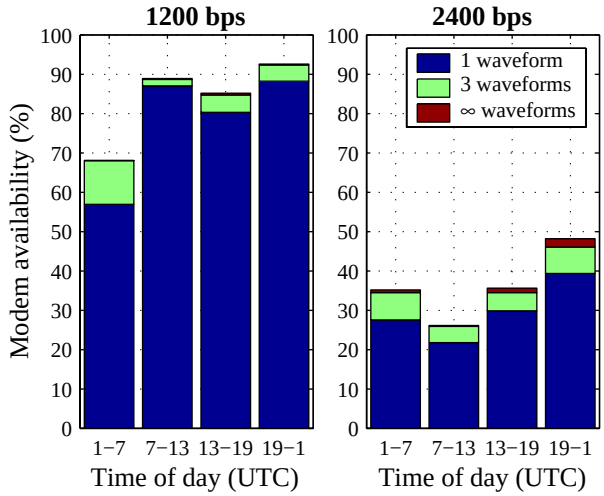


Figure 3.13 Availability at different times of day on the path Harstad-Kiruna in the winter, when the system can choose among all DAMSON frequencies.

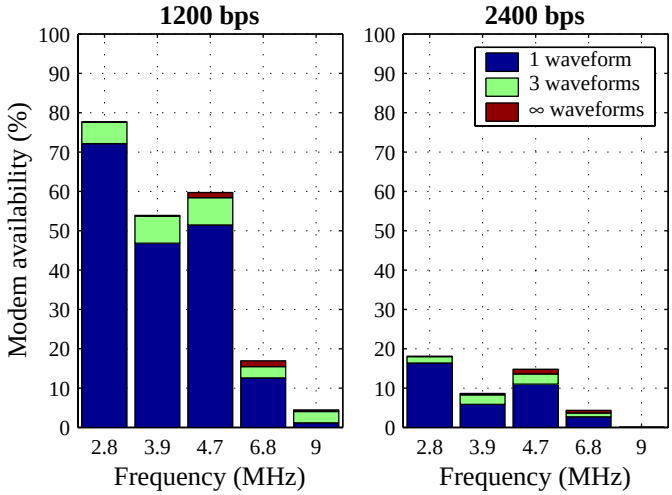


Figure 3.14 Availability at each frequency on the path Harstad-Kiruna in the winter, averaging over all hours of day.

3.4.3 Increase in throughput from improving delay/Doppler performance vs improving SNR performance

In this section, we present the results reported in (Otnes, 2001b).

From the availability of the different data rates, a throughput measure can be calculated as follows: Throughput = sum over all data rates of (data rate · probability that this is the maximum available data rate). For example, if 75 bps had an availability of 30%, 150 bps had an availability of 10%, and all higher data rates were unavailable, the throughput would be $(150 \cdot 0.1 + 75 \cdot (0.3 - 0.1))$ bps = 30 bps.

In Fig. 3.15, this is illustrated using one of the data sets. As could be expected, the throughput depends mostly on the availability of 2400 bps, but it also depends on the availability of lower data rates. It should be kept in mind that the throughput defined here is not the throughput of an operational system (with link setup latency and protocol overhead), but is the time average of the maximum available data rate.

We now want to compare the two approaches to increasing modem availability described in Sec. 3.2. To do this, we first calculate the availability of each data rate and the equivalent throughput using the thresholds shown in Table 3.1. This corresponds to the performance using standard modems/waveforms. In order to investigate the effect of improving delay/Doppler performance of the waveforms, we repeat the calculation with the tolerable delay/Doppler combinations extended as shown in Fig. 3.3. Similarly, the effect of improving the SNR performance is assessed by decreasing the SNR thresholds for all data rates in Table 3.1 by 1, 2, and 3 dB. By doing this,

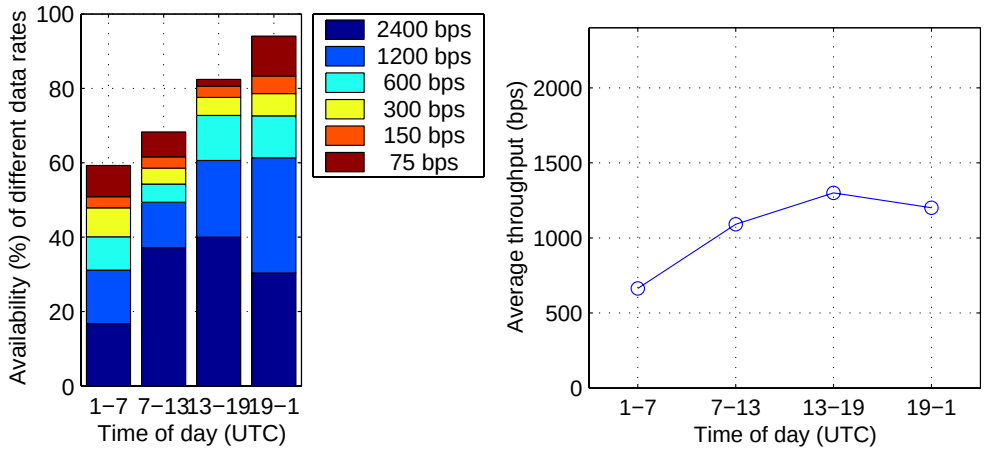


Figure 3.15 Availability of the different data rates on the path Harstad-Kiruna in the disturbed period, and equivalent throughput. The availabilities are calculated using the threshold values in Table 3.1, which correspond to current standard modems.

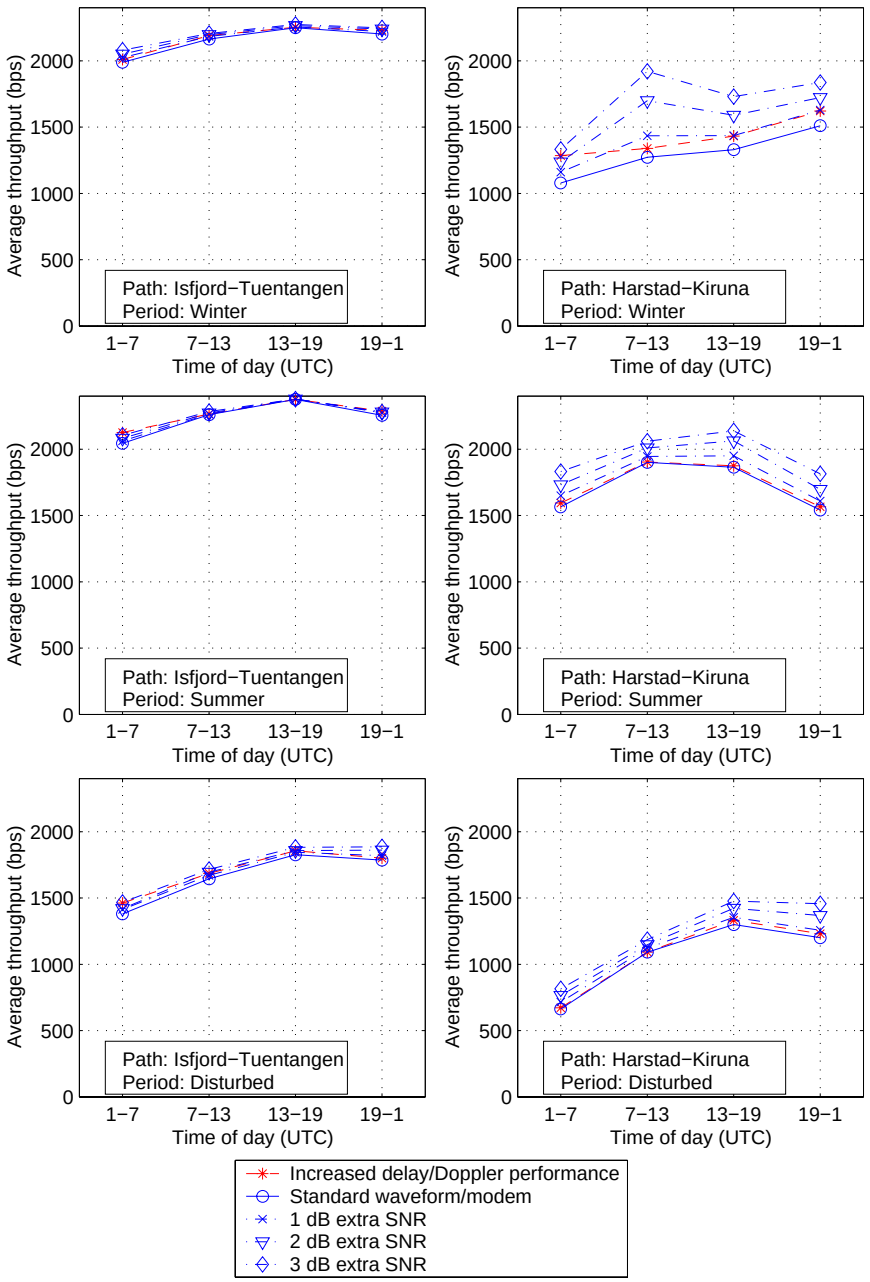


Figure 3.16 Equivalent throughput for the different data sets, and the effect of improving delay/Doppler performance (by choosing among three different frame patterns) or SNR performance.

we can see the effect of improving modem performance by 1, 2, and 3 dB (assuming that the improvement is the same number of dBs for all data rates)

Using the method described in Sec. 3.3, we have calculated the availabilities of the different data rates, for the six data sets under consideration. The resulting equivalent throughput for each data set is shown in Fig. 3.16. The effect of improving delay/Doppler performance as well as SNR performance is included in the plots.

On the long path Isfjord-Tuentangen, improving modem performance, in the delay/Doppler sense as well as in the SNR sense, has small effect on the equivalent throughput. In the winter and summer periods, the throughput is close to the theoretical maximum of 2400 bps even without modem improvements. In the disturbed period, the effect of increasing delay/Doppler performance is the same as improving SNR performance by 3 dB, 1.5 dB, 2 dB, and 0.5 dB, at the different times of day respectively. However, the increase in throughput is marginal for all these cases, so no definite conclusions can be drawn from this.

On the NVIS path Harstad-Kiruna more profound effects are seen from improving modem performance. This path has worse propagation conditions than the long path, and therefore the potential for improvement is also larger. For these data sets, improving SNR performance leads to relatively large increases in equivalent throughput. Improving delay/Doppler performance has virtually no effect except in the winter period, where it has the same effect as improving SNR performance by about 1 dB.

3.5 Conclusions

The conclusions from the three different analyses described in this chapter point in the same direction, as can be expected since they are based on the same channel data.

When the availability is low, there is generally more to gain from improving the SNR requirements than from improving the delay spread or Doppler spread requirements. The increase in availability when decreasing the SNR requirement by 1 dB can be between 0 and 20%, and on the short path Harstad-Kiruna it is most commonly about 3-4%.

Under disturbed conditions, the main problem is absorption, whilst the same availabilities could be achieved with delay spread and Doppler spread requirements below that of the currently standardized waveforms. Therefore, there may be something to gain by using waveforms where Doppler spread and delay spread performance are traded off for better SNR performance.

The requirements to Doppler spread and delay spread reported here are consistent with earlier DAMSON analyses using other data sets (Angling, Cannon, Davies, Willink, Jodalen and Lundborg, 1998; Bröms, Lundborg, Jodalen and Bergsvik, 1997; Willink et al., 1999). These references show larger Doppler and delay spreads than

those reported in this paper, but they consider all frequencies separately and not only the optimal frequency as we have done here. When looking only at the frequencies that are optimal from an SNR point of view, Doppler spreads and delay spreads reported in (Angling et al., 1998) are mainly below 12 Hz and 4 ms, while they are larger at other frequencies. This indicates that the main benefit from using a very robust low-rate waveform like STANAG 4415 at the optimal frequency is the reduced SNR requirement, while the very large Doppler and delay spread capabilities of STANAG 4415 mainly increases its availability at suboptimal frequencies.

The proposed system choosing among different frame patterns would complicate the design of modem hardware and/or software as well as communication protocols significantly. The increase in availability is generally small; the largest increases found are shown in Figs. 3.13 and 3.14. These benefits are hardly large enough to justify the increase in system complexity.

Generally, these channel data show that improving the SNR performance of modems is more important than improving delay/Doppler performance. However, adaptive equalization is still necessary, as the Doppler and delay spread are certainly larger than zero.

Realizing this, we started attacking the SNR performance. For this, we chose to focus on applying turbo equalization to HF modems.

The data used in these analyses have been collected at relatively low sunspot numbers, and it is desirable to perform similar analyses using DAMSON data collected close to solar maximum before drawing definite conclusions.

Chapter 4

Channel models

In this chapter, we present different channel models commonly used for HF communications and other radio communication systems. The chapter mainly describes textbook knowledge in the notation used in this dissertation. The presentation is simplified in order to focus on the points important to this work; for rigorous derivations the reader is referred to text books, e.g. (Steele and Hanzo, 1999; Molisch, 2001; Proakis, 1995), or the classical paper by Bello (1963). A tutorial on modelling fading channels can be also be found in (Sklar, 1997a; Sklar, 1997b).

4.1 Baseband equivalent of bandpass signals and systems

A bandpass signal $s(t)$ with a carrier (center) frequency ω_c is commonly written as

$$s(t) = a(t)\Re\{e^{j(\omega_c t+\phi(t))}\} = \Re\{x(t)e^{j\omega_c t}\} \tag{4.1}$$

where the complex signal $x(t) = x_r(t) + jx_i(t) = a(t)e^{j\phi(t)}$ is called the complex envelope or the baseband equivalent signal of $s(t)$. The amplitude and phase of $s(t)$ will be given by

$$a(t) = |x(t)| = \sqrt{x_r^2(t) + x_i^2(t)} \tag{4.2}$$

$$\phi(t) = \angle x(t) = \arctan \frac{x_i(t)}{x_r(t)} \tag{4.3}$$

If the signal $s(t)$ is transmitted, and the only effects of the channel are to induce a delay τ, a gain $g(t,\tau)$ (which is nonzero only for a single delay τ), and a phase shift $\theta(t,\tau)$, the received signal $r(t)$ can be written

$$r(t) = g(t,\tau)a(t-\tau)\Re\{e^{j\omega_c(t-\tau)+\phi(t-\tau)+\theta(t,\tau)}\} = \Re\{h(t,\tau)x(t-\tau)e^{j\omega_c t}\} \tag{4.4}$$

where $h(t,\tau)$ is called the complex gain of the channel, and is given by

$$h(t,\tau) = g(t,\tau)e^{j(-\omega_c\tau+\theta(t,\tau))} \tag{4.5}$$

In the next section, we extend this example to multipath channels, i.e., $g(t,\tau)$ and $h(t,\tau)$ being nonzero at more than a single delay.

If the received signal is corrupted by additive bandlimited Gaussian noise with an effective noise bandwidth of B (Hz) and a power spectral density of N_0 (W/Hz), the baseband equivalent noise signal $w(t) = w_r(t) + jw_i(t)$ will be circularly symmetric complex Gaussian noise with variance

$$\sigma_w^2 = E\{w(t)w^*(t)\} = N_0 B \tag{4.6}$$

The definition of circularly symmetric complex Gaussian noise is that the real and imaginary part $w_r(t)$ and $w_i(t)$ are independent Gaussian random variables with zero mean and equal variance, i.e.,

$$E\{w_r^2(t)\} = E\{w_i^2(t)\} = \frac{\sigma_w^2}{2} \tag{4.7}$$

$$E\{w_r(t)w_i^*(t)\} = 0 \tag{4.8}$$

When these conditions are fulfilled, $w(t)$ will have the complex Gaussian probability density function given by

$$p_w(w) = \frac{1}{\pi\sigma_w^2} e^{\frac{-|w|^2}{\sigma_w^2}}, \quad w \in \mathbb{C} \tag{4.9}$$

4.2 Tapped delay line channel model

Multipath fading channels are commonly modelled using a tapped delay line, as shown in Fig. 4.1. Here, K is the number of paths, and the jth path exhibits the complex gain $h(t,\tau_j)$ and the delay

$$\tau_j = \sum_{i=0}^{j} \tau_i' \tag{4.10}$$

The received signal is then

$$z(t) = \sum_{i=0}^{K-1} h(t,\tau_i)x(t-\tau_i) + w(t) \tag{4.11}$$

where $w(t)$ is complex (baseband equivalent) additive white Gaussian noise with variance σ_w^2.

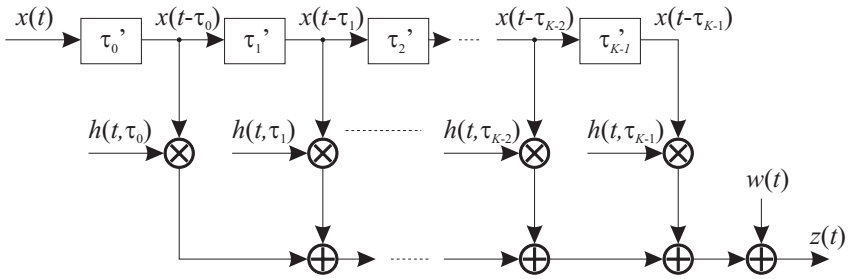


Figure 4.1 Tapped delay line channel model.

The tapped delay line channel model may be seen is a special case of the Bello system function $h(t,\tau)$, called the time-varying impulse response or the input delay-spread function (Bello, 1963). This function is defined for all possible delays $\tau \in [0,\infty)$, such that

$$z(t) = \int_0^\infty h(t,\tau)x(t-\tau)d\tau + w(t) \tag{4.12}$$

When the tapped delay line channel model is used, the connection between (4.11) and (4.12) is

$$h(t,\tau) = \sum_{i=0}^{K-1} h(t,\tau_i)\delta(\tau-\tau_i) \tag{4.13}$$

4.2.1 The WSSUS assumption

When the multipath environment is randomly time-varying, each of the tap gains $h(t,\tau_i)$ will vary with time t. The actual delays τ_i may also vary with t, in which case the model (4.12) should be used rather than the simpler tapped delay line model (4.11).

To make the channel model mathematically tractable, the assumptions of wide sense stationarity (WSS) and uncorrelated scattering (US) are often applied to $h(t,\tau)$. Uncorrelated scattering implies that the channel impulse response is independent for different delays, i.e.,

$$E\{h(t_a,\tau_a)h^*(t_b,\tau_b)\} = E\{h(t_a,\tau_a)\}E\{h^*(t_b,\tau_b)\}, \quad \forall t_a,t_b, \quad \tau_a \neq \tau_b \tag{4.14}$$

Wide sense stationarity implies that the second order statistics of $h(t,\tau)$ does not vary with t. If we also assume US and a tapped delay line channel model (i.e., that $h(t,\tau)$ is only non-zero for discrete delays $\tau = \tau_i$ which are constant with t), the autocorrelation of each tap gain is defined as

$$R_h(u,\tau_i) = E(h(t,\tau_i)h^*(t-u,\tau_i)) \tag{4.15}$$

In reality, neither the assumption of WSS nor of US holds for all t and τ. Therefore, the concept of a quasi-WSSUS channel is used. This means that the WSSUS assumptions do not necessarily hold for all values of t and τ, but do hold for the range of t and τ observed by a communication system. The second-order statistics are assumed to be constant over a "large" time interval under which communications is taking place (but if the channel is used again e.g. an hour later, they may be different), and the bandwidth of the system is assumed to be so small that correlated scatterers with different delays can not be resolved in the receiver, and appear to have the same delay (in which case they just add together).

For a mathematically rigorous definition of WSSUS and quasi-WSSUS channels, the reader is referred to (Bello, 1963).

4.2.2 The Doppler spectrum

The Doppler spectrum $S_h(\nu, \tau_i)$ of a tap gain $h(t, \tau_i)$ is defined as the Fourier transform of the autocorrelation $R_h(u, \tau_i)$, i.e.,

$$S_h(\nu, \tau_i) = \int_{-\infty}^{\infty} R_h(u, \tau_i) e^{-j2\pi\nu u} du \tag{4.16}$$

$$R_h(u, \tau_i) = \int_{-\infty}^{\infty} S_h(\nu, \tau_i) e^{j2\pi\nu u} d\nu \tag{4.17}$$

Generally, a wide Doppler spectrum means that the channel is varying fast, while a narrow Doppler spectrum means that the channel is almost constant. The approximate width of the Doppler spectrum is called the Doppler spread or the fading rate.

The Doppler spectrum can also be viewed as the statistical distribution of the Doppler shift (frequency change) introduced by the channel, hence the name.

The energy (or average power gain) $E_h(\tau_i)$ of the tap is defined as

$$E_h(\tau_i) = \int_{-\infty}^{\infty} S_h(\nu, \tau_i) d\nu = R_h(0, \tau_i) \tag{4.18}$$

and the total energy of a channel modelled as a tapped delay line is given by

$$E_h = \sum_{i=0}^{K-1} E_h(\tau_i) \tag{4.19}$$

4.2.3 Generating the tap gains

In a discrete time simulation setup, the tap gains $h_m(\tau_i) = h(mT_m, \tau_i)$ can be generated as follows. A sampled (discrete time) complex white Gaussian random process

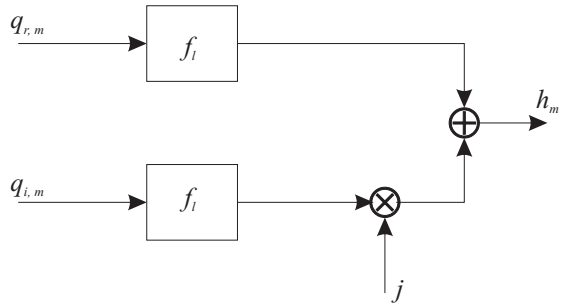


Figure 4.2 Tap gain generation. $q_{r,m}$ and $q_{i,m}$ are independent real-valued white Gaussian noise sequences with variance $E_h/2$, and the filter f_l shapes the Doppler spectrum.

q_m with variance $E_h(\tau_i)$ is passed through a filter with impulse response f_l (normally real-valued), i.e.,

$$h_m(\tau_i) = \sum_{l=-\infty}^{\infty} q_m f_{m-l} \tag{4.20}$$

Equivalently, the real and imaginary part can be generated and filtered independently and added in quadrature, in which case the variance of each of the real-valued Gaussian random variables $q_{r,m}$ and $q_{i,m}$ is $E_h(\tau_i)/2$. This is illustrated in Fig. 4.2.

The frequency response $F(\nu)$ of the filter f_l is given by the discrete time Fourier transform

$$F(\nu) = T_m \sum_{l=-\infty}^{\infty} f_l e^{-j2\pi\nu l T_m}, \quad |\nu| < \frac{1}{2T_m} \tag{4.21}$$

where T_m is the sampling period used in the tap generation. In order to ensure that the variance of $h_m(\tau_i)$ is equal to the variance of q_m, i.e. $E_h(\tau_i)$, the filter must have unit energy gain:

$$\sum_{l=\infty}^{\infty} f_l^2 = 1 \tag{4.22}$$

The Doppler spectrum of a tap gain generated using this method is

$$S_h(\nu, \tau_i) = \frac{E_h(\tau_i)}{T_m} |F(\nu)|^2 \tag{4.23}$$

When the q_m are complex Gaussian distributed, the tap gain $h(t, \tau_i)$ (at a given time t) will also have a complex Gaussian distribution. It can then be shown that the amplitude $g(t, \tau_i)$ is Rayleigh distributed and the phase $\theta(t, \tau_i)$ is uniformly distributed on $[0, 2\pi)$.

The sampling period used in a simulation setup is usually equal to $T_r = T_s/\kappa$, where κ is a small integer and T_s is the symbol period. It will normally be the

case that the Doppler spread is much smaller than $1/T_r$. In this case, the simulation efficiency may be increased by generating the tap gains with a sampling period T_m many times larger than T_r, and using interpolation to increase the sampling rate of the tap gains from $1/T_m$ to $1/T_r$.

4.3 Different Doppler spectra

In this section we present different Doppler spectra and autocorrelation functions used to model randomly time-varying channels. When we consider only a single tap, we drop the indexing on τ_i and write $R_h(u)$ and $S_h(\nu)$. We normalize the expressions such that the energy $E_h = R_h(0) = 1$.

4.3.1 HF channels: The Watterson model

In an ionospheric HF communication system, the transmitter and receiver are not moving (or moving slowly relative to the wavelength), while the radio waves are being reflected by a large number of randomly moving ions. This suggests that the Doppler shift has a Gaussian distribution (McNicol, 1949), as was verified experimentally by Watterson, Juroshek and Bensema (1970) and earlier studied by Shaver, Tupper and Lomax (1967). The model is referred to as the Watterson model.

A Gaussian Doppler spectrum can be written

$$S_h(\nu) = \frac{1}{\sqrt{2\pi\sigma_\nu^2}}\, e^{\frac{-(\nu-\bar{\nu})^2}{2\sigma_\nu^2}} \tag{4.24}$$

where $\bar{\nu}$ is the mean value and σ_ν^2 is the variance of the Doppler shift. For a Gaussian fading spectrum, the Doppler spread ν_d is by convention defined as twice the standard deviation of the Doppler shift, i.e.,

$$\nu_d = 2\sigma_\nu \tag{4.25}$$

Using this definition, and assuming that $\bar{\nu}$ is zero, (4.24) becomes

$$S_h(\nu) = \sqrt{\frac{2}{\pi\nu_d^2}}\, e^{\frac{-2\nu^2}{\nu_d^2}} \tag{4.26}$$

The Fourier transform of a Gaussian function is also a Gaussian function, and the autocorrelation becomes

$$R_h(u) = \sqrt{\frac{2}{\pi\nu_d^2}} \int_{-\infty}^{\infty} e^{\frac{-2\nu^2}{\nu_d^2}} e^{j2\pi\nu u} d\nu = e^{\frac{-(\pi\nu_d u)^2}{2}} \tag{4.27}$$

4.3.2 Mobile channels: Clarke's spectrum

In a mobile radio system, the mobile station is moving with a radial velocity v compared to the base station, while all the reflectors and the base station are assumed not to be moving. If the reflectors are assumed to be uniformly distributed in azimuth seen from the mobile station, the Doppler spectrum will have the classical bowl shape

$$S_h(\nu) = \begin{cases} \frac{1}{\pi\nu_d\sqrt{1-\left(\frac{\nu}{\nu_d}\right)^2}} & , \ |\nu| < \nu_d \\ 0 & , \ |\nu| \geq \nu_d \end{cases} \tag{4.28}$$

where the Doppler spread is now defined as the maximal Doppler shift $\nu_d = v/\lambda$, where λ is the wavelength. This popular model was proposed by Clarke (1968), and a simulator using the model was developed by Jakes (1974). Therefore, it is referred to as Clarke's or Jakes' channel model.

The autocorrelation function corresponding to Clarke's spectrum is

$$R_h(u) = J_0(2\pi\nu_d u) \tag{4.29}$$

where $J_0(\,\cdot\,)$ is the zeroth order Bessel function of the first kind.

4.3.3 Auto-regressive model

A mathematically simple channel model for a discrete time simulation setup is to model the tap gain variation as an auto-regressive (AR) process. In this case, the tap gains are generated as

$$h_m = -\left(\sum_{l=1}^{p} a_l h_{m-l}\right) + \sigma_\varepsilon q_m \tag{4.30}$$

where p is the order of the model, a_l are the AR coefficients, σ_ε^2 is called the prediction error variance (for reasons not detailed here), and q_m is Gaussian distributed with unit variance. An AR process can be real or complex, depending on whether the q_m are real or complex. An AR process of order p is written as AR(p).

The simplest AR model has order one. In this case, we define $\rho = -a_1$ and $\sigma_\varepsilon^2 = \sqrt{1-\rho^2}$, such that

$$h_m = \rho h_{m-1} + \sqrt{1-\rho^2} q_m \tag{4.31}$$

The autocorrelation of an AR(1) process can easily be found as

$$R_h(lT_m) = E(h_m h_{m-l}^*) = \rho^{|l|} \tag{4.32}$$

Channel	$S_h(\nu)$	$R_h(u)$	Definition of ν_d
Watterson	$\sqrt{\frac{2}{\pi\nu_d^2}}\, e^{\frac{-2\nu^2}{\nu_d^2}}$	$e^{\frac{-(\pi\nu_d u)^2}{2}}$	2σ value of Doppler spectrum
Clarke	$\frac{1}{\pi\nu_d\sqrt{1-\left(\frac{\nu}{\nu_d}\right)^2}},\ \lvert\nu\rvert < \nu_d$	$J_0(2\pi\nu_d u)$	Maximum Doppler shift
AR(1)	$T_m \frac{1-\rho^2}{1+\rho^2-2\rho\cos(2\pi\nu T_m)}$	$\rho^{\frac{\lvert u\rvert}{T_m}}$	$R_h\left(\frac{1}{\nu_d}\right) = e^{-1}$

Table 4.1 Characteristics of different models used for fading channels.

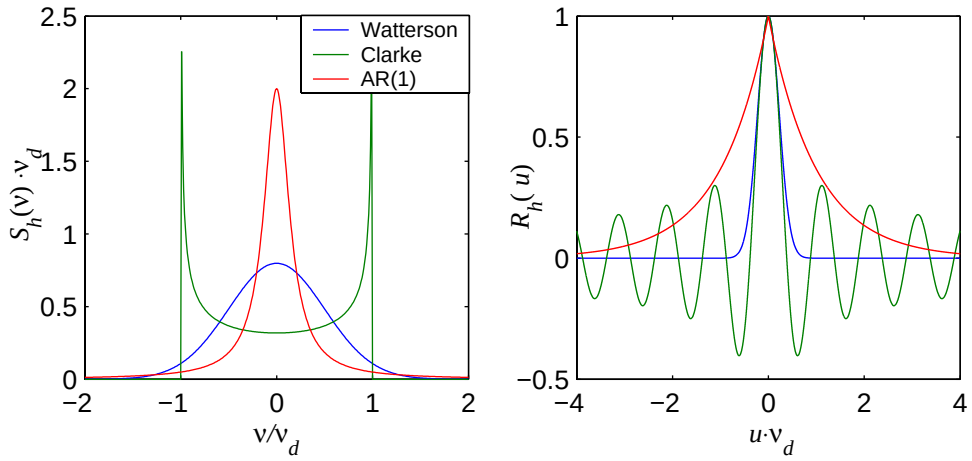


Figure 4.3 The Doppler spectrum $S_h(\nu)$ and the autocorrelation function $R_h(u)$ for different channel models.

and the Doppler spectrum is

$$S_h(\nu) = T_m \sum_{l=-\infty}^{\infty} \rho^{|l|} e^{-j2\pi\nu l T_m} = T_m \frac{1-\rho^2}{1+\rho^2-2\rho\cos(2\pi\nu T_m)} \tag{4.33}$$

In this work we (arbitrarily) define the Doppler spread of an AR(1) process as

$$\nu_d = -\frac{\ln\rho}{T_m} \tag{4.34}$$

or equivalently $\rho = e^{-\nu_d T_m}$. The motivation for this definition is that we then have

$$R_h\left(\frac{1}{\nu_d}\right) = e^{-1} \tag{4.35}$$

4.3.4 Comparing the different Doppler spectra

The properties of the three channel models described above are summarized in Table 4.1. Fig. 4.3 shows the normalized Doppler spectra and autocorrelation functions. Note that the three channel models have quite different statistical properties.

4.4 Standardized test channels for HF communications

The Radio communications group of the International Telecommunications Union (ITU-R, previously known as CCIR) has recommended standard test channels for simulating the performance of HF systems. The test channels are based on the Watterson model.

In the recommendation ITU-R F.520 (ITU, 1992) were defined three test channels, called the ITU-R good, the ITU-R moderate, and the ITU-R poor channel. These channels are still widely used to report the performance of HF modems, but the recommendation is now obsolete and replaced by ITU-R F.1487 (ITU, 2000).

ITU-R F.1487 defines a total of 10 test channels (some of them identical), describing different latitude regions and levels of ionospheric disturbance. Each of the three test channels of F.520 is identical to at least one of the test channels of F.1487.

All the test channels in the ITU-R recommendations are defined as a tapped delay line with only two taps. The delay difference between the two taps is the delay spread τ_m. The taps are fading independently with a Rayleigh (i.e. complex Gaussian) probability density function and a Gaussian fading spectrum, see Sec. 4.3.1. The 2σ Doppler spread ν_d is the same for both taps. Table 4.2 shows the Doppler and delay spread of the different test channels.

A point to note from the original paper of Watterson et al. (1970) as well as from the recommendation ITU-R F.1487, is that the Watterson model is not valid for bandwidths in excess of about 12 kHz. Therefore, other models must be used when investigating wideband HF communications (Mastrangelo, Lemmon, Vogler, Hoffmeyer, Pratt and Behm, 1997). In this dissertation, we concentrate entirely on waveforms designed for a bandwidth of 3 kHz, and we can therefore apply the Watterson model.

4.4.1 Generating the Gaussian spectrum

The tap gains are generated by filtering complex white Gaussian noise, as described in Sec. 4.2.3. By using equations (4.21), (4.23) and (4.26), we find that the sampling period T_m used in the tap gain generation should be much smaller than $1/\nu_d$, and that

Recommendation	Channel	τ_m	ν_d
F.520	ITU-R good	0.5 ms	0.1 Hz
	ITU-R moderate	1 ms	0.5 Hz
	ITU-R poor	2 ms	1 Hz
F.1487	Low lat. quiet cond.	0.5 ms	0.5 Hz
	Low lat. moderate cond.	2 ms	1.5 Hz
	Low lat. disturbed cond.	6 ms	10 Hz
	Mid-lat. quiet cond.	0.5 ms	0.1 Hz
	Mid-lat. moderate cond.	1 ms	0.5 Hz
	Mid-lat. disturbed cond.	2 ms	1 Hz
	Mid-lat. disturbed cond. NVIS	7 ms	1 Hz
	High lat. quiet cond.	1 ms	0.5 Hz
	High lat. moderate cond.	3 ms	10 Hz
	High lat. disturbed cond.	7 ms	30 Hz

Table 4.2 Doppler spread ν_d and delay spread τ_m of the test channels recommended by the ITU-R.

the filter f_l ideally should be[1]

$$f_l = \sqrt[4]{2\pi\nu_d^2 T_m^2}\, e^{-(\pi\nu_d l T_m)^2}, \quad l \in \{-\infty, \ldots, -1, 0, 1, \ldots, \infty\} \tag{4.36}$$

As this filter has infinite length, it can not be perfectly realized in practice.

Two different approaches can be used to implement a filter with an impulse response approximating the one described by (4.36). One approach is to use an infinite impulse response (IIR) filter of low order, e.g. a second order Butterworth filter. The other approach is to use a long finite impulse response (FIR) filter, i.e., truncating or windowing the impulse response of (4.36) to finite length.

Generally, using an IIR filter allows for a smaller filter order (less complexity), while a FIR filter is needed in order to closely match the ideal Gaussian spectrum. Fig. 4.4 compares the magnitude response $|F(\nu)|$ of three different filters: A Gaussian FIR filter truncated to length 43, a fourth order IIR filter designed to have an "almost Gaussian" magnitude response (Willink, Davies, Clarke and Jorgenson, 1996), and a second order IIR filter with a Butterworth response. Note that the IIR filters have more unwanted high frequency components than the FIR filter.

There has been a problem that the fading spectra of different channel simulators are slightly different, such that the modem performances reported by different

[1] The normalization factor is computed from (4.22), $\sum_{l=-\infty}^{\infty} f_l^2 = 1$, using the approximation $\sum_{l=-\infty}^{\infty} e^{-al^2} \approx \int_{-\infty}^{\infty} e^{-al^2} dl = \sqrt{\pi/a}$. This approximation is accurate for values of $a = 2(\pi\nu_d T_m)^2$ positive but close to zero, i.e., $\nu_d T_m << 1$.

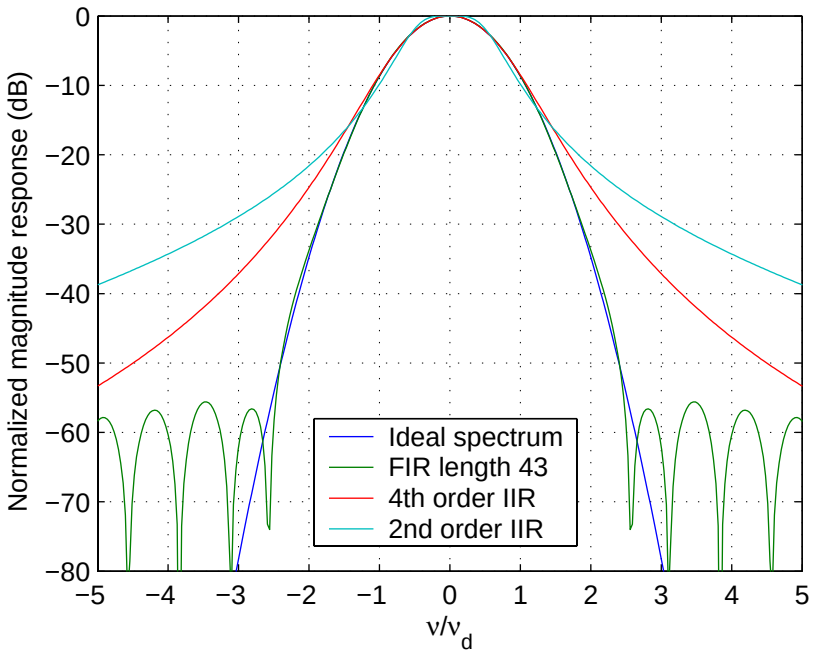


Figure 4.4 Magnitude response of different filters used to generate Gaussian fading spectra. In this example, $1/T_m = 32\nu_d$.

vendors can not be compared directly. To address this problem, Furman and Nieto (2001) have proposed a strict definition on how to generate the tap gains, for possible inclusion in future standards. Among other things, they say that the fading spectrum should be generated using a FIR filter, operating at a sampling rate $1/T_m \geq 32\nu_d$. All filter taps f_l smaller than 1% of the main tap f_0 can be truncated using a rectangular window (for $1/T_m = 32\nu_d$ this means that at least 43 taps must be retained). When the sampling period needs to be decreased from T_m to T_r, this should be done using linear interpolation.

