UNIVERSITÀ DEGLI STUDI DI ROMA

TOR VERGATA

Facoltà di Ingegneria

Dottorato di Ricerca in Sistemi e Tecnologie per lo Spazio (XXI ciclo)

Tesi di Dottorato

IMPLEMENTATION OF ITERATIVE MULTIUSER JOINT DETECTION TECHNIQUES IN A DVB-RCS SATELLITE SCENARIO

Coordinatore/Tutor

Prof. Giancarlo Cardarilli

Dottorando

Bernardo Tacca

Anno Accademico 2008/2009

CONTENTS

ACKNOWLEDGEMENTS

Ringrazio tutti gli ex colleghi della Space Engineering S.p.A. di Roma che mi hanno affiancato ed aiutato nello svolgimento di questa attività di ricerca, in particolare Gennaro Gallinaro e Filippo Di Cecca. Doverosa riconoscenza anche a Pierpaolo Francia per la revisione della forma inglese ed al prof. Giancarlo Cardarilli, che mi ha sempre consigliato saggiamente.

LIST OF ACRONYMS

ACM	Adaptive Code and Modulation
AMPIST	Advanced Modem Prototype for Interactive Satellite Terminals
APP	A Posteriori Probability
APSK	Amplitude Phase Shifting Keying
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
CCI	Co-Channel Interference
CDMA	Code Division Multiple Access
CMMSE-IC	Conditional Minimum Mean Squared Error-Interference Cancellation
COTS	Commercial Off The Shelf
CRDSA	Contention Resolution Diversity Slotted Aloha
C/I	Carrier to Interference Ratio
DFT	Discrete Fourier Transform
DRA	Dynamic Rate Adaptation
DVB-RCS	Digital Video Broadcasting-Return Channel Satellite
DVB-S	Digital Video Broadcasting-Satellite
e.g.	Exempli Gratia
ESA	European Space Agency
EXT	Extrinsic information
FEC	Forward Error Correction
FER	Frame Error Rate
FIFO	First In First Out
FL	Forward Link
FTP	File Transfer Protocol

GSE	Generic Stream Encapsulation
IC-MUD	Interference Cancellation-Multiuser Detection
i.e.	Id Est
IF	Intermediate Frequency
ISDN	Integrated Services Digital Network
LLR	Log Likelihood Ratio
MAC	Media Access Control
MAP	Maximum a Posteriori
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MMSE-IC	Minimum Mean Squared Error-Interference Cancellation
MMSE-SIC	Minimum Mean Squared Error-Serial Interference Cancellation
MPEG	Moving Picture Experts Group
MUD	Multiuser Joint Detection
M&M	Mengali & Morelli
n/a	Not Applicable
NCC	Network Control Centre
O&M	Oerder & Myer
pdf	Probability Density Function
PhD	Doctor of Philosophy
PHY	Physical layer
PIC	Parallel Interference Cancellation
pmf	Probability Mass Function
PSK	Phase Shifting Keying
PSTN	Public Switched Telephone Network
QPSK	Quadrature Phase Shift Keying
RCST	Return Channel Satellite Terminal
RF	Radio Frequency
RL	Reverse Link
SIC	Serial Interference Cancellation
SISO	Soft-Input Soft-Output
SNIR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio

SOVA	Soft-Output Viterbi Algorithm
SUMF	Single User Matched Filter
SUMF-IC	Single User Matched Filter-Interference Cancellation
TDMA	Time Division Multiple Access
UMMSE-IC	Unconditional Minimum Mean Squared Error-Interference Cancellation
UW	Unique Word
VoIP	Voice over IP
VSAT	Very Small Aperture Terminal
WWW	World Wide Web

1. INTRODUCTION

With the explosive growth of telecommunications, the efficient use of available spectrum is becoming increasingly important: tightened frequency reuse planning is a pressing need in satellite networks, as the frequency spectrum has become an extremely precious resource. The main purpose of this work is to develop a novel methodology for improving the bandwidth utilization of a multibeam satellite, depicted in Figure 1.1, i.e. a satellite that generates a network of spot beam coverage areas on the earth instead of the usual single beam. The advantages of a multibeam satellite over a single beam include fuller utilization of the Radio Frequency (RF) spectrum, expanded satellite communication capacity combined with smaller or even mobile ground terminals, improved beam-pointing accuracy, and very narrow beams carrying large blocks of data delivered to specific points on the earth. The drawback, obviously, is the increase of complexity and cost for the satellite system.

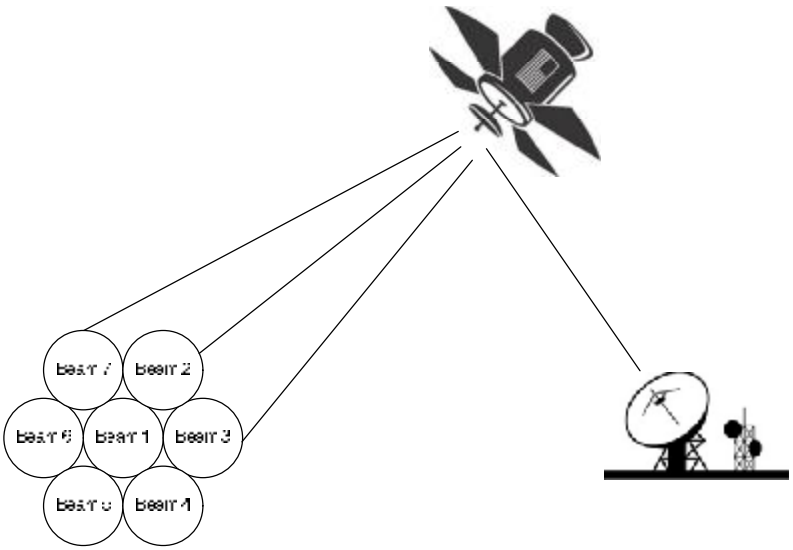


Figure 1.1: Multibeam satellite communication.

One method of achieving the efficient utilization of the available bandwidth is by permitting the frequency reuse for adjacent beams (usually adjacent beams employ different

modulation bands so as not to interfere at the satellite antennas). The resulting *Co-Channel Interference* (CCI) can lead the interference to a Carrier to Interference ratio (C/I) as low as 0 dB if both the user and the interference are at the edge of the user beam. As a consequence, severe degradation in performance is experienced unless efficient interference cancellation schemes are implemented. The usual approach to consider single user demodulation can no longer be used because of the unacceptable performance degradations and *Multiuser Joint Detection* (MUD) algorithms must be considered, where all the channels are demodulated together in order to suppress the CCI interference.

1.1. MULTIUSER JOINT DETECTION OVERVIEW

In the last decade an impressive amount of theoretical investigations have been developed in the field of MUD algorithms; in particular efforts have been focused on Code Division Multiple Access (CDMA) systems for mobile applications, while our case of spatial diversity systems for fixed applications has been considered to a lesser extent. The first studies on MUD techniques were made to maximize the efficiency of CDMA based networks, that have enjoyed success as a strategy for terrestrial cellular systems, partly due to its potential to reuse the same spectrum in each cell. However, these systems are still far from Shannon's theoretical capacity of multiuser systems and to further increase the CDMA capacity there has been much research into multiuser decoding. It is demonstrated in [1] that the optimum multiuser decoder for an asynchronous CDMA system employing Forward Error Correction (FEC) coding combines the trellises of both the asynchronous detector and the FEC code. This decoder is a variation of the Viterbi detector and has a complexity of approximately $O(2^{kK})$ operations per bit where k is the code constraint length and K is the number of users. This exponential complexity makes the system impractical for most CDMA systems. Consequently, there has been considerable research into simpler, suboptimum decoding strategies.

The pioneering work on multiuser detection in [2] was focused on simple uncoded CDMA systems, because they are the "natural scenarios" for the CCI interference, but most of the successive work in this area has addressed multiuser detection for the more realistic coded CDMA systems. Among the several multiuser detection schemes proposed, analyzed in the following Chapters, the so called Serial and Parallel Interference Cancellation MUD schemes (SIC and PIC) are particularly attractive because they process directly the output of a bank of

Single User Matched Filters (SUMF). The receiver front-end is identical to that of a typical detector, therefore these methods can be seen as an “add-on” post-processing to enhance the performance of a conventional receiver when particularly high channel load is needed. It will be shown how the main performance limitation of SIC/PIC schemes is the error propagation caused by feeding back erroneous symbol decisions and the imperfect interference cancellation due to a non ideal estimation of the channel parameters. SIC is both simpler and more robust than PIC with respect to the errors propagation, since users can be ranked according to their Signal to Interference plus Noise Ratio (SNIR) and decoded in sequence [3]-[6]. In early works [3] and [4], SIC is applied to uncoded transmission and hard decisions are used at each stage to remove the already detected users from the received signals. In order to prevent errors propagation, the use of soft (or partial) interference cancellation and iterative SIC schemes has been proposed in different forms and by different authors [6]-[8]. Iterative SIC/PIC schemes can be naturally coupled with iterative parameter estimation in order to improve the estimates with the iterations, as long as the signal is “cleaned-up” from interference.

1.2. PHD RESEARCH ACTIVITY

The PhD activity is part of a larger research program, promoted by the European Space Agency (ESA), named "Advanced Modem Prototype for Interactive Satellite Terminals (AMPIST)", that consists of developing a complete system laboratory hardware prototype with some innovative techniques suitable to enhance the performance of broadband interactive satellite terminals implementing the Digital Video Broadcasting-Return Channel Satellite (DVB-RCS) standard.

The growing interest for multimedia applications is encouraging the deployment of fixed telecommunications satellite systems capable of offering high-speed point-to-point links at competitive service fees. Therefore the next generation of broadband satellite systems must be designed to offer higher throughput than the one provided by current systems. That will be possible by exploiting the higher frequency bands allocated to fixed satellite systems (e.g. the Ka-band), generating a great number of narrow aperture high-gain beams, and reusing the frequency bands as much as possible. But in this way, system throughput will become more and more affected by the CCI interference occurring among signals that share common band portion. A great aid is given by the use of highly efficient coding schemes, such as those

utilized in the Reverse Link (RL) of current DVB-RCS satellite systems and those standardized for the Forward Link (FL) by the Digital Video Broadcasting-Satellite (DVB-S2) working group. These countermeasures can be accompanied by interference and fading mitigation techniques like adaptive coding and modulation, which are being studied and optimized with the aim of providing a higher flexibility and improving the overall system efficiency. It should however be noted that the introduction of those techniques becomes more challenging in the presence of very efficient coded modulations, that is low operating Signal to Noise Ratios (SNR). In such a hostile scenario, new interference mitigation techniques, requiring the receiver to process simultaneously more channels and thus generally identified as MUD techniques, have been proposed as a promising solution to further increase system capacity in an interference-limited system operating under heavy traffic load.

The AMPIST reference scenario will be a satellite system where the gateway station supports a two-way communication with several interactive user terminals equipped with a relatively small aperture antenna and transmitting low RF power. The RL access scheme is assumed to be low-rate TDMA, with data rates from around one hundred kbit/s to few Mbit/s. With this choice, the proposed MUD techniques become applicable to the RL of most current Very Small Aperture Terminals (VSAT), with usual aperture less than 1.5 m, based on the DVB-RCS access scheme as well as on similar proprietary schemes. In particular, iterative MUD algorithms will be evaluated for mitigating the CCI, and this approach will be combined with a spatial Minimum Mean Squared Error (MMSE) linear processing taking place before the MUD receiver, for improved performance.

The MUD concepts have often been analyzed in literature for hardly realistic scenario, proving that only limited effort was devoted to further developing the theoretical background to make it readily applicable to practical systems. Instead, in this work, the algorithms for implementing the selected interference mitigation scheme are evaluated and their performance detailed through physical layer simulations in reference conditions representative of real applications.

The thesis is organized as follows. In Chapter 2 an overview of the DVB-RCS is presented. Chapter 3 outlines the whole AMPIST project, defining the sub activities developed in the PhD, that can be basically divided in two parts: single user scenario analysis and simulations (for a single enhanced DVB-RCS channel) and multiuser scenario analysis and simulations (where different enhanced DVB-RCS channels interfere with each other and

MUD interference mitigation is adopted). Chapter 4 deals with the single DVB-RCS channel decoding. Chapter 5 is an introduction to the turbo decoding, that is necessary for understanding the iterative MUD algorithms presented in the successive Chapter 6. Chapter 7 reports the results obtained for the MUD techniques and finally the conclusions are discussed in Chapter 8.

2. DVB-RCS

2.1. INTRODUCTION

With their tremendous success in broadcasting entertainment services, interactive broadband satellite systems are now also being viewed as viable service delivery vehicles. Many network operators seriously consider satellite-based networks to offer broadband and multimedia services to supplement and enhance their existing terrestrial networks. High speed user terminal interactivity via satellite requires the presence of a satellite return link with broadband capability. Such connectivity is typically offered nowadays via different solutions, some of which proprietary. DVB-RCS [9] is a standard concerning the air interface of the return link of interactive satellite terminals and it defines the end-to-end connectivity link between the satellite operator hub and the user terminal. For this purpose, it embraces the DVB-S and the newly defined DVB-S2 [10] standards as the forward link specifications, while it describes in more details the Physical (PHY) and Media Access Control (MAC) layer aspects of the return channel.

Although there is certainly a market for two-way satellite-based interactive systems, a number of issues will influence the rate of uptake, financing of systems, and ultimately, the number of subscribers. Among these, the cost of the terminal and service (cost per bit delivered to the end user) has to be effective in order to support the financial case for the satellite and ground segments. Therefore the system spectral and energy efficiency has to be improved by using state-of-the-art techniques and solutions. The DVB-S2 standard implements a number of innovative techniques to dramatically improve the forward link system throughput for interactive systems. These techniques comprise an enhanced FEC code, the use of satellite-optimized high order modulations with Adaptive Coding and Modulation (ACM), an efficient framing scheme, transmit signal pre-distortion techniques and direct mapping of variable length packets into the DVB-S2 PHY.

2.2. SYSTEM MODEL

Figure 2.1 shows the system model which is to be used within DVB for interactive services. In the system model, two channels are established between the service provider and the user:

- The Broadcast Channel: an unidirectional broadband broadcast channel including video, audio and data is established from the service provider to the users.
- The Interaction Channel: a bi-directional interaction channel is established between the service provider and the user for interaction purposes. It is formed by:
  - *Return Interaction Path*: from the user to the service provider. It is used to make requests to the service provider, to answer questions or to transfer data.
  - *Forward Interaction Path*: from the service provider to the user. It is used to provide information from the service provider to the user(s) and any other required communication for the interactive service provision. It may be embedded into the Broadcast Channel. It is possible that this channel is not required in some simple implementations which make use of the Broadcast Channel for the carriage of data to the user.

The Return Channel Satellite Terminal (RCST) is formed by the Network Interface Unit (consisting of the Broadcast Interface Module and the Interactive Interface Module) and the Set Top Unit. The RCST provides interface for both Broadcast and Interaction Channels. The interface between the RCST and the interaction network is via the Interactive Interface Module.

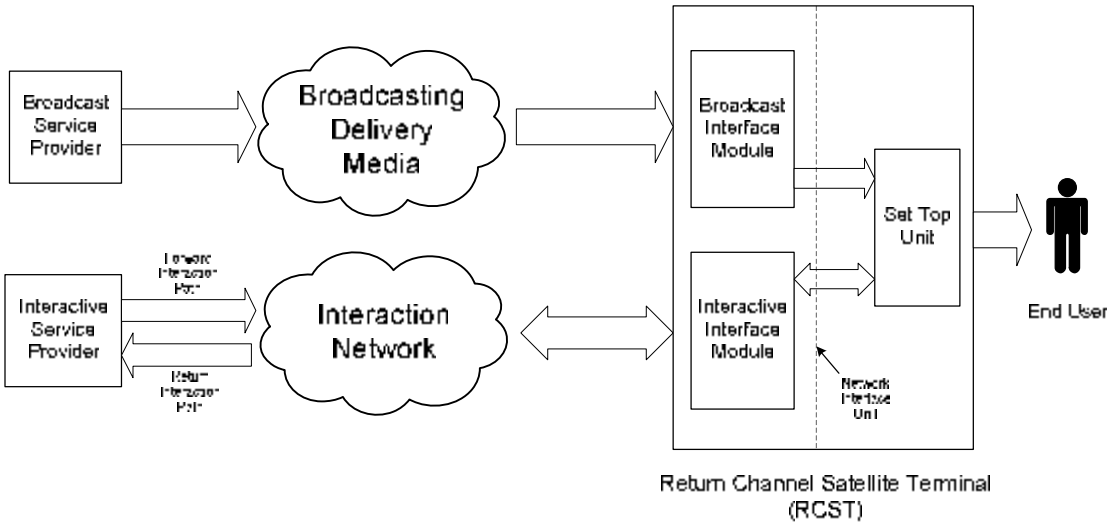


Figure 2.1: Generic system model for interactive systems.