When we report simulation results for a Gaussian fading spectrum in this dissertation, the tap gains have been generated as described in (Furman and Nieto, 2001).

4.5 The equivalent channel impulse response

The signal generated by the baseband modulator, described in Chap. 2, is passed through a radio transmitter for filtering, frequency up-conversion, and power amplification before being transmitted on-air. Similarly, the received signal at the other end of the link is passed through a radio receiver for filtering and frequency down-

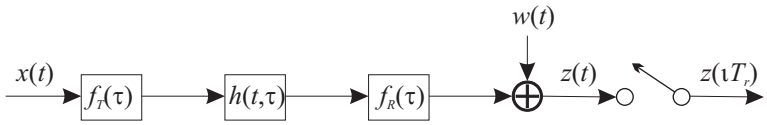


Figure 4.5 Baseband system model including transmitter and receiver filters.

conversion before being delivered to the baseband demodulator, to be described in subsequent chapters.

Assuming that the cascade of all components in the radio transmitter and receiver chain is a linear function, and that the frequency conversions in the transmitter and the receiver cancel exactly, the system can be modelled as shown in Fig. 4.5. The transmitter is represented by a linear filter $f_T(\tau)$, the channel is represented by its time-varying baseband equivalent impulse response $h(t,\tau)$, and the receiver is represented by a linear filter $f_R(\tau)$.

We denote the complex-valued transmitted symbols (before modulation onto subcarrier, i.e., we consider subcarrier modulation and demodulation as part of the radio) as x_n, and the signal input to the radio as the pulse train

$$x(t) = \sum_{n=-\infty}^{\infty} x_n \delta(t - nT_s) \tag{4.37}$$

where T_s is the symbol period.

The radio filters $f_T(\tau)$ and $f_R(\tau)$ are combined with the channel impulse response $h(t,\tau)$, giving the time-varying *equivalent channel impulse response* $h_E(t,\tau)$ which is the convolution

$$\begin{aligned} h_E(t,\tau) &= f_T(\tau) * h(t,\tau) * f_R(\tau) \\ &= f_E(\tau) * h(t,\tau) = \int_{-\infty}^{\infty} h(t,u) f_E(\tau - u) du \end{aligned} \tag{4.38}$$

where $f_E(\tau)$ is the convolution of $f_T(\tau)$ and $f_R(\tau)$.

Note that when radio filters are included, the equivalent channel impulse response is not WSSUS, because the US condition is violated: The radio filters introduce correlation between the value of $h_E(t,\tau)$ for different values of τ.

The baseband equivalent received signal $z(t)$ is given by

$$z(t) = \int_{-\infty}^{\infty} h_E(t,\tau) x(t-\tau) d\tau + w(t) \tag{4.39}$$

The noise $w(t)$ is not necessarily white, because of the presence of $f_R(\tau)$. Inserting

(4.37)-(4.38) into (4.39) gives

$$z(t) = \int_{-\infty}^{\infty} h(t,\tau) \sum_{j=-\infty}^{\infty} x_j f_E(t - jT_s - \tau) d\tau + w(t) \tag{4.40}$$

If the tapped delay line channel model of (4.13) is assumed, (4.40) can be written

$$z(t) = \sum_{i=0}^{K-1} h(t,\tau_i) \sum_{j=-\infty}^{\infty} x_j f_E(t - jT_s - \tau_i) + w(t) \tag{4.41}$$

In this dissertation, we ignore the radio filters in most cases, i.e. $h_E(t,\tau) = h(t,\tau)$, $f_E(\tau) = \delta(\tau)$, and $w(t)$ being white noise.

4.5.1 Discrete-time channel model

The received complex baseband signal is sampled with a sampling period equal to $T_r = T_s/\kappa$, where κ is a small integer called the oversampling rate:

$$z_\iota = z(\iota T_r) \tag{4.42}$$

If the sampling rate $1/T_r$ is larger than the bandwidth B of $z(t)$ (defined such that the frequency range of $z(t)$ is $\langle -B/2, B/2 \rangle$), it is known from the sampling theorem that $z(t)$ can be perfectly reconstructed from the z_ι. The z_ι are then called sufficient statistics, because they include all the information contained in the received signal. In a realistic scenario, B is always somewhat larger than $1/T_s$, such that κ should be 2 or larger unless the simplifying assumptions mentioned in Sec. 4.5.1.1 below can be applied.

Combining (4.42) and (4.40) gives

$$z_\iota = \int_{-\infty}^{\infty} h(\iota T_r,\tau) \sum_{j=-\infty}^{\infty} x_j f_E(\iota T_r - jT_s - \tau) d\tau + w(\iota T_r) \tag{4.43}$$

4.5.1.1 Symbol-spaced channel model

First, assume that no oversampling is used ($\kappa = 1$), such that $T_r = T_s$ and $\iota = n$. Making the substitution $l = n - j$, (4.43) then becomes

$$z_n = \sum_{l=-\infty}^{\infty} x_{n-l} \int_{-\infty}^{\infty} h(nT_s,\tau) f_E(lT_s - \tau) d\tau + w(nT_s) \tag{4.44}$$

We introduce the symbol-spaced discrete-time channel impulse response as

$$h_{n,l} = \int_{-\infty}^{\infty} h(nT_s,\tau) f_E(lT_s - \tau) d\tau \tag{4.45}$$

We further assume that $h_{n,l} = 0$ for $l < 0$ and $l \geq M$, where M is the length of the equivalent CIR measured in symbol intervals, and write the discrete-time CIR in vector form as

$$\mathbf{h}_n = [h_{n,0} \ \cdots \ h_{n,M-1}]^T \tag{4.46}$$

(4.44) then simplifies to

$$z_n = \sum_{l=0}^{M-1} h_{n,l} x_{n-l} + w_n = \mathbf{h}_n^T \mathbf{x}_n + w_n \tag{4.47}$$

where $w_n = w(nT_s)$ is time-discrete complex Gaussian noise with variance $\sigma_w^2 = N_0$, and

$$\mathbf{x}_n = [x_n \ x_{n-1} \ \cdots \ x_{n-M+1}]^T \tag{4.48}$$

is a vector of transmitted symbols.

We have used the symbol-spaced channel model described by (4.47) in most the derivations and simulations presented in this dissertation, unless otherwise noted. Note that in order to ensure that the symbol-spaced samples z_n are sufficient statistics, the filter $f_R(\tau)$ must be a "channel matched filter" matched to the convolution of $f_T(\tau)$ and $h(t, \tau)$, and we must have perfect symbol synchronization between the transmitter and the receiver such that $z(t)$ is sampled at the optimal points in time. In reality we cannot implement $f_R(\tau)$ as a channel matched filter, because the CIR may be unknown and time-varying. We have still chosen to use the symbol-spaced model in this dissertation to keep things simple, but we will also show how some of the proposed algorithms can be extended to the more realistic fractionally spaced channel model described below.

When we present simulation results for a symbol-spaced channel model in this dissertation, we have ignored the radio filters and used a tapped delay line channel model with all τ_i being multiples of T_s. Then, (4.45) simplifies to

$$h_{n,l} = h(nT_s, lT_s), \quad l \in \{0, 1, \ldots, M-1\} \tag{4.49}$$

4.5.1.2 Fractionally spaced channel model

Now, assume that the received signal $z(t)$ is sampled several times per symbol period ($\kappa \geq 2$). We then have κ values of z_ι per symbol period n, which we write

$$z_{n,i} = z_{n\kappa+i}, \quad i \in \{0, 1, \ldots, \kappa-1\} \tag{4.50}$$

i.e. $\iota = n\kappa + i$. Substituting $l = n - j$, (4.43) then becomes

$$z_{n,i} = \sum_{l=-\infty}^{\infty} x_{n-l} \int_{-\infty}^{\infty} h([n\kappa + i]T_r, \tau) f_E([l\kappa + i]T_r - \tau) d\tau + w_{n,i} \tag{4.51}$$

where the indices of $w_{n,i}$ is defined similarly as for $z_{n,i}$. All $w_{n,i}$ are complex Gaussian random variables with variance $\sigma_w^2 = N_0$.

When using fractionally spaced sampling, we find that the discrete-time channel can be represented as κ symbol-spaced time-varying impulse responses, one for each i:

$$h_{n,l,i} = \int_{-\infty}^{\infty} h(nT_s + iT_r, \tau) f_E(lT_s + iT_r - \tau) d\tau, \quad i \in \{0, 1, \ldots, \kappa - 1\} \tag{4.52}$$

or in vector form, assuming $h_{n,l,i} = 0$ for $l < 0$ and $l \geq M$:

$$\mathbf{h}_{n,i} = [h_{n,0,i} \ \cdots \ h_{n,M-1,i}]^T \tag{4.53}$$

Then, (4.51) becomes

$$z_{n,i} = \sum_{l=0}^{M-1} h_{n,l,i} x_{n-l} + w_{n,i} = \mathbf{h}_{n,i}^T \mathbf{x}_n + w_{n,i} \tag{4.54}$$

This can be interpreted as follows: For each transmitted symbol x_n, κ observations $z_{n,i}$ are made of the received signal. Each of these observations is generated by passing the transmitted symbols through one of κ different time-varying filters $\mathbf{h}_{n,i}$.

When using a fractionally spaced receiver ($\kappa \geq 2$), the received samples $z_{n,i}$ will be sufficient statistics, even if $f_R(\tau)$ is not a channel matched filter and/or the symbol synchronization in the receiver is not perfect.

4.6 Chapter summary

In this chapter we have given a short overview of general channel models, and on the specific channel models used for HF communications. We have shown how the Watterson model is implemented in simulation setups, and discussed the underlying assumptions when the channel model and received signal are sampled at the symbol rate or a few times the symbol rate.

Now that we have presented the standardized waveforms in Chap. 2, and the channel models in this chapter, we are ready to discuss the design of receivers for HF waveforms transmitted over HF channels. This will be the topic of the remainder of the dissertation.

URN:NBN:no-3359

Chapter 5

Principles of equalization and turbo equalization

In this chapter, we first present the system model used in the remainder of the dissertation, and give an overview of conventional receiver technology. We then survey the field of turbo equalization, beyond the short introduction given in Sec. 1.2. The chapter is based on vast amounts of literature on adaptive equalization and turbo equalization, which is referenced throughout, and on several fruitful discussions with Michael Tüchler.

5.1 System model

In this dissertation we use turbo equalization as a means to improve receiver performance in communication systems originally not designed with turbo equalization in mind, in particular the standardized HF waveforms presented in Chap. 2.

We use the system model shown in Fig. 5.1 as a reference. The transmitter part encompasses the most important blocks of a transmitter for standardized HF waveforms, see Fig. 2.3 on p. 18.

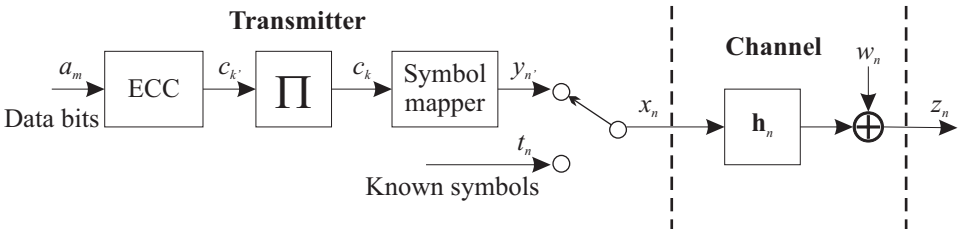


Figure 5.1 Reference system model: Transmitter and channel.

Data bits $a_m \in \{+1, -1\}$ are encoded with an error-correcting code (ECC) with code rate $R_c < 1$. The code bits $c_{k'} \in \{+1, -1\}$ are passed through a block interleaver Π of length N_{int}, giving the interleaved code bits c_k. We consider only one interleaver block at a time, such that $k' \in \{0, \ldots, N_{int} - 1\}$, $k \in \{0, \ldots, N_{int} - 1\}$, and $k = \Pi(k')$. Data symbols $y_{n'}$, $n' \in \{0, \ldots, N_{int}/Q - 1\}$, are generated by mapping Q consecutive interleaved code bits $c_{n'Q+j}$, $j \in \{0, \ldots, Q - 1\}$, onto the 2^Q-ary signal constellation $\mathcal{S} = \{s_0, \ldots, s_{2^Q-1}\}$.

Training and synchronization symbols t_n, which are known to the receiver, are multiplexed with the data symbols $y_{n'}$ to form the transmitted symbols x_n, $n \in \{0, \ldots, N_{block} - 1\}$. N_{block} is the total number of transmitted symbols (training + data) corresponding to one interleaver block. There exists a one-to-one mapping $T(n') = n$ from n' to n which depends on the training pattern, i.e., each $y_{n'}$ is transmitted at a particular symbol interval n. The mapping T is defined for all n', but the inverse mapping $T^{-1}(n) = n'$ is not defined for all n (the values of n corresponding to training symbols are not mapped from/to any n'). The frame pattern efficiency is $R_f = N_{int}/(QN_{block})$, such that the overall data rate of the system is

$$f_a = R_c Q R_f f_s = \frac{R_c N_{int}}{N_{block}} f_s \tag{5.1}$$

The x_n are passed through a discrete-time equivalent channel $\mathbf{h}_n$, which may be time-varying. We use the symbol-spaced channel model of Sec. 4.5.1.1 unless otherwise noted. When the CIR is assumed to be constant, we write $\mathbf{h}_n = \mathbf{h} = [h_0 \ \cdots \ h_{M-1}]^T$. Finally, complex circularly symmetric white Gaussian noise w_n is added. The noise variance $E\{w_n w_n^*\}$ is denoted σ_w^2. In some cases we allow the noise variance to be varying with the time index n, denoted $\sigma_{w,n}^2$.

5.2 Conventional receivers: Separate equalization and decoding

By a conventional receiver we mean a receiver not using turbo equalization, e.g. the receiver in present-day HF modems. Fig. 5.2 shows the block diagram of a conventional receiver for the system model presented in Sec. 5.1. The received signal z_n is passed through an equalizer which reduces the ISI imposed by the channel. When the CIR is unknown and time-varying, the equalizer must be able to adapt to changing channel conditions. The equalizer provides noisy estimates $\tilde{x}_n \in \mathbb{C}$ of the transmitted symbols $x_n \in \mathcal{S}$, which are passed through a symbol demapper providing estimates of the code bits c_k. Those estimates are deinterleaved and input to a decoder for the ECC, giving estimates (decisions) $\hat{a}_m$ of the transmitted data bits.

The estimates of the c_k passed from the demapper to the decoder can either be hard decisions or soft information on the code bits. A hard-decided code bit $\hat{c}_k$ is

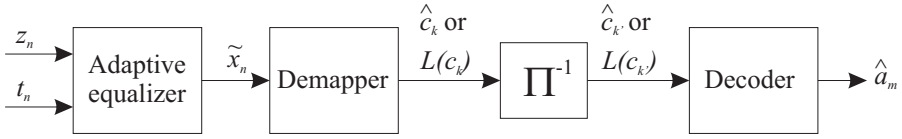


Figure 5.2 Block diagram of a conventional receiver.

either $+1$ or -1, and can be obtained by first making a hard decision $\hat{x}_n$ on the transmitted symbol x_n (i.e., $\hat{x}_n$ equals the $s_i \in \mathcal{S}$ minimizing the distance $|\tilde{x}_n - s_i|$), before performing the inverse operation of the symbol mapper. Soft information, on the other hand, is some measure on the reliability of the decision $\hat{c}_k$. When performing soft-output demapping, the distances $|\tilde{x}_n - s_i|$ for all $s_i \in \mathcal{S}$ are used in the calculation. A popular measure for soft information is the LLR (log likelihood ratio) $L(c_k) = \ln[\Pr(c_k = +1)/\Pr(c_k = -1)]$, to be discussed in Sec. 5.3.1. The relation between hard decisions and soft information is $\hat{c}_k = \text{sgn}[L(c_k)]$, and it is easy to see that information is lost when making a hard decision.

5.2.1 Equalizer types for conventional receivers

Different types of equalizers can be used in a conventional receiver, of which the trellis-based MLSE (maximum likelihood sequence estimator), the DFE (decision feedback equalizer) and the LE (linear equalizer) are most common (Proakis, 1995). In this section we give a survey of those equalizers, because most SISO equalizers used for turbo equalization are extensions of conventional equalizers, with the SISO capability added.

5.2.1.1 Trellis-based equalizers: MLSE and MAP

The equalizers with the best error performance are based on traversing a trellis in order to calculate the most probable sequence of transmitted symbols. The MLSE of Forney (1972), based on the Viterbi algorithm, finds the most likely sequence of transmitted symbols given all received symbols z_n and a known CIR $\mathbf{h}_n$. This algorithm traverses the trellis in one direction only, and minimizes the sequence error probability. In order to rather minimize the error probability of each transmitted symbol, the MAP algorithm of Bahl et al. (1974) should be used. This algorithm traverses the trellis in both directions, and is thus more complicated than the MLSE. In practice, the performance of MLSE and MAP equalization is quite similar for conventional receivers, and therefore the simpler MLSE is more frequenctly used.

The computational complexity and memory requirements of trellis-based equalizers are proportional to 2^{QM}, where Q is the number of bits per symbol and M is

the length of the CIR measured in symbol intervals. They are therefore not used in HF modems, where we typically have $Q = 3$ and M can be as large as 10 or more.

For details on trellis-based equalizers for conventional receivers, we refer to (Forney, 1972; Proakis, 1995).

5.2.1.2 Filter-based equalizers: LE and DFE

When trellis-based equalizers can not be used due to their complexity, equalizers based on filtering is a natural choice. There are two types, the LE which only passes the received symbols through a single transversal filter, and the DFE which also uses previous decisions of the equalizer. The DFE is a nonlinear structure, and can therefore perform closer to the optimal nonlinear trellis-based equalizers than the LE.

Mathematically, the symbol estimates output from an LE are given by

$$\tilde{x}_n = \mathbf{f}_n^H \mathbf{z}_n \tag{5.2}$$

where $\mathbf{f}_n = [f_{n,N_1}\ f_{n,N_1-1} \cdots f_{n,-N_2}]^T$ are the complex coefficients of a linear filter of length $N = N_1 + N_2 + 1$ and $\mathbf{z}_n = [z_{n+N_1} \cdots z_{n-N_2}]^T$ is a vector of received symbols before and after the symbol to be estimated. The filter $\mathbf{f}_n$ is allowed to be time-varying in the case of a varying channel, thus the indexing on n.

Similarly, the symbol estimates output from a DFE are given by

$$\tilde{x}_n = \mathbf{f}_{ff,n}^H \mathbf{z}_{ff,n} + \mathbf{f}_{fb,n}^H \hat{\mathbf{x}}_n = \mathbf{f}_n^H \begin{bmatrix} \mathbf{z}_{ff,n} \\ \hat{\mathbf{x}}_n \end{bmatrix} \tag{5.3}$$

where $\mathbf{f}_{ff,n} = [f_{n,N_1} \cdots f_{n,0}]^T$ is the feedforward filter, $\mathbf{f}_{fb,n} = [f_{n,-1} \cdots f_{n,-N_2}]^T$ is the feedback filter, $\mathbf{z}_{ff,n} = [z_{n+N_1} \cdots z_n]^T$, and $\hat{\mathbf{x}}_n = [\hat{x}_{n-1} \cdots \hat{x}_{n-N_2}]^T$ is a vector of N_2 previous hard-decided symbols output from the equalizer.

The filter coefficients $\mathbf{f}_n$ are found using some optimization criterion, of which the minimum mean square error (MMSE) criterion is most common. In this case, the mean square error (MSE) $E\{|\tilde{x}_n - x_n|^2\}$ is minimized. When the CIR $\mathbf{h}$ and noise variance σ_w^2 are known and constant, and the x_n are assumed to be i.i.d. random variables with zero mean and unit variance, the MMSE-optimal filter coefficients $\mathbf{f}_n$ for an LE are given by

$$\mathbf{f}_n = (\mathbf{H}\mathbf{H}^H + \sigma_w^2 \mathbf{I}_N)^{-1} \mathbf{s} \tag{5.4}$$

where $\mathbf{H}$ is the $N \times (N + M - 1)$ channel convolution matrix

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{M-1} & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{M-1} & 0 & \vdots \\ \vdots & & & \ddots & & & 0 \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{M-1} \end{bmatrix} \tag{5.5}$$

and $\mathbf{s}$ is the (N_1+1)th column of $\mathbf{H}$.

For a DFE, the MMSE-optimal filter coefficients are normally calculated assuming that the past symbol decisions $\hat{\mathbf{x}}_n$ are correct, $\hat{x}_i = x_i$ for $i < n$. Under the same assumptions as used for the LE above, the MMSE-optimal coefficients of the feedforward filter $\mathbf{f}_{ff,n}$ are then

$$\mathbf{f}_{ff,n} = (\mathbf{H}_{ff}\mathbf{V}_{ff}\mathbf{H}_{ff}^H + \sigma_w^2\mathbf{I}_{N_1+1})^{-1}\mathbf{s}_{ff} \tag{5.6}$$

where $\mathbf{H}_{ff}$ is a $(N_1+1)\times(N_1+M)$ channel convolution matrix of the form (5.5), $\mathbf{s}_{ff}$ is the (N_1+1)th column of $\mathbf{H}_{ff}$, and $\mathbf{V}_{ff}$ is a $(N_1+M)\times(N_1+M)$ diagonal matrix where the first N_1+1 diagonal elements are 1 and the remaining $M-1$ diagonal elements are 0. The corresponding coefficients of the feedback filter are

$$\mathbf{f}_{fb,n} = -\mathbf{H}_{fb}^H\mathbf{f}_{ff,n} \tag{5.7}$$

where $\mathbf{H}_{fb}$ is the $(N_1+1)\times N_2$ matrix

$$\mathbf{H}_{fb} = \begin{bmatrix} \vdots & 0 & & 0 \\ 0 & \vdots & \cdots & \\ h_{M-1} & 0 & & \vdots \\ \vdots & h_{M-1} & & \\ h_2 & \vdots & \ddots & \\ h_1 & h_2 & \cdots\ h_{M-1}\ 0 & \cdots\ 0 \end{bmatrix} \tag{5.8}$$

The $M-1$ leftmost columns of $\mathbf{H}_{fb}$ are identical to the $M-1$ rightmost columns of $\mathbf{H}_{ff}$. From (5.7) and (5.8) we see that only the first $M-1$ elements of $f_{fb,n}$ are nonzero, and thus the length N_2 of the feedback filter in a DFE does not need to be larger than $M-1$.

In conventional receivers, the DFE is considered to be a better choice than the LE, since they have comparable complexity and the DFE performs better. This is especially the case if the frequency response of the channel (the Fourier transform of the CIR) exhibits spectral nulls. The DFE is used in conventional HF modems, where typical filter lengths are $N_1 = 15$ and $N_2 = 8$.

The main problem with the DFE is the assumption that past decisions are correct. In reality, some of the hard-decided symbols $\hat{x}_n$ from the equalizer are wrong. These wrong decisions are fed back to the equalizer, increasing the error probability for the following symbols. This phenomenon is called "error propagation"

The performance of DFE and MLSE equalizers for use in HF modems has been compared in (Falconer, Sheikh, Eleftheriou and Tobis, 1985; Shukla and Turner, 1992; Bartlett, Brunt and Darnell, 1999), who all conclude that MLSE performs somewhat better than DFE, but the difference is not large enough to consider implementing the far more complex MLSE algorithm in a HF modem.

5.2.2 Adaptive equalization in conventional receivers

When the CIR is unknown and time-varying, the equalizer must be adapted to the varying channel conditions. There are two fundamentally different approaches to solve this problem: Separate channel estimation and equalization, or direct adaptation of the filter coefficients. Fig. 5.3 illustrates both approaches.

5.2.2.1 Separate channel estimation and equalization

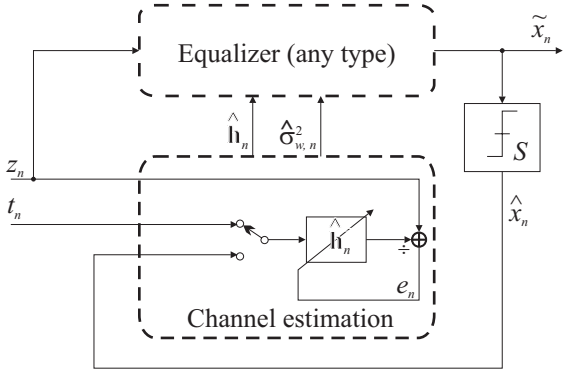
In the first approach, shown in Fig. 5.3(a), a separate channel estimation algorithm provides the equalizer with estimates $\hat{\mathbf{h}}_n$ of the CIR and $\hat{\sigma}^2_{w,n}$ of the noise variance. This approach can be used with trellis-based equalizers as well as with filter-based equalizers. The channel estimation algorithm seeks to minimize the variance of the error signal

$$e_n = z_n - \sum_{l=0}^{M-1} \hat{h}_{n,l} x_{n-l} \tag{5.9}$$

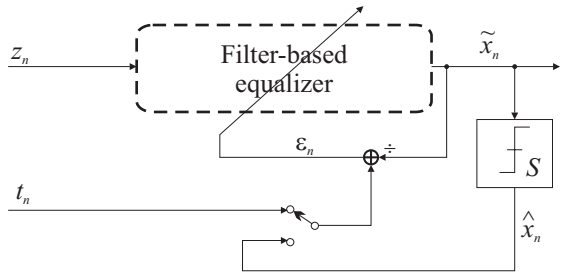
In the nomenclature of adaptive filtering (Haykin, 1996), z_n is here the desired signal, x_n is the input signal, and the estimated CIR $\hat{\mathbf{h}}_n = [\hat{h}_{n,0} \cdots \hat{h}_{n,M-1}]^T$ is the filter to be adapted. The problem with applying (5.9) is that not all the x_n are known to the receiver. When training symbols have been transmitted, x_n can of course be replaced with t_n. Between the training sequences, when data symbols have been transmitted, hard-decided symbols $\hat{x}_n$ generated from the output $\tilde{x}_n$ of the equalizer are often used as an input signal to the channel estimator, in which case error propagation will degrade the performance of the receiver.

If the channel is slowly varying, it may suffice to update the channel estimate only once per some block of received symbols. Then, a "snapshot" channel estimate is calculated from the training symbols t_n only, i.e., one does not need to use the hard-decided symbols $\hat{x}_n$. The least sum of squared errors (LSSE) algorithm (Crozier, Falconer and Mahmoud, 1991) can be used to calculate a channel estimate from a block of received symbols and the corresponding transmitted symbols, which in this case are known training symbols only. The channel estimate can either be kept constant between the training sequences, or interpolation can be used to generate a channel estimate which also varies between the training sequences (Fechtel and Meyr, 1995).

If the channel is varying fast relative to the interval between training sequences, an adaptive algorithm must be used to update the channel estimate every symbol interval. Several algorithms for adaptive filtering may be applied (Haykin, 1996), of which we mention the simple linear mean square (LMS) algorithm (Widrow, McCool, Larimore and Johnson, 1976) and the more complex algorithms of recursive least squares (RLS) (Eleftheriou and Falconer, 1986; Hsu, 1982) and Kalman filtering



(a) Separate channel estimation and equalization.



(b) Direct adaptation of equalizer.

Figure 5.3 Two different approaches to adaptive equalization in conventional receivers.

(Godard, 1974). The RLS algorithm is known to converge faster than the LMS algorithm, and also has better steady-state performance if the input signal is non-white. But in the channel estimation scenario described by eq. (5.9), the input signal is the transmitted symbol sequence, which is almost white. Therefore, the LMS performs almost as well as the RLS algorithm for the purpose of tracking a time-varying channel (Clark and Harun, 1986; Eleftheriou and Falconer, 1986; Bershad, McLaughlin and Cowan, 1990). If one wishes to use LMS rather than RLS, but needs the convergence speed of RLS, an LSSE channel estimate can be calculated from the first training symbols and used to initialize the LMS algorithm.

A time-varying channel estimate can be directly included in the branch metric calculation of trellis-based equalizers, with negligible increase in complexity compared to the case of a constant channel (Magee and Proakis, 1973). For filter-based equalizers, the filter coefficients must be re-calculated every symbol interval using (5.4) or (5.6)-(5.7). This calculation involves a matrix inversion and direct calculation has a computational complexity proportional to N^3 for an LE and N_1^3 for a DFE, which may be prohibitively large when the coefficients are calculated every symbol

interval. Fortunately, the matrix to be inverted exhibits a structure which may be used to decrease the computational complexity to be proportional to N^2 or N_1^2. For a DFE, one may e.g. use the algorithm of (Shukla and Turner, 1991), which is a modified version of the Levinson-Durbin recursion; however this algorithm suffers from numerical instability if the magnitude of the first tap of the estimated CIR, $|\hat{h}_{n,0}|$, is close to zero. Another alternative is to apply algorithms similar to the one described in Sec. 6.1.2, which are numerically more stable.

Receivers using separate channel estimation combined with a linear equalizer have also been proposed for underwater acoustic communications (Eggen, Baggeroer and Preisig, 2000; Eggen, Preisig and Baggeroer, 2001), where the delay and Doppler spreads of the channel may be even more severe than for HF communications.

5.2.2.2 Direct adaptation of equalizer

The second approach, shown in Fig. 5.3(b), is to use an algorithm for adaptive filtering to directly adapt the equalizer coefficients. This approach is often called "adaptive equalization" in the literature, but we use that term in a more general sense also encompassing the approach of separate channel estimation described above.

In the case of direct adaptation, we seek to minimize the variance of the error signal

$$\varepsilon_n = x_n - \tilde{x}_n = \begin{cases} x_n - \mathbf{f}_n^H \mathbf{z}_n, & \text{for an LE} \\ x_n - \mathbf{f}_n^H \begin{bmatrix} \mathbf{z}_{ff,n} \\ \hat{\mathbf{x}}_n \end{bmatrix}, & \text{for a DFE} \end{cases} \tag{5.10}$$

rather than e_n defined in (5.9). The problem formulation has been reversed compared to the case of separate channel estimation: x_n is now the desired signal, z_n (and also past decisions $\hat{x}_{n-i}$ for the case of a DFE) is the input signal, and the equalizer coefficient vector $\mathbf{f}_n$ is the filter to be adapted. An early reference on direct adaptation of a DFE is (George, Bowen and Storey, 1971).

The choice of algorithms for adaptive filtering is the same as in the case of separate channel estimation. Now, the input signal to the adaptive filter is z_n, which in general is not white. Thus the RLS algorithm has better tracking performance than the LMS algorithm when used for direct adaptation. Pseudo-code for effective implementation of the RLS algorithm for adaptive equalization can e.g. be found in (Hsu, 1982).

For filter-based equalizers, direct adaptation is simpler to implement than separate channel estimation because the matrix inversion is avoided. But the ability to track channel variations is worse, for two reasons: Firstly, the number of parameters (coefficients) estimated by the adaptive filter is generally larger (N for direct adaptation as opposed to M for separate channel estimation) because the length of

the equalizer must be a few times larger than the length of the CIR for the equalizer to perform well. Secondly, the input signal to the adaptive filter is not white. The fact that separate channel estimation outperforms direct adaptation in HF communication systems was demonstrated by several researchers in the early 1990's (Shukla and Turner, 1991; Fechtel and Meyr, 1991; McLaughlin, 1991; Farhang-Boroujeny, 1993).

An analogy to direct equalizer adaptation when using trellis-based equalizers, is the concept of per-survivor processing (PSP) (Kubo, Murakami and Fujino, 1994; Raheli, Polydoros and Tzou, 1995). This means that a channel estimate is calculated for every surviving state in the trellis, such that the transmitted symbols leading up to that state can actually be assumed known. Because trellis-based equalizers are too complex for HF communications even without PSP, we do not investigate this matter further here.

5.2.3 Decoders in conventional receivers

The task of the decoder is to exploit the redundancy introduced by the ECC to provide estimates $\hat{a}_m$ of the transmitted data bits. The input to the decoder is either hard-decided code bits $\hat{c}_{k'}$ or soft information $L(c_{k'})$ on the code bits. In the latter case we use the term *soft-input decoding*, but we do not require soft-output capability from the decoder in a conventional receiver.

When the ECC is a convolutional code, the decoding can be done using the trellis-based Viterbi algorithm, similar to MLSE equalization. The number of states in the trellis used in the decoder is 2^{C-1}, where C is the constraint length and $C-1$ is the memory of the convolutional code. For the standardized HF waveforms we have $C = 7$, and thus the code trellis has 64 states. The Viterbi algorithm can be applied with soft information as well as with hard-decided code bits as input.

5.3 Turbo equalization for a known channel

In this section, we give a survey on turbo equalization for the case where the CIR and noise variance are known and constant, as was assumed in the first works on turbo equalization.

Fig. 5.4 shows the block diagram of a receiver employing turbo equalization, as first proposed by (Douillard et al., 1995). The receiver operates by iterating soft information between the equalizer and the decoder through a deinterleaver/interleaver pair. For each iteration, the quality of the soft information will improve, and reliable decisions $\hat{a}_m$ can be made on the transmitted data bits after some iterations.

The motivation for this iterative receiver structure is to approach the performance of the optimal receiver for the system in Fig. 5.1, which would perform equalization

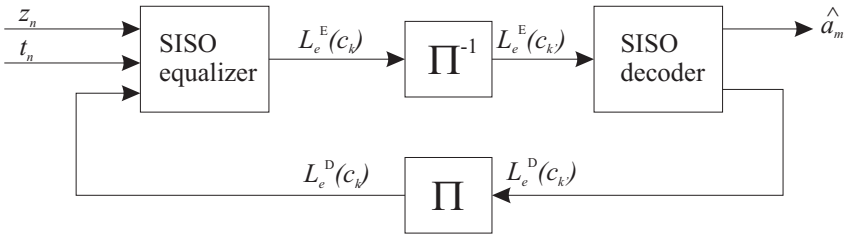


Figure 5.4 Block diagram of a turbo equalization-based receiver for a known channel.

and decoding jointly (Baccarelli, 2000), maximizing $\Pr[\hat{a}_m = a_m]$ given all the received symbols z_n. Joint equalization and decoding is prohibitively complicated, especially in the presence of an interleaver, but can be approximated by letting the equalizer and decoder exchange reliability information in an iterative fashion.

5.3.1 Log likelihood ratios and SISO modules

The soft information iterated between the equalizer and the decoder is usually on the form of log likelihood ratios (LLRs) on the code bits c_k. An LLR is defined as

$$L(c_k) = \ln \frac{\Pr[c_k = +1]}{\Pr[c_k = -1]} \tag{5.11}$$

such that if the receiver is absolutely certain that e.g. $c_k = +1$, we have $L(c_k) = +\infty$. From the LLR we can calculate the probabilities

$$\Pr[c_k = +1] = \frac{1}{2}\left(1 + \tanh\frac{L(c_k)}{2}\right) \tag{5.12}$$

$$\Pr[c_k = -1] = \frac{1}{2}\left(1 - \tanh\frac{L(c_k)}{2}\right) \tag{5.13}$$

The equalizer and the decoder in a turbo equalization-based receiver are SISO (soft-in soft-out) modules, indicating that they can accept soft information (LLRs) as input and provide soft information as output. The equalizer processes the received symbols z_n and *a priori* soft information $L_e^D(c_k)$ on the code bits, fed back from the decoder from the previous iteration, and outputs soft information $L_e^E(c_k)$ on the code bits. Note that soft-output demapping from data symbols to code bits is part of the SISO equalizer.

An output LLR $L_e^E(c_k)$ from a SISO module should not be a function of the *a priori* LLR $L_e^D(c_k)$ for the same code bit in order for the iterative process to converge according to the turbo principle (Hagenauer, 1997). When the SISO equalizer is based on the MAP algorithm (see Sec. 5.3.2), we have the relationship

$$L_e^E(c_k) = L^E(c_k) - L_e^D(c_k) \tag{5.14}$$

where $L^E(c_k)$ is the *a posteriori* LLR calculated by the MAP algorithm, $L_e^D(c_k)$ is the *a priori* input LLR, and $L_e^E(c_k)$ is called the extrinsic LLR. When the equalizer is filter-based, (5.14) does not hold, but we still use the term $L_e^E(c_k)$ to describe an output LLR, *not* necessarily given by (5.14), which is not a function of the *a priori* LLR $L_e^D(c_k)$. Note that in this case $L_e^E(c_k)$ should strictly not be called the extrinsic LLR.

The $L_e^E(c_k)$ are deinterleaved to reverse the interleaving done in the transmitter, providing soft information $L_e^E(c_{k'})$ used as *a priori* information input to the decoder. The decoder computes extrinsic LLRs $L_e^D(c_{k'})$ and also provides estimates $\hat{a}_m$ of the transmitted data bits. The $L_e^D(c_{k'})$ are interleaved again to $L_e^D(c_k)$, used as input to the equalizer in the next iteration.

By convention, the first-time equalization and decoding tasks are called the zeroth iteration, such that a receiver performing three iterations needs to equalize and decode all data four times. In the zeroth iteration, all the *a priori* LLRs $L_e^D(c_k)$ input to the equalizer are zero. If training symbols known to the receiver have been multiplexed into the symbol stream, they are considered as perfect *a priori* information in all iterations including the zeroth (see Sec. 5.3.3.1).

5.3.2 Trellis-based SISO equalizers

The optimal SISO equalizer uses the trellis-based MAP algorithm (Bahl et al., 1974) to calculate *a posteriori* LLRs $L^E(c_k)$ from the the *a posteriori* probabilities (APPs)

$$\Pr[c_k = +1 | z_0, \ldots, z_{N_{block}+M-2}, L_e^D(c_0), \ldots, L_e^D(c_{N_{int}})] \tag{5.15}$$

$$\Pr[c_k = -1 | z_0, \ldots, z_{N_{block}+M-2}, L_e^D(c_0), \ldots, L_e^D(c_{N_{int}})] \tag{5.16}$$

given an entire block of $N_{block} + M - 1$ received symbols and the *a priori* LLRs for the corresponding N_{int} code bits. The MAP algorithm traverses the trellis of the entire block in both directions, called the forward recursion and the backward recursion, before combining the results to form the APPs and output LLRs, from which the extrinsic LLRs are found using (5.14).