An overall Satellite Interactive Network, within which a large number of RCSTs will operate, comprises the following functional blocks, as shown in Figure 2.2:

- Network Control Centre (NCC): a Network Control Centre provides control and monitoring functions. It generates control and timing signals for the operation of the Satellite Interactive Network to be transmitted by one or several Feeder Stations.
- Gateway Station: a Gateway Station receives the RCST return signals, provides accounting functions, interactive services and/or connections to external public, proprietary and private service providers (data bases, pay-per-view TV or video sources, software download, tele-shopping, tele-banking, financial services, stock market access, interactive games, etc.) and networks (Internet, ISDN, PSTN, etc.).
- Feeder Station: a Feeder Station transmits the forward link signal, which is a standard satellite DVB-S uplink, onto which are multiplexed the user data and/or the control and timing signals needed for the operation of the Satellite Interactive Network.

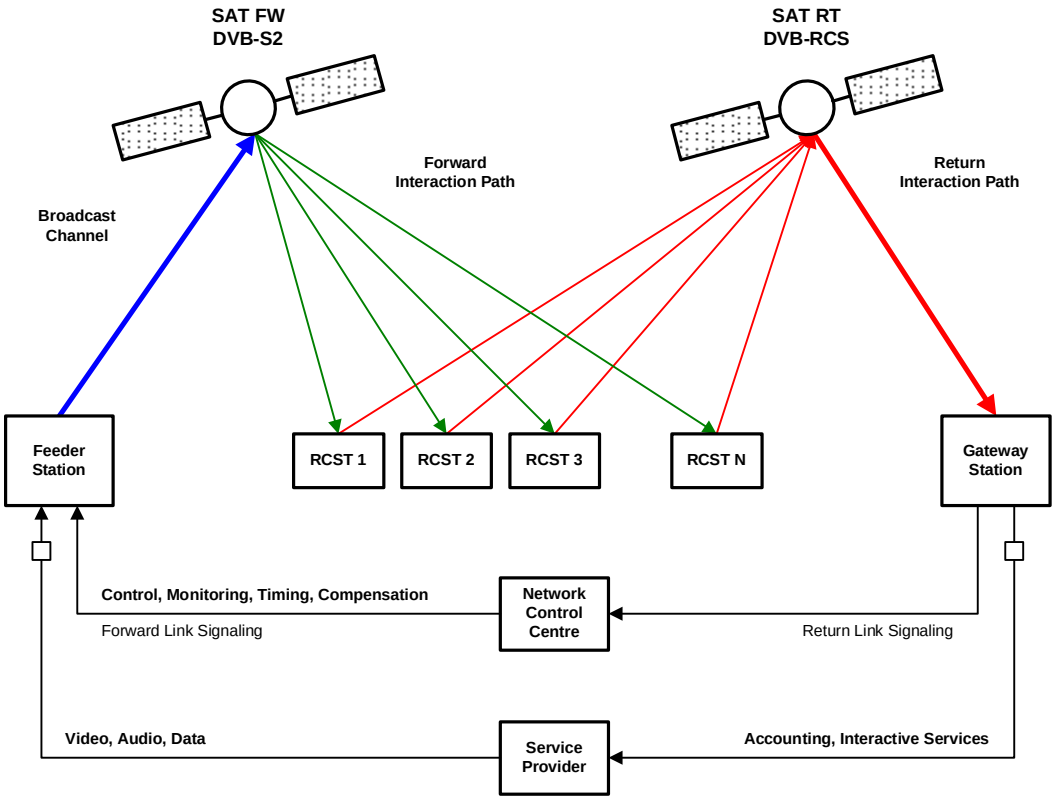


Figure 2.2: Reference model for the Satellite Interactive Network.

The forward link carries user traffic and signalling from the NCC to RCSTs, that is necessary to operate the return link system. Both the user traffic and forward link signalling can be carried over different forward link signals. Several RCST configurations are possible; for this PhD activity the RCSTs operate in a multibeam scenario: TDMA mode in the same beam and different frequencies, in principle, for different beams. In order to simplify the gateway resource scheduling, the additional constraint of constant TDMA slot duration is imposed for the AMPIST project.

3. THE AMPIST PROJECT

3.1. SPECIFICATIONS

The AMPIST hardware simulator will be designed so as to allow full validation of the physical layer and upper layers in a controlled laboratory environment for an interactive satellite scenario that implements modified DVB-RCS PHY and MAC layers on the return link together with a DVB-S2 based forward link (see Figure 2.2). The developed real-time modem demonstrator will be composed of the following parts for the return link:

- Traffic simulator (useful link and other users) including related channel quality reports.
- Reconfigurable advanced coded digital modulator (including framing, coding and channel modulator).
- Reconfigurable advanced digital demodulator and decoder, with Intermediate Frequency (IF) input, including channel estimation.
- Physical layer algorithms.
- Scheduler and MAC layer.
- Ancillary monitoring and control device (system configuration monitoring and control, traffic statistics, modem performance).

In the forward link the advanced modem demonstrator will include:

- User and background traffic simulator including related channel quality reports.
- DVB-S2 Commercial Off the Shelf (COTS) encoder and modulator.
- Satellite channel simulator.
- DVB-S2 COTS decoder and demodulator.
- Ancillary monitoring and control device.

The satellite channel simulator, including as a minimum high power amplifier nonlinearity, local oscillator phase noise, Additive White Gaussian Noise (AWGN), propagation fading and delay, will be provided by ESA. Figure 3.1 clarifies the functions supported by the demonstrator:

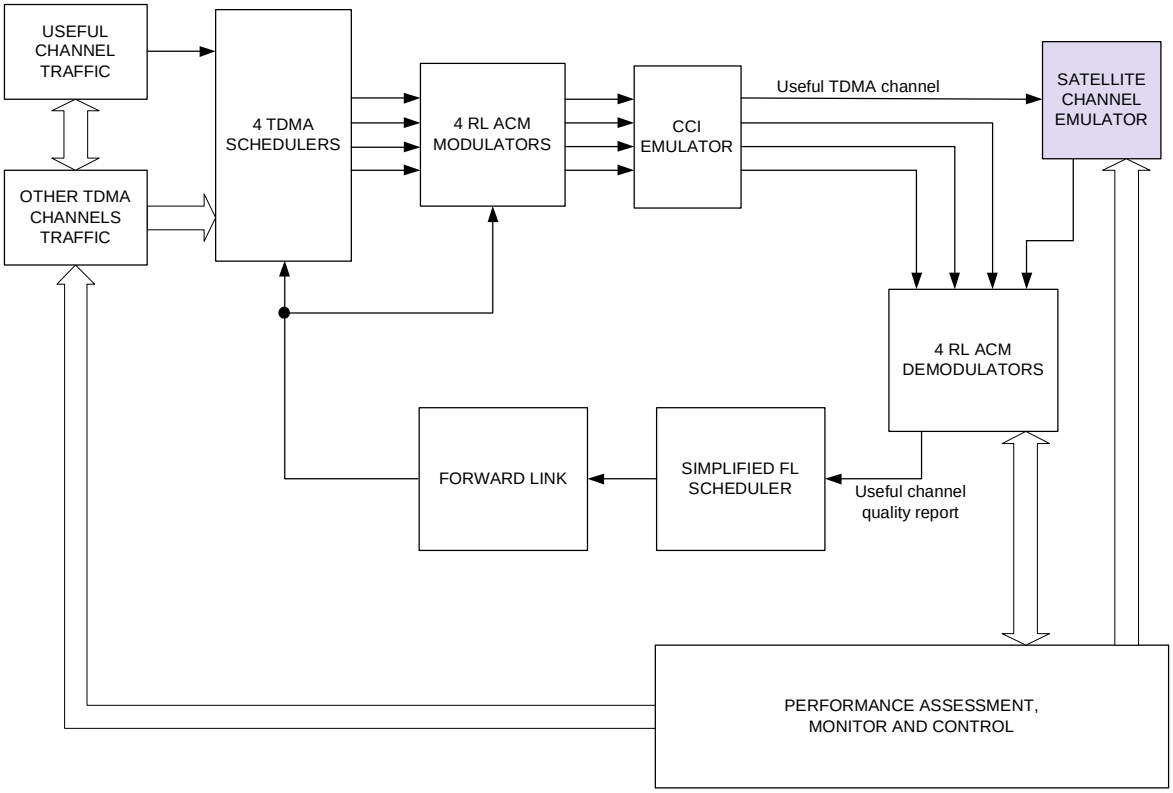


Figure 3.1: AMPIST functional block diagram.

The AMPIST activity has the purpose to implement some innovative PHY and MAC layer algorithms/techniques suitable to enhance the performance of broadband interactive satellite terminals implementing the DVB-RCS standard with the objective to reduce the overall terminal-plus-service cost. In particular, the following algorithms will be studied:

- An enhanced FEC scheme with improved performance with respect to the current DVB-RCS turbo code.
- A novel framing and layer 2 encapsulation scheme optimized for adaptive PHY layer and matching traffic characteristics.
- High order modulation schemes (i.e. 8PSK) with ACM.
- Optimized contention-based access, with a solution based on the Contention Resolution Diversity Slotted Aloha (CRDSA) access scheme.
- Interference cancellation techniques to suppress the CCI interference.

3.1.1. Adaptive Coding and Modulation

The ACM is a sophisticated form of data rate control, where the information bit-rate is adapted by changing modulation and coding rate while maintaining a constant symbol rate over the channel. Preference to QPSK or BPSK has been always given in the system for the higher power efficiency of such modulation formats, however this typically implies a waste of throughput since for most of the time propagation conditions are good and a better usage of the transmitted power is possible. In particular, more bandwidth efficient modulation and coding modes can be used without practical Bit Error Rate (BER) impairment: instead of changing the transmitted power, the gateway control feedback signal (SNIR based) adapts the transmitted modulation and coding to the channel conditions.

The ACM technique strategy naturally tends to optimize the overall system throughput. Some rate adaptation may be performed by changing the code rate only while keeping the same modulation format (e.g. QPSK). However, the total SNIR range that this technique can cover is limited to about 5.5÷6 dB within the current DVB-RCS standard, corresponding to the difference between the decoder thresholds of code rates 1/3 and 6/7. Such a limited range would force one to jointly use Dynamic Rate Adaptation (DRA, i.e. adaptation of the carrier symbol rate) in order to cover the higher SNIR ranges imposed by the typical Ka-band fading. Higher order modulations are thus required, in particular 8PSK and 16APSK constellations. The adopted solution supports high order modulations with a total SNIR range of about 16 dB (11 dB with just QPSK and 8PSK). The ACM functionalities support the following features:

- QPSK, 8PSK, 16APSK and a variety of coding rates with approximately 1 dB step in the code thresholds.
- On-the-fly reconfigurable (from burst to burst) coding scheme (coder and decoder).
- Framing structure which allows to select among different burst types from frame to frame.

The ACM control loop strategy is studied in detail in [11] and [12] where techniques to minimize the residual link margin are investigated and simulation results are shown to prove their efficiency. The support of ACM with high order modulations may complicate the DVB-RCS framing strategy (see [12] for example) but leads to a typical throughput increase between 50% and 100%, depending on system parameters.

3.1.2. Frame Structure

The current DVB-RCS supports two possible packets profiles: ATM or MPEG. In both cases the TDMA burst length has to cope with few possible block lengths as they have to be a multiple of the elementary packet (i.e., for traffic burst, 53 bytes for ATM cells or 188 bytes for MPEG packets plus the FEC code redundancy). Unfortunately this approach makes difficult any extension of the DVB-RCS aimed at the exploitation of ACM, because changing the ACM operating mode would also require to change the terminal burst time duration in order to fulfil the need of having burst sizes (expressed in bits) of constant length (or multiple of the elementary info packet size). For constant baud rate operation, in fact, a constant length (expressed in bits) burst size implies that the burst time duration has to change according to the selected ACM mode. This is quite impractical as the whole carrier time plan should change as a consequence of one terminal changing its ACM mode. Given the impracticality of changing the burst duration, the DRA is typically proposed to achieve some of the ACM advantages while maintaining (almost) backward compatibility with current DVB-RCS specifications.

The DRA requires a static (or slowly varying) partition of the satellite resources in different classes. Partitioning of the resources may be done, for simplicity reason, at the carrier level. Hence, carrier belonging to different classes will be characterized by different code rates and/or different carrier baud rates. A terminal is then dynamically assigned to a carrier class according to its current link quality. This partitioning may greatly reduce the effectiveness of DRA as compared to full ACM (or even adaptive coding only) given the required resource partitioning. To cope with this problem a new DVB-RCS profile, different from the ATM or the MPEG one, is required. This new profile can be based on the Generic Stream Encapsulation (GSE) already being standardized for the DVB-S2 based forward link, but in order to exploit the GSE flexibility, a FEC code allowing much more granularity in the selection of the information bits in each codeword is required. An enhanced FEC with respect to the currently specified DVB-RCS turbo code is a novel turbo code scheme further described in the next paragraph 3.1.3, which shows, in addition to higher flexibility in the selection of the block length, also greater power efficiency as well as suitability for high order modulations.

3.1.3. The Turbo-Φ Code

DVB-RCS FEC specifications currently use either a concatenated code (i.e. a Reed Solomon code cascaded with a convolutional one) or a duobinary, 8-state turbo code. As a matter of fact, the turbo code option is now the one mostly used given its higher power efficiency. Seven possible code rates were defined (1/3, 2/5, 1/2, 2/3, 3/4, 4/5, 6/7) for this turbo code.

A problem with the currently specified turbo code is that it is only compatible with few block lengths comprised between a minimum length of 12 bytes and a maximum one of 216 bytes. This was justified, at the time specifications were drafted, by the fact that the system was conceived to operate with fixed size elementary packets (the 53 bytes ATM cells or 188 bytes MPEG packets). The use of such fixed length elementary packets requires that the turbo encoder must be fed with a reduced set of possible information block lengths and, as already discussed in previous paragraphs, this is a significant limitation when exploitation of ACM is wished. Therefore a new turbo code, nicknamed “Turbo-Φ”, was specifically designed for an ACM enhanced DVB-RCS in [13]. This new code presents both a higher power efficiency (obtained thanks to the doubling of number of states from 8 of the original DVB-RCS code to 16 and to further optimization of the interleaver) and a higher flexibility in the allowed block lengths which fully cover the present ACM requirements (information block lengths variable from 40 to 376 bytes with one byte granularity). This code is intended to offer near-Shannon performance on Gaussian channel, in most situations of block size, coding rate up to 8/9 and associated modulation. It supports QPSK, 8PSK and 16APSK modulations. In addition, the Turbo-Φ gives the natural possibility of using a high level of parallelism in the associated decoder.

Regarding the implementation issues, minimizing the turbo decoder complexity is of paramount importance especially when iterative interference cancellation techniques are employed, since such techniques may require a higher number of iterations with respect to the conventional FEC decoding. In order to reduce the turbo decoder complexity a simplified Max Log MAP algorithm can be used instead of the Log MAP algorithm [14]. In addition to hardware simplifications, the Max Log MAP algorithm has also the advantage of being SNR independent (while the Log MAP is very sensitive to errors in the estimated SNR). The performance penalization of the Max Log MAP algorithm is usually about 0.3 dB and can be reduced as low as 0.1 dB by suitable scaling the extrinsic probabilities computed by the decoder before reusing them for another decoding iteration [15]. Optimal scaling factors are

typically computed by extensive simulations. However, for the Turbo-Φ code an indication of the optimal scaling factors is already contained in the specification document [13]. No optimization of these scaling factors is thus needed. The Max Log MAP algorithm will be further discussed in paragraph 5.2.3.

All the possible AMPIST operative modes are summarized in Table 3.1: the generated I/Q symbols will be then padded with a variable length overhead (necessary for the time, frequency and phase recovery) so as to obtain only one possible fixed length burst for the already mentioned fixed TDMA bursts duration constraint.

Mode	**Payload information bits**	**Rate**	**Coded bits**	**Modulation**	**Waterfall E_s/N_o [dB]**	**I/Q symbols**
S1	312	1/3	936	QPSK	0	468
S2	488	1/2	976	QPSK	2	488
S3	656	2/3	984	QPSK	3.1	492
S4	736	3/4	982	QPSK	4.3	491
S5	848	5/6	1018	QPSK	4.9	509
S6	880	6/7	1028	QPSK	5.9	514
S7	928	2/3	1392	8PSK	6.8	464
S8	1040	3/4	1389	8PSK	8	463
S9	1152	5/6	1386	8PSK	9.9	462
S10	1192	6/7	1392	8PSK	10.8	464

Table 3.1: AMPIST possible operative modes.

3.2. CCI INTERFERENCE

As already discussed in Chapter 1, the CCI scenario envisages a multibeam system. In this scenario it is possible to increase the return link overall bandwidth by reusing the same frequency for every beam (or for a subset of beams jointly decoded by the same gateway, as depicted in Figure 3.2, where the same number beams have the same frequency).

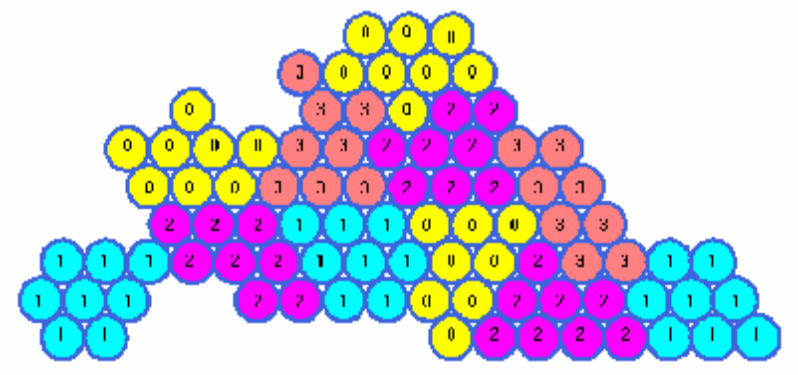


Figure 3.2: Frequency reuse example for the CCI scenario.

In the AMPIST implementation, it will be supposed a cluster of four beams with the same frequency for the CCI case, as shown in Figure 3.3, where all the users U_i from different beams interfere with each other at the transponder antennas:

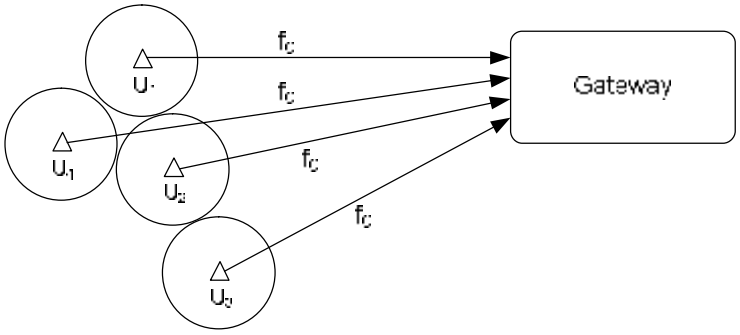


Figure 3.3: AMPIST CCI interference scenario.

3.3. HARDWARE IMPLEMENTATION

The AMPIST modulator hardware implementation, including the CCI generation, is shown in Figure 3.4. The demodulator block diagram will be introduced in Chapter 6.

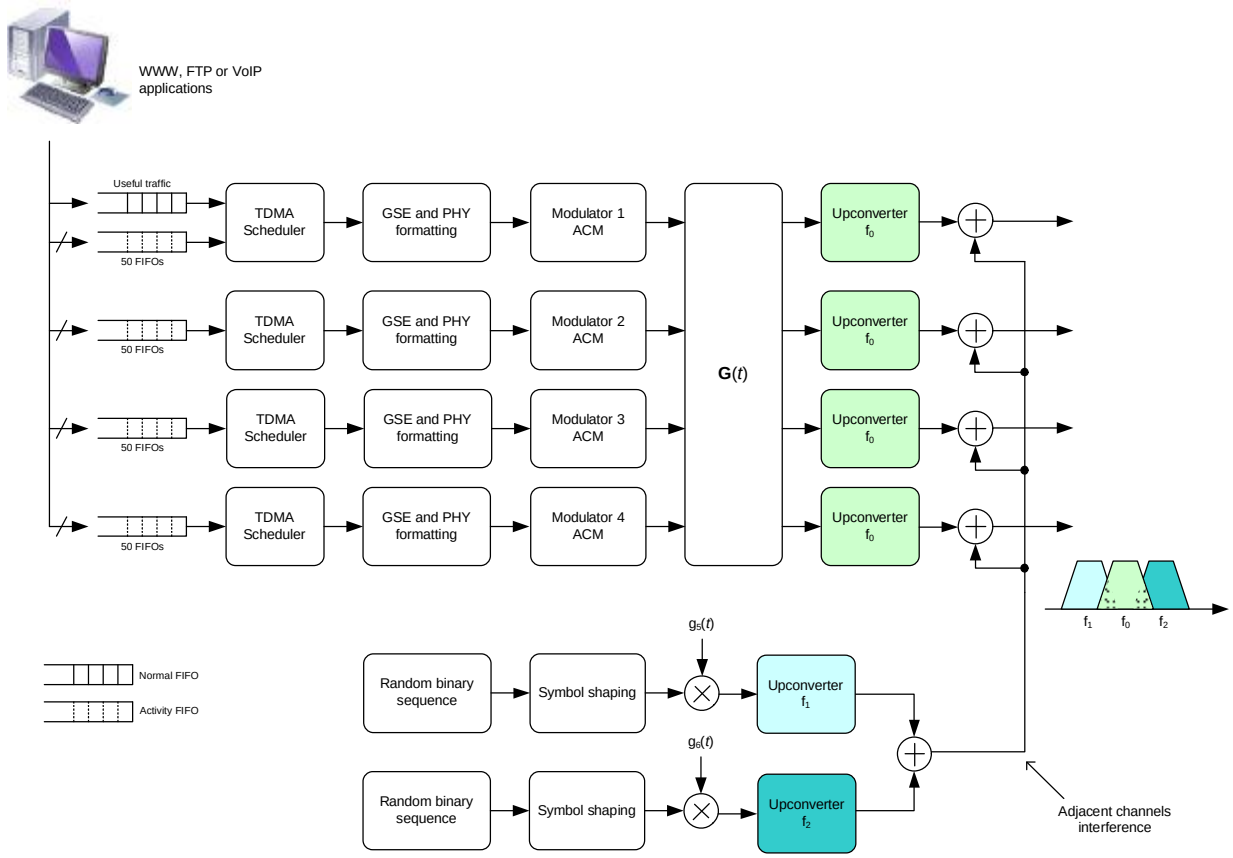


Figure 3.4: AMPIST modulator block diagram.

The useful user and a maximum of 200 interference users are simulated: the number of interference users can be set as well as their characteristic (WWW, FTP or VoIP). These 201 data streams are generated by means of an application running on a PC that fills 201 different First In First Out memories (FIFO). Only the useful traffic FIFO memorizes the data packets, the other FIFOs (called “activity” FIFOs in the figure) just count the packet to be served for the interference users and these packets are then generated as random binary sequences. The TDMA scheduler is in charge of serving the appropriate FIFO with the information from the

forward link. Afterwards the four modulated signals are combined with the $\mathbf{G}(t)$ matrix in order to simulate the CCI interference and the four obtained signals are upconverted on different outputs. The $\mathbf{G}(t)$ matrix also takes into account four independent fading events for the users. The modulators are asynchronous at the symbol level, with different phases and only the first modulator (the one with the useful traffic) is dynamically ACM modulated, while the other three modulators can be varied manually. Two adjacent interference channels are also added, with independent fading. These channels do not need the complete ACM modulation, but only a band shaping at the symbol rate currently adopted. The AMPIST modulator main features are summarized in Table 3.2:

PARAMETER	IMPLEMENTATION	COMMENT
Signal roll-off	0.2 and 0.35	
Supported baud rates	128 kbaud, 512 kbaud and 2 Mbaud	For all the channels
Modulation formats and code rates	Modulations: QPSK, 8PSK Code rates: 1/2, 2/3, 3/4, 6/7	Only the first channel is ACM modulated, for the other three the modulation can be manually changed
Turbo code information block length	Variable from 40 to 376 bytes	
PHY Layer Access Technique	TDMA	
Layer 2 encapsulation	GSE protocol	
Traffic simulator characteristics	Bit stream for the applications: WWW, FTP, VoIP	Maximum 50 users per channel
Co-channel interference	Three carriers with independent fading events and asynchronous with respect to the useful bursts	
Adjacent channel interference	Two carriers with independent fading events	Carrier spacing = 1.3 symbol rate

Table 3.2: AMPIST modulator main features.

3.4. PHD DEVELOPED ACTIVITIES

Within the AMPIST project, the PhD developed activities can be divided into two consequent parts:

- The first one, that will be called *single user*, has the purpose to analyze by means of extensive software simulations the behaviour of the ACM methodology with some of the different operating modes reported in Table 3.1 and to propose a burst format compatible with the recovery algorithms for very low SNR signals and the CCI interference matrix estimation, necessary for the MUD algorithms.
- The second part, *multiuser*, where four single user channels are combined in order to simulate the CCI interference and a novel MUD interference cancellation scheme is adopted. Simulation results will be finally reported.

Therefore in the next Chapters, single user scenario will be presented first and then, exploiting the obtained guidelines, the most complex MUD scenario will be simulated.

4. SINGLE USER DECODING

This Chapter analyzes the ACM bursts decoding (for a single DVB-RCS user), with possible operative modes from S1 to S10 of Table 3.1. Actually, among the ten supported Turbo-Φ modes, only S2, S3, S4, S6, S7 have been implemented for the AMPIST project. This modes selection has been made in order to simplify the demodulator without losing its generality: the result is a 5 dB operative total span for E_s/N_o with steps of about 1 dB, and the 8PSK adoption for the S7 mode.

The burst demodulator is critical because it is necessary to minimize the burst overhead while allowing an acceptable performance at very low SNIR signals. SNIR as low as 2 dB are expected for operation with QPSK $1/2$ (the most hostile mode S2). Actually, operation at even lower SNIR is required when CCI interference is enabled, since SNIR below 0 dB may be expected before the interference cancellation is performed. Figure 4.1 shows the sequence of operations typically performed in a burst demodulator:

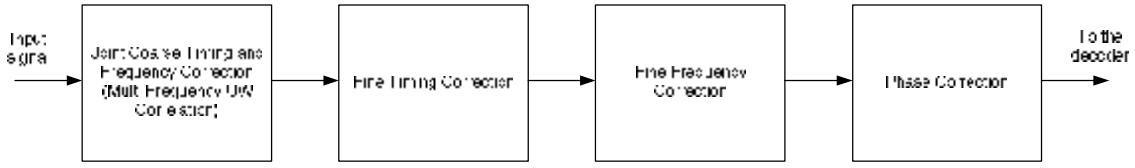


Figure 4.1: Typical burst demodulator.

The first block of joint coarse timing and frequency correction is implemented by correlation techniques exploiting a known pattern (that will be indicated as Unique Word, UW) in the burst. Correlation operation will recover the coarse burst timing with an accuracy related to the number of samples/symbol used in the analog to digital converter. All the presented simulations (both single user and multiuser) operate with 4 samples/symbol. Hence coarse timing recovery can have a maximum error of $\pm 1/8$ symbol (assuming no error in the detection process) and a fine timing correction is typically needed after the burst detection. The timing recovery will be studied in paragraph 4.1.1.

The correlation process used for burst detection can also provide some rough frequency error estimation if parallel frequency search is performed. The actual need for frequency correction will obviously depend on the expected frequency error of the input signal. Experience with DVB-RCS systems suggests that NCC based synchronization mechanisms are able to provide a good frequency reference at the ground terminals. In particular, frequency error at the gateway demodulator input is typically lower than 2 kHz. That means a normalized (to the symbol rate) maximum frequency error of about 1.6% at 128 kbaud and 0.1% at the highest AMPIST baud rate of 2048 kbaud. The analysis of frequency correction possible algorithms is presented in paragraph 4.1.2. Finally phase recovery algorithms are reported in paragraph 4.1.3. Note that it can be demonstrated the equality between the ratio E_s/N_o (where E_s is the symbols energy and N_o the noise spectrum power) and the received SNR, therefore from now on the two terms will be used indifferently.

4.1. RECOVERY ALGORITHMS ANALYSIS

4.1.1. Timing Recovery

A coarse timing recovery is implicit in the burst detection operation. Coarse timing recovery (and burst detection) can be performed using either UW coherent correlation or UW coherent correlation with non coherent post-integration. This second option is interesting only when the UW is split in a “preamble” (i.e. know symbols sequence at the burst beginning) and a “postamble” (know symbols sequence at the burst end). Non coherent post-integration can then be used to combine the coherent integration on the preamble and the postamble.

Figure 4.2 shows the recovery uncertainty curves with UW coherent integration at E_s/N_o = 0 dB for different preamble lengths, with a threshold detection. A preamble length of 40 symbols can be considered sufficient for the burst acquisition. Actually, given the small uncertainty of the start of burst (less than 16 symbols at the highest baud rate), we will not use a threshold based detection, but a more powerful Maximum Likelihood (ML) method (i.e. searching for the maximum correlation over the uncertainty region) that leads to slightly better results.

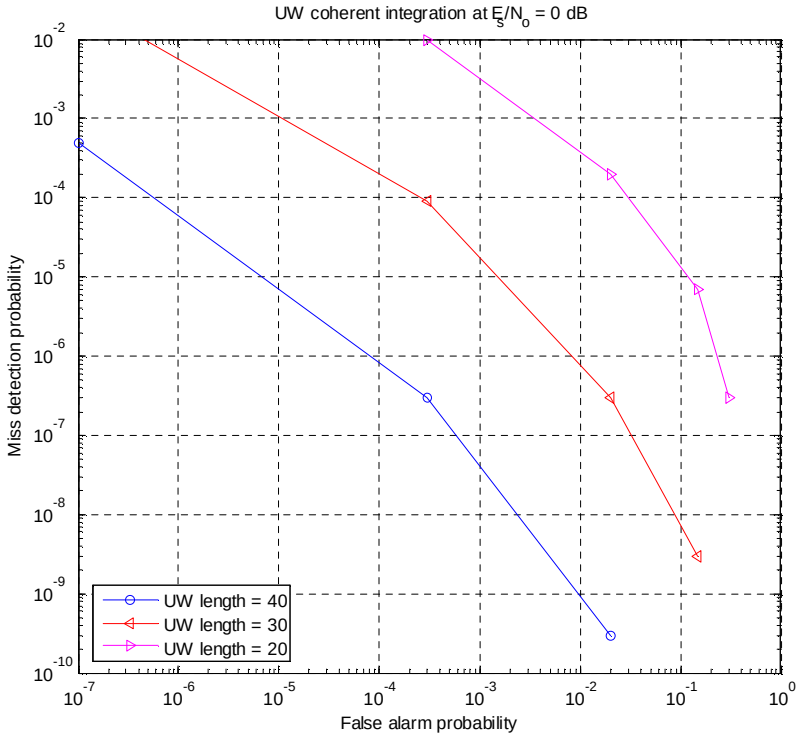


Figure 4.2: Miss detection and false alarm probability with UW coherent integration at E_s / N_o = 0 dB.

It has to be noted that with a 40 symbols coherent integration, the expected maximum frequency error equal to 1.6% of the symbol rate would be destructive and therefore parallel frequency coarse recovery must also be performed (see paragraph 4.2.3).

Regarding the fine timing error recovery, the following approaches can be considered:

- Dicotomic timing search (linear interpolation). For this algorithm, in addition to the best correlation, the second best correlation is also considered. The optimal timing should be comprised between these two time indexes. The preamble is then interpolated to recover a sequence sampled midway the two best timing options. The correlation for this new sampling is then computed and the result compared with the previous best correlation. The procedure can be iterated two or three times in order to get a better timing error estimation.

- Spline interpolation. For this option, the correlation values around the optimal timing index are interpolated via a spline interpolation and the timing corresponding to the maximum of the spline is selected.
- Oerder & Meyr (O&M) algorithm. The O&M algorithm [39] can operate on the whole data burst. This algorithm generates a sequence with a clock component by computing the square norm of the burst symbols. The phase of the clock component is then extracted using a Discrete Fourier Transform (DFT) and used to estimate the fine timing error. Errors up to half the symbol duration can be recovered.

Simulation results have shown that the O&M is the best fine timing recovery methodology when only AWGN noise is present. After the fine timing error measurement, the stored burst samples are interpolated via a Farrow interpolator in order to recover the correct timing sequence. Only one sample per symbol needs to be computed.

4.1.2. Frequency Recovery

Frequency estimation error can be lower bounded by the well known Cramer-Rao bound. The Mengali & Morelli algorithm (M&M) [40] is able to achieve the Cramer-Rao bound when operating on known data (also called “pilots”) at the SNR of interest (greater than 0 dB, for reasonable preamble lengths, see Figure 4.3). The M&M algorithm usually operates on known pilot symbols, but it can process also modulated data, although it is not effective at low SNR, as shown in Figure 4.4.

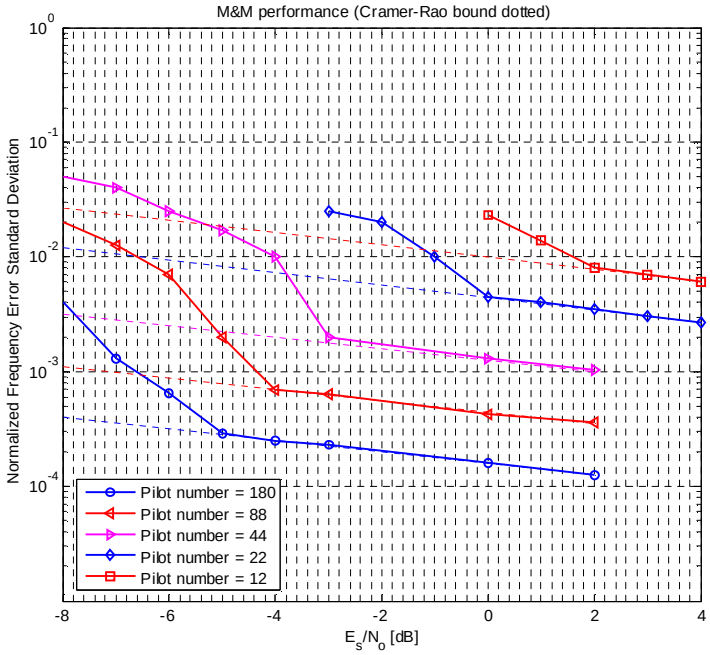


Figure 4.3: M&M algorithm performance with pilot symbols.