When the calculations of the MAP algorithm are performed in the log domain, the corresponding algorithm is referred to as the log-MAP algorithm. The log-MAP algorithm is mathematically equivalent to the MAP algorithm, but has better numerical properties. The first proposals for turbo equalization used the MAP or log-MAP algorithm for equalization (Douillard et al., 1995; Bauch, Khorram and Hagenauer, 1997). Bauch and Franz (1998) also proposed using simpler trellis-based algorithms: The max-log-MAP algorithm, in which some of the computations are approximated, and the soft-output Viterbi algorithm (SOVA) (Hagenauer and Hoeher, 1989), which traverses the trellis in the forward direction only.

The trellis-based equalizers discussed above are too complex to be implemented in HF modems and other communication systems with long impulse responses and/or

high-order signal constellations, for the reasons mentioned in Sec. 5.2.1.1. Therefore, several researchers have investigated suboptimal SISO equalizers with lower computational complexity. Some suggest to use different approximations to reduce the number of states in the trellis (Berthet, Visoz and Tortelier, 2000; Pukkila, 2000; Berthet, Penther, Visoz and Boutros, 2001; Tungsrisaguan and Rajatheva, 2002; Vishwanath, Mansour and Bahai, 2002; Hanzo, Liew and Yeap, 2002), but the number of states required for satisfactory performance may still be prohibitively large. Others suggest to add SISO capability to filter-based equalizers, as we discuss in the following section. In this dissertation, we concentrate on the latter approach.

5.3.3 Filter-based SISO equalizers

5.3.3.1 Linear SISO equalizer

Most proposals for filter-based SISO equalizers are based on a *linear filter*, combined with the concept of *soft ISI cancellation*. We call such an equalizer a linear SISO equalizer, illustrated in Fig. 5.5. The idea is to subtract (cancel) the expected value of the ISI given by the input LLRs, use a linear equalizer to compensate for the remaining ISI, and perform soft-output demapping to calculate the output LLRs. In the following we go through these operations in detail.

From the input LLRs $L_e^D(c_k)$ are calculated the *a priori* mean $\bar{x}_n = E\{x_n\}$ and variance $v_n = E\{x_n x_n^*\} - \bar{x}_n \bar{x}_n^*$ of each transmitted symbol, using the formulas

$$\bar{x}_n = \sum_{s_i \in \mathcal{S}} s_i \Pr[x_n = s_i] \tag{5.17}$$

$$v_n = \sum_{s_i \in \mathcal{S}} s_i s_i^* \Pr[x_n = s_i] - \bar{x}_n \bar{x}_n^* \tag{5.18}$$

where the symbol probabilities $\Pr[x_n = s_i]$ can easily be calculated from the code bit probabilities $\Pr[c_k = \pm 1]$, which are given by the input LLRs through (5.12)-(5.13). When $\mathcal{S}$ is a PSK constellation, we have $s_i s_i^* = 1$ for all $s_i \in \mathcal{S}$, such that (5.18) simplifies to $v_n = 1 - \bar{x}_n \bar{x}_n^*$.

For values of n corresponding to training symbols t_n known to the receiver, we have $\bar{x}_n = x_n = t_n$ and $v_n = 0$, i.e., we have perfect *a priori* information on these symbols in all iterations, including first-time equalization. For values of n corresponding to data symbols $y_{n'}$, we have $\bar{x}_n = \bar{y}_{n'} = E(y_{n'})$, which gives $\bar{x}_n = 0$ and $v_n = 1$ for first-time equalization. As iterations proceed the *a priori* statistics of the data symbols also approach the ideal values $\bar{x}_n = x_n$ and $v_n = 0$.

One main component of the linear SISO equalizer is a linear filter $\mathbf{f}_n = [f_{n,N_1} \; f_{n,N_1-1} \cdots f_{n,-N_2}]^T$ of length $N = N_1 + N_2 + 1$, and each symbol estimate $\tilde{x}_n$ is calculated from the N received symbols in the vector

$$\mathbf{z}_n = [z_{n+N_1} \; \cdots \; z_n \; \cdots \; z_{n-N_2}]^T \tag{5.19}$$

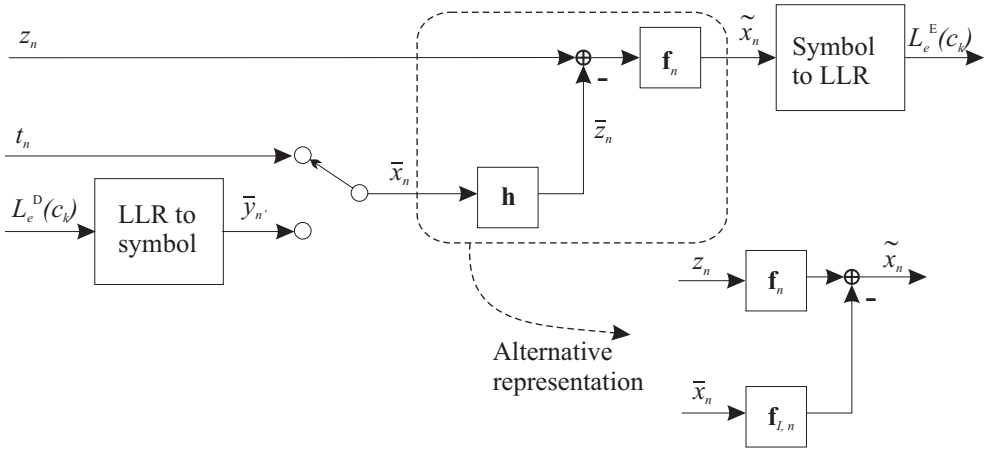


Figure 5.5 Block diagram of a linear SISO equalizer.

The expectation of $\mathbf{z}_n$ given the *a priori* information is obtained by passing the means $\bar{x}_n$ through a filter with impulse response equal to $\mathbf{h}$, and can be written

$$\bar{\mathbf{z}}_n = E\{\mathbf{z}_n\} = \mathbf{H}\bar{\mathbf{x}}_n \tag{5.20}$$

where $\mathbf{H}$ is the $N \times (N + M - 1)$ channel convolution matrix described by (5.5) and $\bar{\mathbf{x}}_n = [\bar{x}_{n+N_1} \cdots \bar{x}_{n-N_2-M+1}]^T$ is a length $N + M - 1$ column vector containing *a priori* means of the transmitted symbols.

Soft ISI cancellation is performed by subtracting the *a priori* average value of the ISI from $\mathbf{z}_n$ before applying the filter $\mathbf{f}_n$. When we wish to estimate the symbol x_n, the *a priori* average value of the ISI is the part of $\bar{\mathbf{z}}_n$ *not dependent* on $\bar{x}_n$, i.e.,

$$\bar{\mathbf{z}}_n - \mathbf{s}\bar{x}_n = \mathbf{H}\bar{\mathbf{x}}_n - \mathbf{s}\bar{x}_n = \mathbf{H}\bar{\mathbf{x}}'_n \tag{5.21}$$

where $\mathbf{s}$ is the $(N_1 + 1)$th column of $\mathbf{H}$ and $\bar{\mathbf{x}}'_n$ is $\bar{\mathbf{x}}_n$ with the $(N_1 + 1)$th element set to zero. These details have been excluded from Fig. 5.5 for illustrational simplicity.

To summarize, the operations of linear filtering and soft ISI cancellation can be written

$$\tilde{x}_n = \mathbf{f}_n^H(\mathbf{z}_n - \mathbf{H}\bar{\mathbf{x}}_n + \mathbf{s}\bar{x}_n) = \mathbf{f}_n^H(\mathbf{z}_n - \mathbf{H}\bar{\mathbf{x}}'_n) \tag{5.22}$$

An alternative representation found in some papers is to move the summation node (see Fig. 5.5), writing

$$\tilde{x}_n = \mathbf{f}_n^H\mathbf{z}_n - \mathbf{f}_{I,n}^H\bar{\mathbf{x}}'_n \tag{5.23}$$

where $\mathbf{f}_{I,n}$ is called the interference cancelling filter. We clearly see that the two representations are equivalent if and only if

$$\mathbf{f}_{I,n} = \mathbf{H}^H\mathbf{f}_n \tag{5.24}$$

Linear SISO equalizers using soft ISI cancellation as described above originally had motivation similar to the "hard" ISI canceller suggested by Gersho and Lim (1981). A natural choice used in the first proposals for linear SISO equalization (Glavieux, Laot and Labat, 1997; Laot, Glavieux and Labat, 2001) was therefore to use the same filters as in (Gersho and Lim, 1981). $\mathbf{f}_n$ is then the matched filter

$$\mathbf{f}_n = \frac{1}{E_h + \sigma_w^2}\mathbf{s} \tag{5.25}$$

The derivation leading to (5.25) is based on the MMSE-criterion, *assuming that* the *a priori* information is perfect, $\bar{x}_n = x_n$. Therefore this approach performs poorly for the first iterations, but performs well when the iterative process is converging and high-quality LLRs are fed back from the decoder.

In order to make the linear SISO equalizer perform well in all iterations, Raphaeli and Saguy (2000) proposed a different optimization criterion to find the coefficients $\mathbf{f}_n$. This proposal, however, seems poorly suited for practical implementation. Michael Tüchler et al. proposed in (Tüchler, 2000; Tüchler, Singer and Koetter, 2002; Tüchler, Koetter and Singer, 2002) to consider the quality of the *a priori* information when using the the MMSE criterion to calculate $\mathbf{f}_n$, which then becomes

$$\begin{aligned}\mathbf{f}_n &= (\mathbf{H}\mathbf{V}_n\mathbf{H}^H + (1-v_n)\mathbf{s}\mathbf{s}^H + \sigma_w^2\mathbf{I}_N)^{-1}\mathbf{s} \\ &= (\mathbf{H}\mathbf{V}_n'\mathbf{H}^H + \sigma_w^2\mathbf{I}_N)^{-1}\mathbf{s}\end{aligned} \tag{5.26}$$

where $\mathbf{V}_n = \text{Diag}[v_{n+N_1} \cdots v_{n-N_2-M+1}]$ is a $(N+M-1)\times(N+M-1)$ diagonal matrix containing *a priori* variances and $\mathbf{V}_n'$ is $\mathbf{V}_n$ with the (N_1+1)th diagonal element set equal to 1. In the next chapter we will rederive (5.26) for the case of a time-varying channel.

Note that when the *a priori* information is perfect, all elements of $\mathbf{V}_n$ are zero and (5.26) becomes the matched filter given by (5.25). Also, when no *a priori* information is present, we have $\mathbf{V}_n = \mathbf{I}_{N+M-1}$, such that (5.26) becomes the expression for a conventional MMSE-optimal linear equalizer given by (5.4). This will be the case for first-time equalization, except for values of n close to a training sequence: When some of the elements of $\bar{\mathbf{x}}_n$ correspond to known symbols, the corresponding diagonal element of $\mathbf{V}_n$ will be zero, and the solution for first-time equalization will be different from (and perform better than) the conventional MMSE-optimal linear equalizer because we cancel the ISI caused by the training symbols.

The equalizer filter $\mathbf{f}_n$ will be different for every symbol interval and for every iteration even though the CIR is constant, because of the dependence on $\mathbf{V}_n$. Since the calculation of $\mathbf{f}_n$ involves a matrix inversion, straightforward implementation would have a computational complexity proportional to N^3 per symbol interval. To address this problem, one can introduce different approximations to make the filter constant over an entire block of data (Tüchler, Singer and Koetter, 2002; Tüchler, Koetter and

Singer, 2002; Tüchler and Hagenauer, 2001; Otnes and Tüchler, 2001). Alternatively, one can use a time-recursive update algorithm for $\mathbf{f}_n$ devised in (Tüchler, Singer and Koetter, 2002) to decrease the computational complexity required to calculate (5.26) to be proportional to N^2 per symbol interval. In Sec. 6.1.2 we will rederive this time-recursive update for the case of time-varying channel conditions.

We call the algorithm given by equations (5.22) and (5.26) the MMSE-optimal linear SISO equalizer. A mathematically equivalent algorithm was derived for multiuser detection in CDMA communications by Wang and Poor (1999) and applied to turbo equalization for a fractionally spaced receiver by Reynolds and Wang (2001). The algorithm has later been rederived by Dejonghe and Vandendorpe (2002) and Raphaeli and Saguy (2002), and can therefore now be considered as widely accepted.

Finally, we must perform soft-output demapping to calculate LLRs on the code bits from the estimates $\tilde{x}_n$ of the transmitted data symbols. In order to do this we need to know the conditional pdf of $\tilde{x}_n$ given that the transmitted symbol was $x_n = s_i$. The exact conditional pdf is difficult to compute and would also complicate the calculation of output LLRs, and is therefore approximated with the complex Gaussian pdf

$$p_{\tilde{x}_n|s_i}(\tilde{x}_n|x_n = s_i) = \frac{1}{\pi\sigma_{n,i}^2} e^{\frac{-|\tilde{x}_n - \mu_{n,i}|^2}{\sigma_{n,i}^2}} \tag{5.27}$$

where $\mu_{n,i}$ and $\sigma_{n,i}^2$ are the mean and variance of the conditional pdf. Using equations (5.21) and (5.22), these statistics can be shown to be

$$\mu_{n,i} = E(\tilde{x}_n|x_n = s_i) = s_i \mathbf{f}_n^H \mathbf{s} = r_n s_i \tag{5.28}$$

$$\sigma_{n,i}^2 = E(\tilde{x}_n \tilde{x}_n^*|x_n = s_i) - \mu_{n,i}\mu_{n,i}^* \tag{5.29}$$

$$= \mathbf{f}_n^H (\mathbf{H}\mathbf{V}_n\mathbf{H}^H - v_n \mathbf{s}\mathbf{s}^H + \sigma_w^2 \mathbf{I}_n)\mathbf{f}_n \tag{5.30}$$

where we have defined $r_n = \mathbf{f}_n^H \mathbf{s}$. These expressions are valid for a general equalizer filter $\mathbf{f}_n$. Note that the variance of the conditional pdf is the same for all $s_i \in \mathcal{S}$. When $\mathbf{f}_n$ is the MMSE-optimal filter given by (5.26), the variance (5.30) of the conditional pdf simplifies to

$$\sigma_{n,i}^2 = \mathbf{f}_n^H \mathbf{s} - \mathbf{s}^H \mathbf{f}_n \mathbf{f}_n^H \mathbf{s} = r_n - r_n^2 \tag{5.31}$$

From eqs. (5.28) and (5.31) we see that r_n can be viewed as a quality measure on the symbol estimate $\tilde{x}_n$, where $r_n = 1$ means that $\tilde{x}_n = x_n$ because the variance becomes zero. We will have $0 \leq r_n \leq 1$ because the variance in (5.31) must be a non-negative real number.

The expression for calculating the output LLRs for each code bit requires detailed notation describing the mapping of code bits to the signal constellation. We do not introduce this notation here, but refer the interested reader to eq. (8) in (Tüchler, Singer and Koetter, 2002). The calculation of the output LLR $L_e^E(c_k)$ of a code

bit $c_k = c_{n'Q+j}$ involves the symbol estimate $\tilde{x}_n$ of the data symbol to which c_k is mapped, the statistics $\mu_{n,i}$ and $\sigma^2_{n,i}$ of the conditional pdf of $\tilde{x}_n$ (which can be represented by the single value r_n for the MMSE-optimal linear SISO equalizer), and the input LLRs $L_e^D(c_{n'Q+j'}), j' \neq j$, of the other code bits mapped to the same data symbol.

5.3.3.2 DFE-based SISO equalizers

Because it is known that DFEs perform better than linear equalizers for a conventional receiver, some may argue that the linear SISO equalizer described above should be replaced with a DFE. Some approaches to DFE-based SISO equalization can be found in (Strauch, Luschi, Sandell and Yan, 1999; Blackmon, Sozer, Murandian, Proakis and Salehi, 2001). Others argue that a DFE should be used for first-time equalization, when no *a priori* information is available, and be replaced with a linear SISO equalizer in subsequent iterations (Langlais and Hélard, 2002).

It is our belief that hard decisions should be avoided altogether in a receiver employing turbo equalization, and therefore the DFE should not be used. When a DFE is used, the performance will be better than for the linear SISO equalizer after first-time equalization and decoding. But as iterations proceed, we find that the receiver using the linear SISO equalizer will perform better. This is because some of the LLRs calculated from the output symbols of a DFE are over-optimistic, because one has to assume that the decisions fed back in the DFE are correct. These over-optimistic LLRs will cause the decoder to be certain about the value of code bits which in reality are uncertain, and all LLRs calculated throughout the iterations will be inexact (a kind of error propagation). This argument is backed by simulations and analysis in (Tüchler, Koetter and Singer, 2002), where the effect of replacing the filter $\mathbf{f}_n$ in Fig. 5.5 with an MMSE-optimal DFE has been investigated.

Actually, the linear SISO equalizer of Fig. 5.5 can be viewed as a "soft DFE", where soft decisions from the decoder rather than hard decisions from the equalizer are passed through a feedback filter. When the iterative process has converged, the LLRs fed back from the decoder will have high quality such that we have $\bar{x}_n \approx x_n$. Then, the ISI caused by soft "decisions", which we know are almost correct, is cancelled. We can then cancel all the ISI, caused by past as well as future symbols relative to the symbol we wish to estimate, such that the performance of a turbo equalization-based receiver using a linear SISO equalizer approaches the performance of a system where the data symbols are transmitted over an AWGN channel with no delay spread (frequency-flat channel). Thus, linear equalization in a turbo equalization-based receiver can cope with spectral nulls in the frequency response of the CIR just as well as a DFE, as opposed to a non-iterative receiver using linear equalization which is known to have inferior performance when the frequency response has spectral nulls.

5.3.4 Neural network-based SISO equalizers

We finally note that SISO equalizers based on the theory of neural networks and radial basis functions have also been proposed in the literature (Yee, Yeap and Hanzo, 2001; Yee, Ng and Hanzo, 2002; Hanzo, Wong and Yee, 2002). Those algorithms are beyond the scope of this work, but are referenced here for completeness.

5.3.5 SISO decoders

The task of the SISO decoder is to use the deinterleaved LLRs $L_e^E(c_{k'})$ from the equalizer to estimate the transmitted data bits a_m (as in a conventional receiver), and also to calculate extrinsic LLRs $L_e^D(c_{k'})$ for the equalizer to use in the next iteration.

The optimal SISO decoder for a convolutional code is the MAP algorithm (Bahl et al., 1974). This algorithm traverses the code trellis for a block of data (e.g. an interleaver block) in both directions, and performs best if the code trellis is terminated to the zero state at the beginning and end of each block. The algorithm can also be applied when tail-biting has been used (as in the high-rate HF waveforms), by traversing the code trellis more than once in each direction, in a circular manner, in order to find the boundary *a posteriori* probabilities of the first and last code bits of the block (Anderson and Hladik, 1998; Weiss, Bettstetter and Riedel, 2001).

The MAP decoder is more complex than the Viterbi decoder used in conventional HF receivers, because the code trellis has to be traversed in both directions and the results from the forward recursion for the entire block must be kept in memory until the backward recursion is performed. Also, the decoding has to be performed several times in a turbo equalization-based receiver. The code trellis of the convolutional code used in standard HF waveforms has 64 states, which is quite large for a MAP decoder. If it turns out to be too complicated to perform MAP decoding in a turbo equalization-based receiver, one has to use a suboptimal SISO decoder. For details on implementation of optimal and suboptimal SISO decoders, we refer to the literature on turbo codes, e.g. (Vucetic and Yuan, 2000; Chugg, Anastasopoulos and Chen, 2001; Benedetto, Divsalar, Montorsi and Pollara, 1996).

In all simulations of turbo equalization-based receivers presented in this dissertation we have used a MAP decoder (implemented as log-MAP), and terminated the code trellis to the zero state at the end of each interleaver block.

5.3.6 The role of the interleaver

When a conventional receiver is used, the motivation for using an interleaver in the transmitter, and a corresponding deinterleaver in the receiver, is to break up burst errors introduced by the channel and equalizer, because convolutional codes do not perform well in burst errors. In other words, the deinterleaver ensures that the corre-

lation between received code bits at the input to the decoder is as small as possible, at least for small lags.

When turbo equalization is used in the receiver, the decorrelating property of the deinterleaver/interleaver becomes even more crucial. The SISO decoding and equalization algorithms are derived assuming that all input LLRs to the SISO modules are independent, and the performance deteriorates when correlations exist. When trying to detect a certain code bit and calculate its output LLR, a SISO module uses the input LLRs for the nearby code bits in the computations. This means that (1) correlation between nearby input LLRs will cause performance degradation, because we may then have several adjacent LLRs of small magnitude (analogous to burst errors), and (2) nearby output LLRs will be correlated, because some of the input LLRs used to compute them are identical. Combining these two observations, we see that a deinterleaver/interleaver pair is certainly required between the SISO modules, especially when a number of iterations are performed.

In the turbo coding literature, there has been much focus on interleaver design, see e.g. (Blackert, Hall and Wilson, 1995; Garello, Montorsi, Benedetto and Cancellieri, 2001; Hokfelt, Edfors and Maseng, 2001; Feng, Yuan and Vucetic, 2002) and references therein. This is harder to do when considering turbo equalization for a time-varying channel, because we can not consider the inner code (i.e. the channel) as known when designing the interleaver. The interleaver should therefore be designed to be "as random as possible". However, our focus is on applying turbo equalization to existing standardized waveforms, and we must therefore consider the interleaver permutation as given. In the simulations in Chap. 9 we have used interleavers from the HF standards, while in Chaps. 6 and 8 we have used so-called S-random interleavers which are known to perform well for turbo codes.

Finally a note on our terminology: Following (Garello et al., 2001), we use the term "block interleaver" to describe an interleaver operating on a block of data at a time, as opposed to a convolutional interleaver. Therefore, we include S-random interleavers into the class of block interleavers. In other works on turbo coding, on the other hand, "block interleaver" is sometimes used to describe a regular interleaver permutation (e.g. a rectangular interleaver), as opposed to a random interleaver permutation.

5.4 Adaptive turbo equalization for unknown/time-varying channels

When the channel impulse response is unknown and/or time-varying, the SISO equalizer must be adapted to the channel conditions. The block diagram of the turbo equalization-based receiver is then as shown in Fig. 5.6, where the only difference from Fig. 5.4 is that the "SISO equalizer" has been replaced by an "adaptive SISO

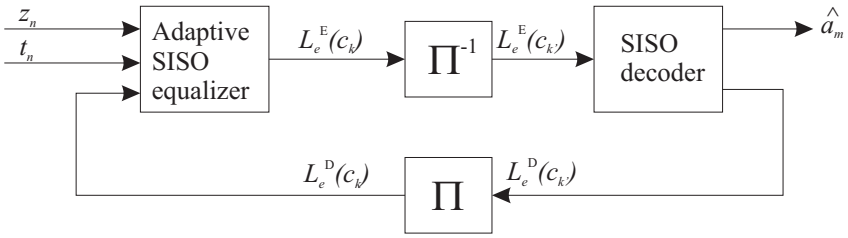


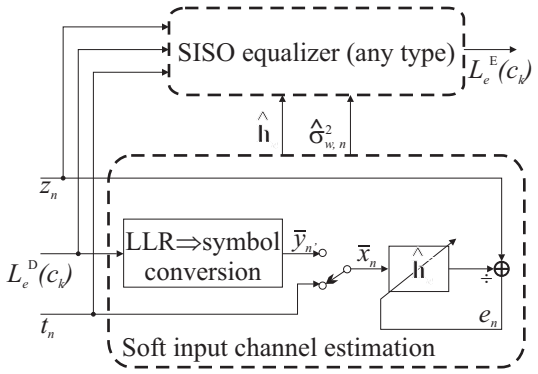
Figure 5.6 Block diagram of a turbo equalization-based receiver for an unknown and/or time-varying channel.

equalizer". We use the general term *adaptive turbo equalization* to describe this receiver structure.

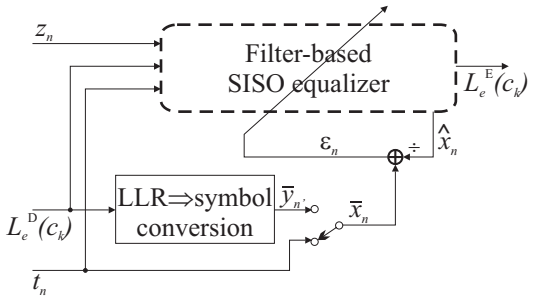
The optimal receiver to detect data bits encoded with an ECC and transmitted over an unknown ISI channel, would jointly perform channel estimation, equalization, and decoding. As for the case of a known channel, this is prohibitively complex when an interleaver is present but can be approximated by performing iterations between the adaptive equalizer and the decoder. The optimal adaptive SISO equalizer is trellis-based and performs channel estimation jointly with the equalization in a PSP-like manner (Davis, Collins and Hoeher, 2001; Wang and Chen, 2001; Anastasopoulos and Chugg, 2000; Vlahoyiannatos and Hanzo, 2001). These approaches are also too complex for our application because a trellis-based equalizer is required, and we thus have to look for suboptimal approaches.

There are several choices to make when implementing a suboptimal adaptive SISO equalizer: The simplest is to estimate the channel impulse response during first-time equalization, and use this estimate for all iterations. Better performance can be achieved by using hard or soft information fed back from the decoder in order to improve the channel estimate in subsequent iterations. This is called iterative channel estimation. If the channel is varying relatively slowly, it may suffice to calculate a channel estimate which is constant over a block of data. Otherwise, the channel estimate should be updated every symbol interval. We also have the choice of direct adaptation vs separate channel estimation discussed in Sec. 5.2.2, which is illustrated in Fig. 5.7 for the case where soft information fed back from the decoder is used to improve the channel estimate in subsequent iterations.

We use the term *iterative channel estimation, equalization, and decoding* to describe a turbo equalization-based receiver where the adaptive SISO equalizer uses separate channel estimation and equalization, as shown in Fig. 5.7(a). Here, the equalizer can be trellis-based or filter-based. Iterative channel estimation implies that the quality of the input reference signal to the channel estimator, which ideally should be equal to the transmitted symbols x_n according to (5.9), is improved for



(a) Separate channel estimation and equalization.



(b) Direct adaptation of equalizer.

Figure 5.7 Two different approaches to adaptive SISO equalization in a turbo equalization-based receiver, using soft information fed back from the decoder as input to the adaptive algorithm.

each iteration. We use the term *hard iterative channel estimation* when the error signal whose variance is minimized by the channel estimator is

$$d'_n = z_n - \sum_{l=0}^{M-1} \hat{h}_{n,l}\hat{x}_{n-l} \tag{5.32}$$

where $\hat{x}_n$ are hard decisions fed back from the decoder (and not from the equalizer) from the previous iteration. The $\hat{x}_n$ are here obtained by mapping hard-decided code bits $\hat{c}_k = \text{sgn}(L_e^D(c_k))$ from the decoder onto the signal constellation $\mathcal{S}$. These hard decisions will have a lower error rate than the hard decisions from the equalizer. When training symbols have been transmitted, $\hat{x}_n$ is of course replaced with t_n. Hard iterative channel estimation has been proposed in (Nefedov and Pukkila, 2000; Visoz, Berthet, Saadani and Penther, 2001; Strauch, Luschi and Kuzminskiy, 2000) where

the channel estimate is constant over a block of data, and in (Tantikovit, Sheikh and Wang, 1998) where the channel estimate is varying from symbol to symbol.

When hard iterative channel estimation is used, a few errors in the code bits fed back from the decoder may seriously degrade the quality of the channel estimate. Therefore it may be wiser to use *soft iterative channel estimation*, where the error signal is defined as

$$d_n = z_n - \sum_{l=0}^{M-1} \hat{h}_{n,l} \bar{x}_{n-l} \tag{5.33}$$

where $\bar{x}_n$ are *a priori* average values of the transmitted symbols, calculated using (5.17) from the LLRs $L_e^D(c_k)$ fed back from the decoder. Soft iterative channel estimation has been proposed in (Kammeyer, Kühn and Petermann, 2001; Strauch et al., 2000; Wong, Yeap and Hanzo, 2000; Yeap, Wong and Hanzo, 2001) where the channel estimate is constant over a block of data, and in (Song, Singer and Sung, 2002; Tüchler, Otnes and Schmidbauer, 2002; Otnes and Tüchler, 2002d) where the channel estimate is varying from symbol to symbol. Soft-input channel estimation has also been proposed in (Nowlan and Hinton, 1993; Baccarelli and Cusani, 1998), where soft symbol estimates from the equalizer rather than from the decoder are fed back to the channel estimator. We also note that iterative channel estimation (hard or soft) has recently been proposed for multiuser detection in CDMA communications (Lampe, 2002; Thomas and Geraniotis, 2002) and for communications using multiple antennas (Sellathurai and Haykin, 2002). In the latter reference, something in between hard and soft iterative channel estimation is used: Only the hard decisions with reliability above a certain threshold are used for channel estimation, where the reliability is calculated from the soft information.

For first-time equalization, where no *a priori* information is present, one may either choose to use hard decisions from the equalizer to update the channel estimate (as in a conventional receiver, see Fig. 5.3(a)), or to keep the channel estimate constant between the training sequences.

For filter-based SISO equalizers, we can also use direct adaptation of the equalizer coefficients, as shown in Fig. 5.7(b). Now, we have the choice of using hard-decided symbols $\hat{x}_n$ or *a priori* means $\bar{x}_n$ as the *desired* signal in the adaptive algorithm. In (Glavieux et al., 1997; Laot et al., 2001) is proposed to use hard-decided symbols as the desired signal in an LMS adaptive filter designed to converge towards the matched filter given by (5.25), and in (Kuhn, 2002) is shown how soft information (*a priori* means) can be used as the desired signal in an LMS or RLS adaptive filter designed to converge towards an approximation to the MMSE-optimal linear SISO equalizer given by (5.26).

The argumentation why separate channel estimation and equalization performs better than direct equalizer adaptation in a conventional receiver (see Sec. 5.2.2.2) should also be valid for turbo equalization-based receivers. Also, soft feedback is

intuitively better than hard feedback because error propagation is avoided. Therefore we have chosen to concentrate on soft iterative channel estimation for our turbo equalization-based receiver, which is then described by the term used in the title of this dissertation: Iterative channel estimation, equalization, and decoding. We also update the channel estimate every symbol interval, in order to be able to track channel variations which are fast compared to the interval between training sequences.

In Chap. 7 we discuss soft iterative channel estimation in more detail, and in Chap. 9 we present our proposed receiver using iterative channel estimation, equalization, and decoding.

5.5 Turbo equalization combined with turbo codes

The performance of a turbo equalization-based receiver, as described previously in this chapter, can never be better than the performance of the ECC transmitted over an AWGN (frequency-flat) channel. In the systems we consider, the ECC is a "traditional" convolutional code, and we could of course achieve better performance by replacing this ECC with a turbo code.

When the ECC of a system employing turbo equalization is a turbo code, two interleavers are needed in the transmitter: One between the two constituent encoders of the turbo code and one between the turbo code and the symbol mapper/channel. In the receiver, iterations must be performed between three SISO modules: The SISO equalizer and the SISO decoders for each constituent code. Such systems are proposed in (Raphaeli and Zarai, 1997; Yeap, Liew, Hámorský and Hanzo, 2002; Hanzo, Liew and Yeap, 2002).

A simpler way to approach the Shannon-limit performance of turbo coded systems is to introduce a rate-1 recursive channel precoder in conjunction with the symbol mapper. Then, the inner code of the serially concatenated system will be a combination of the channel and the precoder, which is a recursive code (i.e. has infinite impulse response). In (Benedetto, Divsalar, Montorsi and Pollara, 1998; ten Brink, 2000b; Tüchler and Hagenauer, 2002) it is shown that the Shannon limit can be approached arbitrarily close in serially concatenated systems where the inner code is a rate-1 recursive code and the outer code is either recursive or non-recursive. The combination of ECC, interleaver, precoder and channel will then be a serially-concatenated turbo code. Turbo equalization with a recursive channel precoder has been studied in (Lee, 2001; Narayanan, 2001).

In this dissertation we seek to improve the receiver performance for existing standardized HF waveforms. We have therefore not considered the approaches discussed in this section further, as they require modifications to standardized transmitter configurations. They should, however, be considered as alternatives if new waveforms for HF communications are standardized in the future.

5.6 Chapter summary

In this chapter, we have given a survey of conventional non-iterative receivers as well as receivers employing turbo equalization, for detection of coded data transmitted over time-invariant or time-varying ISI channels. We have concentrated on approaches that do not require trellis-based equalizers, in particular the linear SISO equalizer. We have also discussed the problem of channel estimation, and introduced the concept of soft iterative channel estimation. Finally, we have given references to works where turbo equalization is used in combination with turbo codes, in contrast to the main focus of this dissertation where the ECC is a single convolutional code.

URN:NBN:no-3359

Chapter 6

Linear SISO equalizer for time-varying channels

In this chapter, we rederive the MMSE-optimal linear SISO equalizer of (Tüchler, Singer and Koetter, 2002), described in Sec. 5.3.3.1, for the case where the channel impulse response is time-varying from one symbol interval to the next. We present derivations and simulations for the case where the channel impulse response is known, and also present simulations where the derived SISO equalizer is used in combination with soft iterative channel estimation. Finally, we discuss how the algorithm can be applied to fractionally spaced equalization. This chapter is mostly based on a paper presented at the IEEE Vehicular Technology Conference in May, 2002 (Otnes and Tüchler, 2002d).

6.1 Linear SISO equalizer for a known, time-varying channel

We assume the system model introduced in Sec. 5.1, with time-varying channel impulse response $\mathbf{h}_n$ and noise variance $\sigma_{w,n}^2$, i.e., the received symbols are given by

$$z_n = \sum_{l=0}^{M-1} h_{n,l} x_{n-l} + w_n \tag{6.1}$$

where w_n is white Gaussian noise, independent from all x_n, with variance $E(w_n w_n^*) = \sigma_{w,n}^2$. In the simulations presented in this chapter we use a BPSK constellation, $Q = 1$ and $\mathcal{S} = \{+1, -1\}$, but the derivations are also valid for higher-order signal constellations. Simulations of the same algorithms using higher-order constellations will be presented in later chapters.

The structure of the linear SISO equalizer is shown in Fig. 6.1, which is identical to Fig. 5.5 except that the calculation of filter coefficients is explicitly shown. The main part is a linear time-varying filter $\mathbf{f}_n$ of length $N = N_1 + N_2 + 1$ with N_1 precursor taps and N_2 postcursor taps. Input to the equalizer are the received symbols z_n as well as the *a priori* information $\bar{x}_n = E\{x_n\}$ (the expectation of x_n), which can be calculated from the LLRs $L_e^D(c_k)$ fed back from the decoder as shown in Sec. 5.3.3.1. The *a priori* variance $v_n = E\{|x_n - \bar{x}_n|^2|$ of each x_n is given by $1 - |\bar{x}_n|^2$ when the signal constellation is PSK, and by (5.18) for other signal constellations. Because the code bits are shuffled by an interleaver, we can assume that $E\{(x_n - \bar{x}_n)(x_j - \bar{x}_j)^*\} = 0$ for all $n \neq j$, at least locally within $N + M - 1$ consecutive symbols.

At time step n, the linear SISO equalizer considers N received symbols $\mathbf{z}_n = [z_{n+N_1} \cdots z_{n-N_2}]^T$ given by

$$\mathbf{z}_n = \mathbf{H}_n \mathbf{x}_n + \mathbf{w}_n \tag{6.2}$$

where $\mathbf{x}_n = [x_{n+N_1} \cdots x_{n-N_2-M+1}]^T$, $\mathbf{H}_n$ is the time-varying $N \times (N + M - 1)$ channel convolution matrix which is now given by

$$\begin{aligned}\mathbf{H}_n &= \begin{bmatrix} \mathbf{h}_{n+N_1}^T & 0 & \cdots & 0 \\ 0 & \mathbf{h}_{n+N_1-1}^T & \cdots & 0 \\ & \ddots & & \\ 0 & \cdots & & \mathbf{h}_{n-N_2}^T \end{bmatrix} \\ &= \begin{bmatrix} h_{n+N_1,0} & h_{n+N_1,1} & \cdots & h_{n+N_1,M-1} & 0 & \cdots & 0 \\ 0 & h_{n+N_1-1,0} & \cdots & & h_{n+N_1-1,M-1} & 0 & \vdots \\ \vdots & & & \ddots & & & 0 \\ 0 & \cdots & 0 & h_{n-N_2,0} & h_{n-N_2,1} & \cdots & h_{n-N_2,M-1} \end{bmatrix}\end{aligned} \tag{6.3}$$

and $\mathbf{w}_n = [w_{n+N_1} \cdots w_{n-N_2}]^T$.

6.1.1 MMSE-optimal solution

We will now rederive the MMSE linear equalization algorithm for the case of a time-varying channel. The derivation follows the line of (Tüchler, Singer and Koetter, 2002), except that we allow $\mathbf{h}_n$ and $\sigma_{w,n}^2$ to be time-varying.