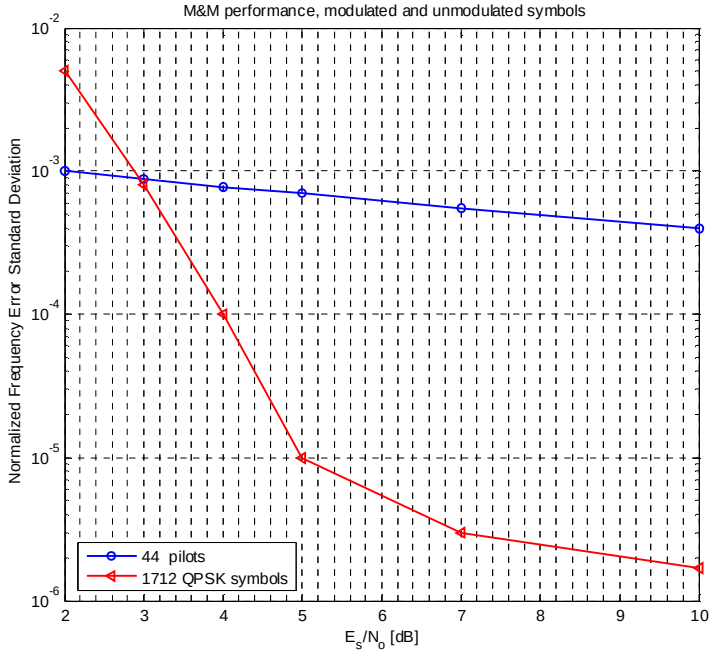


Figure 4.4: M&M algorithm performance with modulated and unmodulated symbols.

The frequency error is normalized to the pilot symbols repetition frequency, so that the absolute frequency error can be decreased if the pilot symbols are not continuous. Operating on a continuous preamble is not sufficient to achieve a good frequency accuracy (for reasonable preamble lengths), hence a distributed know pilot sequence is a requirement for our system at least for the QPSK 1/2 mode or when interference mitigation is adopted (as the initial SNIR may be very low).

Using the M&M algorithm on pilot symbols, from Figure 4.3, at least 22 pilot symbols must be used in order to reach the Cramer-Rao bound at E_s/N_o =0 dB. For the typical AMPIST 490 symbols payload, the overhead which results using a 40 symbols UW plus 20 symbols distributed pilots is about 11%. With this assumption the spacing of pilot symbols would be 20 symbols. Hence, always looking at Figure 4.3, with E_s/N_o =0 dB the resulting frequency error standard deviation (normalized to the symbol rate) would be approximately $0.005 / 20 = 2.5 10^{-4}$.

4.1.3. Phase Recovery

Two options are available for the carrier phase recovery:

- Feedback phase recovery loop (on all the modulated symbols):
  - Single forward loop. This technique employs an usual PLL to perform carrier phase synchronization.
  - Iterated forward and backward loops. This technique employs a PLL iterated forward (from the start of the burst to the end) and backward (from the end of the burst to the start). At each iteration the gain of the loop is reduced.
- Feed-forward carrier phase estimation based on the averaging of the UW and pilot symbols over a predefined length sliding window. This technique requires known pilot symbols, distributed as evenly as possible over the whole burst length.

For the feedback approach, the residual frequency error still affecting the signal after the fine frequency correction imposes a lower limit on the loop bandwidth, or, at least, on its initial bandwidth. Therefore such a large bandwidth can be inadequate for a good phase tracking: in particular the cycle slipping probability may be too high. It can be shown that for SNR = 0 dB, the loop bandwidth requested for a QPSK with a modified Costas loop would

be too narrow for achieving an acceptable cycle slipping probability. An alternative is to perform multiple iteration of phase recovery via a PLL (forward and backward) reducing the loop bandwidth at each iteration. At each iteration (before the last), when reprocessing the UW a check is made about possible cycle slip events, so that the phase can be corrected.

A feed-forward phase recovery based on a sliding window for averaging the pilot phasors is more suitable for our applications because no cycle slipping can happen thanks to the use of pilots. The only problem is the estimation bias which results as a consequence of the frequency error. This bias can be a problem for the burst edge symbols because of the asymmetric pilot location with respect to those symbols. Other feed-forward approaches using also modulated symbols (like the Viterbi & Viterbi [38]) may be adequate for higher SNR but fails at low SNR, particularly if the residual frequency error is not insignificant.

4.2. BURST FORMATS AND ALGORITHMS ASSESSMENT

Burst synchronization at low SNR is challenging, especially for short burst. One of the major problem is represented by the frequency recovery that in these conditions can not efficiently compensate the whole frequency error. Therefore also the carrier phase recovery is problematical because it has to cope with a significant residual frequency error. In particular, classical approaches based on PLLs may fail due to the need of a large loop bandwidth (because of the short time available to converge) and to the resulting non negligible cycle-slipping probability. On the other hand, burst detection and fine timing recovery are generally less critical, although burst detection may also become an issue when very small preambles are considered due to the requirement of minimizing the burst overhead.

In this paragraph, different kinds of burst are analyzed, with UW added for the coarse frequency and timing recovery and pilot symbols added for the fine frequency recovery and phase tracking. A preliminary assessment, separating the two cases of fine timing recovery and fine frequency recovery, will be presented, while the phase recovery algorithms will be tested in the complete receiver simulations reported in paragraph 4.3. Finally, in paragraph 4.4, the burst that best fulfils our requirements will be selected for the multiuser scenario of Chapter 7.

4.2.1. Burst Formats

Four different burst formats have been considered for the synchronization performance analysis and will be described.

- *Burst A*

The simplest burst, called A, is shown in Figure 4.5. It is composed of a preamble UW and a single payload symbols section. Guard symbols are equally inserted before the preamble and after the payload section. The burst A has been used only for preliminary testing purpose, without any frequency and phase errors, and will not be adopted in the real case.

Figure 4.5: Burst format A.

- *Burst B*

The burst format B is reported in Figure 4.6. It is composed of a preamble and several payload symbols sections divided by pilot symbols blocks. Guard symbols are equally inserted before the preamble and after the last payload symbols section. The pilot symbols blocks are evenly spaced and each one is composed by a fixed number of pilot symbols. All the payload symbols sections have the same length except for the last section that can be shorter than the other ones.

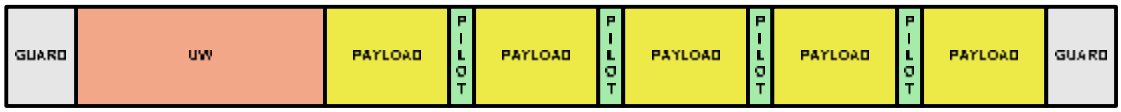


Figure 4.6: Burst format B.

The burst format B has been designed with the aim to exploit the pilot symbols blocks for the fine frequency estimation procedure. Pilots may also be used for carrier phase recovery, if a feed-forward strategy is selected.

- *Burst C*

The burst format C is depicted in Figure 4.7. It is composed of a preamble, a postamble and several payload symbols sections with pilot symbols blocks inserted among the payload sections. Guard symbols are equally inserted before the preamble and after the postamble. The pilot symbols blocks are evenly spaced and each one is composed by a fixed number of pilot symbols. All the payload symbols sections have the same length except for the last section that can be shorter than the other ones.

Figure 4.7: Burst format C.

The burst format C has been designed with the aim to exploit the pilot symbols blocks for the fine frequency estimation procedure (like the burst format B) and to employ the preamble and postamble for obtaining a good phase tracking performance at the edge of the burst when the feed-forward carrier phase estimation using the sliding window technique is selected.

- *Burst D*

The burst format D is shown in Figure 4.8. It is composed of a “middleamble” (UW inserted in the burst middle), several payload symbols sections with pilot symbols blocks inserted among the payload sections and two “edge” symbols blocks respectively inserted before the first payload section and after the last payload section. Guard symbols are equally inserted before the first edge symbols block and after the last edge symbols block. The distance, in symbols, between the first pilot symbols block and the others ones is multiple of a fixed value and each pilot symbol block is composed by a fixed number of pilot symbols.

All the payload sections delimited by the pilot blocks have the same length except for the payload sections adjacent to the middleamble or to the edge symbols blocks. The edge symbols blocks are composed by a fixed number of known symbols.

Figure 4.8: Burst format D.

The burst format D has been designed with the aim to:

- Maximize the number of known symbols that can be used for the channel estimation in case of asynchronous bursts (with respect to the burst format C). In fact the use of a compact middleamble, instead of the pair preamble/postamble, increases the number of overlapping known symbols (see paragraph 6.2).
- Exploit the pilot symbols blocks for the fine frequency estimation procedure.
- Employ the edge symbols blocks for obtaining a good phase tracking performance at the burst edges with the sliding window technique.

4.2.2. Timing Recovery Assessment

The coarse timing recovery is performed on burst formats A, B and D using the coherent correlation with the UW. Instead, for burst format C, the coarse timing recovery employs a separate coherent correlation on the preamble and on the postamble with a non-coherent post integration of the two measurements. Note that a ML burst detection strategy is selected instead of a threshold based one, i.e. the maximum of the correlation process over the guard time uncertainty is declared as the coarse start of burst.

A preliminary analysis revealed that O&M algorithm significantly outperforms the other considered techniques and therefore it has been selected for being used in the simulation campaign. The performance of the O&M is shown in Figure 4.9 for burst formats A, B, C and D considering a 41 symbols length UW and SNR = 2 dB. The optimal sampling instant is equal to 8 and the resulting timing error standard deviation is about $s_T = 0.085$ samples. Note that 4 samples/symbols are assumed in all our simulations, therefore the standard deviation expressed in symbol is $s_T = 0.021$ symbols.

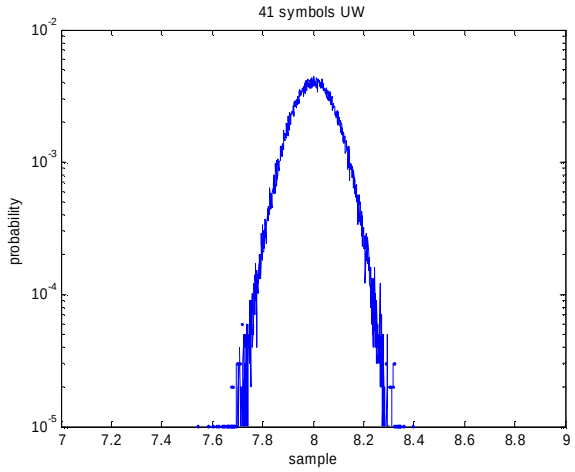


Figure 4.9: O&M fine timing recovery performance, using 41 UW symbols and 488 symbols burst, SNR = 2 dB.

As the O&M timing recovery algorithm uses all the burst symbols, the performance is independent on the burst format and only depends on the number of symbols in the burst. The total number of burst symbols used for O&M timing error estimation was 488 plus UW and pilot symbols. The shown simulation results also consider the effects of the coarse timing estimation done on the UW only. Errors produced by the coarse timing recovery never exceeded half the symbol period (i.e. two samples), hence O&M was able to recover without ambiguity the correct timing.

4.2.3. Frequency Recovery Assessment

Coarse frequency recovery is performed jointly with the coarse timing recovery, employing a bank of correlators tuned on different frequencies. Coarse frequency recovery is performed by selecting the frequency error that results in the highest UW correlation value. For the fine frequency recovery, different techniques have been considered:

- DFT computation on pilot symbols and, optionally, on UW symbols. This technique can be used on all the burst formats.
- M&M algorithm operating on the known symbols. This technique can be used on the burst formats B, C and D exploiting the pilot symbols sequence and the subset of UW symbols that are evenly spaced with the pilot symbol sequence. Also for the minimum

SNIR values of interest this technique can effectively replace the DFT technique that is more complex to implement.

Figure 4.10 illustrates the fine frequency recovery performance using the DFT with SNR = 2 dB for burst format B, burst length of 488 symbols, 41 UW symbols and 41 pilot symbols. The residual coarse frequency error is 2% of the symbol rate (more than the maximum possible for the AMPIST scenario) and the resulting frequency error standard deviation (normalized to the symbol rate) measured after the DFT correction is about $s_F^{\text{norm}} = 10^{-4}$.

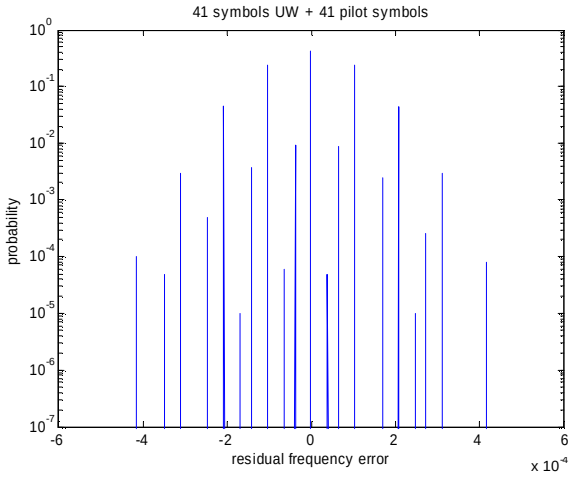


Figure 4.10: Frequency recovery performance with 41 UW symbols and 41 pilot symbols, coarse frequency error 2%, employing the DFT on burst B.

Figure 4.11 shows the frequency recovery performance using the DFT with SNR = 2 dB for burst format C, burst length of 488 symbols, 20 symbols preamble, 20 symbols postamble and 41 pilot symbols. The performance is practically the same as the previous case with burst B.

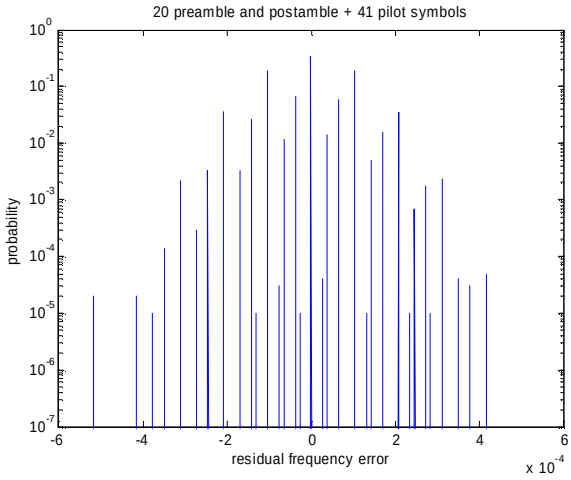


Figure 4.11: Frequency recovery performance with 20 symbols preamble and postamble, 41 pilot symbols, coarse frequency error 2%, employing the DFT on burst C.

Figure 4.12 illustrates the performance of the frequency recovery using the M&M algorithm with SNR = 2 dB for burst format D, burst length of 488 symbols, 41 UW symbols preamble, 40 pilot symbols. Also in this case the result is $s_F^{\text{norm}} = 10^{-4}$.

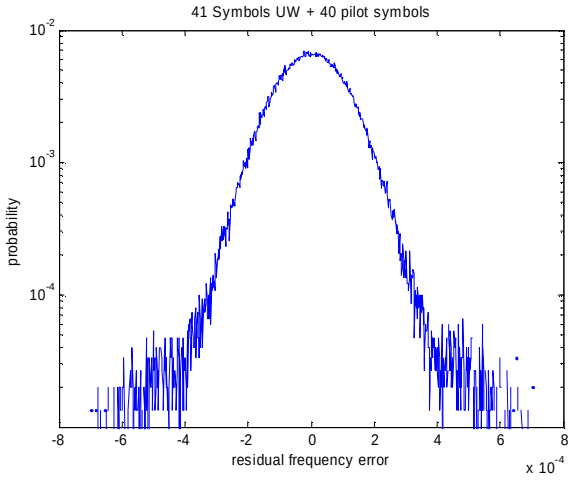


Figure 4.12: Frequency recovery performance with 41 UW symbols and 40 pilot symbols, coarse frequency error 2%, employing M&M on burst D.

For the same overhead, the M&M algorithm achieved basically the same performance as the DFT-based approach. Given its lower complexity, the M&M is the obvious choice for the modem fine frequency recovery with burst format B, C or D.

4.3. SINGLE USER DECODING RESULTS

Extensive software simulations in a C++ environment have been run so as to evaluate the performance of the joint selected timing, frequency and phase synchronization solutions. All the simulations assumptions have been chosen in order to obtain conservative results. The lowest supported symbol rate, i.e. 128 kbaud, has been considered in the simulations because it represents the worst case for the impairment due to the received frequency errors and phase noise. The modulation is the most hostile QPSK 1/2 (S2 mode), with burst payload size equal to 488 symbols. The system operates at SNR = 2 dB, that is the most critical situation expected in a real operational scenario. The following characteristics were implemented:

- Maximum frequency error equal to 2.5 kHz, about 2% of the symbol rate.
- Phase noise mask as specified in the DVB-RCS standard, reported in Table 4.1.
- Shaping filter roll-off 0.35.
- Channel model linear, i.e. no distortion is introduced on the transmission side.

Frequency	DVB-RCS Phase Noise Mask [dBc/Hz]
10 Hz	-16
100 Hz	-54
1 kHz	-64
10 kHz	-74
100 kHz	-89
1 MHz	-106
10 MHz (floor)	-116

Table 4.1: DVB-RCS phase noise mask.

Three different scenarios, each one employing a different burst format, will be studied in the next paragraphs and finally the best burst format for our applications will be selected.

4.3.1. Scenario 1

The scenario 1 employs the burst format B with 41 symbols of UW and 40 pilot blocks, composed by only one pilot symbol each. The resulting pilot symbols block period is 12 symbols. The receiver processing chain for this scenario is shown in Figure 4.13.

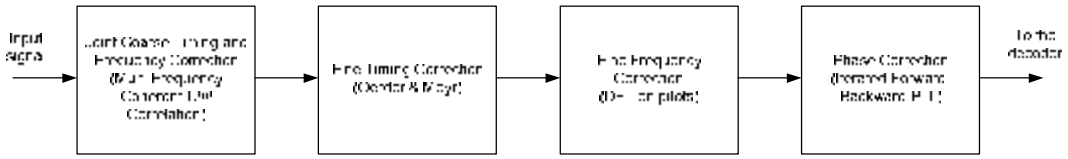


Figure 4.13: Single user scenario 1, receiving processing chain.

The joint coarse timing and frequency recovery is performed employing a multi-frequency coherent UW correlation. The O&M algorithm is used for the fine timing correction. The fine frequency correction is performed using the DFT method on the pilot sequence only. For the phase correction the iterated forward-backward loop is implemented with 16 iterations for a second order PLL. The performance is summarized with the BER and Frame Error Rate (FER) curves, reported in Figure 4.14 and the implementation loss for $FER = 10^{-6}$ is chosen as figure of merit to compare the different considered solutions.

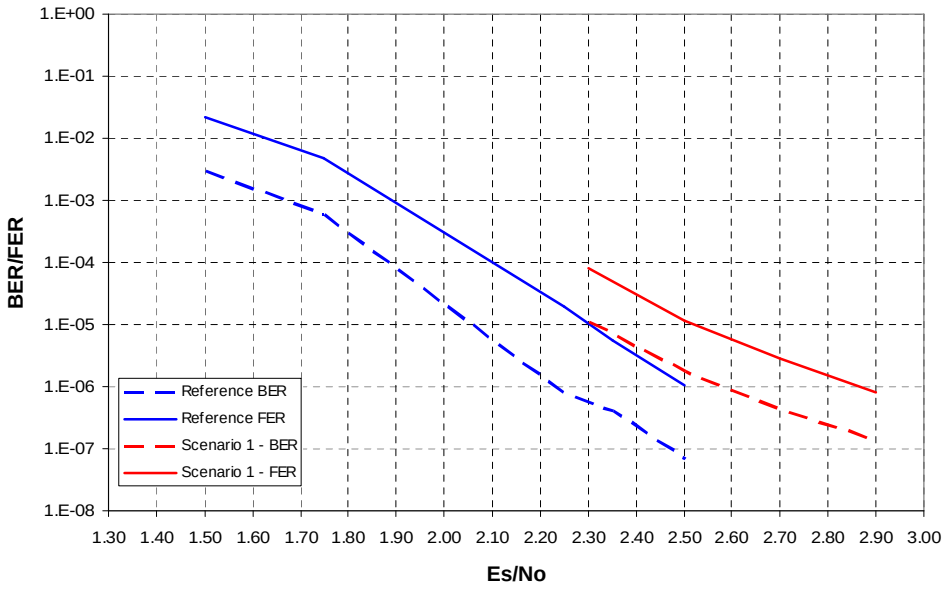


Figure 4.14: Single user, scenario 1 results.

The implementation loss at $FER = 10^{-6}$ is about 0.35 dB. It is worth noting that the BER and FER curves seem to deviate from the reference curves and to exhibit a floor starting from E_s/N_o = 2.5 dB. This is probably due to the cycle slips introduced by the iterated PLL loop. However, for the sake of simplicity, simulations were done without the cycle slipping detection feature and, exploiting the UW knowledge, cycle slipping happened during the initial iterations (where cycle slipping is more likely given the larger loop bandwidth) could have been recovered.

4.3.2. Scenario 2

The scenario 2 employs the burst format C with 20 symbols of preamble, 20 symbols of postamble and 40 pilot blocks, composed by only one pilot symbol each. The resulting pilot symbols block period is 12 symbols. The receiver processing chain for this scenario is shown in Figure 4.15:

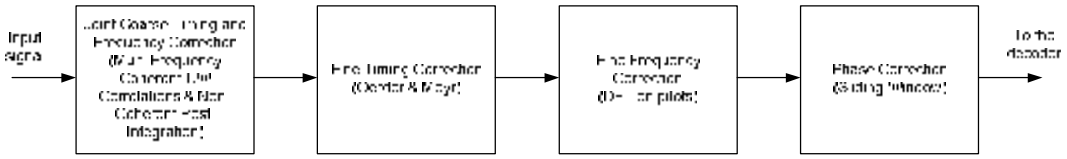


Figure 4.15: Single user scenario 2, receiving processing chain.

The joint coarse timing and frequency recovery is performed employing a multi-frequency coherent UW correlation plus a non coherent post-integration. The O&M algorithm is used for fine timing correction. The fine frequency correction is performed using the DFT method on the pilot sequence only. For the phase correction, the feed-forward sliding window technique is used. The window size is 229 symbols and then it includes 20 pilot blocks. No weighting of the pilot symbols in the window is done (i.e. a rectangular window shape is used). From the BER/FER curve, reported in Figure 4.16, the implementation loss at $\mathrm{FER} = 10^{-6}$ is about 0.35 dB. In this case the BER and FER curves do not show any floor.

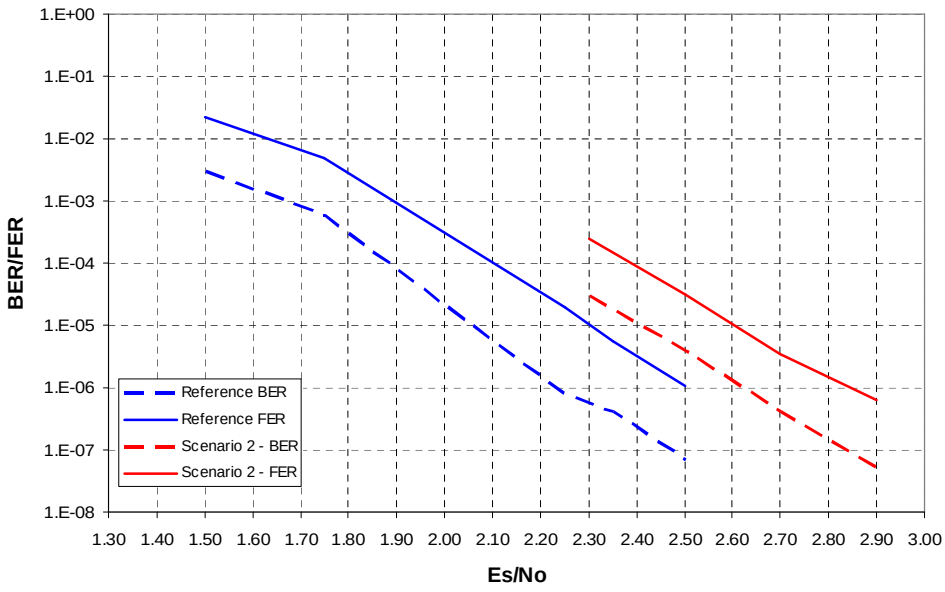


Figure 4.16: Single user, scenario 2 results.

4.3.3. Scenario 3

The scenario 3 employs the burst format D with 41 symbols of middleamble, 38 pilot blocks, composed by only one pilot symbol each, and 2 edge blocks. The resulting pilot symbols block period is 12 symbols. The receiver processing chain for this scenario is shown in Figure 4.17:

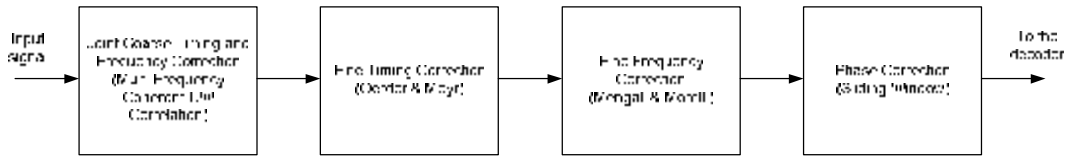


Figure 4.17: Single user scenario 3, receiving processing chain.

The joint coarse timing and frequency recovery is performed employing a multi-frequency coherent UW correlation. The O&M algorithm is used for the fine timing correction. The fine frequency correction is performed using the M&M algorithm on the pilot symbols and on the UW symbols that are evenly spaced with the pilot symbol sequence. For the phase correction the sliding window technique is employed. The window size is 229 symbols, that covers a span of 20 pilot symbol blocks, and the window shape is rectangular.

Figure 4.18 shows the simulation results: it appears that the modem loss at $\mathrm{FER} = 10^{-6}$ can be limited to about 0.5 dB. The label numbers for the various curves represent the number of pilot fields (each one containing one symbol), the number of edge symbols and the phase recovery sliding window size respectively. Note that the sliding window performance is better for burst C with respect to burst D, due to the presence of preamble and postamble that helps the phase recovery at the burst edges. To achieve results comparable with the ones obtained for burst C using the sliding window phase recovery method, a significant number of edge symbols is required.

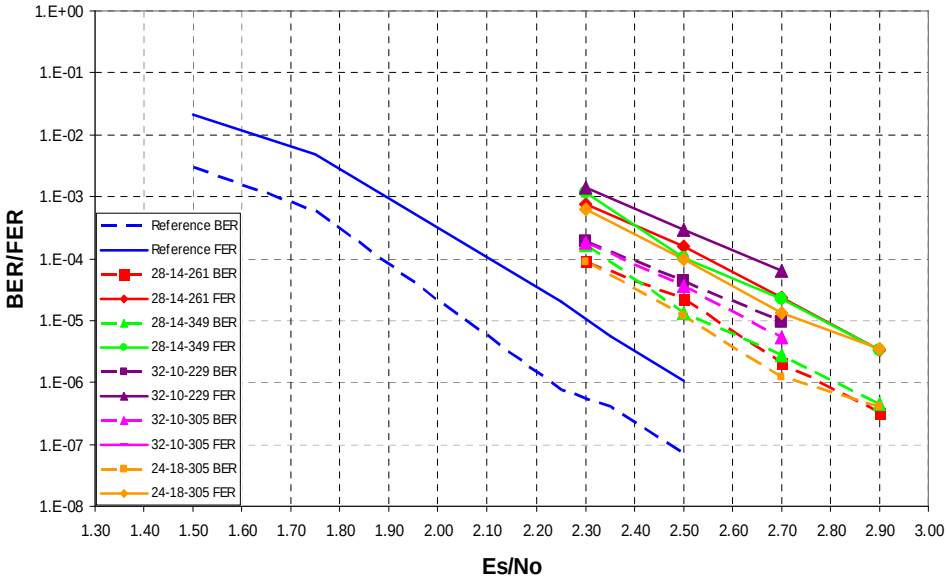


Figure 4.18: Single user, scenario 3 results.

4.4. CONCLUSIONS

In this Chapter an exhaustive analysis of applicable recovery methodologies has been carried out with simulations of the most hostile AMPIST mode S2 (QPSK 1/2). Acceptable performance is obtained with an 80 symbols overhead for 488 symbols payload (14% overhead). For modes operating at higher SNR, it could be possible to reduce the overhead removing the pilot symbols and employing simpler algorithms, but, since the number of total burst symbols must be the same for all the modes, overhead symbols will be inserted anyway. Therefore the same recovery algorithms are employed for all the modes, with reduced loss with respect to the QPSK 1/2 case. The next Figure 4.19 shows the amount of overhead for the AMPIST operative modes:

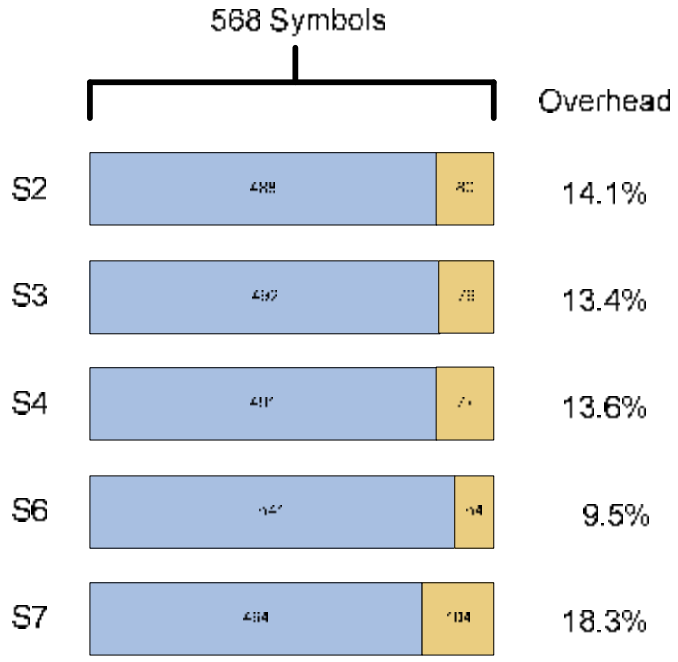


Figure 4.19: Proposed overhead for the AMPIST operative modes.

The burst format C will be adopted, since it has been demonstrated its ability to track the carrier phase on the burst edges. The M&M algorithm will be used for the fine frequency recovery as it has basically the same performance as the more complex DFT-based approach. The sliding window technique is selected for the phase recovery, because the iterative PLL solution would be too complex and would lead to the same performance (although it may allow the reduction of the distributed pilots). Table 4.2 summarizes the obtained guidelines:

ISSUE	SELECTION
Burst format	Burst C
Joint coarse timing and coarse frequency recovery	UW multi-frequency coherent correlation with non coherent post-integration
Fine frequency recovery	M&M algorithm on pilot symbols
Fine timing recovery	O&M algorithm on all the symbols
Phase recovery	Sliding window (229 symbols)
Preamble length	20 symbols
Postamble length	20 symbols
Pilots number	40 symbols
Guard symbols number	2 symbols at the burst beginning 2 symbols at the burst end

Table 4.2: Adopted recovery algorithms and burst format.

5. TURBO CODES

Berrou, Glavieux and Thitimajshima [20] introduced in 1993 a revolutionary error-control coding technique, which they called turbo coding. This coding technique consists essentially of a parallel concatenation of two binary convolutional codes, decoded by an iterative decoding algorithm that makes the resulting BER performance be close to the Shannon limit.

5.1. TURBO ENCODING

In the turbo encoder structure, two recursive convolutional systematic encoders are arranged in parallel concatenation, so that each input element is encoded twice, but the input to the second encoder passes first through a random interleaver, designed to make the encoder output sequences statically independent from each other. Systematic encoders have the property of repeating the first uncoded bit as the first coded bit in output. The systematic encoders adopted in the turbo encoding have usually code rate 1/2. As a result of the systematic form of the coding scheme and the double encoding of each input bit, the resulting code rate should be 1/3. In order to control the rate, the puncturing of the encoder outputs is usually included.

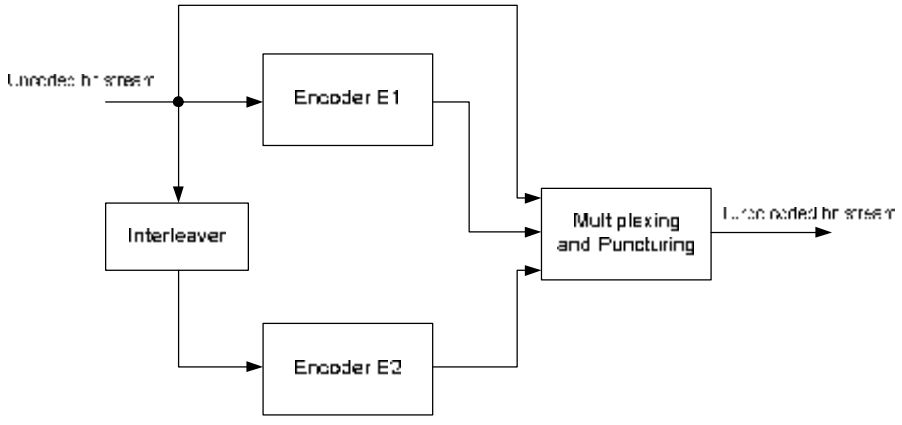


Figure 5.1: Turbo encoder scheme.

5.2. TURBO DECODING

The turbo decoding is obviously much more difficult than the encoding and involves the corresponding decoders of the two convolutional codes iteratively exchanging soft-decision information. Therefore the decoders operate in a Soft-Input Soft-Output mode (SISO), that is the resulting decoding output must be estimated and the usual Viterbi algorithm can not be used. Both decoders operate by utilizing what is called the *a priori* information (i.e. the probabilities $P(b_i = +1)$ of the sent bits b_i) and together with the channel information provided by the samples of the received sequence and information about the structure of the code, they produce an estimate of the message bits.