The coefficients $\mathbf{f}_n$ are calculated according to the MMSE-criterion, i.e., to minimize $E\{|\tilde{x}_n - x_n|^2\}$. We first ignore the constraint that $\tilde{x}_n$ should not be a function of $\bar{x}_n$, in which case the MMSE-optimal linear estimate of x_n given the observed vector $\mathbf{z}_n$ is (Poor, 1994)

$$\tilde{x}_n = E\{x_n\} + Cov\{\mathbf{z}_n, x_n\}^H Cov\{\mathbf{z}_n, \mathbf{z}_n\}^{-1}(\mathbf{z}_n - E\{\mathbf{z}_n\}) \tag{6.4}$$

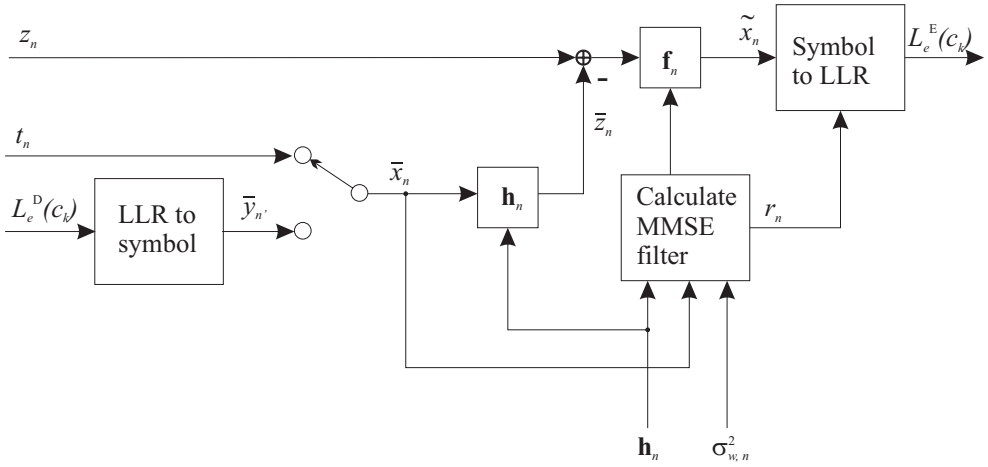


Figure 6.1 Structure of the linear SISO equalizer.

where $Cov\{\mathbf{a},\mathbf{b}\} = E\{\mathbf{a}\mathbf{b}^H\} - E\{\mathbf{a}\}E\{\mathbf{b}^H\}$. From (6.2) we find that

$$E\{x_n\} = \bar{x}_n \tag{6.5}$$
$$E\{\mathbf{z}_n\} = \bar{\mathbf{z}}_n = \mathbf{H}_n\bar{\mathbf{x}}_n \tag{6.6}$$
$$Cov\{\mathbf{z}_n,\mathbf{z}_n\} = \mathbf{H}_n\mathbf{V}_n\mathbf{H}_n^H + \mathbf{\Sigma}_n \tag{6.7}$$
$$Cov\{\mathbf{z}_n,x_n\} = \mathbf{H}_nE\{(\mathbf{x}_n - \bar{\mathbf{x}}_n)x_n^*\} = \mathbf{s}_n v_n \tag{6.8}$$

where

$$\bar{\mathbf{x}}_n = E\{\mathbf{x}_n\} = [\bar{x}_{n+N_1} \ \cdots \ \bar{x}_n \ \cdots \ \bar{x}_{n-N_2-M+1}]^T \tag{6.9}$$
$$\mathbf{V}_n = Cov\{\mathbf{x}_n,\mathbf{x}_n\} = \mathrm{Diag}[v_{n+N_1} \ \cdots \ v_n \ \cdots \ v_{n-N_2-M+1}] \tag{6.10}$$
$$\mathbf{\Sigma}_n = Cov\{\mathbf{w}_n,\mathbf{w}_n\} = \mathrm{Diag}[\sigma^2_{w,n+N_1} \ \cdots \ \sigma^2_{w,n} \ \cdots \ \sigma^2_{w,n-N_2}] \tag{6.11}$$

and $\mathbf{s}_n$ is the (N_1+1)th column of $\mathbf{H}_n$. By inserting equations (6.5)-(6.8) into (6.4), we obtain

$$\tilde{x}_n = \bar{x}_n + v_n\mathbf{g}_n^H(\mathbf{z}_n - \mathbf{H}_n\bar{\mathbf{x}}_n) \tag{6.12}$$
$$\mathbf{g}_n = (\mathbf{H}_n\mathbf{V}_n\mathbf{H}_n^H + \mathbf{\Sigma}_n)^{-1}\mathbf{s}_n \tag{6.13}$$

which is the MMSE-optimal linear estimate of x_n without the constraint that $\tilde{x}_n$ should not be a function of the *a priori* information $\bar{x}_n$ and v_n for the same symbol. The constraint is fulfilled by forcing $\bar{x}_n$ to 0 and v_n to 1 when calculating $\tilde{x}_n$. Noting that $\bar{x}_n$ is the (N_1+1)th entry in $\bar{\mathbf{x}}_n$, (6.12)-(6.13) then becomes

$$\tilde{x}_n = \mathbf{f}_n^H(\mathbf{z}_n - \mathbf{H}_n\bar{\mathbf{x}}_n + \bar{x}_n\mathbf{s}_n) \tag{6.14}$$
$$\mathbf{f}_n = (\mathbf{H}_n\mathbf{V}_n\mathbf{H}_n^H + (1-v_n)\mathbf{s}_n\mathbf{s}_n^H + \mathbf{\Sigma}_n)^{-1}\mathbf{s}_n \tag{6.15}$$

Note the similarities with the MMSE-optimal linear equalizer for a time-invariant channel, given by equations (5.22) and (5.26).

Using the matrix inversion lemma (Haykin, 1996, page 385), it turns out that $\mathbf{f}_n$ is simply a scaled version of $\mathbf{g}_n$:

$$\mathbf{f}_n = \frac{\mathbf{g}_n}{1 + (1 - v_n)\mathbf{g}_n^H \mathbf{s}_n} \tag{6.16}$$

This result is useful because $\mathbf{g}_n$ is easier to compute than $\mathbf{f}_n$, using the time-recursive algorithm described in the next section.

6.1.2 Time-recursive update

The computational complexity to directly compute the filter coefficients $\mathbf{f}_n$, or equivalently $\mathbf{g}_n$, is proportional to N^3 per time step n. Fortunately, a time-recursive algorithm similar to the one for a time-invariant channel presented in (Tüchler, Singer and Koetter, 2002) can be derived, with a complexity proportional to N^2. The algorithm exploits the structured time-dependence of the Hermitian $N \times N$ matrix

$$\mathbf{\Pi}_n = \mathbf{H}_n \mathbf{V}_n \mathbf{H}_n^H + \mathbf{\Sigma}_n \tag{6.17}$$

and computes $\mathbf{g}_n$ from $\mathbf{g}_{n-1}$ and $\mathbf{f}_n$ from $\mathbf{g}_n$ through (6.16). From the definition of $\mathbf{\Pi}_n$ follows that $\mathbf{g}_n = \mathbf{\Pi}_n^{-1}\mathbf{s}_n$, and that

$$\mathbf{\Pi}_{n-1} = \left[\begin{array}{c|c} \mathbf{A}_n & \mathbf{b}_n \\ - + - & \\ \mathbf{b}_n^H & b_n \end{array}\right], \; \mathbf{\Pi}_n = \left[\begin{array}{c|c} a_n & \mathbf{a}_n^H \\ - + - & \\ \mathbf{a}_n & \mathbf{A}_n \end{array}\right] \tag{6.18}$$

with the common $(N-1) \times (N-1)$ submatrix $\mathbf{A}_n$. The new elements in the first column and row of $\mathbf{\Pi}_n$ are given by

$$a_n = \sigma_{w,n+N_1}^2 + \sum_{l=0}^{M-1} |h_{n+N_1,l}|^2 v_{n+N_1-l} \tag{6.19}$$

$$\mathbf{a}_n = [\alpha_{n,1} \; \cdots \; \alpha_{n,N-1}]^T \tag{6.20}$$

$$\alpha_{n,j} = \sum_{l=0}^{M-1-j} h_{n+N_1-j,l} h_{n+N_1,l+j}^* v_{n+N_1-l-j} \tag{6.21}$$

Because the inverse of a Hermitian matrix is also Hermitian, we can split the inverse matrix $\mathbf{U}_n = \mathbf{\Pi}_n^{-1}$ in a similar way, but without a common submatrix:

$$\mathbf{U}_{n-1} = \left[\begin{array}{c|c} \mathbf{D}_n & \mathbf{d}_n \\ - + - & \\ \mathbf{d}_n^H & d_n \end{array}\right], \; \mathbf{U}_n = \left[\begin{array}{c|c} c_n & \mathbf{c}_n^H \\ - + - & \\ \mathbf{c}_n & \mathbf{C}_n \end{array}\right] \tag{6.22}$$

The recursive algorithm calculates $\mathbf{A}_n^{-1}$ from $\mathbf{U}_{n-1}$ using the identity $\mathbf{\Pi}_{n-1}\mathbf{U}_{n-1} = \mathbf{I}_N$. The matrix $\mathbf{U}_n$ is then calculated from $\mathbf{A}_n^{-1}$ using the identity $\mathbf{\Pi}_n\mathbf{U}_n = \mathbf{I}_N$, and used to start the next recursion and to compute the new coefficient vector $\mathbf{g}_n$:

$$\mathbf{A}_n^{-1} = \mathbf{D}_n - \mathbf{d}_n\mathbf{d}_n^H/d_n \tag{6.23}$$
$$\boldsymbol{\psi}_n = \mathbf{A}_n^{-1}\mathbf{a}_n \tag{6.24}$$
$$c_n = 1/(a_n - \mathbf{a}_n^H\boldsymbol{\psi}_n) \tag{6.25}$$
$$\mathbf{c}_n = -\boldsymbol{\psi}_n c_n \tag{6.26}$$
$$\mathbf{C}_n = \mathbf{A}_n^{-1} + c_n\boldsymbol{\psi}_n\boldsymbol{\psi}_n^H \tag{6.27}$$
$$\mathbf{g}_n = \mathbf{U}_n\mathbf{s}_n \tag{6.28}$$

The intermediate vector $\boldsymbol{\psi}_n$ is introduced in order to simplify the calculation. Note that no matrix inversion is required when using these update equations.

The recursion can be initialized during the initial training sequence, at a time step where $N + M - 1$ consecutive symbols x_n are known training symbols yielding $\mathbf{V}_n = \mathbf{0}_{N+M-1}$. Then, $\mathbf{U}_n = \mathbf{\Pi}_n^{-1} = \mathbf{\Sigma}_n^{-1}$, which is trivial to compute because the matrix to be inverted is diagonal.

In the simulations of MMSE-optimal linear SISO equalizers presented in this dissertation, we have implemented this recursion as C code.

6.1.3 Calculation of output LLRs

As mentioned in Sec. 5.3.3.1, the code bit LLRs $L_e^E(c_k)$ after equalization are calculated from $\tilde{x}_n$ by approximating the (unknown) conditional probability density function (pdf) $p_{\tilde{x}_n|s_i}(\tilde{x}_n|x_n = s_i)$, $s_i \in \mathcal{S}$, of $\tilde{x}_n$ given that s_i was transmitted, as a Gaussian distribution with mean $\mu_{i,n}$ and variance $\sigma_{i,n}^2$ given by

$$\mu_{i,n} = E\{\tilde{x}_n|x_n = s_i\} = s_i \cdot r_n \tag{6.29}$$
$$\sigma_{i,n}^2 = E\{\tilde{x}_n\tilde{x}_n^*|x_n = s_i\} - \mu_{i,n}\mu_{i,n}^* = r_n - r_n^2 \tag{6.30}$$

where $r_n = \mathbf{f}_n^H\mathbf{s}_n$. For the case where $\mathcal{S}$ is a BPSK constellation, we have $c_k = x_n$ such that the LLRs $L_e^E(c_k)$ can be calculated as:

$$\begin{aligned} L_e^E(c_k) = L_e^E(x_n) &= \ln\frac{\Pr(x_n = +1|\tilde{x}_n)}{\Pr(x_n = -1|\tilde{x}_n)} - L_e^D(c_k) \\ &= \ln\frac{p_{\tilde{x}_n|+1}(\tilde{x}_n)}{p_{\tilde{x}_n|-1}(\tilde{x}_n)} = \ln\frac{\exp\left(\frac{-|\tilde{x}_n - r_n|^2}{2(r_n - r_n^2)}\right)}{\exp\left(\frac{-|\tilde{x}_n + r_n|^2}{2(r_n - r_n^2)}\right)} = \frac{2\,\tilde{x}_n}{1 - r_n} \end{aligned} \tag{6.31}$$

where the *a priori* LLR $L_e^D(c_k)$ fed back from the decoder is subtracted out since the SISO equalizer should output the extrinsic information only. For higher-order signal constellations the expression gets more complicated, and we again refer to (Tüchler, Singer and Koetter, 2002) for details.

6.1.4 Simulation results for known channel conditions

In (Otnes and Tüchler, 2002d), we presented simulations using the following setup: Binary data is systematic recursively encoded to blocks of $N_{int} = 2048$ code bits $c_{k'}$ using a rate $1/2$ convolutional ECC with generator polynomials $D^4 + D + 1$ and $D^4 + D^3 + D^2 + 1$, which are permuted using an S-random interleaver (Heegard and Wicker, 1999; Vucetic and Yuan, 2000) and mapped to symbols y_n from $\mathcal{S} = \{+1, -1\}$. We use the first-order real-valued AR model

$$\mathbf{h}_n = \rho \cdot \mathbf{h}_{n-1} + \sqrt{1-\rho^2} \cdot [q_{n,0} \cdots q_{n,M-1}]^T \tag{6.32}$$

to generate a time-varying CIR $\mathbf{h}_n$ of length $M = 5$. The parameter ρ is set to $\sqrt{0.999}$, corresponding to a fading rate of $\nu_d \approx f_s/2000$ according to (4.34), and the $q_{i,n}$ are real zero-mean independent Gaussian noise samples with variance $1/M$ yielding a unit average energy $\sum_{l=0}^{M-1} E(|h_{n,l}|^2) = 1$ of the CIR. The impulse response is initialized to $\mathbf{h}_0 = [1\,1 \cdots 1]/\sqrt{M}$ at the beginning of the first block.

A BPSK signal constellation is used, such that the transmitted data symbols $y_{n'}$ are equal to the interleaved code bits c_k. Training sequences of 15 known symbols t_n are regularly inserted into the block, with 64 data symbols between each training sequence. The training sequences are used to aid the channel estimation process for the case of an unknown channel, to be discussed in the next section, and are also used by the SISO equalizer as *a priori* information (even during first-time equalization).

The receiver in the simulations employs a turbo equalizer structure with 4 iterations after initial equalization and decoding. When using the linear SISO equalizer, the filter length is $N = 13$ ($N_1 = 8$ and $N_2 = 4$).

All error rates are defined for data bits out of the decoder, and have been calculated by simulating 10000 interleaver blocks, i.e. about 1 million data bits. The frame error rate indicates the fraction of interleaver blocks with at least one bit error. Fig. 6.2 shows the simulated bit error rate (BER) and frame error rate (FER) for the case that the noise variance and the channel impulse response $\mathbf{h}_n$ are known for every n. The linear SISO equalizer presented in this chapter is compared to the BER-optimal approach using a MAP equalizer (Douillard et al., 1995). We note that the although MAP equalization is better than linear equalization after one-time equalization and decoding, the bit error rate and frame error rate when using linear equalization converges to almost the same error rates as when using MAP equalization.

These simulation results demonstrate that, after convergence of the iterative process, a turbo equalization-based receiver using a linear SISO equalizer can have performance similar to one using the optimal MAP-based SISO equalizer. We will later come to the same conclusion using EXIT chart analysis, in Sec. 8.5.

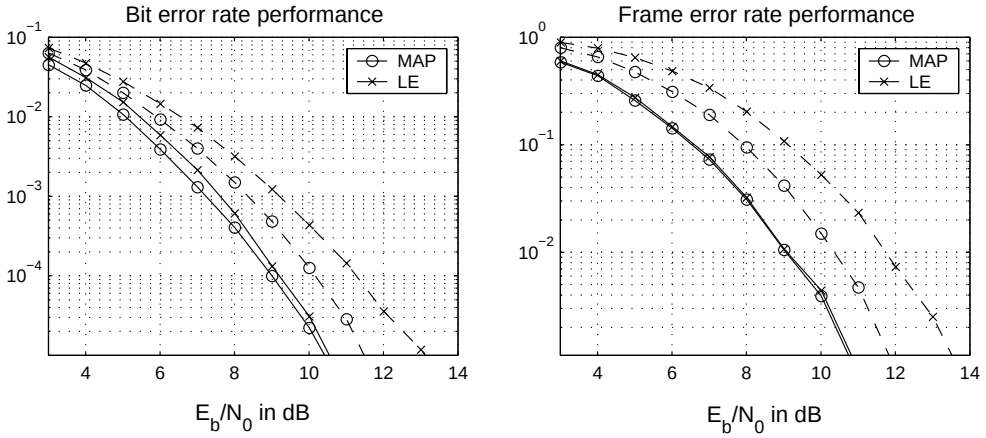


Figure 6.2 Simulated performance of turbo equalization when the channel impulse response (CIR) is time-varying and known, using a BER-optimal (MAP) SISO equalizer or low-complexity linear SISO equalizer (LE) after one-time equalization and decoding (dashed lines) and after four iterations (solid lines).

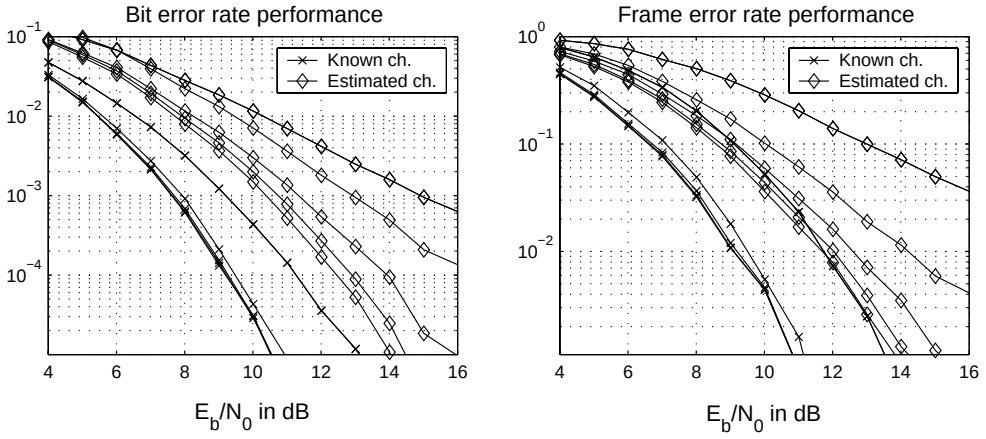


Figure 6.3 Performance of Turbo equalization after different numbers of iterations using a linear SISO equalizer when the channel impulse response (CIR) is time-varying. The noise variance and the CIR are either known or estimated using soft iterative channel estimation. All error rates are decreasing when more iterations are added, i.e., the right-most curves correspond to one-time equalization and decoding.

6.1.5 Linear SISO equalizer combined with soft iterative channel estimation

We also included soft iterative channel estimation in the simulation setup described above. In this case, the CIR $\mathbf{h}_n$ and noise variance $\sigma_{w,n}^2$ are unknown to the receiver, and are therefore replaced with time-varying estimates $\hat{\mathbf{h}}_n$ and $\hat{\sigma}_{w,n}^2$, calculated using a channel estimation algorithm. Note that the simulation results presented here are better than those published in (Otnes and Tüchler, 2002d), for reasons outlined in Sec. 7.4.1.

We use the exponentially windowed RLS algorithm (Haykin, 1996) with forgetting factor $\lambda = 0.96$ for channel estimation. The RLS algorithm is chosen rather than the LMS algorithm due to the faster convergence rate at the beginning of the block, as the initial training sequence in this setup is only 15 symbols in length. The error variance $\sigma_{w,n}^2$ is estimated using an exponentially windowed average of the squared error

$$|d_n|^2 = \left| z_n - \sum_{l=0}^{L-1} \hat{h}_{n,l} \check{x}_{n-l} \right|^2 \tag{6.33}$$

where $\check{x}_n$ are the input symbols considered by the channel estimator. For first-time equalization, the training symbols t_n and the hard decisions $\check{x}_n = \hat{x}_n = \text{sgn}(\tilde{x}_n)$ from the linear SISO equalizer are used as input to the channel estimator[1]. In subsequent iterations, the t_n and the soft symbols $\check{x}_n = \bar{x}_n = \tanh(L_d(y_n))$ fed back from the decoder are used as described in (Tüchler, Otnes and Schmidbauer, 2002) and in the next chapter.

Fig. 6.3 shows how the error rates using the linear SISO equalizer change as iterations are performed, for the case of a known channel and for the case of soft iterative channel estimation. Note that when the channel is known, the error rates do not improve noticeably after two iterations (three times equalization and decoding). When the channel is estimated, the turbo equalizer performs several dB worse than when the channel is known. However, there is more to be gained from iterations when the channel is estimated. This is because the soft symbols fed back to the channel estimator from the decoder becomes better when iterations are added, improving the quality of the channel estimate.

These simulation results demonstrate the advantage of using soft iterative channel estimation in combination with turbo equalization. In later chapters, we will investigate this matter further.

[1] We have later found that it may be a better idea to keep the channel estimate constant between the training sequences during first-time equalization, but this was not done in (Otnes and Tüchler, 2002d)

6.2 Fractionally spaced linear SISO equalizer

Until now in this chapter, the received signal has been sampled at the symbol rate. As discussed in Sec. 4.5.1.2, this is suboptimal unless some idealizing assumptions can be applied. We therefore derive the linear SISO equalizer for the case when the received signal is sampled twice per symbol period, called a fractionally spaced linear SISO equalizer.

When the received signal is sampled twice per symbol period, the two samples corresponding to one transmitted symbol will be given by (4.54):

$$z_{n,i} = \sum_{l=0}^{M-1} h_{n,l,i} x_{n-l} + w_{n,i}, \quad i \in \{0,1\} \tag{6.34}$$

The time-varying fractionally spaced channel impulse response is denoted $\mathbf{h}_{n,i} = [h_{n,0,i} \ \cdots \ h_{n,M-1,i}]^T$. We still define N as the length of the equalizer measured in symbol periods, such that the actual number of (fractionally spaced) filter taps is $2N$, and the estimate $\tilde{x}_n$ of a transmitted symbol is calculated from $2N$ received samples $z_{n,i}$ and *a priori* information on $N+M-1$ transmitted symbols x_n. We redefine the vectors $\mathbf{z}_n$ and $\mathbf{w}_n$ to have length $2N$

$$\mathbf{z}_n = [z_{n+N_1,1} \ z_{n+N_1,0} \ z_{n+N_1-1,1} \ \cdots \ z_{n-N_2,1} \ z_{n-N_2,0}]^T \tag{6.35}$$
$$\mathbf{w}_n = [w_{n+N_1,1} \ w_{n+N_1,0} \ w_{n+N_1-1,1} \ \cdots \ w_{n-N_2,1} \ w_{n-N_2,0}]^T \tag{6.36}$$

and the channel convolution matrix to have size $2N \times (N+M-1)$:

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{h}_{n+N_1,1}^T & 0 & \cdots & 0 \\ \mathbf{h}_{n+N_1,0}^T & 0 & \cdots & 0 \\ 0 \ \mathbf{h}_{n+N_1-1,1}^T & \cdots & 0 \\ 0 \ \mathbf{h}_{n+N_1-1,0}^T & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \mathbf{h}_{n-N_2,1}^T \\ 0 & \cdots & \mathbf{h}_{n-N_2,0}^T \end{bmatrix}$$

$$= \begin{bmatrix} h_{n+N_1,0,1} & h_{n+N_1,1,1} & \cdots & h_{n+N_1,M-1,1} & 0 & \cdots & 0 \\ h_{n+N_1,0,0} & h_{n+N_1,1,0} & \cdots & h_{n+N_1,M-1,0} & 0 & \cdots & 0 \\ 0 & h_{n+N_1-1,0,1} & \cdots & & h_{n+N_1-1,M-1,1} & 0 & \vdots \\ 0 & h_{n+N_1-1,0,0} & \cdots & & h_{n+N_1-1,M-1,0} & 0 & \\ \vdots & & & \ddots & & & 0 \\ 0 & \cdots & 0 & h_{n-N_2,0,1} & h_{n-N_2,1,1} & \cdots & h_{n-N_2,M-1,1} \\ 0 & \cdots & 0 & h_{n-N_2,0,0} & h_{n-N_2,1,0} & \cdots & h_{n-N_2,M-1,0} \end{bmatrix}$$

We allow the noise variance $\sigma^2_{w,n,i} = E\{w_{n,i}w^*_{n,i}\}$ to be different for every sample and assume that all noise samples are independent, such that the noise covariance matrix $\mathbf{\Sigma}_n = Cov\{\mathbf{w}_n, \mathbf{w}_n\}$ becomes

$$\mathbf{\Sigma}_n = \text{Diag}[\sigma^2_{w,n+N_1,1}\ \sigma^2_{w,n+N_1,0}\ \cdots\ \sigma^2_{w,n-N_2,1}\ \sigma^2_{w,n-N_2,0}] \tag{6.37}$$

With these redefinitions, we have

$$\mathbf{z}_n = \mathbf{H}_n\mathbf{x}_n + \mathbf{w}_n, \tag{6.38}$$

This expression is identical to (6.2), which was the basis for the derivation of the linear SISO equalizer in Sec. 6.1.1. Thus, the same derivation can be applied, and the solution for the MMSE-optimal fractionally spaced linear SISO equalizer is given by the same equations (6.14)-(6.15) as for the symbol-spaced case, except that the dimension of the matrix to be inverted is $2N \times 2N$ rather than $N \times N$.

In order to avoid computing the matrix inverse explicitly, we need to derive a time-recursive update similar to the one in Sec. 6.1.2. We still define $\mathbf{\Pi}_n = \mathbf{H}_n\mathbf{V}_n\mathbf{H}_n^H + \mathbf{\Sigma}_n$ and $\mathbf{U}_n = \mathbf{\Pi}_n^{-1}$, which are now $2N \times 2N$ matrices. We use the partitions

$$\mathbf{\Pi}_{n-1} = \left[\begin{array}{c|c|c} \mathbf{A}_n & \mathbf{G}_n & \\ \hline & & \\ \mathbf{G}_n^H & \mathbf{B}_n & \end{array}\right],\ \mathbf{\Pi}_n = \left[\begin{array}{c|c|c} \mathbf{E}_n & \mathbf{F}_n^H & \\ \hline & & \\ \mathbf{F}_n & \mathbf{A}_n & \end{array}\right] \tag{6.39}$$

$$\mathbf{U}_{n-1} = \left[\begin{array}{c|c|c} \mathbf{D}_n & \mathbf{N}_n & \\ \hline & & \\ \mathbf{N}_n^H & \mathbf{M}_n & \end{array}\right],\ \mathbf{U}_n = \left[\begin{array}{c|c|c} \mathbf{K}_n & \mathbf{L}_n^H & \\ \hline & & \\ \mathbf{L}_n & \mathbf{C}_n & \end{array}\right] \tag{6.40}$$

where $\mathbf{B}_n$, $\mathbf{E}_n$, $\mathbf{K}_n$, and $\mathbf{M}_n$ are 2×2 matrices, $\mathbf{F}_n$, $\mathbf{G}_n$, $\mathbf{L}_n$, and $\mathbf{N}_n$ are $(2N-2) \times 2$ matrices, and $\mathbf{A}_n$, $\mathbf{C}_n$, and $\mathbf{D}_n$ are $(2N-2) \times (2N-2)$ matrices. The time-recursive algorithm exploits the fact that we also now have a structured time-dependence between $\mathbf{\Pi}_{n-1}$ and $\mathbf{\Pi}_n$: They share a common submatrix $\mathbf{A}_n$, as can be verified by writing out the expression $\mathbf{\Pi}_n = \mathbf{H}_n\mathbf{V}_n\mathbf{H}_n + \mathbf{\Sigma}_n$ elementwise. We also find that the "new" parts of $\mathbf{\Pi}_n$ are

$$\mathbf{E}_n = \begin{bmatrix} \alpha_{n,0,11} & \alpha_{n,0,10} \\ \alpha^*_{n,0,10} & \alpha_{n,0,00} \end{bmatrix} \tag{6.41}$$

$$\mathbf{F}_n = \begin{bmatrix} \alpha_{n,1,11} & \alpha_{n,1,10} \\ \alpha_{n,1,01} & \alpha_{n,1,00} \\ \alpha_{n,2,11} & \alpha_{n,2,10} \\ \alpha_{n,2,01} & \alpha_{n,2,00} \\ \vdots & \vdots \\ \alpha_{n,N-1,11} & \alpha_{n,N-1,10} \\ \alpha_{n,N-1,01} & \alpha_{n,N-1,00} \end{bmatrix} \tag{6.42}$$

where

$$\alpha_{n,0,11} = \sigma^2_{w,n+N_1,1} + \sum_{l=0}^{M-1} |h_{n+N_1,l,1}|^2 v_{n+N_1-l} \tag{6.43}$$

$$\alpha_{n,0,00} = \sigma^2_{w,n+N_1,0} + \sum_{l=0}^{M-1} |h_{n+N_1,l,0}|^2 v_{n+N_1-l} \tag{6.44}$$

$$\alpha_{n,0,10} = \sum_{l=0}^{M-1} h_{n+N_1,l,1} h^*_{n+N_1,l,0} v_{n+N_1-l} \tag{6.45}$$

$$\alpha_{n,j,11} = \sum_{l=0}^{M-1-j} h_{n+N_1-j,l,1} h^*_{n+N_1,l+j,1} v_{n+N_1-l-j},\ 1 \le j \le N-1 \tag{6.46}$$

$$\alpha_{n,j,00} = \sum_{l=0}^{M-1-j} h_{n+N_1-j,l,0} h^*_{n+N_1,l+j,0} v_{n+N_1-l-j},\ 1 \le j \le N-1 \tag{6.47}$$

$$\alpha_{n,j,10} = \sum_{l=0}^{M-1-j} h_{n+N_1-j,l,1} h^*_{n+N_1,l+j,0} v_{n+N_1-l-j},\ 1 \le j \le N-1 \tag{6.48}$$

$$\alpha_{n,j,01} = \sum_{l=0}^{M-1-j} h_{n+N_1-j,l,0} h^*_{n+N_1,l+j,1} v_{n+N_1-l-j},\ 1 \le j \le N-1 \tag{6.49}$$

The time-recursive update is derived from the fact that $\mathbf{\Pi}_n \mathbf{U}_n = \mathbf{\Pi}_{n-1} \mathbf{U}_{n-1} = \mathbf{I}_{2N}$, which implies

$$\mathbf{A}_n \mathbf{D}_n + \mathbf{G}_n \mathbf{N}_n^H = \mathbf{I}_{2N-2} \tag{6.50}$$

$$\mathbf{A}_n \mathbf{N}_n + \mathbf{G}_n \mathbf{M}_n = \mathbf{0}_{2N-2} \tag{6.51}$$

$$\mathbf{E}_n \mathbf{K}_n + \mathbf{F}_n^H \mathbf{L}_n = \mathbf{I}_2 \tag{6.52}$$

$$\mathbf{F}_n \mathbf{K}_n + \mathbf{A}_n \mathbf{L}_n = \mathbf{0}_{2N-2} \tag{6.53}$$

$$\mathbf{F}_n \mathbf{L}_n^H + \mathbf{A}_n \mathbf{C}_n = \mathbf{I}_{2N-2} \tag{6.54}$$

Solving these equations for $\mathbf{C}_n$, $\mathbf{K}_n$ and $\mathbf{L}_n$ gives the algorithm to calculate $\mathbf{U}_n$ from $\mathbf{U}_{n-1}$, $\mathbf{E}_n$ and $\mathbf{F}_n$:

$$\mathbf{A}_n^{-1} = \mathbf{D}_n - \mathbf{N}_n \mathbf{M}_n^{-1} \mathbf{N}_n^H \tag{6.55}$$

$$\mathbf{\Psi}_n = \mathbf{A}_n^{-1} \mathbf{F}_n \tag{6.56}$$

$$\mathbf{K}_n = (\mathbf{E}_n - \mathbf{F}_n^H \mathbf{\Psi}_n)^{-1} \tag{6.57}$$

$$\mathbf{L}_n = -\mathbf{\Psi}_n \mathbf{K}_n \tag{6.58}$$

$$\mathbf{C}_n = \mathbf{A}_n^{-1} + \mathbf{\Psi}_n \mathbf{K}_n \mathbf{\Psi}_n^H \tag{6.59}$$

Note the similarities to the update equations (6.23)-(6.27) for the symbol-spaced case. Now, we need to do two matrix inversions in the time-recursive update. The matrices to be inverted are only size 2×2, such that the computationally most expensive tasks are still to calculate matrix products. The computational complexity will be proportional to $(2N)^2$, i.e. about 4 times larger than for the symbol-spaced case for the same value of N.

After computing $\mathbf{U}_n$ using the time-recursive update presented above, the filter $\mathbf{f}_n$ is calculated using the same expressions as for the symbol-spaced case:

$$\mathbf{g}_n = \mathbf{U}_n \mathbf{s}_n \tag{6.60}$$

$$\mathbf{f}_n = \frac{\mathbf{g}_n}{1 + (1 - v_n)\mathbf{g}_n^H \mathbf{s}_n} \tag{6.61}$$

The algorithm presented here, for fractionally spaced MMSE-optimal linear SISO equalization, has not been implemented during this doctoral work. Implementation is left for future work, as an exercise in keeping track of indices in large matrices.

6.3 Conclusions

In this chapter the MMSE-optimal linear SISO equalizer has been derived for the case of a time-varying channel. Simulations show that also in this case turbo equalization using an MMSE-optimal linear SISO equalizer can converge to error rates close to what is achieved by using a BER-optimal MAP equalizer. The computational complexity of calculating the time-varying equalizer filter is proportional to N^2 per symbol interval, where N is the length of the filter.

When the channel impulse response and the noise variance are known, the turbo equalizer converges to error rates close to the optimum after two iterations (three times equalization and decoding) in the simulation setup used in this chapter. When the channel impulse response and the noise variance are estimated, and the channel estimate is improved for each iteration using soft iterative channel estimation, the error rates can still be improved after the third iteration (see Fig. 6.3).

We have also shown how the MMSE-optimal linear SISO equalizer can be applied when the received signal is sampled twice per symbol interval rather than at the symbol rate. This causes the computational complexity to increase by about a factor of 4.

Chapter 7

Algorithms for soft iterative channel estimation

In this chapter, we compare different channel estimation algorithms for soft iterative channel estimation, using *a priori* means of the transmitted symbols rather than the transmitted symbols themselves as the input ("training") signal. We consider four different algorithms: The LMS algorithm, the RLS algorithm, a new modified RLS algorithm using a novel cost function, and a Kalman-based channel algorithm proposed in (Song et al., 2002). This chapter is mostly based on a paper presented at IEEE International Conference on Communication Systems in November, 2002 (Otnes and Tuchler, 2002e).

7.1 Introduction

In a receiver employing turbo equalization for an initially unknown time-varying frequency-selective channel, the quality of the channel estimate can be improved for each iteration by using soft information fed back from the previous iteration as input to the channel estimator. An introduction and references for this approach, which we call iterative channel estimation, equalization, and decoding, was given in Sec. 5.4.

The setup for iterative channel estimation, equalization, and decoding is shown in Fig. 7.1, which is a combination of Figs. 5.1, 5.6, and 5.7(a). The soft information fed back from the decoder is used to improve the channel estimate in subsequent iterations, known as soft iterative channel estimation.

In classical theory on channel estimation (system identification) (Haykin, 1996), the transmitted symbols are either considered completely known to the receiver (or estimated using hard decisions), or completely unknown (blind channel estimation). In soft iterative channel estimation, the mean and variance of each transmitted symbol (fed back from the previous iteration), rather than the exact symbol itself, is known to

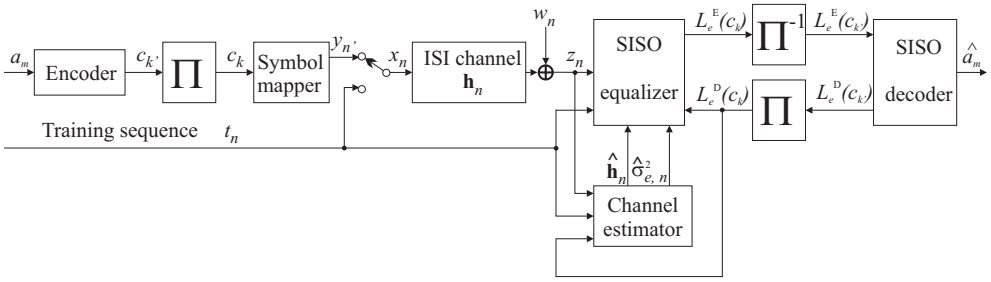


Figure 7.1 Block diagram of turbo equalization for a time-varying channel: Iterative channel estimation, equalization, and decoding.

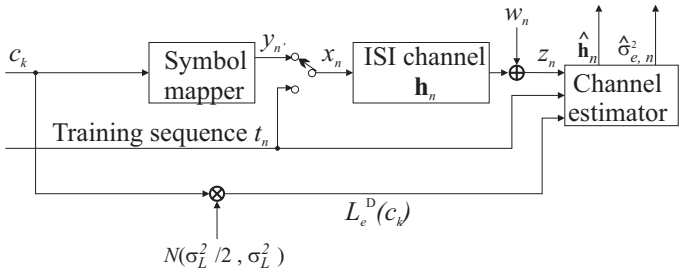


Figure 7.2 Simplified system model used to investigate soft iterative channel estimation.

the channel estimator. Therefore, channel estimation theory needs to be reexamined for this case.

The most common algorithms to estimate and track a time-varying CIR are the least mean square (LMS) algorithm and the (exponentially windowed) recursive least squares (RLS) algorithm (Haykin, 1996). In this chapter, we also introduce a new modified RLS algorithm and investigate a Kalman-based algorithm proposed in (Song et al., 2002). The latter two algorithms take into account the statistics of the soft input symbols. We use simulations to compare these four algorithms with respect to their performance in soft iterative channel estimation.

7.2 System model

In this chapter we concentrate on channel estimation algorithms applied to turbo equalization, and consider the simplified system model shown in Fig. 7.2 where all blocks which are not important for the problem of channel estimation are omitted.