They are also able to produce an estimate called the *extrinsic* information (EXT), which is passed to the other decoder, information that in the following iteration will be used as the *a priori* information of the other decoder. Thus the first decoder generates extrinsic information that is taken by the second decoder as its a *priori* information. This procedure is repeated in the second decoder, which by using the *a priori* information, the channel information and the code information generates again an estimation of the message information, and also an extrinsic information that is now passed to the first decoder. The first decoder then takes the received information as its *a priori* information for the new iteration and operates in the same way described before, and so on.

The iterative passing of information between the first and the second decoders continues until a given number of iterations is reached. With each iteration the estimate of the message bits improve and it usually converges to a correct estimate of the message. However the improvement of the estimate does not increase linearly and so in practice it is enough to utilize a reasonable small number of iterations to achieve acceptable performance. The most suitable decoding algorithms that performs SISO decisions is the BCJR algorithm [22].

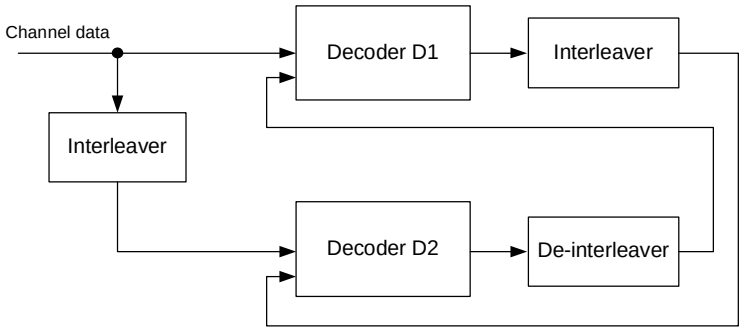


Figure 5.2: Turbo decoder scheme.

The turbo decoder is shown in Figure 5.2: in the decoding procedure each decoder takes into account the information provided by the samples of the channel, which correspond to the systematic (uncoded bits) and parity bits, together with the *a priori* information that was provided by the other decoder, which was calculated as its extrinsic information in the previous iteration.

The soft-decision information provided by the BCJR decoders is an estimate of the corresponding bit being a '1' or a '0'; that is, it is a measure of the probability that the decoded bit is a '1' or a '0'. This information is more conveniently evaluated in logarithmic form, by using what is known as a Log Likelihood Ratio (LLR), defined in the next paragraph 5.2.1. This measure is very suitable because it is a signed number, and its sign directly indicates whether the bit being estimated is a '1' (positive sign) or a '0' (negative sign), whereas its magnitude gives a quantitative measure of the probability that the decoded bit is a '1' or a '0'. The BCJR algorithm is then presented in paragraph 5.2.2.

5.2.1. The Log Likelihood Ratio

The LLR is the most common information measure or metric used in iterative decoding algorithms. The LLR for a bit b_i is denoted as $L(b_i)$, and it is defined as the natural logarithm of the quotient between the probabilities that the bit is equal to '1' or '0'. Since this is a signed number, the sign can be directly considered as representative of the symbol which is being estimated, and so the decision is taken over alphabet $\{-1, +1\}$, rather than over the binary information alphabet $\{0, 1\}$. This estimate is then defined as:

$$L(b_i) = \ln\left(\frac{P(b_i = +1)}{P(b_i = -1)}\right) \tag{5.1}$$

This definition will be found more convenient in the description of the decoding algorithms, where the sign of the LLR is directly used as the hard decision of the estimate, and its value is utilized as the magnitude of the estimate reliability.

Figure 5.3 shows the LLR as a function of $P(b_i = +1)$, which is positive if $P(b_i = +1) > 0.5$ (bit '1' is more likely than bit '0') and it is negative if $P(b_i = +1) < 0.5$ (bit '0' is more likely than bit '1'). The magnitude of this amount is a measure of the probability that the estimated bit adopts one of these two value.

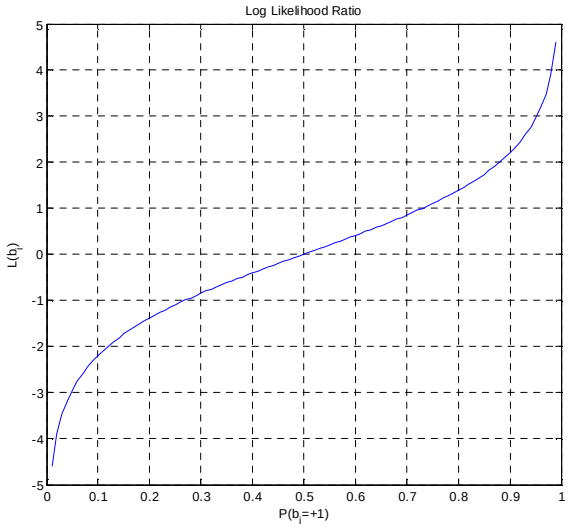


Figure 5.3: LLR as a function of the bit probability of the symbol +1.

An useful propriety can be derived for the LLR, that is:

$$\tanh(L/2) = \frac{e^{\ln\left(\sqrt{\frac{P_{+1}}{P_{-1}}}\right)} - e^{-\ln\left(\sqrt{\frac{P_{+1}}{P_{-1}}}\right)}}{e^{\ln\left(\sqrt{\frac{P_{+1}}{P_{-1}}}\right)} + e^{-\ln\left(\sqrt{\frac{P_{+1}}{P_{-1}}}\right)}} = 2P_{+1} - 1 \tag{5.2}$$

where $P_{+1} = P(b_i = +1)$ and $P_{-1} = P(b_i = -1)$. The (5.2) will be used for translating the logarithmic variables calculated in the BCJR algorithm and exchanged inside the turbo decoder into the linear domain, where the signals are subtracted in the MUD algorithms. In particular, $\tanh(L/2)$ is +1 when $P_{+1} = 1$ and -1 if $P_{+1} = 0$, so that it can be adopted for reconstructing the soft estimated sent bit.

5.2.2. The BCJR Algorithm

The LLR can also be defined for conditional probabilities. The Maximum a Posteriori (MAP) decoding algorithm performs a soft decision of a given bit conditioned or based on the reception of a block **Y** of *n* vectors, $P(b_i \mid \mathbf{Y})$, with $\mathbf{Y} = \{Y_1, Y_2, \mathbf{K}, Y_n\}$. It will supposed that $X_i = \{x_{i1}, \mathbf{K}, x_{iN}\}$, where x_{ik} are the transmitted encoded bits (with code rate $1/N$) relative to the information bit b_i, and $Y_i = \{y_{i1}, \mathbf{K}, y_{iN}\}$ is the received vector after the channel.

When the source output $\mathbf{X} = \{X_1, X_2, \mathbf{K}, X_n\}$ is a sequence of independent discrete data, then the source is considered to be a discrete memoryless source. However, this is not the most suitable model for the encoded output sequence generated by a trellis encoder, since the output symbols are related and the sequence contains some degree of memory. The behaviour of this sort of encoded sequences is more appropriately described by the model of the so called discrete hidden Markov source. This sequence is input to a discrete memoryless channel and the resulting output sequence **Y** can also be considered as a discrete hidden Markov source.

When dealing with a given discrete hidden Markov source, the intention is to determine the hidden variables as a function of the observable variables. In this case it means to estimate the sequence **X** as a function of the observation of the sequence **Y**. An iterative solution for the above problem is the BCJR algorithm. The aim of this algorithm is to determine the following LLR for all the transmitted information bit b_i:

$$L(b_i \mid \mathbf{Y}) = \ln\left(\frac{P(b_i = +1 \mid \mathbf{Y})}{P(b_i = -1 \mid \mathbf{Y})}\right) \tag{5.3}$$

This conditioned probability $P(b_i \mid \mathbf{Y})$ is the *a posteriori* probability (APP) of b_i and therefore the BCJR algorithm decodes b_i by choosing the maximum between $P(b_i = +1 \mid \mathbf{Y})$ and $P(b_i = -1 \mid \mathbf{Y})$, performing a MAP decoding. Another two parameters involved in the turbo decoding process are the *a priori* LLR $L(b_i)$ and the constant $L_c = 2\,\mathrm{E_b} / \sigma^2$, a measure of the channel SNR.

The BCJR was implemented first for the trellis decoding of both block and convolutional codes, and, in comparison with the well known Viterbi decoding algorithm, the proposed algorithm did not provide any particular advantage, as its complexity was higher than that of the Viterbi decoder. However, this is a decoding algorithm that inherently utilizes SISO decisions, and that becomes a decisive factor for its application in the iterative decoding of turbo codes. The BCJR MAP decoding algorithm determines $L(b_i \mid \mathbf{Y})$ as a function of the $\mathbf{Y}$ sequence, the encoding trellis, $L(b_j)$ ($j = 1, \mathbf{K}, n$) and L_c. It can be shown that an *extrinsic* LLR $L_e(b_i)$, depending neither on the channel information y_{i1} of the message bit b_i nor on the *a priori* $L(b_i)$ can be obtained from the APP LLR:

$$L_e(b_i) = L(b_i \mid \mathbf{Y}) - L(b_i) - L_c\, y_{i1} \tag{5.4}$$

where y_{i1} is the first received bit corresponding to the transmitted bit $x_{i1} = b_i$, as it is a systematic encoding.

Finally, referring to Figure 5.2, the iterative turbo decoding algorithm can be deeper explained using the BCJR terminology as shown in Figure 5.4, where the superscript in parentheses indicates the iteration number, the subscript the decoder number and I is the decoder interleaving function. Note that for the first decoder 1 iteration, no information is available for its *a priori* information, hence it will be set to the initial value of 0, i.e. $P(b_i = -1) = P(b_i = +1) = 0.5$ for any $i = 1, \mathbf{K}, n$.

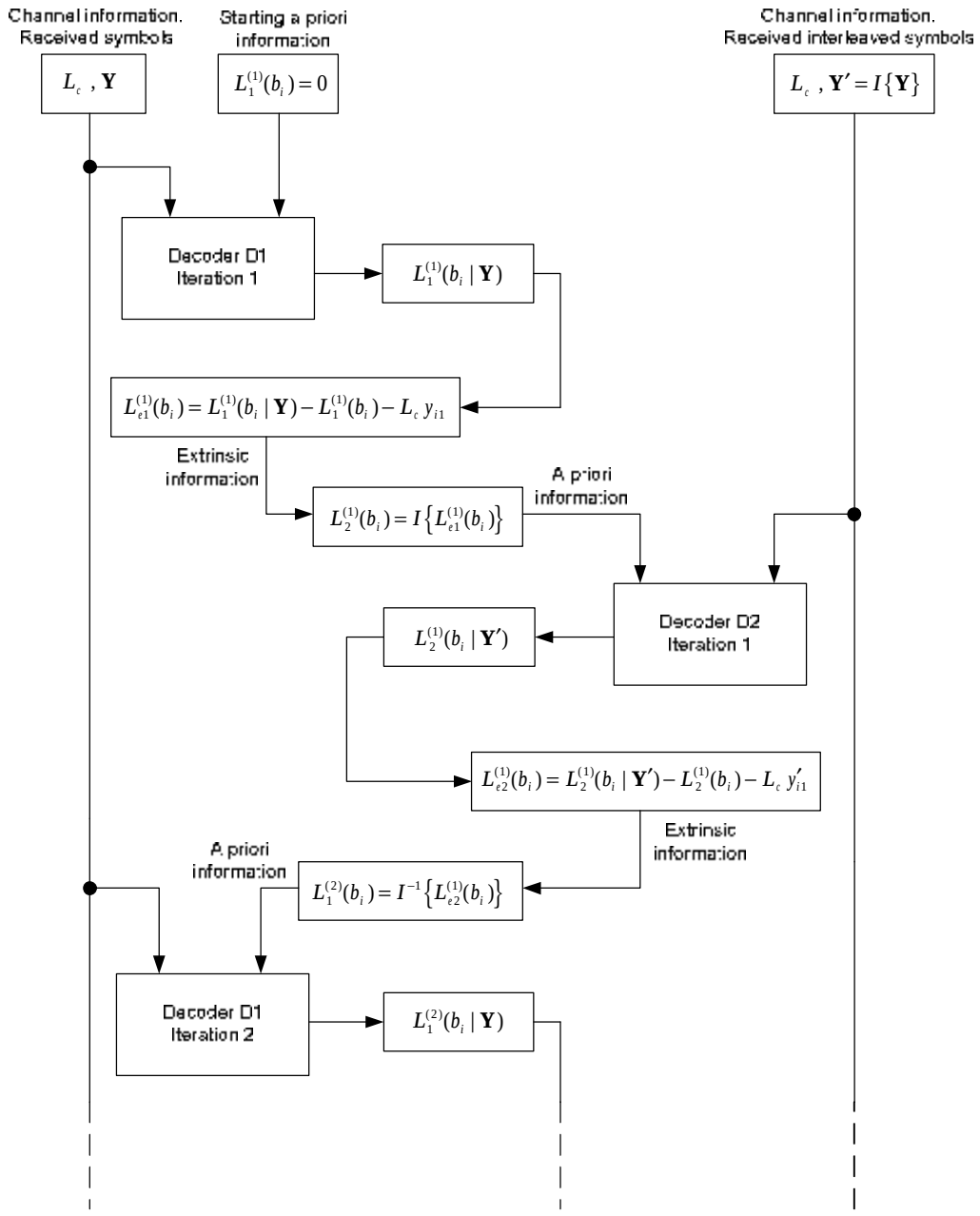


Figure 5.4: Iterative decoding of turbo codes.

5.2.3. Simplified Decoding Algorithms for Turbo Decoding

The decoding algorithm already introduced in this Chapter for turbo codes decoding is the MAP BCJR algorithm. This algorithm is in general of high complexity, and, on the other hand, sums and products involved in its calculation can lead to underflow and overflow problems in practical implementations. These calculations also require considerable amount of memory to store all the values, until a decoding decision is taken. A logarithmic version of this algorithm appears to be a solution for many of the above calculation problems that the original version of the BCJR algorithm faces. The basic idea is that by converting calculations into their logarithmic form, products convert into sums. The logarithm of a sum of two or more terms seems to be a new complication, but this operation is solved by using the following equation:

$$\ln(e^{A}+e^{B}) = \max(A, B) + \ln(1+e^{-|A-B|}) = \max(A, B) + f_c(|A-B|) \qquad (5.5)$$

where $f_c(|A-B|)$ is a correction factor that can be either exactly calculated or, in practical implementations of this algorithm, obtained from a look-up table.

The logarithmic version of the MAP BCJR algorithm greatly reduces the overflow and underflow effects in its application. This logarithmic version is known as the Log MAP BCJR or MAX* algorithm. As explained above, the correction term in equation (5.5) can be appropriately taken from a look-up table. Another and even simpler version of the Log MAP BCJR algorithm is the so called Max Log MAP BCJR algorithm, or MAX, in which the correction term is omitted in the calculation, and the equation (5.5) is used by simply evaluating the max value of the involved quantities. A detailed analysis can be found in [37], where it is shown that the MAX algorithm and the Soft-Output Viterbi Algorithm (SOVA) are those of minimal complexity, but with a level of degradation in BER performance with respect to the MAX* algorithm. Therefore, as usual, decoding complexity is in a trade-off with BER performance. The degradation in BER performance is around 0.6 dB between the best decoder, the MAX* decoder, and the worst, the SOVA decoder.

6. MULTIUSER DETECTION

The first studies on MUD were collected in [2], where in addition to the already known linear methodologies (such as the MMSE, or the decorrelating filter), a decision feedback algorithms was proposed, i.e. the demodulated symbols were remodulated and subtracted in some way to the received signals in order to delete the mutual interference. Later this approach benefited from the adoption of the turbo coding error correction scheme, by exploiting its iterative methodology. The main MUD algorithms will be described in this Chapter.

6.1. THE MMSE ALGORITHM

For the sake of simplicity, the satellite channel model for a two channels CCI scenario will be studied. The extension to the general *K* users scenario is straightforward. The channel model is shown in Figure 6.1:

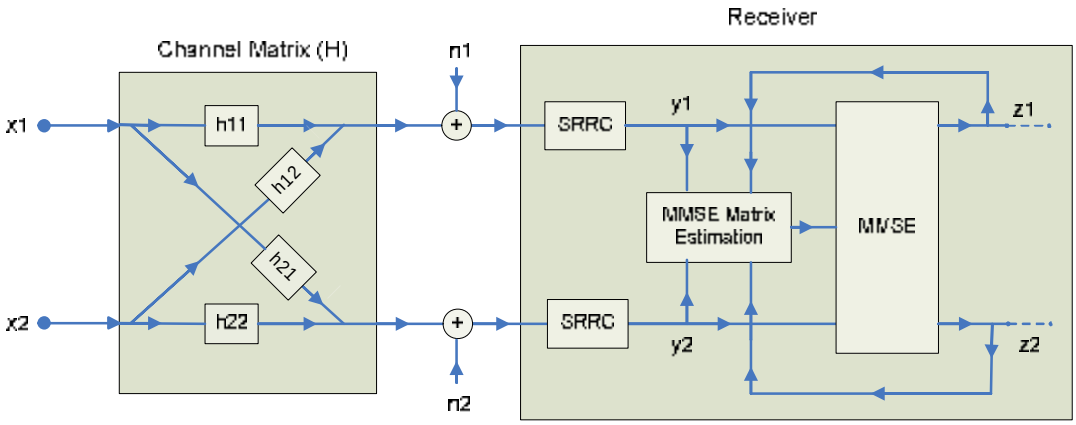


Figure 6.1: CCI channel model.