We model the interleaved code bits $c_k \in \{+1, -1\}$ as independent Bernoulli trials, i.e., as far as estimation is concerned, the redundancy contained in the c_k is

neglected. Symbols $y_{n'}$ from the 2^Q-ary signal constellation $\mathcal{S}$ are generated by mapping Q consecutive c_k onto each data symbol $y_{n'}$. Sequences of known training symbols $t_n \in \mathcal{S}$ are multiplexed into this symbol stream as described in Sec. 5.1. Thus, transmitted is the symbol x_n, which is either a training symbol t_n or a data symbol $y_{n'}$. We assume a coherent symbol-spaced receiver front end and precise knowledge of the system timing. We also assume that the overall CIR, including transmitter and receiver filters, can be modelled as a symbol-spaced time-varying linear filter $\mathbf{h}_n = [h_{n,0} \cdots h_{n,M-1}]^T$ of length M. In this chapter, we define $\mathbf{x}_n$ to be a length M vector (in contrast to length $N + M - 1$ in the other chapters of the dissertation):

$$\mathbf{x}_n = [x_n \ \cdots \ x_{n-M+1}]^T \tag{7.1}$$

Received are the symbols

$$z_n = \sum_{i=0}^{M-1} h_{n,i} x_{n-i} + w_n = \mathbf{h}_n^T \mathbf{x}_n + w_n \tag{7.2}$$

where the w_n are circularly symmetric complex zero-mean white Gaussian noise samples with variance $E\{w_n w_n^*\} = \sigma_w^2$. Note that $\mathbf{h}_n$ is transposed to $\mathbf{h}_n^T$, and not conjugate transposed to $\mathbf{h}_n^H$, in (7.2). This is to be consistent with the rest of the dissertation, but is inconsistent with other literature on channel estimation, e.g. (Haykin, 1996). For this reason, complex conjugates are seen in some "odd-looking" places throughout this chapter, e.g. in (7.10).

In the iterative receiver of Fig. 7.1, soft information on the c_k is fed back from the decoder from the previous iteration. This information is usually a log-likelihood ratio (LLR) $L_e^D(c_k) = \ln[\Pr(c_k = +1)/\Pr(c_k = -1)]$ on the code bit. When we use the simplified system model of Fig. 7.2, the iterative receiver is left out, and the LLRs $L_e^D(c_k)$ fed back from the decoder are rather modelled as independent real Gaussian random variables with average value $c_k \cdot \sigma_L^2/2$ and variance σ_L^2. This is a common model for the LLRs produced by a decoder during iterative decoding (ten Brink, 2001), and the actual distribution of LLRs computed directly from bits transmitted over a simple AWGN channel. We note that $\sigma_L = 0$ corresponds to poor soft information, all $L_e^D(c_k)$ are 0, and an increasing σ_L corresponds to more and more reliable soft information (see also Fig. 8.2 in Sec. 8.4).

From the code bit LLRs $L_e^D(c_k)$, the mean $\bar{y}_{n'} = E\{y_{n'}\}$ and the variance $v_{y_{n'}} = E\{y_{n'} y_{n'}^*\} - \bar{y}_n \bar{y}_n^*$ of each data symbol $y_{n'}$ can be calculated using equations (5.17)-(5.18). Using the statistics $\bar{y}_{n'}$ and $v_{y_{n'}}$, the transmitted symbols x_n can be written

$$x_n = \bar{x}_n + \chi_n \tag{7.3}$$

where χ_n is a discrete-valued noise variable with variance $E\{\chi_n \chi_n^*\} = v_n$ and zero mean. We have $\bar{x}_n = \bar{y}_{n'}$ and $v_n = v_{y_{n'}}$ when data symbols have been transmitted,

and $\bar{x}_n = t_n$ and $v_n = 0$ when training symbols have been transmitted. We assume that $E\{\chi_n \chi_k^*\} = 0$ for $k \neq n$, which is plausible because an interleaver has been used in the turbo equalization setup. It follows that the covariance matrix of $\mathbf{x}_n$ given the LLRs $L_e^D(c_k)$ is a diagonal matrix of the form $\mathbf{V}_n = \text{Diag}[v_n \ \cdots \ v_{n-M+1}]$. We define the length M vectors $\bar{\mathbf{x}}_n$ and $\boldsymbol{\chi}_n$ with indices similar to $\mathbf{x}_n$. We will then have

$$E\{\boldsymbol{\chi}_n \boldsymbol{\chi}_n^H\} = \mathbf{V}_n \tag{7.4}$$

We also consider the case of hard iterative channel estimation. The channel estimator uses in this case the symbol estimates $\hat{x}_n$, which are obtained by mapping the hard decisions $\hat{c}_k = \text{sgn}[L_e^D(c_k)]$ onto the signal constellation $\mathcal{S}$. As for soft iterative channel estimation, we define the length M vector $\hat{\mathbf{x}}_n$ with indices similar to $\mathbf{x}_n$.

The channel estimator provides a time-varying channel estimate $\hat{\mathbf{h}}_n$ at each time step n. Three different error signals, measured at time η using the channel estimate at time n, are defined:

$$e_{\eta,n} = z_\eta - \hat{\mathbf{h}}_n^T \mathbf{x}_\eta \tag{7.5}$$

$$d_{\eta,n} = z_\eta - \hat{\mathbf{h}}_n^T \bar{\mathbf{x}}_\eta \tag{7.6}$$

$$d'_{\eta,n} = z_\eta - \hat{\mathbf{h}}_n^T \hat{\mathbf{x}}_\eta \tag{7.7}$$

We also use the shorthand notations $e_n = e_{n,n-1}$, $d_n = d_{n,n-1}$, and $d'_n = d'_{n,n-1}$. The ultimate goal of any channel estimation algorithm is to minimize the variance of each e_n, but this is complicated by the fact that $\mathbf{x}_n$ is not perfectly known when data symbols have been transmitted. Therefore, a common approach is to minimize the variance of d_n (soft feedback) or d'_n (hard feedback) instead, since $\bar{\mathbf{x}}_n$ and $\hat{\mathbf{x}}_n$ are available.

7.3 Channel estimation algorithms

7.3.1 LMS channel estimation

The simplest channel estimation algorithm is the LMS algorithm, which is a suboptimal version of the stochastic gradient (SG) algorithm (Haykin, 1996). For the case that all x_n are known, the LMS algorithm computes

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + (1-\lambda) e_n \mathbf{x}_n^* \tag{7.8}$$

where the parameter $\beta = 1 - \lambda$ is called the "step size", and governs the convergence speed and stability of the algorithm. Our motivation for using λ rather than β as a parameter here will be apparent in Sec. 7.3.4.

7.3.2 RLS channel estimation

The RLS algorithm has been extensively studied in the literature, and several implementations with different complexity and numerical properties exist. First, we assume that all the x_n are known such that the error signal $e_{\eta,n}$ can be used. The cost function minimized by the RLS algorithm, an exponentially windowed sum of squared error, is in this case:

$$\varphi_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} e_{\eta,n} e_{\eta,n}^* \tag{7.9}$$

The vector $\hat{\mathbf{h}}_n$ minimizing φ_n is given by (Haykin, 1996)

$$\hat{\mathbf{h}}_n = \left(\boldsymbol{\Phi}_n^{-1}\boldsymbol{\theta}_n\right)^* \tag{7.10}$$

where

$$\boldsymbol{\Phi}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} \mathbf{x}_\eta \mathbf{x}_\eta^H = \lambda \boldsymbol{\Phi}_{n-1} + \mathbf{x}_n \mathbf{x}_n^H \tag{7.11}$$

$$\boldsymbol{\theta}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} \mathbf{x}_\eta z_\eta^* = \lambda \boldsymbol{\theta}_{n-1} + \mathbf{x}_n z_n^* \tag{7.12}$$

The equation to compute $\hat{\mathbf{h}}_n$ can be rewritten to a recursive formula yielding a computationally efficient version of the RLS algorithm (Haykin, 1996). The recursion is based on the fact that (7.11) is a so-called rank-1 update, because the matrix $\mathbf{x}_n \mathbf{x}_n^H$ has rank 1.

A straightforward extension of the LMS and RLS algorithms to the case of soft feedback is to replace e_n with d_n and $\mathbf{x}_n$ with $\bar{\mathbf{x}}_n$. Hard feedback can similarly be applied by replacing e_n with d'_n and $\mathbf{x}_n$ with $\hat{\mathbf{x}}_n$. The performance of RLS channel estimation using soft feedback has been analyzed in (Tüchler, Otnes and Schmidbauer, 2002).

For the case of known data symbols or using training symbols, only, it is known that the RLS algorithm performs only slightly better than the LMS algorithm when used to estimate and track a time-varying channel (Eleftheriou and Falconer, 1986; Bershad et al., 1990). Besides that, the major benefit of the RLS algorithm is that it converges faster than the LMS algorithm, i.e., it produces a good channel estimate $\hat{\mathbf{h}}_n$ in less time steps.

7.3.3 Modified RLS channel estimation

When soft symbols are used as input to the ordinary RLS algorithm, the exponentially windowed sum of the squared error $d_{\eta,n} d_{\eta,n}^*$ is minimized. This is obviously

suboptimal for $v_n \neq 0$, since the observed noise variance when the real CIR $\mathbf{h}_n$ is replaced with $\hat{\mathbf{h}}_{n-1}$ is $E\{e_n e_n^*\}$ and not $E\{d_n d_n^*\}$, and we are thus minimizing the "wrong" error variance.

We define a novel cost function to address this problem. First, we note that the cost function (7.9) for the ordinary RLS algorithm, when all x_n are known, can be written as

$$\varphi_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} E\{e_{\eta,n} e_{\eta,n}^* | z_\eta, \mathbf{x}_\eta; \hat{\mathbf{h}}_n\} \tag{7.13}$$

because the known squared error $e_{\eta,n} e_{\eta,n}^*$ can be written as the expectation of the squared error conditioned on all known values in the expression for $e_{\eta,n}$.

Our new idea for soft feedback is to condition the expectation of the squared error on values that are actually known (fed back from the decoder). The cost function is then given by

$$\psi_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} E\{e_{\eta,n} e_{\eta,n}^* | z_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta; \hat{\mathbf{h}}_n\} \tag{7.14}$$

From (7.5) we find that

$$\begin{aligned} &E\{e_{\eta,n} e_{\eta,n}^* | z_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta; \hat{\mathbf{h}}_n\} \\ &= z_\eta z_\eta^* + \hat{\mathbf{h}}_n^T \mathbf{U}_\eta \hat{\mathbf{h}}_n^* - 2\Re[\hat{\mathbf{h}}_n^T (z_\eta^* \bar{\mathbf{x}}_\eta + E\{\boldsymbol{\chi}_\eta z_\eta^* | z_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta\})] \end{aligned} \tag{7.15}$$

where $\mathbf{U}_\eta$ is the correlation matrix

$$\mathbf{U}_\eta = E\{\mathbf{x}_\eta \mathbf{x}_\eta^H\} = \bar{\mathbf{x}}_\eta \bar{\mathbf{x}}_\eta^H + \mathbf{V}_\eta \tag{7.16}$$

To calculate $E\{\boldsymbol{\chi}_\eta z_\eta^*\}$, we assume for a moment that $\mathbf{h}_\eta$ is known in the underlying system model (7.2). Then we can condition the expectation on $\mathbf{h}_\eta$ rather than on z_η and obtain

$$E\{\boldsymbol{\chi}_\eta z_\eta^* | z_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta\}) = E\{\boldsymbol{\chi}_\eta z_\eta^* | \mathbf{h}_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta\}) = \mathbf{V}_\eta \mathbf{h}_\eta^* \tag{7.17}$$

Of course, $\mathbf{h}_\eta$ is not known, but it is estimated by the estimation algorithm. Therefore, we introduce the approximation $\mathbf{h}_\eta \approx \hat{\mathbf{h}}_{\eta-1}$ (in this expression only) to be able to evaluate $E\{\boldsymbol{\chi}_\eta z_\eta^* | z_\eta, \bar{\mathbf{x}}_\eta, \mathbf{V}_\eta\})$. Inserting (7.17) into (7.15) then gives

$$E\{e_{\eta,n} e_{\eta,n}^* | z_\eta, \hat{\mathbf{x}}_\eta, \mathbf{V}_\eta; \hat{\mathbf{h}}_n\} \approx z_\eta z_\eta^* + \hat{\mathbf{h}}_n^T \mathbf{U}_\eta \hat{\mathbf{h}}_n^* - 2\Re[\hat{\mathbf{h}}_n^T (z_\eta^* \bar{\mathbf{x}}_\eta + \mathbf{V}_\eta \hat{\mathbf{h}}_{\eta-1}^*)] \tag{7.18}$$

Following the derivation for the case of known data symbols as in (Haykin, 1996), we find that the channel estimate minimizing the cost function ψ_n, given by equations (7.14) and (7.18), can still be written as

$$\hat{\mathbf{h}}_n = (\boldsymbol{\Phi}_n^{-1} \boldsymbol{\theta}_n)^* \tag{7.19}$$

where we now have

$$\mathbf{\Phi}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} \mathbf{U}_\eta = \lambda \mathbf{\Phi}_{n-1} + \mathbf{U}_n \tag{7.20}$$

$$\boldsymbol{\theta}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta} (\bar{\mathbf{x}}_\eta z_\eta^* + \mathbf{V}_\eta \hat{\mathbf{h}}_{\eta-1}^*) = \lambda \boldsymbol{\theta}_{n-1} + \mathbf{U}_n \hat{\mathbf{h}}_{n-1}^* + d_n^* \bar{\mathbf{x}}_n \tag{7.21}$$

The matrix $\mathbf{U}_n$ has full rank, in contrast to the rank-1 matrix $\mathbf{x}_n \mathbf{x}_n^H$ used in the ordinary RLS update. Therefore, there remains a challenge to derive an efficient update rule to calculate $\mathbf{\Phi}_n^{-1}$ from $\mathbf{\Phi}_{n-1}^{-1}$ for the modified RLS algorithm, since a straightforward implementation has a computational complexity proportional to M^3.

7.3.4 Approximated modified RLS channel estimation

The modified RLS algorithm can be simplified by ignoring the off-diagonal terms in the correlation matrix $\mathbf{U}_\eta$. This approximation is based on the fact that when the transmitted symbols are uncorrelated, the off-diagonal terms are on average zero and finally do not contribute anything to $\mathbf{\Phi}_n$, which is an exponentially windowed average of different $\mathbf{U}_\eta$. This approximation becomes less plausible when λ is much smaller than 1.

When the off-diagonal terms of $\mathbf{U}_\eta$ and $\mathbf{\Phi}_n$ are ignored, computing $\mathbf{\Phi}_n^{-1}$ becomes trivial because the matrix to be inverted is diagonal. If the signal constellation $\mathcal{S}$ is a PSK constellation, we furthermore have $E\{x_\eta x_\eta^*\} = 1$ regardless of the LLRs $L_e^D(c_k)$, such that $\mathbf{U}_\eta \approx \mathbf{I}_M$ after removing the off-diagonal terms. In this case, we find that

$$\mathbf{\Phi}_n \approx \sum_{\eta=1}^{n} \lambda^{n-\eta} \mathbf{I}_M \approx \frac{1}{1-\lambda} \mathbf{I}_M \tag{7.22}$$

and using equations (7.19) and (7.21), the estimated CIR becomes

$$\hat{\mathbf{h}}_n = (1-\lambda)\boldsymbol{\theta}_n^* = (1-\lambda)(\lambda \boldsymbol{\theta}_{n-1}^* + \hat{\mathbf{h}}_{n-1} + d_n \bar{\mathbf{x}}_n^*) = \hat{\mathbf{h}}_{n-1} + (1-\lambda) d_n \bar{\mathbf{x}}_n^* \tag{7.23}$$

Thus, the modified RLS algorithm is reduced to the ordinary LMS algorithm (7.8) using soft input symbols. Note that for non-PSK signal constellations, ignoring the off-diagonal terms of $\mathbf{U}_\eta$ does not reduce the modified RLS algorithm to the LMS algorithm, but still simplifies the algorithm tremendously such that the computational complexity per symbol interval is proportional to M.

7.3.5 Kalman-based channel estimation

A soft-input channel estimation algorithm based on Kalman filtering has been proposed by Song et al. (2002). We wish to determine the relationship between this algorithm and RLS channel estimation algorithms.

Kalman filters are based on a state-space model of the channel variation. We assume the state-space model of (Haykin, 1996, Sec. 6.9), which can be written *in the notation of (Song et al., 2002)*:

$$\mathbf{F} = \mathbf{I} \tag{7.24}$$

$$\mathbf{G}\mathbf{Q}_v(n)\mathbf{G}^H = q\mathbf{P}(n|n) \tag{7.25}$$

$$\mathbf{P}(n|n) = [\mathbf{I} - \mathbf{k}(n)\bar{\mathbf{b}}^H(n)]\mathbf{P}(n|n-1) \tag{7.26}$$

where q is a scaling factor.

When this state-space model is assumed, we can use the relationship between Kalman filtering and the RLS algorithm described in (Haykin, 1996, page 407). Careful calculations and comparisons, not included here, can then be applied to show that the channel estimate computed by the Kalman-based algorithm of (Song et al., 2002) is equivalently given by

$$\mathbf{\Phi}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta}\bar{\mathbf{x}}_\eta\bar{\mathbf{x}}_\eta^H\alpha_\eta^{-1} = \lambda\mathbf{\Phi}_{n-1} + \bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H\alpha_n^{-1} \tag{7.27}$$

$$\boldsymbol{\theta}_n = \sum_{\eta=1}^{n} \lambda^{n-\eta}\bar{\mathbf{x}}_\eta z_\eta^*\alpha_\eta^{-1} = \lambda\boldsymbol{\theta}_{n-1} + \bar{\mathbf{x}}_n z_n^*\alpha_n^{-1} \tag{7.28}$$

$$\hat{\mathbf{h}}_n = (\mathbf{\Phi}_n^{-1}\boldsymbol{\theta}_n)^* \tag{7.29}$$

where α_n is a time-varying scaling factor depending on the *a priori* information:

$$\alpha_n = 1 + \frac{\hat{\mathbf{h}}_{n-1}^T\mathbf{V}_n\hat{\mathbf{h}}_{n-1}^*}{\sigma_w^2} + \frac{(\mathrm{Diag}[\mathbf{V}_n])^T\mathrm{Diag}[\mathbf{\Phi}_{n-1}^{-1}]}{\lambda} \tag{7.30}$$

Now, the update from $\mathbf{\Phi}_{n-1}$ to $\mathbf{\Phi}_n$ is a rank-1 update, such that $\mathbf{\Phi}_n^{-1}$ can be computed from $\mathbf{\Phi}_{n-1}^{-1}$ with the same computational complexity as in the ordinary RLS algorithm. The only problem is that knowledge of σ_w^2 is required for the calculation of α_n. In our implementation of this algorithm, we estimate σ_w^2 during the initial training sequence, and replace σ_w^2 in (7.30) with this estimate for the remainder of the block.

Note that the Kalman-filter based channel estimator of (Song et al., 2002) would lead to other update equations if a state space-model different from (7.24)-(7.26) is assumed. Also, all RLS channel estimation algorithms can be viewed as "Kalman-based", assuming this state-space model. In this chapter, however, we use the term *Kalman-based channel estimator* to describe the algorithm given by (7.27)-(7.30), to distinguish it from the other algorithms.

7.3.6 Relationship between soft-input channel estimation algorithms

We have now described four different RLS-like channel algorithms using soft input symbols $\bar{x}_n$: The ordinary RLS algorithm, the modified RLS algorithm, the approximated modified RLS algorithm (which is reduced to LMS when the signal constellation is PSK), and the Kalman-based algorithm derived from (Song et al., 2002).

All those algorithms compute the channel estimate as $\hat{\mathbf{h}}_n = (\mathbf{\Phi}_n^{-1}\boldsymbol{\theta}_n)^*$, where $\mathbf{\Phi}_n$ and $\boldsymbol{\theta}_n$ are time-varying estimates of the correlations $\mathbf{R}_n$ and $\mathbf{r}_n$, respectively, defined as

$$\mathbf{R}_n = E\{\mathbf{x}_n\mathbf{x}_n^H\} \tag{7.31}$$

$$\mathbf{r}_n = E\{\mathbf{x}_n z_n^*\} \tag{7.32}$$

The differences between the algorithms are summarized in Table 7.1, and lies in how $\mathbf{\Phi}_n$ and $\boldsymbol{\theta}_n$ are updated from one symbol interval to the next, when soft symbols $\bar{x}_n$ rather than known symbols x_n are used as input signal.

When the transmitted symbols are uncertain such that $\bar{\mathbf{x}}_n \neq \mathbf{x}_n$, the ordinary RLS algorithm will calculate wrong estimates of $\mathbf{R}_n$ and $\mathbf{r}_n$. The other algorithms seek to compensate for this by weighting the terms $\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H$ and $\bar{\mathbf{x}}_n z_n^*$ lower when the uncertainty, represented by the diagonal terms of $\mathbf{V}_n$, is large. The modified RLS algorithm performs the compensation by adding terms depending on $\mathbf{V}_n$, such that $\mathbf{\Phi}_n$ and $\boldsymbol{\theta}_n$ still approach the true values of $\mathbf{R}_n$ and $\mathbf{r}_n$. The Kalman-based algorithm performs the compensation by downscaling the uncertain terms with a factor $\alpha_n \geq 1$ depending on $\mathbf{V}_n$, such that the old covariance estimates $\mathbf{\Phi}_{n-1}$ and $\boldsymbol{\theta}_{n-1}$ are weighted higher compared to the new terms.

The benefits of the Kalman-based algorithm compared to the modified RLS algorithm is that the computational complexity per symbol interval is proportional to M^2 rather than M^3, but the Kalman-based algorithm as presented here has one major weakness: When the *a priori* information is poor, i.e. all $\bar{x}_n \approx 0$, $\mathbf{\Phi}_n$ as well as $\boldsymbol{\theta}_n$ will decay exponentially with time n. The estimate $\hat{\mathbf{h}}_n = (\mathbf{\Phi}_n^{-1}\boldsymbol{\theta}_n)^*$ will still be "correct" because both factors decay equally fast, but the numerical properties

	$\mathbf{\Phi}_n =$	$\boldsymbol{\theta}_n =$	Complexity
RLS	$\lambda\mathbf{\Phi}_{n-1}+\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H$	$\lambda\boldsymbol{\theta}_{n-1}+\bar{\mathbf{x}}_n z_n^*$	$O(M^2)$
Modified RLS	$\lambda\mathbf{\Phi}_{n-1}+\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H+\mathbf{V}_n$	$\lambda\boldsymbol{\theta}_{n-1}+\bar{\mathbf{x}}_n z_n^*+\mathbf{V}_n\mathbf{h}_{n-1}^*$	$O(M^3)$
App.mod. RLS	$\lambda\mathbf{\Phi}_{n-1}+\text{Diag'}[\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H]+\mathbf{V}_n$	$\lambda\boldsymbol{\theta}_{n-1}+\bar{\mathbf{x}}_n z_n^*+\mathbf{V}_n\mathbf{h}_{n-1}^*$	$O(M)$
Kalman	$\lambda\mathbf{\Phi}_{n-1}+\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^H\alpha_n^{-1}$	$\lambda\boldsymbol{\theta}_{n-1}+\bar{\mathbf{x}}_n z_n^*\alpha_n^{-1}$	$O(M^2)$

Table 7.1 Update equations for $\mathbf{\Phi}_n$ and θ_n in different soft-input channel estimation algorithms, and the corresponding computational complexity per symbol interval. Here, the operator Diag'$[\mathbf{X}]$ denotes the matrix $\mathbf{X}$ with all off-diagonal elements set equal to zero.

may be poor after some symbol intervals: $\boldsymbol{\theta}_n$ will become very small, and $\boldsymbol{\Phi}_n^{-1}$ will become very large.

7.4 Using the channel estimate

Before applying the channel estimate to an equalization algorithm, it is desirable to investigate the correlation between the error signal $e_{\eta,n}$ and the transmitted signal x_n. Most equalization algorithms assume that the error (noise) signal is uncorrelated with the transmitted symbols. First, we note that for all channel estimation algorithms considered here, $\hat{\mathbf{h}}_n$ is a function of x_i (through $\bar{x}_i$) only for $i \leq n$. Assuming that the channel estimate is unbiased, $E\{\hat{\mathbf{h}}_n\} = \mathbf{h}_n$, and that the channel is varying slowly such that $\mathbf{h}_\eta \approx \mathbf{h}_n$ for small $n - \eta$, we find that $E\{(\mathbf{h}_\eta - \hat{\mathbf{h}}_n)^T x_i\} = 0$ for $i > n$ but not necessarily for $i \leq n$. The error $e_{\eta,n}$ can be written as

$$e_{\eta,n} = (\mathbf{h}_\eta - \hat{\mathbf{h}}_n)^T \mathbf{x}_\eta + w_\eta \tag{7.33}$$

and, using the assumptions above, we find that we must have $i > n$ or $\eta > n + M - 1$ in order to ensure that $E\{e_{\eta,n} x_i^* = 0\}$. For example, if the equalizer requires the error signal e_n to be uncorrelated with x_n, but not necessarily with x_{n-1}, a proper choice is the channel model

$$z_n = \hat{\mathbf{h}}_{n-1}^T \mathbf{x}_n + e_n \tag{7.34}$$

where $e_n = e_{n,n-1}$ as previously defined. This is the error signal studied in the simulations in Sec. 7.5. A point to note from the simulations is that the error signal $e_{n,n}$, which might seem more natural to use, can have a variance much smaller than σ_w^2 due to the correlation with x_n. Thus, the uncorrelated part of the error signal is dominated by the correlated part, which is hard to handle properly for an equalizer. This effect is not observed for the error signal $e_{n,n-1}$.

If we wish to ensure zero correlation between the error signal and the x_i for all $i > n - N + 1$, we must use the channel model

$$z_n = \hat{\mathbf{h}}_{n-N}^T \mathbf{x}_n + e_{n,n-N} \tag{7.35}$$

However, the benefit of this approach must be weighed against the increase in error variance caused by delaying the channel estimate.

7.4.1 Notes on simulation results presented in our papers

The importance of avoiding correlation between $\hat{\mathbf{h}}_n$ and x_n, or equivalently between $\hat{\mathbf{h}}_n$ and $\bar{x}_n$, is illustrated by some unfortunate errors in the simulation results presented in the published versions of the papers on which Chaps. 6 and 9 are based. In (Otnes and Tüchler, 2002d) we used the channel estimate $\hat{\mathbf{h}}_n$ directly rather than

delaying it and use $\hat{\mathbf{h}}_{n-1}$. When we discovered that it was wiser to use $\hat{\mathbf{h}}_{n-1}$, the simulated performance got even better than in the original paper, and a one-page correction was handed out to those interested at the conference.

For (Otnes and Tüchler, 2002c), we had time to make the correction before the final version of the paper was submitted, and the simulated performance was improved compared to the draft version of the paper. However, by a mistake we performed the correction in the wrong direction: We used $\hat{\mathbf{h}}_{n+1}$ rather than $\hat{\mathbf{h}}_{n-1}$. The correlation between $\hat{\mathbf{h}}_{n+1}$ and $\bar{x}_n$ is smaller than between $\hat{\mathbf{h}}_n$ and $\bar{x}_n$, but it is not zero (the correlation between $\hat{\mathbf{h}}_{n-1}$ and $\bar{x}_n$ is zero as discussed above). The mistake was not discovered before the final weeks of writing this dissertation, and we then saw significant improvements by correcting the error. Actually, in Fig. 9.4 (right), the error floor was decreased from 10^{-4} to being unobservable.

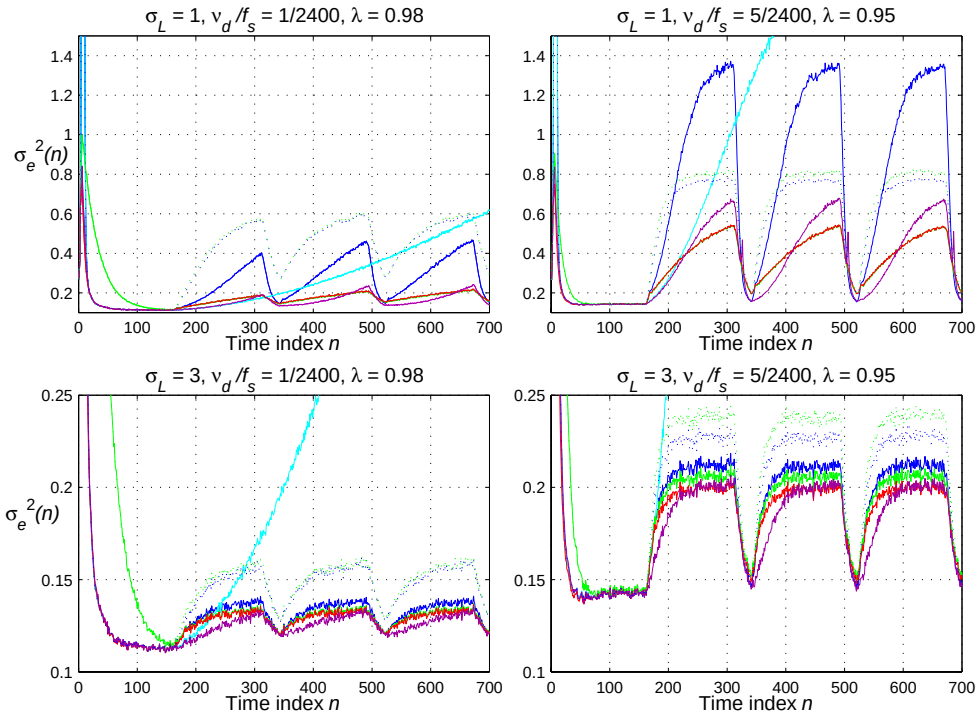
The reason why it is important that the channel estimate used to compute $\tilde{x}_n$ is as uncorrelated with $\bar{x}_n$ as possible, is that no correlation at all should exist between $\bar{x}_n$ and $\tilde{x}_n$ according to the turbo principle. And we have actually learnt from our errors that the performance does improve as the correlation decreases.

7.5 Simulation results

The channel estimation algorithms discussed in this chapter have been compared through simulations to give an indication of their relative performance. Analytical results are left for further study.

The following simulation setup has been used. Data symbols y_n from a Gray-coded 8-PSK constellation $\mathcal{S}$ have been generated randomly as discussed in Sec. 7.2. Random training symbols t_n have been generated in the same manner. A frame of symbols consists of 160 initial training symbols followed by three sequences of 150 data symbols and 30 training symbols. The energy in each transmitted symbol is $x_n x_n^* = E_s = 1$. The signalling rate (channel symbols per second) is denoted f_s. The symbols x_n are transmitted over a time-varying ISI channel of length $M' = 6$. Each of the 6 taps $h_{n,i}$ suffers from independent Rayleigh fading (circularly symmetric complex Gaussian distribution), with a Gaussian fading spectrum. The fading rate ν_d is defined as the 2σ value of the fading spectrum. We normalize $\mathbf{h}_n$ such that $E_h = E\{\mathbf{h}_n^H \mathbf{h}_n\} = 1$. Complex Gaussian noise with variance $\sigma_w^2 = N_0$ is added to the received symbols.

The LLRs $L_e^D(c_k)$, fed back from the decoder, are modelled as independent real Gaussian random variables with average value $c_k \cdot \sigma_L^2/2$ and variance σ_L^2, and used to calculate the *a priori* statistics $\bar{x}_n$ and v_n of each symbol, as discussed in Sec. 7.2. Each channel estimation algorithm has been used to generate a time-varying channel estimate $\hat{\mathbf{h}}_n$ of length $M = M' + 2 = 8$ (to account for the fact that M' is not known in reality) and the error signal e_n has been recorded.



Dotted dark blue:	Ordinary RLS, hard feedback
Solid dark blue:	Ordinary RLS, soft feedback
Dotted green:	LMS, hard feedback
Solid green:	LMS, soft feedback (= approximated modified RLS)
Solid red:	Modified RLS, soft feedback
Solid magenta:	Kalman-based, soft feedback
Solid light blue:	Constant channel estimate after initial training sequence

Figure 7.3 Simulated error variance σ_e^2 at each symbol interval n in a frame of training and data symbols. Simulation setup: $E_s/N_0 = 10$dB; σ_L, ν_d/f_s and λ as shown above the plots.

In total 10000 frames have been simulated, with $\mathbf{h}_n$, t_n, c_k, and $L_e^D(c_k)$ being randomly different for each frame (with the statistical properties described above). This has been used to estimate the (ensemble) error variance $\sigma_e^2(n) = E\{e_n e_n^*\}$ at each symbol interval, averaged over all frames. Simulations have been performed with different combinations of E_s/N_0 (signal to noise ratio), σ_L (quality of information fed back from decoder), ν_d/f_s (fading rate relative to signalling rate), and λ (forgetting factor).

In the following, a few of the simulation results are presented. Fig. 7.3 shows examples on how σ_e^2 varies throughout the frame when using different algorithms, for some representative combinations of simulation parameters. The variance σ_e^2 is

decreasing with time when known symbols are used for estimation, and increasing when (soft or hard) data symbols fed back from the decoder are used. We have also shown the effect of keeping the channel estimate constant after the initial training sequence, to illustrate the rate of the channel variations relative to the frame pattern. We make the following observations from these plots and other simulation results not shown here:

- The LMS algorithm converges slower than the other algorithms during the initial training sequence, but certainly deserves comparison during the remainder of the block.

- At $\sigma_L = 1$, the performance of the modified RLS algorithm is indistinguishable from the performance of the much simpler LMS algorithm using soft feedback.

- At $\sigma_L = 3$, the modified RLS algorithm performs only slightly better than the LMS algorithm using soft feedback.

- The LMS algorithm using soft feedback is generally a better choice than the ordinary RLS algorithm using soft or hard feedback.

- The Kalman-based algorithm always performs better than the other algorithms at the beginning of each data block, but does sometimes converge towards an error variance larger than the soft feedback LMS and modified RLS algorithms at the end of each data block. This is particularly seen at $\sigma_L = 1$.

- In some cases, keeping the channel estimate constant while data symbols are transmitted is a good alternative to hard or soft iterative channel estimation. This is particularly the case for relatively low fading rates and/or small σ_L (upper left plot in Fig. 7.3).

- For the LMS algorithm, soft feedback is always better than hard feedback.

- For the ordinary RLS algorithm, soft feedback is better than hard feedback for some combinations of channel parameters, while hard feedback is better for other parameter combinations (in particular, at high fading rates and/or small σ_L). This phenomenon is explained in (Tüchler, Otnes and Schmidbauer, 2002).

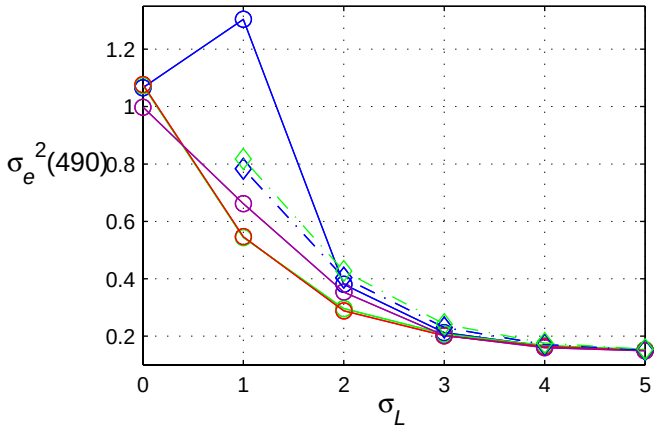


Figure 7.4 Simulated error variance at symbol interval 490 in the frame pattern, as a function of σ_L. Simulation setup: $E_s/N_0 = 10$dB, $\nu_d/f_s = 5/2400$, and $\lambda = 0.95$. Legend as in Fig. 7.3.

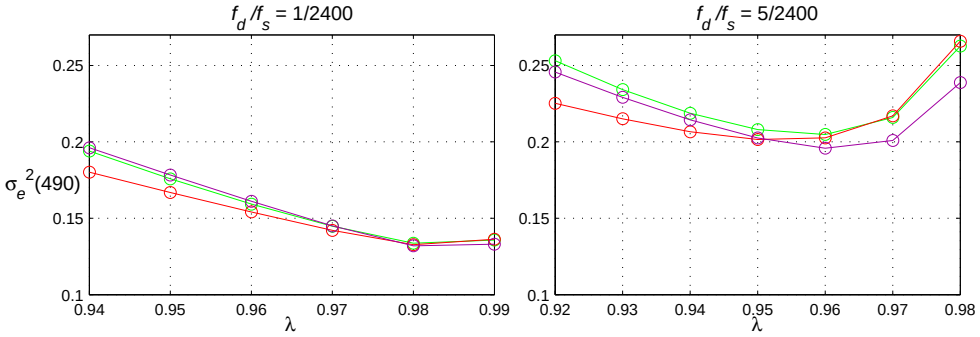


Figure 7.5 Simulated error variance at symbol interval 490 using the Kalman-based algorithm (magenta), the modified RLS algorithm (red), or the LMS algorithm (green) with soft symbols fed back from the decoder, as a function of the forgetting factor λ. Simulation setup: $E_s/N_0 = 10$dB, $\sigma_L = 3$, f_d/f_s as shown above the plots.

Fig. 7.4 shows the error variance at symbol interval 490 (the last data symbol before the third training sequence) as a function of σ_L, for a particular combination of simulation parameters. The performance difference between the algorithms declines with increasing σ_L and is finally negligible for $\sigma_L \geq 5$. At $\sigma_L = 0$, all the channel estimation algorithms are essentially keeping the channel estimate constant between the training sequences, and therefore have almost identical performance (although the Kalman-based channel estimator for some reason has an error variance slightly

below the other algorithms in this case). For small but nonzero σ_L, modified RLS and LMS using soft feedback have the smallest error variance at this time step, followed by the Kalman-based channel estimator and the RLS and LMS algorithms using hard feedback.

The reason why the modified RLS algorithm performs only slightly better than soft feedback LMS, is that soft feedback LMS is equivalent to an approximation to the modified RLS algorithm when PSK signal constellations are used. As discussed in Sec. 7.3.4, this approximation worsens with decreasing λ. For all algorithms, the optimum λ is a function of ν_d/f_s, E_s/N_0, and σ_L. Therefore, it may be difficult to choose the optimum λ in a practical system. In Fig. 7.5 we have investigated how sensitive three of the algorithms are to the choice of λ. Of these three algorithms, the soft feedback LMS algorithm is most sensitive and the modified RLS algorithm is least sensitive to the choice of λ.

Drawing conclusions from these simulation results, three of the considered algorithms perform relatively well in soft iterative channel estimation: The Kalman-based algorithm, the modified RLS algorithm, and the soft feedback LMS algorithm. There are many parameters to vary in this simulation setup, and which of these algorithms performs best depends e.g. on the frame pattern and the fading rate. But in terms of computational complexity, the LMS algorithm is certainly the best choice, followed by the Kalman-based algorithm and the modified RLS algorithm.

7.6 Estimating the error variance

7.6.1 Direct estimation of error variance

The derivation of the SISO equalizer for known channel conditions in Chap. 6 is based on the channel model

$$z_n = \mathbf{h}_n^T \mathbf{x}_n + w_n \tag{7.36}$$

where the w_n are independent zero-mean complex Gaussian random variables with time-varying variance $\sigma_{w,n}^2$, which must be known for the calculation of filter coefficients. When the channel is estimated rather than known, (7.36) can be replaced by (7.34):

$$z_n = \hat{\mathbf{h}}_{n-1}^T \mathbf{x}_n + e_n \tag{7.37}$$

Note that the noise signal w_n has been replaced by the error signal e_n, and therefore $\sigma_{w,n}^2$ should be replaced by $\sigma_{e,n}^2$, the variance of e_n. $\sigma_{e,n}^2$ will be time-varying even when $\sigma_{w,n}^2$ is constant; it will be smaller during training symbols than during data symbols. The problem is that this variance is not known, and we must therefore use an estimate:

$$\hat{\sigma}_{e,n}^2 \approx \sigma_{e,n}^2 = E\{e_n e_n^*\} \tag{7.38}$$

Estimating this variance would be simple if all x_n were known, but in soft iterative channel estimation we only know the *a priori* means $\bar{x}_n$ and variances v_n. Then we do not know the error signal e_n, but only the error signal d_n given by (7.6):

$$z_n = \hat{\mathbf{h}}_{n-1}^T \bar{\mathbf{x}}_n + d_n \tag{7.39}$$

We have devised an *ad hoc* algorithm to estimate $\sigma_{e,n}^2$ from the error signal d_n. The algorithm has a weak mathematical foundation, but performs satisfactory in our simulation setups. The algorithm is as follows:

- Initialization:

$$\hat{\sigma}_{e,0}^2 = \hat{\sigma}_{\text{init}}^2 \tag{7.40}$$

- During training symbols ($\mathbf{V}_n = \mathbf{0}_M$):

$$\hat{\sigma}_{e,n}^2 = (1-\lambda) d_n d_n^* + \lambda \hat{\sigma}_{e,n-1}^2 \tag{7.41}$$
$$\hat{\sigma}_{\text{low}}^2 = \hat{\sigma}_{e,n}^2 \tag{7.42}$$

- During data symbols ($\mathbf{V}_n \neq \mathbf{0}_M$):

$$\hat{\sigma}_{\text{new}}^2 = (1-\lambda)(d_n d_n^* - \hat{\mathbf{h}}_{n-1}^T \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^*) + \lambda \hat{\sigma}_{e,n-1}^2 \tag{7.43}$$
$$\hat{\sigma}_{e,n}^2 = \max(\hat{\sigma}_{\text{new}}^2, \hat{\sigma}_{\text{low}}^2) \tag{7.44}$$

Having produced a time-varying estimate of the variance of e_n, we further assume that this estimate is correct, and that e_n is a zero-mean white Gaussian sequence uncorrelated with all x_n. These assumptions can be summarized as

$$E\{e_n\} = 0 \tag{7.45}$$
$$E\{e_n e_n^*\} = \hat{\sigma}_{e,n}^2 \tag{7.46}$$
$$E\{e_n e_k^*\} = 0, \quad n \neq k \tag{7.47}$$
$$E\{e_n x_k^*\} = 0, \quad \forall\, n,k \tag{7.48}$$

such that SISO equalization algorithms derived for known channel conditions, e.g. the one in Chap. 6, can be applied directly by replacing $\mathbf{h}_n$ with $\hat{\mathbf{h}}_{n-1}$, w_n with e_n, and $\sigma_{w,n}^2$ with $\hat{\sigma}_{e,n}^2$. Note that in all chapters except the current we write the used channel estimate $\hat{\mathbf{h}}_n$ and the used noise variance $\hat{\sigma}_{w,n}^2$ (they should really have been $\hat{\mathbf{h}}_{n-1}$ and $\hat{\sigma}_{e,n}^2$), in order to ease the discussion.

We will now outline the motivation for the variance estimation algorithm given by equations (7.40)-(7.44). When $\mathbf{x}_n$ consists of known training symbols only, we have $e_n = d_n$ such that $\sigma_{e,n}^2 = \sigma_{d,n}^2 = E\{d_n d_n^*\}$. In (7.41) we estimate this variance

by an exponentially weighted time average of $d_n d_n^*$, where we for simplicity have chosen to use the same forgetting factor λ as in the channel estimation algorithm.

When some of the symbols in $\mathbf{x}_n$ are data symbols, inserting $\mathbf{x}_n = \bar{\mathbf{x}}_n + \boldsymbol{\chi}_n$ into equations (7.37) and (7.39) gives

$$d_n = e_n + \hat{\mathbf{h}}_{n-1}^T \boldsymbol{\chi}_n \tag{7.49}$$

from which we find the following relationships between $\sigma_{d,n}^2$ and $\sigma_{e,n}^2$:

$$\sigma_{d,n}^2 = \sigma_{e,n}^2 + \hat{\mathbf{h}}_{n-1}^T \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^* + 2\Re[\hat{\mathbf{h}}_{n-1}^T E\{\boldsymbol{\chi}_n e_n^*\}] \tag{7.50}$$

$$\sigma_{e,n}^2 = \sigma_{d,n}^2 + \hat{\mathbf{h}}_{n-1}^T \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^* - 2\Re[\hat{\mathbf{h}}_{n-1}^T E\{\boldsymbol{\chi}_n d_n^*\}] \tag{7.51}$$

From simulations we find that $E\{\boldsymbol{\chi}_n d_n^*\}$ is generally much larger in magnitude than $E\{\boldsymbol{\chi}_n e_n^*\}$, and we therefore introduce the approximations

$$E\{\boldsymbol{\chi}_n e_n^*\} \approx \mathbf{0}_{M\times 1} \tag{7.52}$$

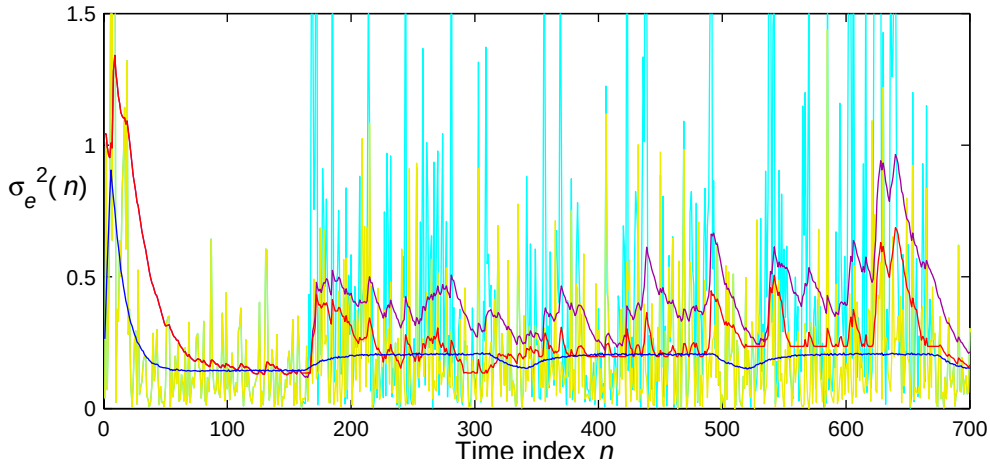
$$E\{\boldsymbol{\chi}_n d_n^*\} \approx \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^* \tag{7.53}$$

such that (7.50) and (7.51) both simplifies to

$$\sigma_{e,n}^2 \approx \sigma_{d,n}^2 - \hat{\mathbf{h}}_{n-1}^T \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^* \tag{7.54}$$

In (7.43) we try to estimate (7.54) by an exponentially weighted time average. We do however perform an odd operation here: The covariance matrix $\hat{\mathbf{h}}_{n-1}^T \mathbf{V}_n \hat{\mathbf{h}}_{n-1}^*$, which can be viewed as an ensemble average, is subtracted from the measurement (realization) $d_n d_n^*$. This will occasionally produce a negative number, which is improper when used to estimate a variance. We have circumvented this problem by introducing the equations (7.42) and (7.44) into the variance estimation algorithm, forcing the estimated variance $\hat{\sigma}_{e,n}^2$ during data symbols never to be smaller than $\hat{\sigma}_{e,n}^2$ at the end of the last training sequence.

We leave for further study the derivation of an algorithm with better mathematical foundation than the one devised here, to estimate the variance $\sigma_{e,n}^2$ of the actual error signal e_n from the observed error signal d_n.



Light blue: $d_n d_n^*$
Magenta: $\hat{\sigma}^2_{d,n}$
Yellow: $e_n e_n^*$
Red: $\hat{\sigma}^2_{e,n}$
Dark blue: $\sigma^2_{e,n}$, ensemble average from Sec. 7.5

Figure 7.6 Estimation of error variance, when the CIR is estimated using the LMS algorithm with soft feedback. Simulation setup: $E_s/N_0 = 10$dB, $\sigma_L = 3$, $\nu_d/f_s = 5/2400$, and $\lambda = 0.95$.

Fig. 7.6 illustrates the operation of our proposed error variance estimation algorithm, using the simulation setup from Sec. 7.5. This plot shows data for one block only, i.e., there is no ensemble averaging except for the dark blue line. We note that the real squared error $e_n e_n^*$ is generally smaller than the "observed" squared error $d_n d_n^*$, except during training where $e_n = d_n$. Accordingly, the estimated error variance $\hat{\sigma}^2_{e,n}$, calculated using (7.40)-(7.44), is generally smaller than the estimated error variance $\hat{\sigma}^2_{d,n}$, calculated using (7.41) during data as well as training symbols. We also see how the estimate $\hat{\sigma}^2_{e,n}$ hits the "floor" defined by $\hat{\sigma}^2_{\text{low}}$. Be aware that the ensemble average $\sigma^2_{e,n}$ is shown here only to outline the frame pattern, and cannot be directly compared with $\hat{\sigma}^2_{e,n}$ for one block only because the channel will be different from one block to the next.

7.6.2 Channel estimation error

An alternative to modelling or estimating the error signal e_n directly, as was done above, is to investigate the channel estimation error vector $\mathbf{d}_n$ defined as

$$\mathbf{d}_n = \hat{\mathbf{h}}_{n-1} - \mathbf{h}_n \tag{7.55}$$

and the corresponding $M \times M$ channel estimation error covariance matrix

$$\mathbf{\Sigma}_{h,n} = E\{\mathbf{d}_n \mathbf{d}_n^H\} \tag{7.56}$$

The estimation error variance of each channel tap is

$$\sigma_{h,n,l}^2 = E\{|\hat{h}_{n,l} - h_{n,l}|^2\} \tag{7.57}$$

and if the channel estimation error for different taps are assumed independent, $\mathbf{\Sigma}_{h,n}$ will be a diagonal matrix with the variances $\sigma_{h,n,l}^2$ on the main diagonal.

In (Tüchler, Otnes and Schmidbauer, 2002), the channel estimation error of the ordinary RLS algorithm using soft feedback has been analyzed, and formulas to efficiently calculate an approximation to $\mathbf{\Sigma}_{h,n}$ have been derived. Once $\mathbf{\Sigma}_{h,n}$ is known or estimated, the channel estimation error can be taken into consideration by the SISO equalizer. For MAP-based equalizers, the channel estimation error variances can be directly included in the branch metric computation (Otnes and Tüchler, 2003). Another approach, which can be used for any equalizer, is to compute the "signal" error variance $\sigma_{e,n}^2$ from the channel estimation error covariance matrix. If the signal constellation is PSK such that $E\{\mathbf{x}_n \mathbf{x}_n^H\} \approx \mathbf{I}_M$, and we assume that $\mathbf{\Sigma}_{h,n}$ is diagonal, we have

$$\sigma_{e,n}^2 = E\{e_n e_n^*\} = E\{|(\hat{\mathbf{h}}_{n-1} - \mathbf{h}_n)^T \mathbf{x}_n + w_n|^2\} \approx \sigma_{w,n}^2 + \sum_{l=0}^{M} \sigma_{h,n,l}^2 \tag{7.58}$$

from which we see that $\sigma_{e,n}^2$ is always larger than $\sigma_{w,n}^2$, with an amount equal to the sum of the channel tap error variances.

The problem with applying (7.58) in practice, is that two additional quantities must be estimated in this case: The noise variance $\sigma_{w,n}^2$, and the channel estimation error covariance matrix $\mathbf{\Sigma}_{h,n}$. We have therefore rather chosen to estimate $\sigma_{e,n}^2$ directly, using the algorithm described in Sec. 7.6.1.

7.7 Fractionally spaced channel estimation

When the receiver employs fractional sampling of the received signal $z(t)$, the fractionally spaced channel impulse response can be estimated by running κ channel estimation algorithms in parallel, where κ is the oversampling ratio (number of samples per symbol period).

As discussed in Sec. 4.5.1.2, the fractionally spaced received signal $z_{n,i}$ can be written

$$z_{n,i} = \mathbf{h}_{n,i}^T \mathbf{x}_n + w_{n,i}, \quad i \in \{0, 1, \ldots, \kappa - 1\} \tag{7.59}$$

If we assume statistical independence for $i \neq j$ between $\mathbf{h}_{n,i}$ and $\mathbf{h}_{n,j}$, and between $w_{n,i}$ and $w_{n,j}$, the κ "subchannels" defined by (7.59) will be decoupled and can therefore be estimated independently. We can use any channel estimation algorithm, e.g. one of the algorithms investigated in this chapter, to provide κ different time-varying channel estimates $\hat{\mathbf{h}}_{n,i}$. The channel model using the estimated channel is then

$$z_{n,i} = \hat{\mathbf{h}}_{n-1,i}^T \mathbf{x}_n + e_{n,i}, \quad i \in \{0, 1, \ldots, \kappa - 1\} \tag{7.60}$$

where $e_{n,i}$ is the error signal corresponding to channel estimator number i.

Let us now concentrate on the case $\kappa = 2$. If all $e_{n,i}$ can be assumed independent, the fractionally spaced linear SISO equalizer of Sec. 6.2 can be applied directly, replacing $\mathbf{h}_{n,i}$ with $\hat{\mathbf{h}}_{n-1,i}$ and $\sigma^2_{w,n,i}$ with $\hat{\sigma}^2_{e,n,i}$. However, correlation may exist between $e_{n,0}$ and $e_{n,1}$, because the same "soft training symbols" $\bar{x}_n$ have been used to compute both estimates. If we were able to model or estimate this correlation as $\hat{\rho}_{n,10} \approx \rho_{n,10} = E(e_{n,1} e^*_{n,0})$, it could be taken into consideration in the linear SISO equalizer by replacing (6.45) with

$$\alpha_{n,0,10} = \hat{\rho}_{n+N_1,10} + \sum_{l=0}^{M-1} h_{n+N_1,l,1} h^*_{n+N_1,l,0} v_{n+N_1-l} \tag{7.61}$$

The possible correlation $\rho_{n,10}$ has not been further investigated in this dissertation.

7.8 Conclusions

The critical reader will find that this chapter is only an initial study of algorithms for soft iterative channel estimation, which is a field in which few papers have been previously published. Several threads have been left open for further research. We can however draw some conclusions, which are useful for the design of a receiver using iterative channel estimation, equalization, and decoding:

- Soft iterative channel estimation is better than hard iterative channel estimation, if the channel estimation algorithm is chosen wisely.
- The ordinary RLS algorithm is a bad choice for soft iterative channel estimation.
- The LMS algorithm is a good choice for soft iterative channel estimation. It is very simple and has performance comparable to the more complex modified RLS algorithm and Kalman-based algorithm. The only prerequisite for using the LMS algorithm is that the initial training sequence is long enough to account for the slower convergence rate of the LMS algorithm (if the initial

training sequence is too short for the LMS algorithm to settle, an LSSE channel estimate based on the initial training sequence can be used to initialize the LMS algorithm).

- When using soft iterative channel estimation, the same channel estimation algorithm can be used in the zeroth iteration as in later iterations. When all input LLRs are zero, the channel estimation algorithms do nothing more wrong than keeping the channel estimate constant between training sequences.
- When using an estimated channel, the SISO equalizer should replace the noise variance $\sigma_{w,n}^2$ with the estimated error variance $\hat{\sigma}_{e,n}^2$, e.g. estimated using the algorithm in Sec. 7.6.1.

URN:NBN:no-3359

Chapter 8

EXIT chart analysis applied to adaptive turbo equalization

In this chapter, we show how a semianalytical technique called EXIT charts (extrinsic information transfer charts), originally developed by S. ten Brink for analysis of turbo codes (ten Brink, 2001), can be used to predict the performance of an iterative receiver using adaptive turbo equalization. To demonstrate the usefulness of the technique, we use EXIT charts to address interesting questions about adaptive turbo equalization for time-varying channels: Which pattern of training sequences should be used, what is the performance difference between optimal MAP equalization and suboptimal linear equalization, and between the case of a known channel and the case of an estimated channel, and what can be gained by using a recursive precoder in conjunction with the symbol mapper. This chapter is mostly based on a paper presented at Nordic Signal Processing Symposium in October, 2002 (Otnes and Tüchler, 2002b).

8.1 Introduction

Analytical expressions for the performance of a receiver employing turbo equalization are not known, and the task of deriving analytical expressions is even more complicated when channel estimation is included in the iterative process. Therefore, the performance of turbo equalization for a particular system is usually studied via time-consuming simulations. It is, however, desirable to use other methods to better understand the operation of iterative receivers, in particular the convergence behavior of the iterative process.

The same lack of analytical tools for studying the convergence of iterative decoders has been a problem in the study of turbo codes and low-density parity-check

(LDPC) codes[1]. In the last years, useful methods have been introduced based on the evolution of the "density" (pdf) of LLRs as the iterations proceed. The correlations between LLRs for different code bits are ignored, and all those methods are therefore only valid for infinite, or in practice very large, interleaver lengths (or block lengths for LDPC codes).

In this chapter we use the shorthand notation $p(l|c)$ for the conditional pdf $p_{L|c}(L_e(c_k)|c_k = c)$ of the extrinsic LLRs $L_e(c_k)$, conditioned on the corresponding transmitted code bit c_k. As iterations proceed, the conditional pdf $p(l|c)$ will "improve" in the sense that the overlap between $p(l|1)$ and $p(l|-1)$ becomes smaller, such that it is possible to make a correct decision on c_k from $L_e(c_k)$ with high probability.

For LDPC codes, it is under certain circumstances possible to analytically compute the evolution of $p(l|c)$ for every iteration, called density evolution (Richardson, 2000; Richardson, Shrokrollahi and Urbanke, 2001). For turbo codes, "practical" density evolution is sometimes used in conjunction with simulations, showing a simulated estimate of $p(l|c)$ after each iteration, see e.g. (Divsalar, Dolinar and Pollara, 2001). Computing the conditional pdf $p(l|c)$ analytically for every iteration is a very complicated problem for turbo decoding and turbo equalization, and a simpler approach which is still reasonably accurate is to investigate the evolution of a single parameter derived from $p(l|c)$. This approach was pioneered by ten Brink in (ten Brink, 1999; ten Brink, 2000b; ten Brink, 2000a; ten Brink, 2001), using the mutual information between $L_e(c_k)$ and c_k as the parameter derived from $p(l|c)$, to study the convergence behavior of iterative demapping, and turbo decoding of serially and parallel concatenated turbo codes. The method of ten Brink is called the EXIT chart. Another option is to use an "equivalent SNR" as the parameter derived from $p(l|c)$ (Divsalar et al., 2001; Gamal and Hammons, 2001; Chung, Richardson and Urbanke, 2001), but the benefits of rather using the mutual information are that it is bounded on $[0, 1]$ such that it is easier to visualize, and that it is unambiguously defined for any shape of conditional pdf.

In practice, the EXIT chart is used as a semianalytical tool to investigate the performance of iterative detectors. Separate simulations are performed on each SISO module to obtain *mutual information transfer functions*, which describe the mapping from the mutual information between the code bits and the LLRs input to the SISO module, to the mutual information between the code bits and the LLRs output from the SISO module. Because the output of one SISO module is input to the other SISO module and vice versa, the two transfer functions can be drawn in a two-dimensional chart and the performance of the receiver is given by a trajectory between the two

[1]LDPC codes are linear codes introduced by Gallager (1963), which can be decoded using an iterative procedure similar to turbo decoding (although the component decoders are vastly different). They were mostly forgotten until the advent of turbo codes.

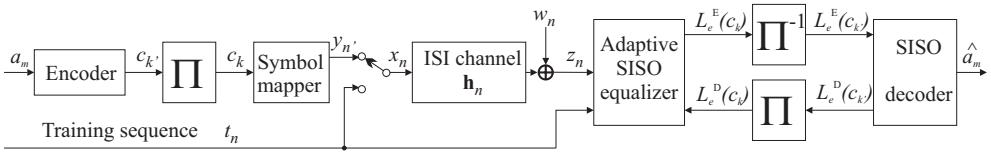


Figure 8.1 Block diagram of a communication system employing adaptive turbo equalization in the receiver.

curves. The benefits of this method are to predict the system performance without actually simulating data transmission, and to gain insight about the convergence of the iterative procedure. The benefit of using EXIT charts to analyze the performance of turbo equalization was presented in (Tüchler, Koetter and Singer, 2002) for the case of a known, time-invariant channel. In this chapter, we consider unknown, time-varying channels, and show how EXIT charts can be applied to receivers performing iterative equalization, estimation, and decoding, which is referred to as adaptive turbo equalization in the remainder of this chapter.

8.2 System model

Consider the system model in Fig. 8.1, which should by now be familiar to the reader of this dissertation. A block of data bits a_m is encoded with a rate-R_c convolutional code to N_{int} code bits $c_{k'} \in \{+1, -1\}$ including trellis termination. The bits $c_{k'}$ are interleaved to c_k, which are mapped to N_{int}/Q data symbols $y_{n'}$ from a 2^Q-ary signal constellation. The data symbols $y_{n'}$ are multiplexed with N_t training symbols t_n known to the receiver to form a block of $N_{block} = N_{int}/Q + N_t$ symbols x_n. As in (5.1), the overall code rate including coding and training overhead is

$$R = R_c \frac{N_{int}}{N_{int} + QN_t} \tag{8.1}$$

The x_n are transmitted over the time-varying ISI channel with equivalent symbol-spaced discrete-time CIR $\mathbf{h}_n = [h_{n,0} \cdots h_{n,M-1}]^T$. The length of the CIR is M symbol intervals. The received symbols z_n are given by

$$z_n = \sum_{l=0}^{M-1} h_{n,l} x_{n-l} + w_n \tag{8.2}$$

where the w_n are complex white Gaussian noise samples with variance σ_w^2, i.e., their probability density function (pdf) is $p(w) = \exp(-|w|^2/\sigma_w^2)/(\pi\sigma_w^2)$.

The (adaptive) SISO equalizer processes the received symbols z_n, the training symbols t_n, and the *a priori* LLRs $L_e^D(c_k)$ fed back from the decoder. During first-time equalization, all $L_e^D(c_k)$ are zero. The equalizer outputs the LLRs $L_e^E(c_k)$,

which, deinterleaved to $L_e^E(c_{k'})$, are input to the SISO decoder. The SISO decoder computes estimates $\hat{a}_m$ of the transmitted data bits and also outputs the LLRs $L_e^D(c_{k'})$, which, interleaved to $L_e^D(c_k)$, are used as *a priori* LLRs by the SISO equalizer in the next iteration.

8.3 The mutual information

The EXIT chart describes the convergence of the iterative receiver algorithm by investigating the exchange of mutual information between the equalizer and the decoder. Observed is the mutual information $I_e = I(L_e^E(c_k); c_k)$ at the equalizer output and $I_d = I(L_e^D(c_{k'}); c_{k'})$ at the decoder output, where the mutual information is defined as

$$I(L;C) = \frac{1}{2} \sum_{c \in \{+1,-1\}} \int_{-\infty}^{\infty} p(l|c) \log_2 \frac{2p(l|c)}{p(l|+1) + p(l|-1)} dl \tag{8.3}$$

where $p(l|c)$ denotes the pdf of the LLRs modelled as outcomes of the random variable (r.v.) L conditioned on its corresponding transmitted code bit c represented by the r.v. C. The range of I_d and I_e is $[0, 1]$, where $I_d = 0$ or $I_d = 1$ correspond to no or perfect knowledge about $c_{k'}$ given $L_e^D(c_{k'})$.

We can usually assume that the symmetry constraint $p(l|c) = p(-l|-c)$ holds, such that (8.3) simplifies to

$$I(L;C) = \int_{-\infty}^{\infty} p(l|+1) \log_2 \frac{2p(l|+1)}{p(l|+1) + p(-l|+1)} dl \tag{8.4}$$

The mutual informations I_e and I_d can be found by calculating the integral (8.4) numerically from the conditional pdfs $p(l|+1)$ output from the equalizer and the decoder, which are estimated as histograms of $c_k L_e^E(c_k)$ and $c_{k'} L_e^D(c_{k'})$, respectively.

When the LLRs have been computed exactly, the conditional pdf $p(l|c)$ is subject to a second constraint, called the consistency condition:

$$p(l|+1) = p(-l|+1) \cdot e^l \tag{8.5}$$

When the consistency condition also holds, (8.4) simplifies further to (Tüchler and

Hagenauer, 2002)

$$I(L;C) = 1 - \int_{-\infty}^{\infty} p(l|+1)\log_2(1+e^{-l})dl \tag{8.6}$$

which can be approximated by the time average

$$I(L;C) \approx 1 - \frac{1}{N_{tot}} \sum_{k=1}^{N_{tot}} \log_2(1+e^{-L_e(c_k)}) \tag{8.7}$$

where N_{tot} is the number of observed LLRs $L_e(c_k)$.

Of course, calculating the time average (8.7) is much simpler than generating a histogram and calculating the integral (8.4) numerically. Unfortunately, our simulations have shown that the consistency condition does not always hold, e.g., it does not hold when the channel is estimated. We must therefore resort to the more complicated histogram method in those cases.

8.4 EXIT charts for adaptive turbo equalization

The evolution of I_d and I_e over the iterations in a real receiver is called the system trajectory of the iterative algorithm. The purpose of the EXIT chart is to predict the system trajectory from *mutual information transfer functions*, without performing simulations on the complete iterative receiver. When we simulate each SISO module separately, we do not know the conditional pdf $p(l|c)$ of the input LLRs to the SISO module, and they are therefore modelled as independent Gaussian variables with average value $c\sigma_L^2/2$ and variance σ_L^2, which is a common distribution of LLRs in an iterative receiver (ten Brink, 2001), see also Sec. 7.2. We then have

$$p(l|+1) = \frac{1}{\sqrt{2\pi\sigma_L^2}}\, e^{\frac{-(l-\sigma_L^2/2)^2}{2\sigma_L^2}} \tag{8.8}$$

A one-to-one relationship exists between σ_L^2 and $I(L;C)$: By inserting (8.8) into (8.4), we obtain

$$I(L;C) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_L^2}}\, e^{\frac{-(l-\sigma_L^2/2)^2}{2\sigma_L^2}} \log_2 \frac{2}{1+e^{-l}} dl \tag{8.9}$$

which can be calculated through numerical integration. The result is shown in Fig. 8.2.

The simulation setup to generate the transfer function of the SISO decoder is shown in Fig. 8.3(a). Code bits $c_{k'}$ are generated by encoding a block of independent

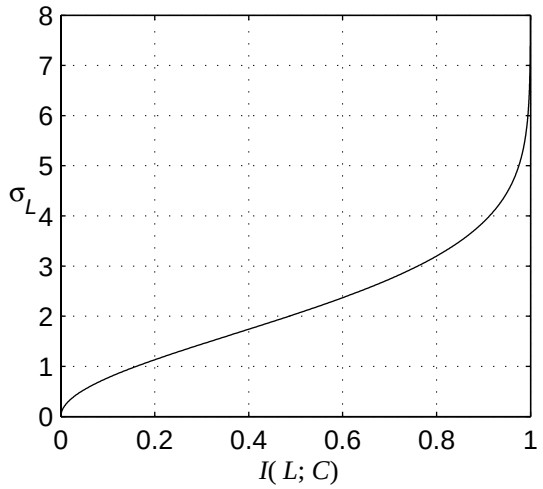
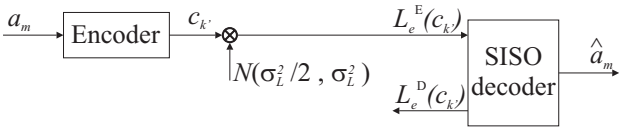
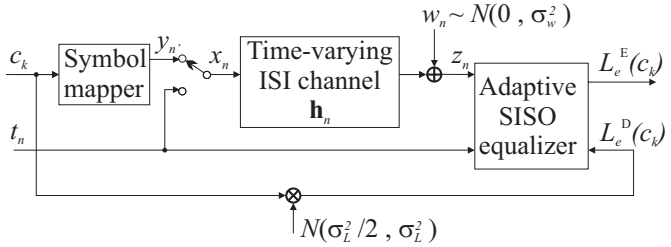


Figure 8.2 Relationship between mutual information and σ_L (the square root of σ_L^2) when the LLRs have the conditional pdf given by (8.8).



(a) Decoder



(b) Equalizer

Figure 8.3 Simulation setup for generating mutual information transfer functions.

data bits a_m with the ECC. The input LLRs $L_e^E(c_{k'})$ to the decoder are drawn independently from a normal distribution with average value $c_{k'}\sigma_L^2/2$ and variance σ_L^2 as discussed above. We use the MAP algorithm (Bahl et al., 1974) for SISO decoding. A number of blocks is simulated for different values of σ_L^2 in order to estimate the output mutual information I_d and the bit error rate of the data bit estimates $\hat{a}_m$ output from the decoder. The result is shown in Fig. 8.4 for two different rate-1/2 convolutional codes: a memory-2 code with generator polynomials $1 + D^2$ and $1 + D + D^2$, and a memory-6 code with generator polynomials $1 + D + D^3 + D^4 + D^6$ and $1 + D^3 + D^4 + D^5 + D^6$ (the code used in the HF standards, see Chap. 2). Note that the stronger code (memory-6) has a steeper transfer function than the weaker code, and therefore requires a lower input mutual information I_e in order to produce high-quality output LLRs and low bit error rates. The weaker code, however, performs better when I_e is below 0.5, and may therefore be a better choice than the stronger code in terms of initial convergence of the iterative receiver. An increase/decrease in code rate R_c would move all curves in Fig. 8.4 towards the right/left.

The transfer function of the equalizer is similarly generated using the setup shown in Fig. 8.3(b). A block of N_{int} independent code bits c_k is passed through all blocks in Fig. 8.1 from the symbol mapper to the received symbols z_n. The ECC generating the $c_{k'}$ is omitted and dependencies within the interleaved code bits c_k, which exist in a real system, are neglected. In a parallel path, the c_k are multiplied by independent samples from a Gaussian distribution with average value $\sigma_L^2/2$ and variance σ_L^2, to form the input LLRs $L_e^D(c_k)$ with conditional pdf given by (8.8). Since the adaptive SISO equalizer also processes received symbols z_n, its transfer function will depend on E_b/N_0 in contrast to the SISO decoder. Fig. 8.5 shows the transfer function for the following example: The block of transmitted symbols x_n consists of 248 initial known training symbols, followed by 256 data, 31 training, 256 data, 31 training symbols and so on, until the end of the block. The signal constellation is Gray-coded 8-PSK, and the CIR is of the form $\mathbf{h}_n = [h_{n,0}\ 0\ 0\ 0\ 0\ h_{n,5}]^T$ of length $M' = 6$. The two nonzero taps $h_{n,0}$ and $h_{n,5}$ are drawn according to independent Rayleigh fading with a Gaussian Doppler spectrum having a 2σ Doppler spread (fading rate) of $\nu_d = 1/(2400T_s)$, where T_s is the symbol period. The adaptive SISO equalizer uses separate soft-in channel estimation and equalization as discussed in Chap. 7, where the least mean squares (LMS) algorithm is used to generate a time-varying channel estimate and the MMSE-optimal linear SISO equalizer of Chap. 6 is used for equalization, with filter length parameters $N_1 = 15$ and $N_2 = 8$. The step size of the LMS algorithm is 0.02, and the length of the channel estimate is $M = 8$, i.e. two symbol intervals longer than the length M' of the actual CIR to account for the fact that the length of the CIR is not known in advance.

The EXIT chart in Fig. 8.6 (left) combines the equalizer transfer function at 10 dB E_b/N_0 and the decoder transfer function for the memory-2 code. Since the output

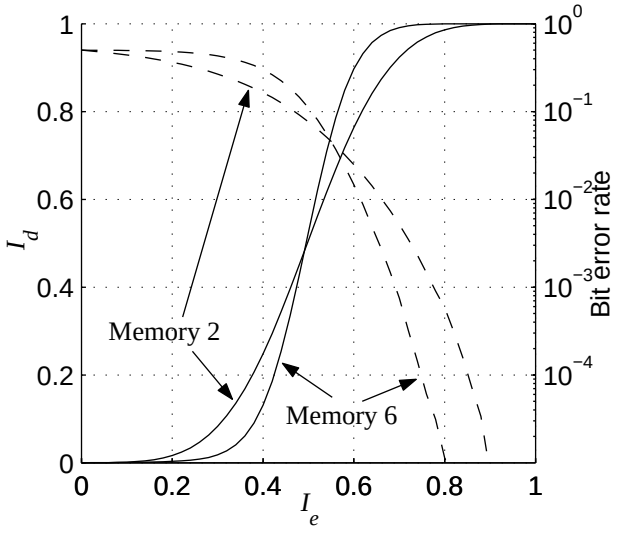


Figure 8.4 Transfer function (solid lines) from I_e (input) to I_d (output), and bit error rate (dashed lines) as function of I_e, of a MAP-based decoder for rate-1/2 convolutional codes with memory 2 or 6.

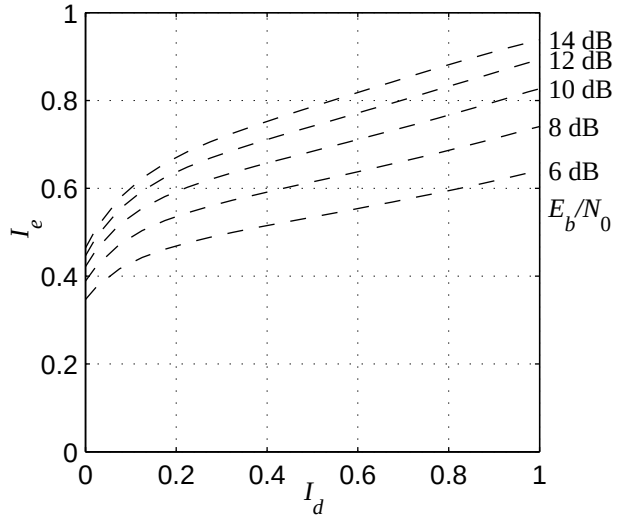


Figure 8.5 Mutual information transfer function of the linear SISO equalizer receiving CIR estimates from an LMS-based channel estimator.

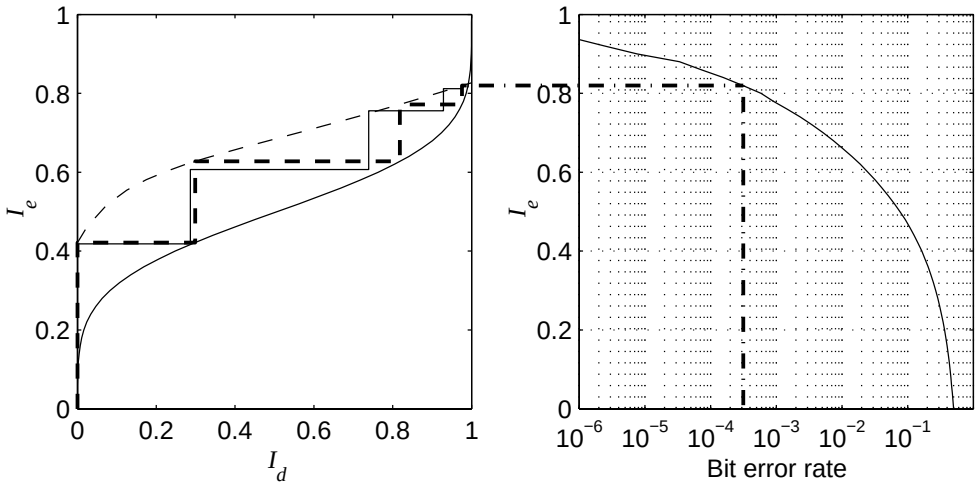


Figure 8.6 Left: EXIT chart for the example in the text at 10 dB E_b/N_0. The predicted system trajectory is shown as *dashed* vertical and horizontal lines, and the actual system trajectory obtained from simulations on the complete iterative receiver is shown as *solid* vertical and horizontal lines. Right: Bit error rate out of the decoder as a function of I_e from Fig. 8.4 (with flipped axes), showing the predicted bit error rate after convergence of the iterative receiver.