Therefore the satellite channel can be represented as a channel matrix **H**:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{6.1}$$

where:

x is the column vector $\{x_1, x_2\}^T$, representing the transmitted signals: x_1 and x_2 carry different information using the same frequency band on different beams.

y = $\{y_1, y_2\}^T$, represents the corresponding signal as received on the two receiver arms: y_1 and y_2 are affected by the channel noise and the CCI effect.

n is the column vector whose elements n_1 and n_2 are the thermal noise affecting y_1 and y_2 respectively. The noise covariance matrix $\mathbf{\Sigma}^2 = E\left[\mathbf{n}\mathbf{n}^H\right]$ is diagonal as the noise on the two channels is supposed uncorrelated.

H is the CCI coupling matrix or channel matrix, with element h_{ij} representing the complex gain with which the signal transmitted on beam *j* is cross-coupled with the receiver tuned on beam *i*.

The MMSE methodology estimates the original transmitted signals by a linear transformation of the received signal vector **y** which minimizes the mean squared error between the estimated signals and the real ones. This linear transformation **M** is the MMSE solution to the problem. The expression of the transformation **M** is:

$$\mathbf{M}_{MMSE} = \arg\min_{\mathbf{M}} E\left[\left\|\mathbf{x} - \mathbf{M}\mathbf{y}\right\|^2\right] \tag{6.2}$$

but

$$\begin{aligned} E\left[\left\|\mathbf{x} - \mathbf{M}\mathbf{y}\right\|^2\right] &= E\left[\operatorname{trace}\left\{(\mathbf{x} - \mathbf{M}\mathbf{y})(\mathbf{x} - \mathbf{M}\mathbf{y})^H\right\}\right] \\ &= \operatorname{trace}\left\{E\left[(\mathbf{x} - \mathbf{M}\mathbf{y})(\mathbf{x} - \mathbf{M}\mathbf{y})^H\right]\right\} \\ &= \operatorname{trace}\left\{\operatorname{cov}(\mathbf{x} - \mathbf{M}\mathbf{y})\right\} \end{aligned} \tag{6.3}$$

and the covariance can be further expanded as:

$$\mathrm{cov}(\mathbf{x}-\mathbf{My}) = E\left[\mathbf{xx}^H\right] - E\left[\mathbf{xy}^H\right]\mathbf{M}^H - \mathbf{M}E\left[\mathbf{yx}^H\right] + \mathbf{M}E\left[\mathbf{yy}^H\right]\mathbf{M}^H \tag{6.4}$$

where (supposing unitary transmitted signal power: the gains are incorporated in the **H** matrix):

$$\begin{aligned} E\left[\mathbf{xx}^H\right] &= \mathbf{I} \\ E\left[\mathbf{xy}^H\right] &= \mathbf{H}^H \\ E\left[\mathbf{yx}^H\right] &= \mathbf{H} \\ E\left[\mathbf{yy}^H\right] &= \mathbf{HH}^H + \boldsymbol{\Sigma}^2 \end{aligned} \tag{6.5}$$

Substituting, we have:

$$\mathrm{cov}(\mathbf{x}-\mathbf{My}) = \mathbf{I} - \mathbf{H}^H\mathbf{M}^H - \mathbf{MH} + \mathbf{M}(\mathbf{HH}^H + \boldsymbol{\Sigma}^2)\mathbf{M}^H \tag{6.6}$$

Computing the derivative of the (6.3) with respect to **M** and equating to zero, one can find the transformation which minimizes the mean squared error. It is:

$$\mathbf{M}_{MMSE} = \mathbf{H}^H(\mathbf{HH}^H + \boldsymbol{\Sigma}^2)^{-1} \tag{6.7}$$

Hence the matrix transformation **M** can be readily computed once the channel matrix **H** and the noise covariance matrix $\boldsymbol{\Sigma}^2$ are known. While it may be expected that the noise covariance matrix is almost stable with time, this is certainly not true for the channel matrix whose estimation has to be updated. In practical implementations, from (6.5) the term $(\mathbf{HH}^H + \boldsymbol{\Sigma}^2)$ is the covariance matrix of the whole received signal including noise and it will be computed by suitably averaging the matrix $\mathbf{yy}^H$ over a sufficient time interval, while the channel matrix **H** will be separately estimated (see paragraph 6.2).

Finally, note that in absence of noise the signal transmitted **x** is perfectly extracted from the received signal **y** filtered by the MMSE transformation:

$$\mathbf{My} = \mathbf{MHx} = \mathbf{H}^H(\mathbf{HH}^H)^{-1}\mathbf{Hx} = \mathbf{H}^H(\mathbf{H}^H)^{-1}\mathbf{H}^{-1}\mathbf{Hx} = \mathbf{x}$$

Instead, in the actual case, the noise is present and the SNIR obtainable after the MMSE filtering is the maximum obtainable for any linear transformation.

6.2. ESTIMATION ISSUES

6.2.1. Channel and MMSE Matrixes Estimation

The channel matrix **H** is not known at the receiver and needs to be estimated by sending a sequence of known data called training sequence. For the AMPIST project, the UW, necessary for the coarse timing and frequency recovery, will be also used as training sequence. Note that the bursts in the DVB-RCS scenario are not perfectly synchronous for the different users, but can have a shift of a few symbols.

Let $c_i(t)$ be the training sequence adopted by user *i*. This sequence is generated starting from a sequence c_{ik}, $k = (1, 2, \mathbf{K}, L)$, given a pulse waveform $g(t)$ with symbol period *T*, as:

$$c_i(t) = \sum_{k=1}^{L} c_{ik}\, g(t - kT) \tag{6.8}$$

For example, c_{ik} may assume complex values equal to $\pm 0.707 \pm j\,0.707$. Due to the time shift, frequency error and unknown signal phase, the actual received UW is:

$$c_i^{act}(t) = c_i(t - \tau_i)e^{j(2\pi f_i t + \phi_i)} \tag{6.9}$$

Let us now suppose to sample the above sequences at the same time and produce a matrix **C** having on each row the sampled representation of the *i*-th sequence. Matrix **C** will have *K* rows, the number of beams and users, and a number of column *S* which is equal to the UW length *L* multiplied by the number of samples per symbol. The matrix of the received sampled UWs after the channel is then:

$$\mathbf{Y}^{UW} = \mathbf{HC} + \mathbf{N} \tag{6.10}$$

where $\mathbf{N}$ is the $K \times S$ matrix for the noise contribution. Post-multiplying $\mathbf{Y}^{UW}$ by the $\mathbf{C}^{+}$ Moore-Penrose pseudo-inverse of the matrix $\mathbf{C}$, defined as:

$$\mathbf{C}^{+} = \mathbf{C}^{H}(\mathbf{C}\mathbf{C}^{H})^{-1} \tag{6.11}$$

we get:

$$\mathbf{Y}^{UW}\mathbf{C}^{+} = \mathbf{H} + \mathbf{N}\mathbf{C}^{+} = \hat{\mathbf{H}} \tag{6.12}$$

Hence, apart for the noise term, we are able to get a good estimate $\hat{\mathbf{H}}$ of the channel matrix $\mathbf{H}$.

Walsh-Hadamard sequence UWs are the best choice as their inverse is easy to compute (because it is equal to the hermitian of the matrix to be inverted) and in addition orthogonal matrixes can be shown not to produce noise enhancement (see paragraph 6.2.2). Walsh-Hadamard sequence are however impractical because sequences transmitted by different terminals are not perfectly synchronous and are subjected to different frequency errors: these time and frequency errors make Hadamard sequences no longer orthogonal when they are received. The best choice is to use random sequences as UW for channel estimation. Regarding the complexity of the pseudo-inverse in (6.11), it shall be noted that $\mathbf{C}\mathbf{C}^{H}$ has size $K \times K$, therefore the inverse computation has complexity K^3. There are then two matrix products each one having $S \times K^2$ multiplications. With $K = 4$, as requested in our scenario, the complexity appears manageable even for large *S*.

A problem with the above approach is that one has to know the actual received matrix $\mathbf{C}$ and this implies that we need a good estimation of the timing and frequency error of each user expressed in (6.9). It is easy to understand how an increase of the UW length *L* leads to a better $\mathbf{H}$ estimate, but at the expense of the effective data throughput. In our case, as we have already assessed in Chapter 4, $L = 40$ with 4 points per symbols, and therefore *S* = 160.

In order to obtain the MMSE matrix **M**, after calculating the estimated $\hat{\mathbf{H}}$ matrix, the second step is to compute the last term in (6.7) with the input signals covariance matrix:

$$E\left[\mathbf{y}\mathbf{y}^{H}\right] = \mathbf{H}\mathbf{H}^{H} + \Sigma^{2} = \mathbf{R} \tag{6.13}$$

As the process is ergodic, the **R** matrix elements can be obtained by averaging a sufficient number of input samples. Finally the MMSE matrix **M** can be calculated as:

$$\mathbf{M} = \hat{\mathbf{H}}^{H}\,\mathbf{R}^{-1} \tag{6.14}$$

The usual MMSE estimate for the two users case, coherently with Figure 6.1, is summarized in Figure 6.2.

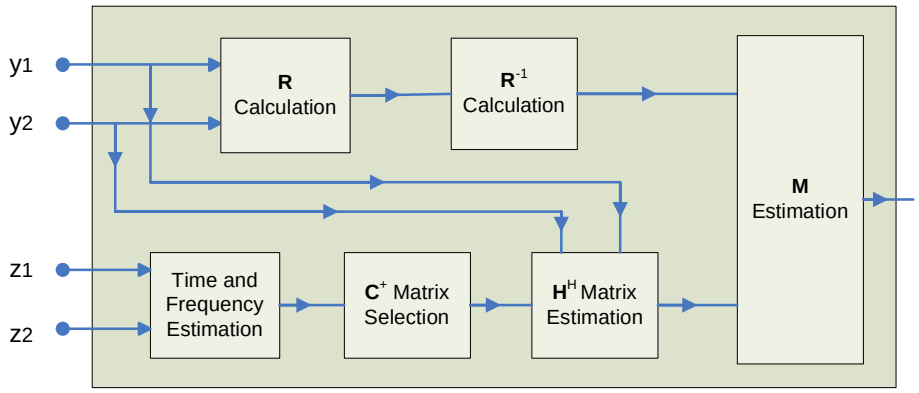


Figure 6.2: MMSE matrix estimation for the two users case.

6.2.2. Channel Matrix Estimation Error

It is well known that optimal training sequences can be constructed using orthogonal sequences. This can be obtained by using Walsh-Hadamard codes. Although this setting in a reverse link satellite communication is not possible due to the users asynchronicity, the optimal training case can be used as an upper bound and will be useful for comparing the case

of random sequence training. The Walsh-Hadamard sequence is a particular code in which the size L is a power of 2 and it is generated as follows:

$$\mathbf{W}_0 = 1$$

$$\mathbf{W}_1 = \begin{pmatrix} \mathbf{W}_0 & \mathbf{W}_0 \\ \mathbf{W}_0 & -\mathbf{W}_0 \end{pmatrix}$$

$$\mathbf{M}$$

$$\mathbf{W}_{2n} = \begin{pmatrix} \mathbf{W}_n & \mathbf{W}_n \\ \mathbf{W}_n & -\mathbf{W}_n \end{pmatrix}$$

Since $K \leq L$, we extract from the $L \times L$ Walsh-Hadamard matrix K codes (in rows) of length L so as to obtain a $\mathbf{C}$ matrix $K \times L$. The specific feature of this code is that $\mathbf{C}\mathbf{C}^H = L\,\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In this case, supposing that users are synchronized, the estimated channel matrix can be rewritten from (6.12) as:

$$\hat{\mathbf{H}} = \mathbf{H} + \frac{1}{L}\mathbf{N}\,\mathbf{C}^H \tag{6.15}$$

Reminding that

$$\text{trace}\left\{\mathbf{A}\,\mathbf{A}^H\right\} = \sum_{i=1}^{K}\sum_{j=1}^{K}\left|a_{ij}\right|^2 \tag{6.16}$$

where $\mathbf{A}$ is a general squared $K \times K$ matrix, with values a_{ij}, the $\hat{\mathbf{H}}$ elements mean error variance is then given by:

$$\sigma^2_{\hat{\mathrm{H}}} = \frac{1}{K^2}\,\mathrm{trace}\left\{E\left[\frac{1}{L^2}\,\mathbf{N}\,\mathbf{C}^H\mathbf{C}\,\mathbf{N}^H\right]\right\}$$

$$= \frac{1}{K^2}\,\mathrm{trace}\left\{\frac{1}{L^2}\,\mathbf{C}^H\mathbf{C}\,E\left[\mathbf{N}^H\mathbf{N}\right]\right\}$$

$$= \frac{1}{K^2}\,\mathrm{trace}\left\{\frac{1}{L^2}\,\mathbf{C}^H\mathbf{C}\,K\,\sigma^2\right\} \qquad (6.17)$$

$$= \frac{\sigma^2}{K\,L^2}\,\mathrm{trace}\left\{\mathbf{C}\,\mathbf{C}^H\right\}$$

$$= \frac{\sigma^2}{L}$$

where σ^2 is the single user noise power, supposed identical for all the users and the property $\mathrm{trace}\{\mathbf{A}\,\mathbf{B}\} = \mathrm{trace}\{\mathbf{B}\,\mathbf{A}\}$ has been used. From (6.17) it is clear that the longer the training sequence L, the better the noise immunity of the channel estimation. However the price to be paid is the transmission efficiency reduction.

As we stated before, due to the lack of synchronization of the users, the training sequences can not be fully orthogonal and will be modeled by random sequences. In this case, the (6.12) can no longer be simplified and

$$\hat{\mathbf{H}} = \mathbf{H} + \mathbf{N}\mathbf{C}^{+} = \mathbf{H} + \mathbf{N}\,\mathbf{C}^H(\mathbf{C}\,\mathbf{C}^H)^{-1} \qquad (6.18)$$

The $\hat{\mathbf{H}}$ elements mean error variance is now given by:

$$\sigma^2_{\hat{\mathrm{H}}} = \frac{1}{K^2}\mathrm{trace}\left\{E\left[\mathbf{N}\mathbf{C}^H(\mathbf{C}\mathbf{C}^H)^{-2}\mathbf{C}\mathbf{N}^H\right]\right\}$$

$$= \frac{1}{K^2}\mathrm{trace}\left\{\mathbf{C}^H(\mathbf{C}\mathbf{C}^H)^{-2}\mathbf{C}E\left[\mathbf{N}^H\mathbf{N}\right]\right\}$$

$$= \frac{1}{K^2}\mathrm{trace}\left\{(\mathbf{C}\mathbf{C}^H)^{-1}K\sigma^2\right\}$$

$$= \frac{\sigma^2}{K}\int_0^\infty \frac{1}{\lambda} f(\lambda)d\lambda$$

$$= \frac{\sigma^2}{K}\frac{1}{\frac{L}{K}-1}$$

$$= \frac{\sigma^2}{L}\frac{1}{1-\frac{K}{L}} \qquad (6.19)$$

where $f(\lambda)$ is the eigenvalue distribution of $\mathbf{C}\mathbf{C}^H$ given by the Marchenko-Pastur law [19]: if we define $\alpha = \frac{L}{K}$, then, only for $\alpha \geq 1$ (usually verified) and random $\mathbf{C}$ matrix, the distribution has a bounded support on $[(\sqrt{\alpha}-1)^2;(\sqrt{\alpha}+1)^2]$ and $f(\lambda) = \sqrt{4\alpha-(\lambda-1-\alpha)^2}\ /\ 2\pi\lambda$, that leads to the reported result. Note that the above (6.19) can be only applied for a random $\mathbf{C}$ matrix, while the relationship $\sigma^2_{\hat{\mathrm{H}}} = \frac{\sigma^2}{K}\mathrm{trace}\left\{(\mathbf{C}\mathbf{C}^H)^{-1}\right\}$ is always valid for any $\mathbf{C}$ and shows that the noise will be more enhanced the more the $\mathbf{C}$ rows (i.e. the UWs of the users) are correlated. As one can see, random and orthogonal training sequences have the same performance up to the scalar factor $\frac{1}{1-\frac{K}{L}}$. In the usual cases this factor is nearly 1 and therefore there is no performance loss.

6.3. SERIAL INTERFERENCE CANCELLATION

The SIC approach is based on a simple and natural idea: if a decision has been made about an interfering user bit, then that interfering signal can be recreated at the receiver and subtracted from the received waveform. This will cancel the interfering signal providing that the decision was correct; otherwise it will double the contribution of the interferer. This kind of non linear methodologies are referred as decision-driven (or decision-feedback) detectors, since they use a previous decision (hard or soft) of the decoders. In this paragraph only the hard decision-feedback is discussed, while the soft decision feedback will be analyzed later.