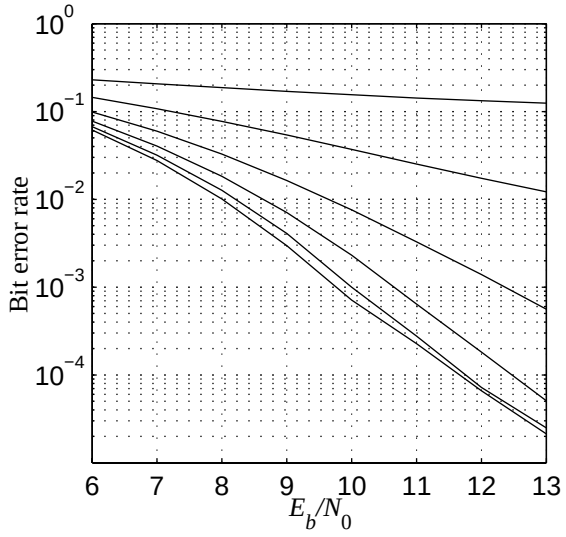


Figure 8.7 Simulated bit error rate as function of E_b/N_0 for the example in the text, after 0 (upper line) until 5 (bottom line) iterations.

LLRs from the equalizer are input to the decoder and vice versa, both transfer functions are drawn in the same plot with the axes being flipped for the decoder transfer function. The system trajectory of the iterative receiver can be predicted by representing each equalization task as a vertical line and each decoding task as a horizontal line, bouncing between the two transfer functions (the dashed lines in Fig. 8.6). The trajectory starts in the (0,0) point and approaches a fixed point, which is the leftmost crossing point of the two transfer functions. Each input mutual information I_e to the decoder is associated with a bit error rate as shown in Fig. 8.4 and (with flipped axes) in Fig. 8.6 (right). From the EXIT chart in Fig. 8.6, we predict that at 10 dB E_b/N_0, iterative decoding converges after 4 times equalization and decoding to a bit error rate of $3 \cdot 10^{-4}$ corresponding to $I_e = 0.82$ in the fixed point.

To verify the predicted performance, we track the trajectory of a real system by calculating the mutual information at the output of the equalizer and the decoder after each iteration. We use an S-random interleaver of length $N_{int} = 24576$. The real system trajectory at 10 dB E_b/N_0 is shown as solid lines in Fig. 8.6. The actual performance is close to the predicted performance for the first few iterations, but they depart at later iterations. In reality, equalization and decoding tasks must be performed 5 or 6 times in order for the receiver to converge, i.e. one or two times more than the prediction of 4. The simulated bit error rate is shown in Fig. 8.7. At 10 dB E_b/N_0 the bit error rate after convergence is $7 \cdot 10^{-4}$, which is a bit more that the predicted bit error rate. The EXIT chart analysis assumes that all LLRs are independent, and is therefore in theory valid only for an infinite interleaver length N_{int}. This is one reason for the slight differences between predicted and actual performance in the example discussed above, and we find that the differences become smaller when N_{int} is increased. For time-varying channels, N_{int} should be large compared to the fading rate, i.e., there should be several fades within each interleaver block, in order to ensure that all blocks have the same statistical properties such that the EXIT chart analysis becomes exact. For smaller N_{int}, the performance predicted by the EXIT chart can be viewed as a practical bound on the achievable performance.

In Fig. 8.8 we show the actual density evolution for the simulation setup used to generate the EXIT chart in Fig. 8.6, where the conditional pdf $p(l| + 1)$ has been estimated using histograms of $c_k L_e^E(c_k)$ out of the equalizer and $c_{k'} L_e^D(c_{k'})$ out of the decoder. We also show Gaussian pdfs on the form (8.8), having the same mutual information as the simulated conditional pdf. This figure may suggest the approximation of the conditional pdf as Gaussian is bad, especially for the output from the equalizer. However, ten Brink (1999) has found that the output mutual information from a SISO module is relatively insensitive to the actual *shape* of the conditional pdf of the input LLRs, as long as the input mutual information is unchanged, due to what he calls the "robustness" of the entropy measure used in the definition of mutual information. Therefore, the deviation between prediction and reality in Fig.

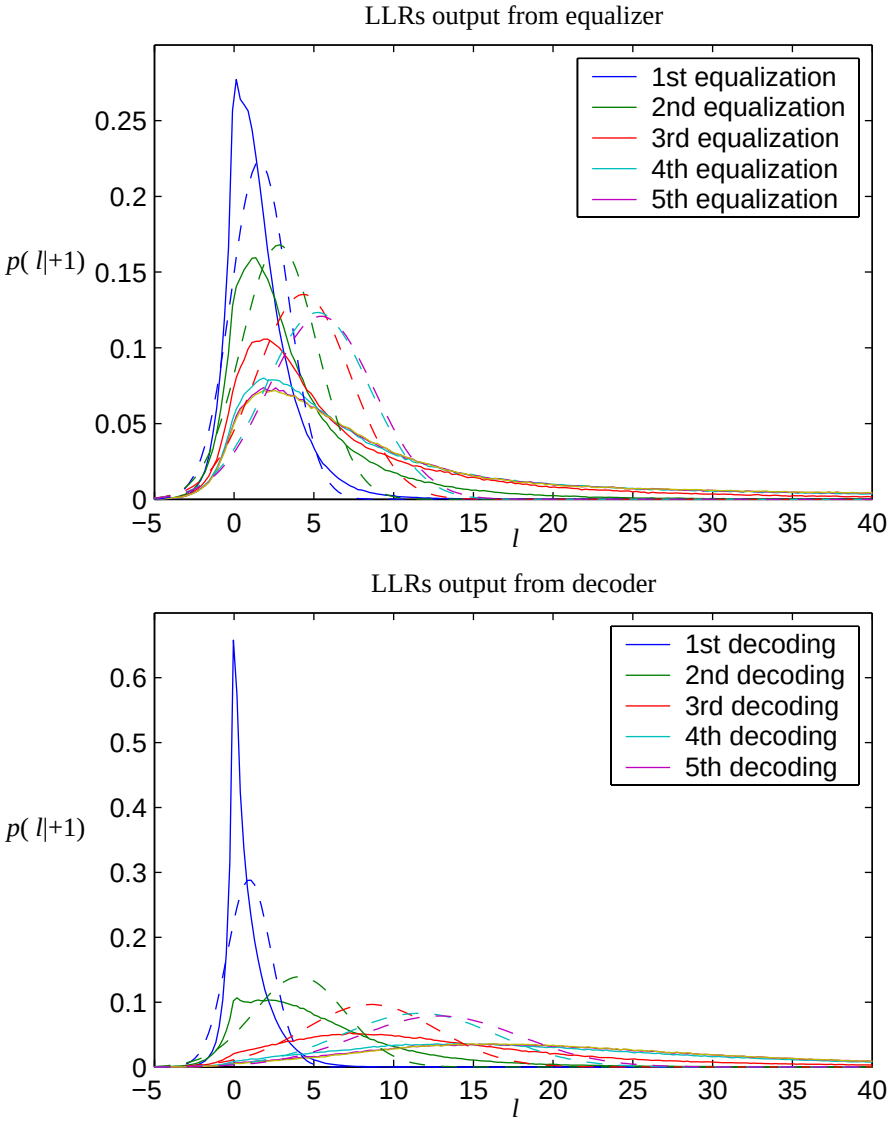


Figure 8.8 Simulated conditional pdf $p(l|c)$ after each iteration (solid lines) obtained from histograms, and (dashed lines) Gaussian pdf $p(l|c)$ on the form (8.8) having the same mutual information as the simulated conditional pdf.

8.6 is relatively small even though the input conditional pdfs to the SISO modules in reality are quite different from those used when generating the mutual information transfer functions.

8.5 Using the EXIT chart

Now that we have demonstrated by an example how EXIT charts can be applied to adaptive turbo equalization, we are ready to discuss their benefits, some of which are

1. The EXIT chart is a nice way to visualize what happens in an iterative receiver, which may otherwise seem like "magic" to people not familiar to the subject.
2. We can investigate the effect of modifying one of the two SISO algorithms without performing simulations on the complete iterative receiver.
3. We can see how well the ECC/decoder and channel/equalizer ("outer" and "inner" code) are fitted to each other.
4. The transfer function of the equalizer can be used to calculate the maximum achievable code rate for specific channel conditions.

The first point was demonstrated thoroughly by the example in the previous section. To demonstrate the remaining points, we give some further examples.

In Fig. 8.9 we show the effect of changing the pattern of training symbols. When the interval between training sequences is increased, the overall code rate R will increase, causing the curves to move upward at high I_d. However, the channel estimate will be poorer when I_d is low because it fails in tracking the channel variations between training sequences, and the curves with the most training have the highest I_e in the left part of the EXIT chart. Finding the optimal training pattern is therefore a trade-off between initial convergence at low E_b/N_0 and error rates after convergence at high E_b/N_0, and for this example 128 or 256 data symbols between each training sequence seem to be good choices. The transfer function for the two rate-1/2 codes discussed earlier are also shown in this EXIT chart. Note that the memory-2 code has a transfer function almost parallel to most equalizer transfer functions, while the memory-6 code has a "knee" at low I_d which may introduce a fixed point at low I_d when E_b/N_0 is decreased compared to this plot. Therefore, a strong outer code is not always a good choice when using turbo equalization in the receiver: When E_b/N_0 is decreased, the stronger code will fail before the weaker code in terms of initial convergence because the weaker code is better matched to the equalizer. It is possible to construct a code matched to a given equalizer transfer function according to some optimization criterion, e.g., to obtain values of I_d and I_e as high as possible after a certain number of iterations (Tüchler, 2002).

In Fig. 8.10 we have compared the equalizer transfer function for different scenarios. The example is the same as in Sec. 8.4, except that the length of the estimated

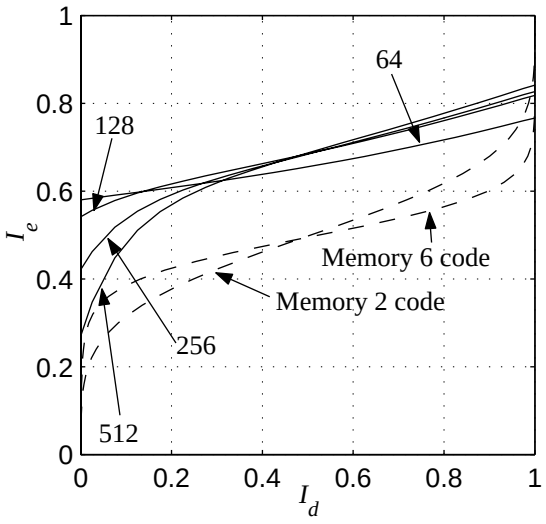


Figure 8.9 EXIT chart for different frame patterns, having 64, 128, 256, and 512 data symbols between each training sequence of length 31 symbols. $E_b/N_0 = 10$ dB for all curves.

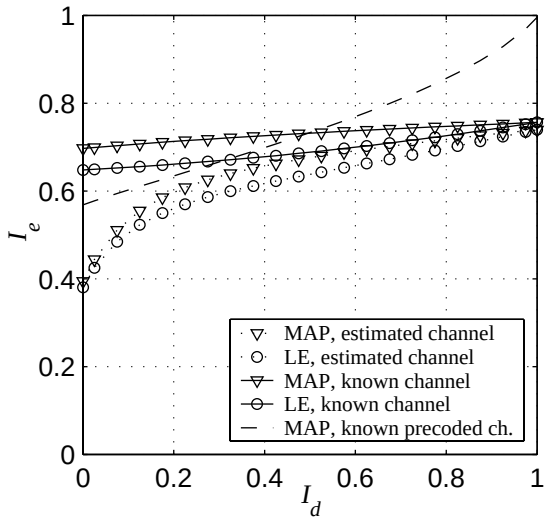


Figure 8.10 Equalizer transfer function for different scenarios: Linear equalization (LE) vs optimal MAP equalization, known vs estimated channel, and a system using a recursive precoder, at $E_b/N_0 = 8$ dB.

and actual CIR has been reduced to $M = M' = 3$, i.e. $\mathbf{h}_n = [h_{n,0}\ 0\ h_{n,2}]$, to make MAP equalization feasible. We see that at $I_d = 1$, MAP equalization has the same performance as the much simpler approach of soft ISI cancellation combined with linear equalization (LE), for the case of a known channel as well as for an estimated channel. Thus, the performance using LE is similar to the performance using MAP equalization, after convergence of the iterative receiver. The major differences between LE and MAP equalization are therefore the number of iterations needed before convergence, and the initial convergence at low E_b/N_0. We also find that the transfer functions when the channel is known at all times are much flatter than when the channel is estimated, indicating that there is much more to gain from doing iterations when the channel estimate is improved over the iterations. At $I_d = 1$, the curves for a known and for an estimated channel are quite close.

We have in Fig. 8.10 also investigated the use of a recursive precoder in conjunction with the symbol mapper, see also Sec. 5.5. In (Lee, 2001; Narayanan, 2001) is shown that turbo equalization can have performance similar to turbo-coded systems when a recursive precoder is used: A "turbo cliff" at a certain E_b/N_0, above which the bit error rate is very small (negligible compared to the error rates in Fig. 8.7). Here we have used the simple precoder defined by $s_{n,i} = s_{n-1,i} \cdot c_{3n+i}$, $i \in \{0, 1, 2\}$, $s_{n,i} \in \{+1, -1\}$, where three precoded bits $s_{n,i}$ are mapped onto each symbol y_n using an 8-PSK constellation. Note that the transfer function goes to the (1,1) point when using a precoder (here, the channel is known). This explains the turbo cliff behavior: When the iterative decoding does converge, it converges to the (1,1) point where error rates are negligible. The price to pay compared to the non-precoded case is that I_e is decreased at low I_d, such that the E_b/N_0 required for convergence is increased if the same ECC is used. Also, non-precoded turbo equalization can be used as a receiver technology in present-day communication systems designed without turbo equalization in mind, whereas the introduction of recursive precoding requires changes to the transmitter. Future work in precoded turbo equalization includes complexity reduction by deriving algorithms based on linear filters, and the inclusion of channel estimation.

It can be shown (Tüchler and Hagenauer, 2002; Ashikhmin, Kramer and ten Brink, 2002) that the maximum rate of an outer code which stays underneath the transfer function of the equalizer, such that the iterative receiver can converge, is equal to the area under that transfer function. An interesting point to note is that the area under the curve for a precoded system is exactly the same as the area under the curve for the corresponding non-precoded system (Tüchler, 2002), such that the same overall rate can be achieved for both systems if the ECC is optimized.

8.6 Chapter summary

In this chapter, we have shown how the well-known and often cited method called "EXIT charts" can be applied to analyze the convergence behavior of a receiver employing iterative channel estimation, equalization, and decoding (adaptive turbo equalization). We have presented in detail the simulation setups used to generate mutual information transfer functions for the considered application.

By an example, we show that the performance predicted by the EXIT chart is quite close to the "real" simulated performance of the iterative receiver, even though two underlying assumptions for the EXIT chart analysis are invalid: (1) The interleaver length is not infinite, a well-known deficiency of the EXIT chart method, and (2) the conditional pdf of LLRs output from the SISO modules in the turbo equalization setup are quite different from the conditional pdf which is assumed when generating the mutual information transfer functions.

Finally, we have used the EXIT chart to investigate the effect of the frame pattern on the performance of adaptive turbo equalization, and to compare optimal and suboptimal equalization algorithms and the case of knowing the channel exactly vs soft iterative channel estimation.

URN:NBN:no-3359

Chapter 9

Turbo equalization-based receiver for HF modems

After delving into details for some chapters, we will now use the obtained knowledge on iterative channel estimation, equalization, and decoding, to design a turbo equalization-based receiver for HF serial-tone modems. The proposed receiver is implemented in a simulation setup, and we have also implemented a conventional receiver using decision feedback equalization in the same setup. Simulation results show that the turbo equalization-based receiver performs more than 2 dB better than the conventional receiver over an ITU-R poor channel, when considering the required SNR to obtain a certain bit error ratio. This chapter is mostly based on a paper presented at MILCOM in October, 2002 (Otnes and Tüchler, 2002c). In this chapter, and the paper it is based on, we have deliberately excluded all equations in order to point out the essential results.

9.1 Applicability of turbo equalization to serial-tone HF waveforms

In Chap. 2 we described standardized serial-tone HF waveforms, and summarized their properties in Table 2.3 on page 31. We will now discuss the applicability of turbo equalization to the different waveforms.

Of particular interest is the interleaver. Generally, an increasing interleaver length improves the ability to break up burst errors in a conventional receiver, as well as the performance of turbo equalization. However, the system delay also increases with increasing interleaver length.

The interleaver particularly affects the performance and applicability of turbo equalization, since the code bit LLRs are deinterleaved and interleaved several times during the iterative process. A convolutional interleaver causes a system delay equal to the interleaver length multiplied by the number of iterations; the delay using a block-based interleaver is the same regardless of the number of iterations. Therefore, STANAG4285 is not suitable for turbo equalization. Neither is the high-rate data link (HDL) traffic data waveform "BW2" in STANAG4538, since it does not use interleaving at all. Waveforms with an interleaver suitable for turbo equalization are, thus: MIL-STD-188-110A/B, STANAG4539, STANAG4415, and the low-latency data link (LDL) traffic data waveform "BW3" in STANAG4538. The latter two, which are the "robust" waveforms, use Walsh functions in the modulator. Turbo equalization can also be applied to such waveforms (Herzog, Hagenauer and Schmidbauer, 1997; Herzog, Schmidbauer and Hagenauer, 1997), but they do not fit into the framework used when deriving algorithms in this dissertation: We must include despreading of the Walsh codes, and channel estimation has to be performed differently. The receiver topology we propose here can therefore be applied to the medium- and high-rate waveforms described in MIL-STD-188-110 and STANAG 4539.

Not mentioned in Table 2.3, the transmitted symbols are scrambled in order to appear more random on-air. This feature does not affect turbo equalization as long as descrambling and scrambling operations are appropriately inserted while performing iterations. Those operations are not shown in the figures in this chapter, but have been included in the simulations.

The decoder for the ECC is more complex when turbo equalization is applied than for a conventional receiver, due to the SISO capability. Such a SISO decoder traverses the code trellis in both directions (Bahl et al., 1974) in contrast to the forward-only operation of a Viterbi decoder. The code trellis the decoder operates on can be terminated for a given block size, either by encoding some extra zero data bits or by forcing each path in the trellis to start and end in the same state, which does not introduce a rate loss due to termination. The latter technique is called "tail-biting". STANAG4539 uses tail-biting to terminate the trellis for each interleaver block, while the other standards terminate the trellis by flushing zeros at the end of each message.

In this chapter, we have performed simulations using the 2400 bps waveform in MIL-STD-188-110A/B. For simplicity we have considered a message size exactly equal to one interleaver block, such that the code trellis termination is for each interleaved block. The message size is then 1434 data bits for the short interleaver and 11514 data bits for the long interleaver setting, plus 6 zeros for trellis termination.

9.2 DFE-based receiver

For comparison, we have implemented a DFE-based receiver in our simulation setup. The implementation is shown in Fig. 9.1. The least-mean-square (LMS) channel estimation algorithm is used to provide an estimate $\hat{\mathbf{h}}_n$ of the time-varying channel impulse response (CIR) $\mathbf{h}_n$. The channel estimator uses the known training symbols and hard decisions from the DFE as a reference input signal. We have also implemented the recursive-least-squares (RLS) algorithm for channel estimation, but the simulated performance is very similar using the much simpler LMS algorithm. This follows from the covariance matrix of the input signal $\hat{x}_n$ to the channel estimator being close to an identity matrix (Eleftheriou and Falconer, 1986; Bershad et al., 1990).

The CIR estimate is updated once per symbol interval and used to calculate the MMSE-optimal DFE coefficients using the algorithm devised in (Shukla and Turner, 1991). It is similar to the Levinson-Durbin algorithm (Haykin, 1996) and has a computational complexity proportional to the square of the number N of filter coefficients, i.e. N^2. The algorithm requires the magnitude of the first coefficient of the CIR estimate to be significantly larger than zero, which is achieved with a suitable symbol timing in the receiver. We assume perfect knowledge of system timing in our simulations, such that symbol synchronization is not an issue.

We pass soft information on the code bits (rather than hard decisions) from the equalizer to the decoder through the deinterleaver, to ensure that the comparison with the turbo equalization-based receiver is fair.

The simulated performance of our DFE-based receiver over an ITU-R poor channel, depicted in Fig. 9.3, is 3 dB better than the requirements in the standard at a bit error rate (BER) of 10^{-5}. This result is as expected, since a portion of the shown gain is due to idealizations made in the simulation setup.

9.3 Turbo equalization-based receiver

The structure of the turbo equalization-based receiver we propose is shown in Fig. 9.2. The SISO equalizer is based on soft ISI cancellation and linear filtering as described in Chap. 6, since a trellis-based SISO equalizer would be too complex for typical HF system parameters. We will now give a brief and simplified summary of the operation of this equalizer: After first-time equalization and decoding, the SISO equalizer uses the LLRs (soft information) fed back from the decoder to compute expected values (means) of each transmitted symbol. For the known training symbols, the mean is equal to the symbol itself. These means, which change during the iterations, are passed through a filter with impulse response equal to the estimated CIR, yielding the expected values of the received symbols. These expected values are then subtracted from the received symbols (“soft ISI cancellation”) and the remain-

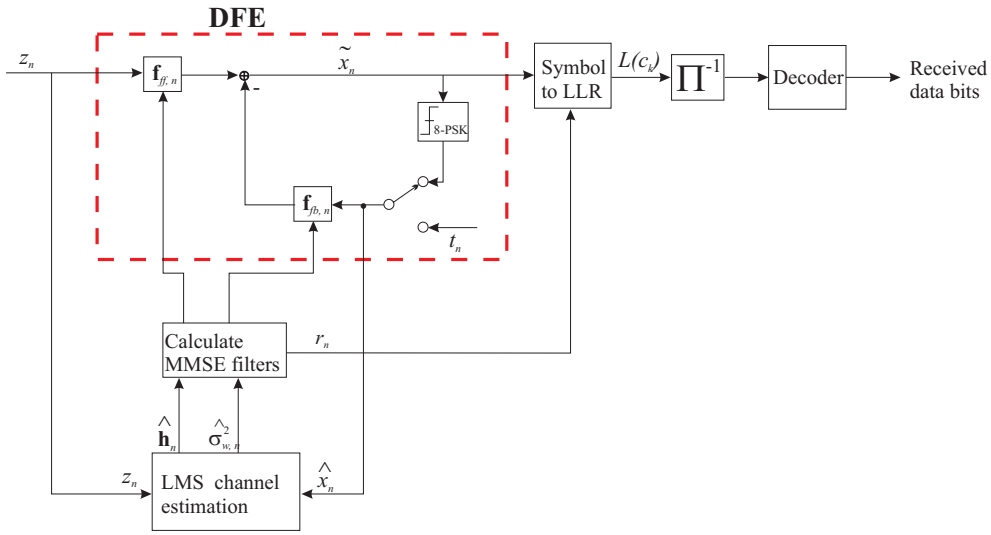
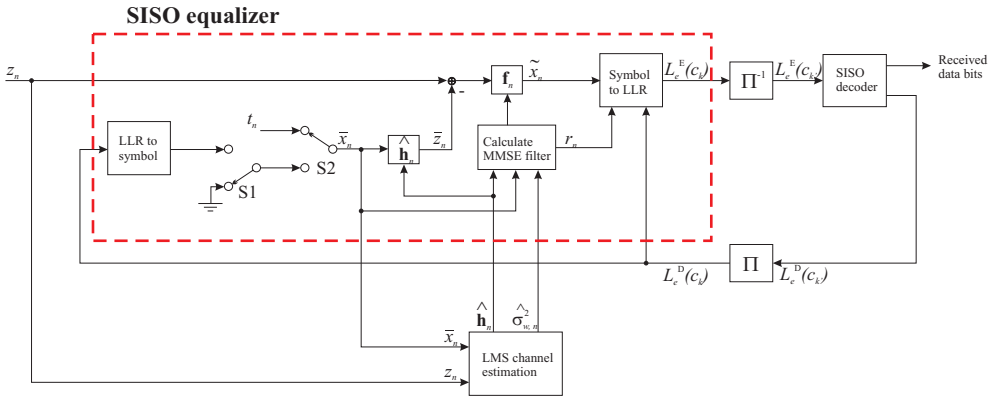


Figure 9.1 DFE-based receiver.



The switch denoted S1 shall be in the position shown during first-time equalization, and in the opposite position during later iterations
The switch denoted S2 shall be in the position shown during known symbols, and in the opposite position during data symbols

Figure 9.2 Turbo equalization-based receiver.

ing signal is passed through a linear filter. The coefficients of this length N filter are calculated using the minimum mean squared error (MMSE) criterion. Because the means vary with time and iteration number, the filter coefficients must be recomputed at each symbol interval and iteration, which can be done recursively with a computational complexity proportional to N^2 per symbol interval (see Sec. 6.1.2). Finally, the symbols at the filter output are converted back to LLRs by assuming a Gaussian distribution on the equalized symbols with known mean and variance, calculated from the value r_n which is a quality measure on the equalized symbol $\tilde{x}_n$.

It should be noted that during first-time equalization (0th iteration), our turbo equalization-based receiver performs worse than the DFE-based receiver. This is partly because the quality of the channel estimate used for first-time equalization is better in the DFE-based receiver (see next paragraph), and partly because the chosen SISO equalizer is a plain MMSE linear equalizer without using soft information from the decoder in the 0th iteration. However, information about the known training symbols is incorporated in a DFE-like manner. After several equalization and decoding tasks, the performance improves tremendously and surpasses the performance of the non-iterative DFE-based receiver as shown in Sec. 9.4.

We have also included soft iterative channel estimation, using an LMS-based channel estimator. When data symbols have been transmitted, soft symbols (the expected values or means of each transmitted symbol) $\bar{x}_n$ fed back from the decoder are used as reference signal to the estimator. This improves the quality of the CIR estimate, since no error propagation can occur when using soft symbols instead of hard decisions. In this way, the quality of the channel estimate improves for each iteration, as the reference symbols become more and more correct. During the 0th iteration, the expected value of each data symbol fed into the channel estimator is zero. It can be shown that the LMS algorithm keeps the channel estimate constant when the reference signal is zero, and therefore in the 0th iteration the channel estimate will only be updated when training symbols have been transmitted. The channel estimate for the 0th iteration will therefore be poorer than when feeding back tentative hard decisions as in the DFE-based receiver, but the calculated LLRs output from the equalizer will be more exact when no hard decisions are used. Therefore, the performance after doing iterations will actually be better when the poorer channel estimate is used for the 0th iteration.

We have used the *ad hoc* error variance estimation algorithm of Sec. 7.6.1 to estimate the time-varying "equivalent" noise variance, caused by channel noise and channel estimation error, when using the estimated CIR rather than the actual CIR.

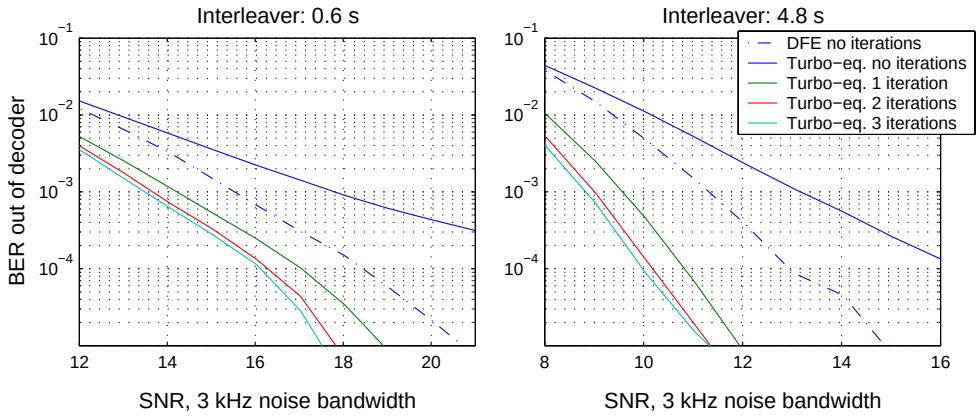


Figure 9.3 Simulated bit error rate performance of the proposed turbo equalization-based receiver, compared to a DFE-based receiver, of MIL-STD-188-110 2400 bps over an approximated ITU-R poor channel (1 Hz Doppler spread, 2.1 ms delay spread). Note the different scaling of the abscissas.

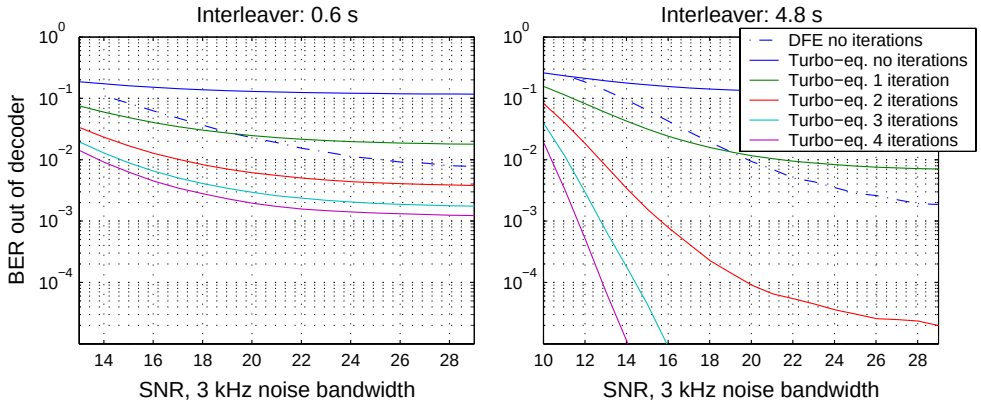


Figure 9.4 Simulated bit error rate performance of the proposed turbo equalization-based receiver, compared to a DFE-based receiver, of MIL-STD-188-110 2400 bps over a channel with 6 Hz Doppler spread and 2.9 ms delay spread. Note the different scaling of the abscissas.

9.4 Simulation results

We have performed simulations of the MIL-STD-188-110A/B 2400 bps waveform over an approximated ITU-R poor channel. The channel model is symbol-spaced and a delay spread of 2.1 ms (exactly 5 symbol intervals) was used instead of the 2.0 ms defined for the ITU-R poor channel. The channel consists of two independent Rayleigh fading taps with a Gaussian fading spectrum with a 2σ Doppler spread of 1 Hz, generated as described in (Furman and Nieto, 2001). No transmit or receive filters were included in the simulations. Note that the simulation results presented here are better than those published in (Otnes and Tüchler, 2002c), for reasons outlined in Sec. 7.4.1.

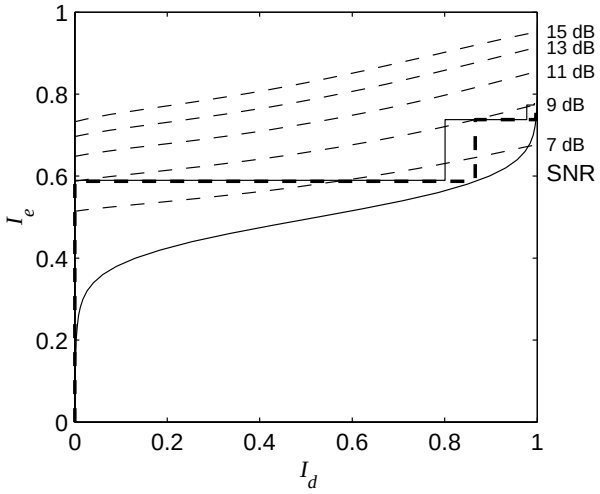
Fig. 9.3 shows the simulated BER performance of our proposed receiver, compared to the DFE-based receiver. The filter length $N = 24$ is the same for both receivers, 16 feedforward and 8 feedback filter taps for the DFE and 15 precursor and 8 postcursor taps for the SISO equalizer (i.e. $N_1 = 15$, $N_2 = 8$). The LMS channel estimation algorithm uses a step size of $\beta = 0.03$, which was found by simulations to be close to the optimum for both receivers on this channel.

The simulation results for the turbo equalization-based receiver show that for an ITU-R poor channel, two iterations (three times equalization and decoding) are enough, i.e., the performance is only marginally improved by using more than two iterations. The required SNR to achieve a certain BER over an ITU-R poor channel is more than 2 dB lower using a turbo equalization-based receiver compared to the DFE-based receiver.

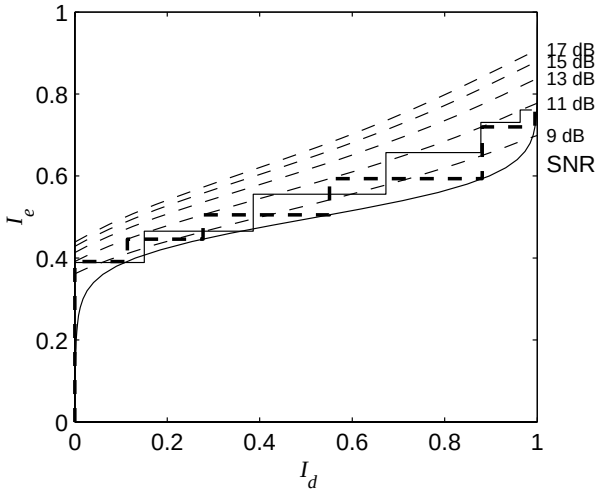
We also performed simulations on a channel varying 6 times faster than the ITU-R poor channel, with results as shown in Fig. 9.4. The step size β of the LMS algorithm was increased in order to track the faster variations of the channel, to 0.05 for the turbo equalization-based receiver and 0.06 for the DFE-based receiver. For this channel, an error floor is seen for both receivers (i.e., the bit error rate does not decrease below some limit no matter how large the SNR is) at the short interleaver setting, because of channel estimation errors when tracking a channel varying this fast. After the 0th iteration, the error floor of the turbo equalization-based receiver is far above the DFE-based receiver. But after some iterations, the error floor of the turbo equalization-based receiver is almost a decade below the DFE-based receiver. For the long interleaver setting, on the other hand, while the error floor of the DFE-based receiver is about 10^{-3}, the error floor of the turbo equalization-based receiver is unobservable in this plot and we see tremendous gains from doing iterations.

9.4.1 EXIT charts

Fig. 9.5 shows EXIT charts, generated as described in Sec. 8.4, corresponding to the simulation results presented in the previous section for the turbo equalization-based



(a) Approximated ITU-R poor channel: 1 Hz Doppler spread, 2.1 ms delay spread. Trajectories shown for 9 dB SNR.



(b) Fast-varying channel: 6 Hz Doppler spread, 2.9 ms delay spread. Trajectories shown for 11 dB SNR.

Figure 9.5 EXIT charts corresponding to the simulations in Figs. 9.3-9.4. Solid curve is transfer function for decoder, dashed curves are transfer functions for equalizer at different SNRs. The predicted trajectory is shown as dashed lines, and the actual simulated trajectory using the long interleaver (23040 code bits) is shown as solid lines.

receiver. We note that the difference between prediction and reality is now larger than in Fig. 8.6, even though the interleaver lengths are comparable. We have not investigated this matter in detail, but suspect that the reason is one (or both) of the following: We now use a relatively regular interleaver permutation, whereas an S-random interleaver was used in Chap. 8. Also, the constraint length 6 code used in the HF standards may be more sensitive to correlations in LLRs than the constraint length 2 code used in Chap. 8.

Still, the predicted and real trajectories follow roughly the same paths through the EXIT charts, and we can therefore draw general conclusions on the convergence behavior from the EXIT chart. For example, we note that the slope of the equalizer transfer function is steeper when the channel is varying faster, because of problems in tracking the channel in the left-hand part of the chart (where data symbols can not be used as a reference signal between training sequences). Therefore, for the fast-varying channel, the gain from doing iterations is larger but more iterations are required to reach convergence. By comparing the two EXIT charts of Fig. 9.5, we can extrapolate the effect of increasing the Doppler spread further beyond 6 Hz: The equalizer and decoder transfer functions will start intersecting in the left-hand part of the EXIT chart, such that the iterative receiver will not converge. This will be the limiting factor for the Doppler spread capability of our proposed receiver.

9.5 Conclusions and future work

We have proposed a turbo equalization-based receiver for serial-tone HF waveforms. The SISO equalizer used in the turbo equalization setup is a suboptimal equalizer with reasonable complexity, based on a linear filter and soft ISI cancellation.

Simulation results for the ITU-R poor channel show that our proposed receiver using turbo equalization performs more than 2 dB better than a conventional DFE-based receiver over an ITU-R poor channel. In Chap. 3 we have found, using databases of HF channel measurements, that a sensitivity gain of 2 dB can yield significant improvements in the availability of the waveform.

The turbo equalization-based receiver is more complex than the DFE-based receiver: Equalization and decoding has to be performed several times, and each equalization and decoding task is more complex than for the DFE-based receiver because of the SISO capability. We still think that the necessary computational burden is manageable.

Future work includes simulations on other waveforms (in particular the high-rate waveforms of STANAG 4539) and/or channel conditions. Also, the receiver should be modified to accommodate fractionally-spaced input symbols, as outlined at the end of Chaps. 6 and 7, and algorithms for the recovery of the symbol timing. Practical implementation and complexity issues should also be addressed.

We finally note that application of turbo equalization to HF waveforms has also been discussed in (Langlais and Hélard, 2002), and in a recent paper (Nieto, 2002) the performance of turbo equalization for the 9600 bps waveform of STANAG 4539 has been investigated. Without disclosing any details about the implementation, they report gains of 2-5 dB (for different types of channels) by introducing turbo equalization, compared to a conventional receiver. This is similar to the gains reported in our work for a 2400 bps waveform.

Chapter 10

Concluding remarks

This doctoral work was started in January 2000 with the vaguely defined goal of increasing the availability of HF communications at high latitudes, and an idea that this could be achieved using some kind of iterative processing in the receiver. We went down two parallel paths to address the problem: The channel characteristics were studied using measured data, and an extensive literature search on iterative processing was conducted. Also, the contact with Michael Tüchler gave a boost to the understanding of turbo equalization.

At the time when conclusions could be drawn from the channel analyses, the understanding of iterative processing in general and turbo equalization in particular had reached a point where we could start experimenting with turbo equalization applied to HF waveforms. By the end of 2001, we had simulations running where a conventional DFE-based receiver could be compared to a turbo equalization-based receiver using soft iterative channel estimation.

The year 2002 was used to write conference papers addressing different aspects of iterative channel estimation, equalization, and decoding for time-varying channels, to improve the turbo equalization-based receiver by applying new knowledge, and to prepare this dissertation. Because of a strong intent to finish the doctoral work on time, several threads have been left open for further research, giving a dissertation which is quite broad but could have delved more deeply into details.

We do however consider the achievements to be encouraging: We have designed the baseband signal processing required for applying turbo equalization to HF serial-tone waveforms, and to do that we had to address two problems which have received little attention in the literature on turbo equalization: SISO equalization algorithms for time-varying channels, and the concept of soft iterative channel estimation.

Our future goal is now to see the proposed algorithms implemented in real-world modems, and not just running as software on a computer.

URN:NBN:no-3359

References

Anastasopoulos, A. and Chugg, K. M.: 2000, Adaptive soft-input soft-output algorithms for iterative detection with parametric uncertainty, *IEEE Trans. Communications* **48**(10), 1638–1649.

Anderson, J. B. and Hladik, S. M.: 1998, Tailbiting MAP decoders, *IEEE J. Sel. Areas in Communications* **16**(2), 297–302.

Angling, M. J. and Cannon, P. S.: 1996, An analysis of DAMSON data applicable to the design of the robust waveform, *Technical Report DRA/CIS(CIS1)/TP96006/1.0*, Defence Research Agency, Malvern, UK.

Angling, M. J., Cannon, P. S., Davies, N. C., Willink, T. J., Jodalen, V. and Lundborg, B.: 1998, Measurements of Doppler and multipath spread on oblique high-latitude HF paths and their use in characterizing data modem performance, *Radio Science* **33**(1), 97–107.

Arthur, P. C. and Maundrell, M. J.: 1997, Multi-dimensional HF modem performance characterisation, *IEE Conf. Publ. No. 411: 7th Int. Conf. on HF Radio Systems and Techniques*, IEE, Nottingham, UK, pp. 154–158.

Ashikhmin, A., Kramer, G. and ten Brink, S.: 2002, Extrinsic information transfer functions: A model and two properties, *Proc. Conf. Information Sciences and Systems, CISS 2002*, Princeton University, NJ, USA, pp. 742–747.

Baccarelli, E.: 2000, On the performance limits of TCM in fast-fading multipath channels with combined equalization / decoding, *IEEE Trans. Communications* **48**(12), 1957–1964.

Baccarelli, E. and Cusani, R.: 1998, Combined channel estimation and data detection using soft statistics for frequency-selective fast-fading digital links, *IEEE Trans. Communications* **46**(4), 424–427.

Bahl, L. R., Cocke, J., Jelinek, F. and Raviv, J.: 1974, Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Information Theory* **IT-20**, 284–287.

Bartlett, A., Brunt, S. M. and Darnell, M.: 1999, Comparison of DFE and MLSE equalisation in a HF serial tone modem and implications for frequency selection, *IEE Seminar Digest 017: Frequency Selection and Management Techniques for HF Communications*, IEE, London, UK, pp. 15.1–15.7.

Bauch, G. and Franz, V.: 1998, A comparison of soft-in/soft-out algorithms for "turbo-detection", *Proc. 5th Int. Conf. on Telecommunications*, Porto Carras, Greece, pp. 259–263.

Bauch, G., Khorram, H. and Hagenauer, J.: 1997, Iterative equalization and decoding in mobile communications systems, *ITG-Fachbericht 145 (1997): 2nd Eur. Personal Mobile Communications Conf. together with 3rd ITG-Fachtagung "Mobile Kommunikation"*, ITG, Bonn, Germany, pp. 307–312.

Bello, P. A.: 1963, Characterization of randomly time-variant linear channels, *IEEE Trans. Communication Systems* **11**(4), 360–393.

Benedetto, S., Divsalar, D., Montorsi, G. and Pollara, F.: 1996, Algorithm for continous decoding of turbo codes, *IEE Electronics Letters* **32**(4), 314–315.

Benedetto, S., Divsalar, D., Montorsi, G. and Pollara, F.: 1998, Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding, *IEEE Trans. Information Theory* **44**(3), 909–926.

Benedetto, S. and Montorsi, G.: 1996, Serial concatenation of block and convolutional codes, *IEE Electronics Letters* **32**(10), 887–888.

Berrou, C., Glavieux, A. and Thitimajshima, P.: 1993, Near Shannon limit error-correcting coding and decoding: Turbo-codes, *Proc. IEEE Int. Conf. on Communications, ICC '93*, Vol. 2, IEEE, Geneva, Switzerland, pp. 1064–1070.

Bershad, N. J., McLaughlin, S. and Cowan, C. F. N.: 1990, Performance comparison of RLS and LMS algorithms for tracking a first order Markov communications channel, *Proc. IEEE Int. Symp. on Circuits and Systems*, Vol. 1, IEEE, New Orleans, LA, USA, pp. 266–270.

Berthet, A. O., Penther, B., Visoz, R. and Boutros, J. J.: 2001, A new reduced complexity turbo-detector for highly selective long delay spread ISI channels: A solution for 4G mobile systems?, *Proc. 53rd IEEE VTS Vehicular Tech. Conf., VTC 2001 Spring*, Vol. 3, IEEE, Rhodes, Greece, pp. 1654–1658.

Berthet, A., Visoz, R. and Tortelier, P.: 2000, Sub-optimal turbo-detection for coded 8-PSK signals over ISI channels with application to EDGE advanced mobile system, *Proc. 11th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, PIMRC 2000*, Vol. 1, IEEE, London, UK, pp. 151–157.

Blackert, W. J., Hall, E. K. and Wilson, S. G.: 1995, Turbo code termination and interleaver conditions, *IEE Electronics Letters* **31**(24), 2082–2084.

Blackmon, F., Sozer, E., Murandian, M., Proakis, J. and Salehi, M.: 2001, Performance comparison of iterative/integral equalizer/decoder structures for underwater acoustic channels, *Proc. MTS/IEEE Conf. OCEANS 2001*, Vol. 4, IEEE, Honolulu, HI, USA, pp. 2191–2200.

Bröms, M., Lundborg, B., Jodalen, V. and Bergsvik, T.: 1997, Doppler effects on high latitude HF paths during an ionospheric disturbance, *IEE Conf. Publ. No. 411: 7th Int. Conf. on HF Radio Systems and Techniques*, IEE, Nottingham, UK, pp. 75–79.

Cannon, P. S., Angling, M. J., Davies, N. C., Willink, T., Jodalen, V., Jacobsen, B., Lundborg, B. and Bröms, M.: 2000, DAMSON HF channel characterization - a review, *Proc. MILCOM 2000*, Vol. 1, IEEE, Los Angeles, CA, USA, pp. 59–64.

Chamberlain, M. W.: 2002, Proposed data link protocol enhancements for STANAG-4538, *Proc. MILCOM 2002*, IEEE, Anaheim, CA, USA.

Chugg, K., Anastasopoulos, A. and Chen, X.: 2001, *Iterative Detection*, Kluwer Academic Publishers, Boston, MA, USA.

Chung, S.-Y., Richardson, T. J. and Urbanke, R. L.: 2001, Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation, *IEEE Trans. Information Theory* **47**(2), 657–670.

Clark, A. P. and Harun, R.: 1986, Assessment of Kalman-filter channel estimators for an HF radio link, *IEE Proc. Pt. F* **133**(6), 513–521.

Clarke, R. H.: 1968, A statistical theory of mobile-radio reception, *Bell Syst. Tech. J.* **47**, 957–1000.

Crozier, S. N., Falconer, D. D. and Mahmoud, S. A.: 1991, Least sum of squared errors (LSSE) channel estimation, *IEE Proc. Pt. F* **138**(4), 371–378.

Davies, K.: 1990, *Ionospheric Radio*, Peter Peregrinus Ltd., Stevenage, UK.

Davis, L. M., Collins, I. B. and Hoeher, P.: 2001, Joint MAP equalization and channel estimation for frequency-selective and frequency-flat fast-fading channels, *IEEE Trans. Communications* **49**(12), 2106–2114.

Dejonghe, A. and Vandendorpe, L.: 2002, Turbo-equalization for multilevel modulation: An efficient low-complexity scheme, *Proc. IEEE Int. Conf. on Communications, ICC 2002*, Vol. 3, IEEE, New York, NY, USA, pp. 1863–1867.

Divsalar, D., Dolinar, S. and Pollara, F.: 2001, Iterative turbo decoder analysis based on density evolution, *IEEE J. Sel. Areas in Communications* **19**(5), 891–907.

Douillard, C., Jézéquel, M., Berrou, C., Picart, A., Didier, P. and Glavieux, A.: 1995, Iterative correction of intersymbol interference: Turbo-equalization, *Eur. Trans. Telecommunications* **6**(5), 507–511.

Eggen, T. H., Baggeroer, A. B. and Preisig, J. C.: 2000, Communication over Doppler spread channels - part I: Channel and receiver presentation, *IEEE J. Oceanic Engineering* **25**(1), 62–71.

Eggen, T. H., Preisig, J. C. and Baggeroer, A. B.: 2001, Communication over Doppler spread channels - II: Receiver characterization and practical results, *IEEE J. Oceanic Engineering* **26**(4), 612–621.

Eleftheriou, E. and Falconer, D. D.: 1986, Tracking properties and steady-state performance of RLS adaptive filter algorithms, *IEEE Trans. Acoustics, Speech and Signal Processing* **ASSP-34**(5), 1097–1110.

Elvy, S. J.: 2001, Comparison of second and third generation HF communication links, *RTO MP-065: RTO IST Symp. on "Military Communications"*, NATO, Warsaw, Poland, pp. 16.1–16.8.

ETS: 2001, ETSI TS 101 980 v1.1.1: Digital Radio Mondiale (DRM); system specification.

Falconer, D. D., Sheikh, A. U. H., Eleftheriou, E. and Tobis, M.: 1985, Comparison of DFE and MLSE receiver performance on HF channels, *IEEE Trans. Communications* **COM-33**(5), 484–486.

Farhang-Boroujeny, B.: 1993, Channel equalization via channel identification for rapidly fading HF channels, *Proc. IEEE Singapore Int. Conf. on Information Engineering '93*, Vol. 2, IEEE, Singapore, pp. 563–567.

Fechtel, S. A. and Meyr, H.: 1991, An investigation of channel estimation and equalization techniques for moderately rapid fading HF-channels, *Proc. IEEE Int. Conf. on Communications, ICC '91*, Vol. 2, IEEE, Denver, CO, USA, pp. 768–772.

Fechtel, S. A. and Meyr, H.: 1995, Optimal linear interpolative channel estimation for TDMA-based personal mobile communications over frequency-selective channels, *Proc. 45th IEEE VTS Vehicular Tech. Conf., VTC 1995*, Vol. 1, IEEE, Chicago, IL, USA, pp. 321–325.

FED-STD-1052: Telecommunications: HF Radio Modems: 1996, General Services Administration, Office of Information Resource Management, USA.

Feher, K. (ed.): 1987, *Advanced Digital Communications*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Feng, W., Yuan, J. and Vucetic, B. S.: 2002, A code-matched interleaver design for turbo codes, *IEEE Trans. Communications* **50**(6), 926–937.

Forney, Jr., G. D.: 1972, Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference, *IEEE Trans. Information Theory* **IT-18**(3), 363–378.

Furman, W. N. and Nieto, J. W.: 2001, Understanding HF channel simulator requirements in order to reduce HF modem performance measurement variability, *Proc. 6th Nordic Shortwave Conf., HF 01*, Nordic Radio Society, Fårö, Sweden, pp. 6.4.1–6.4.13.

Gallager, R. G.: 1963, *Low-Density Parity-Check Codes*, MIT Press, Cambridge. MA, USA.

Gamal, H. E. and Hammons, Jr., A. R.: 2001, Analyzing the turbo decoder using the Gaussian approximation, *IEEE Trans. Information Theory* **47**(2), 671–686.

Garello, R., Montorsi, G., Benedetto, S. and Cancellieri, G.: 2001, Interleaver properties and their applications to the trellis complexity analysis of turbo codes, *IEEE Trans. Communications* **49**(5), 793–807.

George, D. A., Bowen, R. R. and Storey, J. R.: 1971, An adaptive decision feedback equalizer, *IEEE Trans. Communication Technology* **COM-19**(3), 281–293.

Gersho, A. and Lim, T. L.: 1981, Adaptive cancellation of intersymbol interference for data transmission, *Bell Syst. Tech. J.* **60**(11), 1997–2021.

Giles, T. C.: 1995, *On the Design of HF Radio Modems*, PhD thesis, The University of Adelaide, Australia.

Glavieux, A., Laot, C. and Labat, J.: 1997, Turbo equalization over a frequency selective channel, *Proc. Int. Symp. on Turbo Codes & Related Topics*, ENST Bretagne, Brest, France, pp. 96–102.

Godard, D.: 1974, Channel equalization using a Kalman filter for fast data transmission, *IBM J. Res. Develop.* pp. 267–273.

Goodman, J. M.: 1992, *HF Communications. Science and Technology*, Van Nostrand Reinhold, New York, NY, USA.

Hagenauer, J.: 1997, The turbo principle: Tutorial introduction and state of the art, *Proc. Int. Symp. on Turbo Codes and Related Topics*, ENST Bretagne, Brest, France, pp. 1–11.

Hagenauer, J. and Hoeher, P.: 1989, A Viterbi algorithm with soft-decision outputs and its applications, *Proc. IEEE Global Telecommunications Conf., GLOBECOM '89*, Vol. 3, IEEE, Dallas, TX, USA, pp. 1680–1686.

Hall, E. K. and Wilson, S. G.: 1998, Design and analysis of turbo codes on Rayleigh fading channels, *IEEE J. Sel. Areas in Communications* **16**(2), 160–174.

Hall, M. P. M., Barclay, L. W. and Hewitt, M. T. (eds): 1996, *Propagation of Radiowaves*, The Institution of Electrical Engineers, London, UK.

Hanzo, L., Liew, T. H. and Yeap, B. L.: 2002, *Turbo Coding, Turbo Equalization and Space-Time Coding for Transmission over Fading Channels*, John Wiley & Sons Ltd, Chichester, UK.

Hanzo, L., Webb, W. and Keller, T.: 2000, *Single- and Multi-Carrier Quadrature Amplitude Modulation*, John Wiley & Sons, Chichester, UK.

Hanzo, L., Wong, C. H. and Yee, M. S.: 2002, *Adaptive Wireless Transceivers: Turbo-Coded, Turbo-Equalized and Space-Time Coded TDMA, CDMA and OFDM Systems*, John Wiley & Sons Ltd, Chichester, UK.

Haykin, S.: 1996, *Adaptive Filter Theory, 3rd Edition*, Prentice Hall, Upper Saddle River, NJ, USA.

Heegard, C. and Wicker, S. B.: 1999, *Turbo Coding*, Kluwer Academic Publishers, Boston, MA, USA.

Herzog, R., Hagenauer, J. and Schmidbauer, A.: 1997, Soft-in/soft-out Hadamard despreader for iterative decoding in the IS-95(A) system, *Proc.47th IEEE VTS Vehicular Technology Conf., VTC 1997*, Vol. 2, IEEE, Phoenix, AZ, USA, pp. 1219–1222.

Herzog, R., Schmidbauer, A. and Hagenauer, J.: 1997, Iterative decoding and despreading improves CDMA-systems using M-ary orthogonal modulation and FEC, *Proc. IEEE Int. Conf. on Communications, ICC '97*, Vol. 2, IEEE, Montreal, Que., Canada, pp. 909–913.

Hokfelt, J., Edfors, O. and Maseng, T.: 2001, A turbo code interleaver design criterion based on the performance of iterative decoding, *IEEE Communications Letters* **5**(2), 52–54.

Hsu, F. M.: 1982, Square root Kalman filtering for high-speed data received over fading dispersive HF channels, *IEEE Trans. Information Theory* **IT-28**(5), 753–763.

ITU: 1992, Rec. ITU-R F.520 (withdrawn): Use of high frequency ionospheric channel simulators.

ITU: 2000, Rec. ITU-R F.1487: Testing of HF modems with bandwidths of up to about 12 KHz using ionospheric channel simulators.

Jakes, W. C.: 1974, *Microwave Mobile Communications*, Wiley, New York, NY, USA.

Jodalen, V.: 1996, A study of observed and predicted HF propagation characteristics at high latitudes, *Publication FFI/PUBLICATION-96/01107*, Norwegian Defence Research Establishment, Kjeller, Norway.

Jodalen, V., Bergsvik, T., Cannon, P. S. and Arthur, P. C.: 2001, Performance of HF modems on high-latitude paths using multiple frequencies, *Radio Science* **36**(6), 1687–1698.

Johnson, E. E.: 2002, ARQ protocol features for HF radio links, *Proc. MILCOM 2002*, IEEE, Anaheim, CA, USA.

Johnson, E. E., Desourdis, Jr., R. I., Earle, G. D., Cook, S. C. and Ostergaard, J. C.: 1997, *Advanced High-Frequency Radio Communications*, Artech House, Boston, MA, USA.

Jorgenson, M. B., Johnson, R. and Moreland, K. W.: 1999, Beyond 9600 bps at HF, *RTO MP-026: RTO IST Symp. on "Tactical Mobile Communications"*, NATO, Lillehammer, Norway, pp. 34.1–34.8.

Jorgenson, M. B., Johnson, R. W., Moreland, K. W., Bova, W. M. and Jones, P. F.: 2000, Meeting military requirements for increased data rates at HF, *Proc. MILCOM 2000*, Vol. 2, IEEE, Los Angeles, CA, USA, pp. 1149–1153.

Jorgenson, M. B. and Moreland, K. W.: 1999, HF serial-tone waveform design, *RTO MP-026: RTO IST Symp. on "Tactical Mobile Communications"*, NATO, Lillehammer, Norway, pp. 33.1–33.10.

Jursa, A. S.: 1985, *Handbook of Geophysics and the Space Environment*, Air Force Geophysics Laboratory, Air Force Systems Command, US Air Force, USA.

Kallgren, D. G. and Smaal, J.-W.: 2001, IP unicast/multicast operation over STANAG 5066, *Proc. MILCOM 2001*, Vol. 1, IEEE, McLean, VA, USA, pp. 501–505.

Kammeyer, K.-D., Kühn, V. and Petermann, T.: 2001, Blind and nonblind turbo estimation for fast fading GSM channels, *IEEE J. Sel. Areas in Communications* **19**(9), 1718–1728.

Kotlowski, A., Brakemeier, A., Mouy, B., Coutulleau, M., Vila, P. and Mérel, D.: 1998, High data rate modem : Comparison of serial and parallel waveforms for shortwave communications, *Proc. 5th Nordic Shortwave Conf., HF 98*, Nordic Radio Society, Fårö, Sweden.

Kubo, H., Murakami, K. and Fujino, T.: 1994, An adaptive maximum-likelihood sequence estimator for fast time-varying intersymbol interference channels, *IEEE Trans. Communications* **42**(2/3/4), 1872–1880.

Kuhn, C.: 2002, *Iterative Kanalschätzung, Kanalentzerrung und Decodierung für den EDGE-Standard*, Master's thesis, Munich University of Technology, Munich, Germany.

Lampe, A.: 2002, Iterative multiuser detection with integrated channel estimation for coded DS-CDMA, *IEEE Trans. Communications* **50**(8), 1217–1223.

Langlais, C. and Hélard, M.: 2002, Optimization of the equalization and the interleaving in turbo-equalization for a frequency-selective fading channel, *Proc. IEEE Int. Conf. on Communications, ICC 2002*, Vol. 3, IEEE, New York, NY, USA, pp. 1868–1872.

Laot, C., Glavieux, A. and Labat, J.: 2001, Turbo equalization: Adaptive equalization and channel decoding jointly optimized, *IEEE J. Sel. Areas in Communications* **19**(9), 1744–1752.

Lee, E. A. and Messerschmitt, D. G.: 1994, *Digital Communication*, 2nd edn, Kluwer Academic Publishers, Boston, MA, USA.

Lee, I.: 2001, The effect of a precoder on serially concatenated coding systems with an ISI channel, *IEEE Trans. Communications* **49**(7), 1168–1175.

Lin, S., Costello, Jr., D. J. and Miller, M. J.: 1984, Automatic-repeat-request error-control schemes, *IEEE Communications Magazine* **22**(12), 5–17.

Magee, Jr., F. R. and Proakis, J. G.: 1973, Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference, *IEEE Trans. Information Theory* **IT-19**(1), 120–124.

Marsland, I. D. and Mathiopoulos, P. T.: 1998, Multiple differential detection of parallel concatenated convolutional (turbo) codes in correlated fast Rayleigh fading, *IEEE J. Sel. Areas in Communications* **16**(2), 265–275.

Mastrangelo, J. F., Lemmon, J. J., Vogler, L. E., Hoffmeyer, J. A., Pratt, L. E. and Behm, C. J.: 1997, A new wideband high frequency channel simulation system, *IEEE Trans. Communications* **45**(1), 26–34.

McLaughlin, S.: 1991, Adaptive equalisation via Kalman filtering techniques, *IEE Proc. Pt. F* **138**(4), 388–396.

McNicol, R. W. E.: 1949, The fading of radio waves of medium and high frequencies, *Proc. of the IEE* **96, pt. III**, 517–524.

MIL-STD-188-110B: Interoperability and Performance Standards for Data Modems: 2000, US Department of Defense, USA.

MIL-STD-188-141B: Interoperability and Performance Standards for Medium and High Frequency Radio Systems: 1999, US Department of Defence, USA.

Molisch, A. F. (ed.): 2001, *Wideband Wireless Digital Communications*, Prentice-Hall, Upper Saddle River, NJ, USA.

Narayanan, K. R.: 2001, Effect of precoding on the convergence of turbo equalization for partial response channels, *IEEE J. Sel. Areas in Communications* **19**(4), 686–698.

Nefedov, N. and Pukkila, M.: 2000, Turbo equalization and iterative (turbo) estimation techniques for packet data transmission, *Proc. 2nd Int. Symp. on Turbo Codes & Related Topics*, ENST Bretagne, Brest, France, pp. 423–426.

Nieto, J. W.: 2001, Does modem performance really matter on HF channels? An investigation of serial-tone and parallel-tone waveforms, *Proc. 6th Nordic Shortwave Conf., HF 01*, Fårö, Sweden, pp. 3.4.1–3.4.15.

Nieto, J. W.: 2002, Iterative equalization and decoding of STANAG 4539 9600 bps HF waveforms, *Proc. 3rd Int. Symp. on Communications Systems, Networks and Digital Signal Processing, CSNDSP 2002*, Staffordshire, UK.

Nowlan, S. J. and Hinton, G. E.: 1993, A soft decision-directed LMS algorithm for blind equalization, *IEEE Trans. Communications* **41**(2), 275–279.

Otnes, R.: 2001a, Factors affecting the availability of medium data rate waveforms at high latitude HF channels, *Proc. 6th Nordic Shortwave Conf., HF 01*, Nordic Radio Society, Fårö, Sweden, pp. 6.1.1–6.1.10.

Otnes, R.: 2001b, Guidelines for increasing the availability of low and medium data rates at high latitude HF channels, *RTO MP-065: RTO IST Symp. on "Military Communications"*, NATO, Warsaw, Poland, pp. 19.1–19.8.

Otnes, R. and Jodalen, V.: 2001, Increasing the availability of medium data rates at high latitude HF channels, *Proc. MILCOM 2001*, Vol. 1, IEEE, McLean, VA, USA, pp. 437–441.

Otnes, R. and Maseng, T.: 2001, Adaptive data rate using ARQ and nonuniform constellations, *Proc. 53rd IEEE VTS Vehicular Tech. Conf., VTC 2001 Spring*, Vol. 2, IEEE, Rhodes, Greece, pp. 1211–1215.

Otnes, R. and Tüchler, M.: 2001, Block SISO linear equalizers for turbo equalization in serial-tone HF modems, *Proc. Norwegian Signal Processing Symp., NORSIG-2001*, NORSIG, Trondheim, Norway, pp. 93–98.

Otnes, R. and Tüchler, M.: 2002a, Adaptive turbo equalization for time-varying frequency-selective channels. Submitted to IEEE Trans. Wireless Communications.

Otnes, R. and Tüchler, M.: 2002b, EXIT chart analysis applied to adaptive turbo equalization, *Proc. 5th Nordic Signal Processing Symp., NORSIG-2002*, NORSIG, Hurtigruten Tromsø–Trondheim, Norway.

Otnes, R. and Tüchler, M.: 2002c, Improved receivers for digital high frequency waveforms using turbo equalization, *Proc. MILCOM 2002*, IEEE, Anaheim, CA, USA.

Otnes, R. and Tüchler, M.: 2002d, Low-complexity turbo equalization for time-varying channels, *Proc. 55th IEEE VTS Vehicular Technology Conf., VTC 2002 Spring*, Vol. 1, IEEE, Birmingham, AL, USA, pp. 140–144.

Otnes, R. and Tuchler, M.: 2002e, Soft iterative channel estimation for turbo equalization: Comparison of channel estimation algorithms, *Proc. 8th IEEE Int. Conf. on Communication Systems, ICCS 2002*, IEEE, Singapore, pp. 72–76.

Otnes, R. and Tüchler, M.: 2003, On iterative equalization, estimation, and decoding, *to be presented at Int. Conf. on Communications, ICC 2003*, IEEE, Anchorage, AK, USA.

Poor, H. V.: 1994, *An Introduction to Signal Detection and Estimation*, 2nd edn, Springer-Verlag, New York, NY, USA.

Proakis, J. G.: 1995, *Digital Communications*, 3rd edn, McGraw-Hill, New York, NY, USA.

Pukkila, M.: 2000, Turbo equalisation for the enhanced GPRS system, *Proc. 11th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications, PIMRC 2000*, Vol. 2, IEEE, London, UK, pp. 893–897.

Raheli, R., Polydoros, A. and Tzou, C.-K.: 1995, Per-survivor processing: A general approach to MLSE in uncertain environments, *IEEE Trans. Communications* **43**(2/3/4), 354–364.

Raphaeli, D. and Saguy, A.: 2000, Linear equalizers for turbo equalization: A new optimisation criterion for determining the equalizer taps, *Proc. 2nd Int. Symp. on Turbo Codes & Related Topics*, ENST Bretagne, Brest, France, pp. 371–374.

Raphaeli, D. and Saguy, A.: 2002, Reduced complexity APP for turbo equalization, *Proc. IEEE Int. Conf. on Communications, ICC 2002*, Vol. 3, IEEE, New York, NY, USA, pp. 1940–1943.

Raphaeli, D. and Zarai, Y.: 1997, Combined turbo equalization and turbo decoding, *Proc. Int. Symp. on Turbo Codes & Related Topics*, ENST Bretagne, Brest, France, pp. 180–183.

Renfree, P.: 2001, The U.S. navy returns to HF with STANAG 5066 as the path, *Proc. MILCOM 2001*, Vol. 1, IEEE, McLean, VA, USA, pp. 471–476.

Reynolds, D. and Wang, X.: 2001, Low-complexity turbo equalization for diversity channels, *Signal Processing* **81**, 989–995.

Richardson, T.: 2000, The geometry of turbo-decoding dynamics, *IEEE Trans. Information Theory* **46**(1), 9–23.

Richardson, T. J., Shrokrollahi, M. A. and Urbanke, R. L.: 2001, Design of capacity-approaching irregular low-density parity-check codes, *IEEE Trans. Information Theory* **47**(2), 619–637.

Sellathurai, M. and Haykin, S.: 2002, TURBO-BLAST for wireless communications: Theory and experiments, *IEEE Trans. Signal Processing* **50**(10), 2538–2546.

Shaver, H. N., Tupper, B. C. and Lomax, J. B.: 1967, Evaluation of a Gaussian HF channel model, *IEEE Trans. Communication Technology* **COM-15**(1), 79–88.

Shukla, P. K. and Turner, L. F.: 1991, Channel-estimation-based adaptive DFE for fading multipath radio channels, *IEE Proc. Pt. I* **138**(6), 525–543.

Shukla, P. K. and Turner, L. F.: 1992, Examination of an adaptive DFE and MLSE/near-MLSE for fading multipath radio channels, *IEE Proc. Pt. I* **139**(4), 418–428.

Sklar, B.: 1997a, Rayleigh fading channels in mobile digital communication systems. Part I: Characterization, *IEEE Communications Magazine* (7), 90–100.

Sklar, B.: 1997b, Rayleigh fading channels in mobile digital communication systems. Part II: Mitigation, *IEEE Communications Magazine* (7), 102–109.

Song, S., Singer, A. C. and Sung, K.-M.: 2002, Turbo equalization with an unknown channel, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 2002*, Vol. 3, IEEE, Orlando, FL, USA, pp. 2805–2808.

Stallings, W.: 1997, *Data and Computer Communications*, 5th edn, Prentice Hall, Upper Saddle River, NJ, USA.

STANAG 4285: Characteristics of 1200/2400/3600 Bits Per Second Single Tone Modulators/Demodulators for HF Radio Links: 1989, NATO.

STANAG 4415: Characteristics of a Robust, Non-Hopping, Serial Tone Modulator/Demodulator for Severely Degraded HF Radio Links: 1999, NATO.

STANAG 4444: Technical Standards for a Slow-Hop HF EPM Communications System: 1999, NATO.

STANAG 4538: Technical Standards for an Automatic Radio Control System (ARCS) for HF Communication Links: 2000, NATO.

STANAG 4539: Technical Standards for Non-Hopping HF Communications Waveforms: 2000, NATO.

STANAG 5066: Profile for High Frequency (HF) Radio Data Communications: 1999, NATO. Version 1.2.

Steele, R. and Hanzo, L. (eds): 1999, *Mobile Radio Communications: Second and Third-Generation Cellular and WATM Systems*, 2nd edn, John Wiley & Sons Ltd, Chichester, UK.

Strauch, P., Luschi, C. and Kuzminskiy, A. M.: 2000, Iterative channel estimation for EPGRS, *Proc. 52nd IEEE VTS Vehicular Tech. Conf., VTC 2000 Fall*, Vol. 5, IEEE, Boston, MA, USA, pp. 2271–2277.

Strauch, P., Luschi, C., Sandell, M. and Yan, R.: 1999, Turbo equalization for an 8-PSK modulation scheme in a mobile TDMA communication system, *Proc. 50th IEEE VTS Vehicular Tech. Conf., VTC 1999 Fall*, Vol. 3, IEEE, Amsterdam, Netherlands, pp. 1605–1609.

Tantikovit, S., Sheikh, A. U. H. and Wang, M. Z.: 1998, Code-aided adaptive equalizer for mobile communication systems, *IEE Electronics Letters* **34**(17), 1638–1640.

ten Brink, S.: 1999, Convergence of iterative decoding, *IEE Electronics Letters* **35**(10, 13), 806–808, 1117–1119.

ten Brink, S.: 2000a, Iterative decoding trajectories of parallel concatenated codes, *ITG-Fachbericht 159 (2000): 3rd IEEE/ITG Conf. on Source and Channel Coding*, ITG, Munich, Germany, pp. 75–80.

ten Brink, S.: 2000b, Rate one-half code for approaching the Shannon limit by 0.1 dB, *IEE Electronics Letters* **36**(15), 1293–1294.

ten Brink, S.: 2001, Convergence behavior of iteratively decoded parallel concatenated codes, *IEEE Trans. Communications* **49**(10), 1727–1737.

Thomas, J. and Geraniotis, E.: 2002, Space-time iterative receivers for narrowband multichannel networks, *IEEE Trans. Communications* **50**(7), 1049–1054.

Trinder, S. E. and Gillespie, A. F. R.: 2001, Optimisation of the STANAG 5066 ARQ protocol to support high data rate HF communications, *Proc. MILCOM 2001*, Vol. 1, IEEE, McLean, VA, USA, pp. 482–486.

Tüchler, M.: 2000, *Iterative equalization using priors*, Master's thesis, University of Illinois at Urbana-Champaign.

Tüchler, M.: 2002, Design of serially concatenated systems depending on the block length. Submitted to IEEE Trans. Communications.

Tüchler, M. and Hagenauer, J.: 2001, Linear time and frequency domain turbo equalization, *Proc. 53rd IEEE VTS Vehicular Tech. Conf., VTC 2001 Spring*, Vol. 2, IEEE, Rhodes, Greece, pp. 1449–1453.

Tüchler, M. and Hagenauer, J.: 2002, EXIT charts of irregular codes, *Proc. Conf. Information Sciences and Systems, CISS 2002*, Princeton University, NJ, USA, pp. 748–753.

Tüchler, M., Koetter, R. and Singer, A. C.: 2002, Turbo equalization: Principles and new results, *IEEE Trans. Communications* **50**(5), 754–767.

Tüchler, M., Otnes, R. and Schmidbauer, A.: 2002, Performance of soft iterative channel estimation in turbo equalization, *Proc. IEEE Int. Conf. on Communications, ICC 2002*, Vol. 3, IEEE, New York, NY, USA, pp. 1858–1862.

Tüchler, M., Singer, A. C. and Koetter, R.: 2002, Minimum mean squared error equalization using *a priori* information, *IEEE Trans. Signal Processing* **50**(3), 673–683.

Tungsrisaguan, E. and Rajatheva, R. M. A. P.: 2002, Turbo equalization with sequential sequence estimation over multipath fading channels, *IEEE Communications Letters* **6**(3), 93–95.

Vishwanath, S., Mansour, M. and Bahai, A.: 2002, Complexity based design for iterative joint equalization and decoding, *Proc. 55th IEEE VTS Vehicular Tech. Conf., VTC 2002 Spring*, Vol. 4, IEEE, Birmingham, AL, USA, pp. 1699–1704.

Visoz, R., Berthet, A. O., Saadani, A. and Penther, B.: 2001, Turbo equalization and incremental redundancy for advanced TDMA systems, *Proc. 53rd IEEE VTS Vehicular Tech. Conf., VTC 2001 Spring*, Vol. 3, IEEE, Rhodes, Greece, pp. 1629–1633.

Vlahoyiannatos, S. and Hanzo, L.: 2001, Blind PSP-based turbo equalization, *Proc. 53rd IEEE VTS Vehicular Tech. Conf., VTC 2001 Spring*, Vol. 3, IEEE, Rhodes, Greece, pp. 1858–1862.

Vucetic, B. and Yuan, J.: 2000, *Turbo Codes: Principles and Applications*, Kluwer Academic Publishers, Boston, USA.

Wang, X. and Chen, R.: 2001, Blind turbo equalization in Gaussian and impulsive noise, *IEEE Trans. Vehicular Technology* **50**(4), 1092–1105.

Wang, X. and Poor, H. V.: 1999, Iterative (turbo) soft interference cancellation and decoding for coded CDMA, *IEEE Trans. Communications* **47**(7), 1046–1061.

Warrington, E. M., Jackson, C. A., Stocker, A. J., Jones, T. B. and Lundborg, B.: 2000, Observations of the directional characteristics of obliquely propagating HF radio signals and simultaneous HF radar measurements, *IEE Conf. Publ. No. 474: 8th Int. Conf. on HF Radio Systems and Techniques*, IEE, Guildford, UK, pp. 243–247.

Watterson, C. C., Juroshek, J. R. and Bensema, W. D.: 1970, Experimental confirmation of an HF channel model, *IEEE Trans. Communication Technology* **COM-18**(6), 792–803.

Weiss, C., Bettstetter, C. and Riedel, S.: 2001, Code construction and decoding of parallel concatenated tail-biting codes, *IEEE Trans. Information Theory* **47**(1), 366–386.

Widrow, B., McCool, J. M., Larimore, M. G. and Johnson, Jr., C. R.: 1976, Stationary and nonstationary learning characteristics of the LMS adaptive filter, *Proc. of the IEEE* **64**(8), 1151–1162.

Willink, T. J., Davies, N. C., Angling, M. J., Jodalen, V. and Lundborg, B.: 1999, Robust HF data communications at high latitudes, *IEE Proc. Microwaves Antennas and Propagation* **146**(4), 263–269.

Willink, T. J., Davies, N. C., Clarke, J. and Jorgenson, M. B.: 1996, Validation of HF channel simulators, *IEE Seminar Digest 024: Frequency Selection and Management Techniques for HF Communications*, IEE, London, UK, pp. 13.1–13.6.

Wong, C. H., Yeap, B. L. and Hanzo, L.: 2000, Wideband burst-by-burst adaptive modulation with turbo equalization and iterative channel estimation, *Proc. 51st IEEE VTS Vehicular Tech. Conf., VTC 2000 Spring*, Vol. 3, IEEE, Tokyo, Japan, pp. 2044–2048.

Yeap, B. L., Liew, T. H., Hámorský, J. and Hanzo, L.: 2002, Comparative study of turbo equalization schemes using convolutional, convolutional turbo, and block-turbo codes, *IEEE Trans. Wireless Communications* **1**(2), 266–273.

Yeap, B. L., Wong, C. H. and Hanzo, L.: 2001, Reduced complexity in-phase/quadrature-phase turbo equalization using iterative channel estimation, *Proc. IEEE Int. Conf. On Communications, ICC 2001*, Vol. 2, IEEE, Helsinki, Finland, pp. 393–397.

Yee, M. S., Ng, S. X. and Hanzo, L.: 2002, Iterative radial basis function assisted turbo equalization of various coded modulation schemes, *Proc. 55th IEEE VTS Vehicular Tech. Conf., VTC 2002 Spring*, Vol. 4, IEEE, Birmingham, AL, USA, pp. 1705–1709.

Yee, M. S., Yeap, B. L. and Hanzo, L.: 2001, Turbo equalization of convolutional coded and concatenated space time trellis coded systems using radial basis function aided equalizers, *Proc. 54th IEEE VTS Vehicular Tech. Conf., VTC 2001 Fall*, Vol. 2, IEEE, Atlantic City, NJ, USA, pp. 882–886.

Yuan, J., Feng, W. and Vucetic, B.: 2002, Performance of parallel and serial concatenated codes on fading channels, *IEEE Trans. Communications* **50**(10), 1600–1608.