It is clear how the demodulation order of the channel plays now a very important role, as the decoding error will propagate: the best ordering rule should obviously be SNIR based, but it is not easy to practical implement it and instead the most popular approach is to demodulate the users in the order of decreasing received powers. The matrix **H** is therefore ordered according to the channel strength, i.e. $|h_{11}| \geq |h_{22}| \geq \mathbf{K} \geq |h_{KK}|$, where *K* is the number of users.

The SIC algorithm can be also combined with the MMSE in order to add their benefits: the resulting MMSE-SIC [16]-[17] has triggered a lot of implementation research schemes as it was shown in [18] to be optimal. In the following, it will be assumed that $\mathbf{m}_i$ is the *i*-th row of the matrix **M**, $\mathbf{m}^j$ the *j*-th column and the index in parentheses indicates the number of iteration (e.g. $\mathbf{M}(i)$ is the matrix **M** at the step *i*).

The algorithm relies on a sequential detection of the received blocks: at the first step the usual MMSE matrix $\mathbf{M}(1) = \mathbf{M}$ is computed from the channel matrix $\mathbf{H}(1) = \hat{\mathbf{H}}$ with the (6.14), then the received $K \times 1$ column vector $\mathbf{y}(1)$ of the *K* users is filtered with $\mathbf{m}_1(1)$ (note that the number of iteration in parentheses is also equal to the channel that is being demodulated) and the resulting z_1 is decoded by the first decoder. The decoded sequence $\hat{z}_1$ is then re-modulated, subtracted to the other $K-1$ channels and the process is repeated until the last user is demodulated. The $\mathbf{H}(i+1)$ and $\mathbf{M}(i+1)$ $(i = 1,2,\mathbf{K},K-1)$ matrixes will be obtained from the previous $\mathbf{H}(i)$ and $\mathbf{M}(i)$ by deleting the first row and column or will be recalculated starting from the $K-i+1$ dimensional $\mathbf{y}(i)$, relying on the refinement due to the cancellation of an interferer. Therefore, for the previous defined vectors and matrixes, at the step *i* we get the following dimensions (where *S* was defined in paragraph 6.2.1):

$\mathbf{H}(i)$ and $\mathbf{M}(i)$ = $(K-i+1)$ x $(K-i+1)$

$\mathbf{y}(i)$ = $(K-i+1)$ x 1

$\mathbf{Y}^{UW}(i)$= $(K-i+1)$ x S

$\mathbf{C}(i)$= $(K-i+1)$ x S

The MMSE-SIC algorithm will be initialized with:

$$\mathbf{Y}^{UW}(1)\mathbf{C}^{+}(1)=\mathbf{H}(1)$$

$$E\left[\mathbf{y}(1)\mathbf{y}^{H}(1)\right]=\mathbf{R}(1)$$

$$\mathbf{M}(1)=\mathbf{H}^{H}(1)\mathbf{R}^{-1}(1)$$

and at the iteration i:

$$z_i=\mathbf{m}_1(i)\mathbf{y}(i)$$

$$\hat{z}_i=\text{Decoded}[z_i]$$

$$\mathbf{y}(i+1)=\text{Delete first point of }\left[\mathbf{y}(i)-\mathbf{h}^1(i)\hat{z}_i\right]$$

$$\mathbf{H}(i+1)=\begin{cases}\text{Delete first row and column of }\mathbf{H}(i)\\ \text{or}\\ \mathbf{Y}^{UW}(i+1)\mathbf{C}^{+}(i+1)\end{cases}$$

$$\mathbf{R}(i+1)=\begin{cases}\text{Delete first row and column of }\mathbf{R}(i)\\ \text{or}\\ E\left[\mathbf{y}(i+1)\mathbf{y}^{H}(i+1)\right]\end{cases}$$

$$\mathbf{M}(i+1)=\mathbf{H}^{H}(i+1)\mathbf{R}^{-1}(i+1).$$

The resulting algorithm is showed in Figure 6.3. As already discussed, the MMSE filtering can be avoided for a simpler implementation and the SIC scheme is obtained.

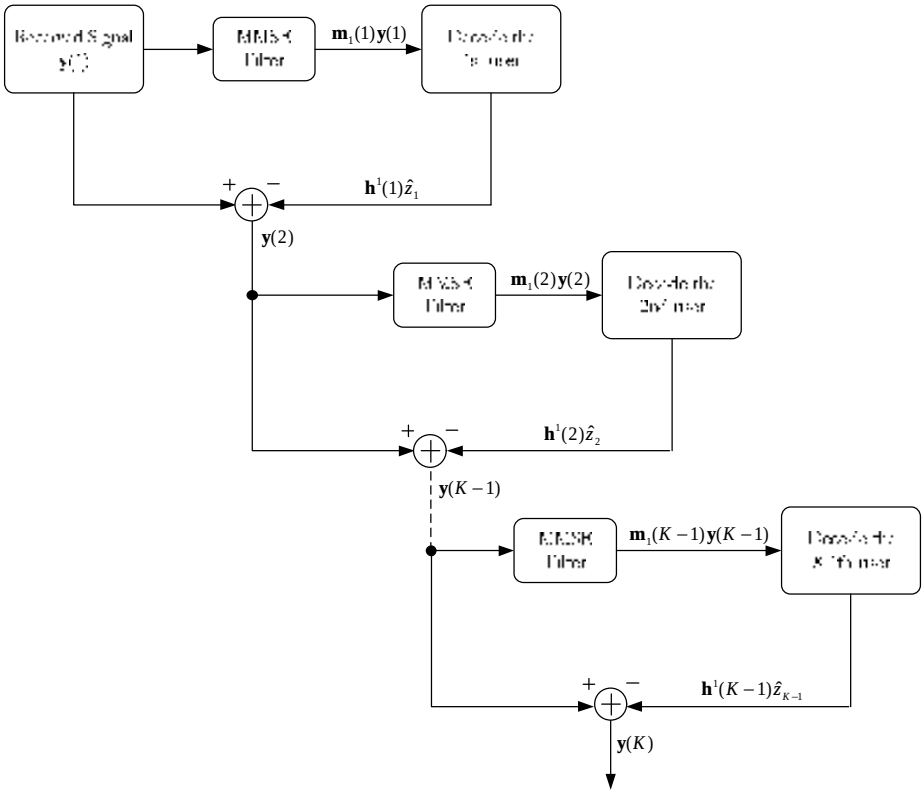


Figure 6.3: MMSE hard-SIC block diagram.

The main shortcomings of the decision-driven algorithms (both parallel and serial interference cancellation) are:

- The relative timing and frequency offsets among the users must be carefully evaluated in order to re-modulate the decoded bits with the appropriate time and frequency.
- The channel matrix must be well estimated in order to have a good interference cancellation.
- The performance is very sensitive to the power users spread: it is mediocre when the received powers are comparable and it is usually better as the power differences increase. This happens because the most powerful user can be perfectly demodulated, then subtracted to the others and so on.

6.4. ITERATIVE MUD ALGORITHMS

In order to analyze the performance for the iterative MUD schemes that will be introduced in this paragraph, the *optimum decoder* for multiuser scenario is presented first. It will be assumed that the receiver not only knows the impulse waveform and timing of every active user, but it also knows (or can estimate) the received amplitudes of all users and the noise level. We will refer to the transmitted signal vector $(y_1, \mathbf{K}, y_K)$ generated as follows:

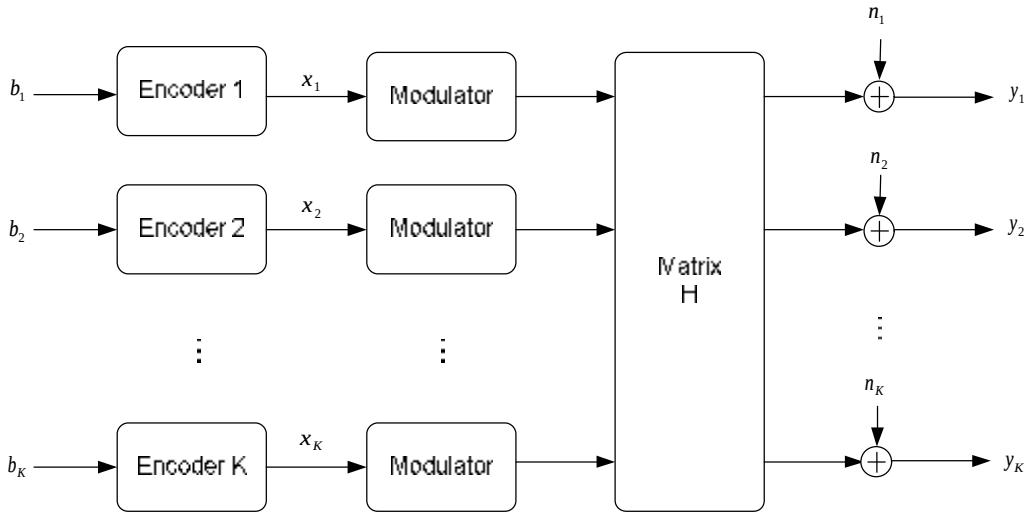


Figure 6.4: Transmitted signals and channel effect.

6.4.1. Optimum Decoder and Belief Propagation

For some time, it was widely believed that the decisions of the conventional SUMF filtering were, if not optimal, almost optimal for channels with a large number of equal-powers users. The wrong conclusion of the near optimality of the SUMF originates from the implicit assumption that the observable used to demodulate user 1 must be restricted to its matched filter output. Instead, although $(y_1, \mathbf{K}, y_K)$ is a sufficient statistics for the data $(b_1, \mathbf{K}, b_K)$, it is not true that y_K is a sufficient statistic for b_K.

Let us start with the uncoded case, where $x_i = b_i$ and $\mathbf{y} = (y_1, \mathbf{K}, y_K)$. The minimum probability of error decision for the user k is obtained by selecting the value b_k^0 of $b_k \in \{-1, +1\}$ that maximizes the APP:

$$\Omega\left[b_k^0 \mid \mathbf{y}\right] \tag{6.20}$$

We could pose a different optimum detection problem by requiring that the receiver selects the vector $(b_1^0, \mathbf{K}, b_K^0)$ that maximizes the joint APP:

$$\Omega\left[(b_1^0, \mathbf{K}, b_K^0) \mid \mathbf{y}\right] \tag{6.21}$$

We can write (6.20) in terms of (6.21):

$$\Omega\left[b_k^0 \mid \mathbf{y}\right] = \sum_{\substack{b_1, \mathbf{K}, b_K \\ b_k = b_k^0}} \Omega\left[(b_1, \mathbf{K}, b_K) \mid \mathbf{y}\right] \tag{6.22}$$

Those optimum detection strategies, which we will refer to as individually optimum and jointly optimum, respectively, need not result in the same decisions. The underlying reason is that the transmitted bits b_i are not independent when conditioned in the observed waveform.

In our case the channels are encoded: the optimal multiuser sequence estimator for coded systems was introduced first by Giallorenzi and Wilson in [1]. They formulated the multiuser ML receiver for convolutionally coded non-dispersive AWGN links. The receiver performs both the function of equalization of the multiuser interference and decoding of the code together using a Viterbi algorithm. The ML decoder has a complexity of approximately $O(2^{kK})$ operations per bit, where k is the convolutional code constraint length. The exponential complexity makes the optimum receiver impractical even for small systems. This motivated the research of sub-optimum strategies, approximating joint decoding, practically viable and able to provide performance close to the optimum. Moher in [23] proposed an iterative structure based on cross entropy minimization techniques or equivalently on the so

called sum-product algorithms on factor graph. Since it has been shown in [24] that the second approach provides a general framework for a large class of iterative joint decoders we will introduce the *Belief Propagation* (BP) algorithm in [23] making use of factor graphs.

Let $b_{k,j}$ denote the *j*-th information bit of user *k* and $\mathbf{b}^k = (b_{k,1}, b_{k,2}, \mathbf{K}, b_{k,M})$ the vector of information bits of user *k*. The corresponding codeword will be the *N* symbols $\mathbf{x}^k = (x_{k,1}, x_{k,2}, \mathbf{K}, x_{k,N})$. The multiuser channel with coding is completely described by the a posteriori probability density function (pdf) of $\{\mathbf{b}^1, \mathbf{K}, \mathbf{b}^K\}$ given the received $K \times N$ matrix $\mathbf{Y}$. Referring to (6.22), the (individually) optimal receiver for coded symbols is given by

$$\hat{b}_{k,j} = \arg \max_{b \in \{-1,+1\}} \sum_{\substack{\mathbf{b}^1, \mathbf{K}, \mathbf{b}^K \\ b_{k,j}=b}} \Omega(\mathbf{b}^1, \mathbf{K}, \mathbf{b}^K \mid \mathbf{Y}) \tag{6.23}$$

Applying the sum-product algorithm on the factor graph we obtain a general method for approximating (6.23). Let us specialize the concept of sum-product factor graph to equation (6.23) as in [24]. By using the fact that the channel is memoryless, that user codewords are independently generated, and that the user information bits have uniform *a priori* probability we can write

$$\hat{b}_{k,j} \propto \arg \max_{b \in \{-1,+1\}} \sum_{\substack{\text{codewords } \mathbf{x}^k \text{ corresponding} \\ \text{to messages } \mathbf{b}^k \text{ with } b_{k,j}=b}} \prod_{n=1}^{N} Q_{k,n}(x_{k,n}) \tag{6.24}$$

where the *marginal probabilities* $Q_{k,n}(x)$:

$$Q_{k,n}(x) \propto \sum_{\substack{\mathbf{X}^n \in A^K \\ x_{k,n}=x}} \exp(-|\mathbf{Y}^n - \mathbf{H}\mathbf{X}^n|^2) \prod_{\substack{j=1 \\ j \neq k}}^{K} P_{j,n}(x_{j,n}) \tag{6.25}$$

with $\mathbf{Y}^n$ the *n*-th column of $\mathbf{Y}$, $\mathbf{X}^n = (x_{1,n}, \mathbf{K}, x_{K,n})^T$ the transmitted vector at symbol *n*, *A* the modulation symbols set, and

$$P_{k,n}(x) \propto \sum_{\substack{\text{codewords } \mathbf{x}^k \\ \text{with } x_{k,n}=x}} \prod_{\substack{j=1 \\ j\neq n}}^{N} Q_{k,j}(x_{k,j}) \tag{6.26}$$

$P_{k,n}(x)$ is the extrinsic information for channel k at symbol n, that is the information provided about a symbol from the other received symbols due to the constraints imposed by the FEC code.

The previous (6.24), that is an approximation of the optimum decoder (6.23) using the sum-product algorithm, is called the BP algorithm and it is an iterative joint decoder with a parallel scheduling with structure shown in Figure 6.5:

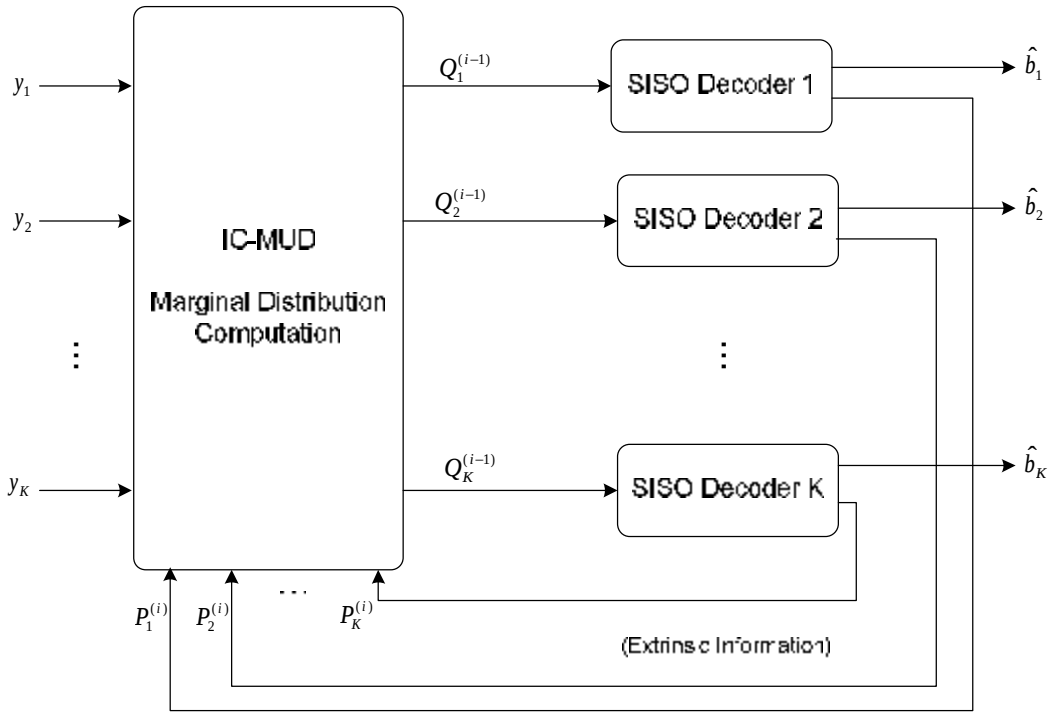


Figure 6.5: Iterative MUD decoder.

The algorithm is intrinsically iterative, that is by iteratively computing the marginal distributions and extrinsic information the $\hat{b}_i$ estimates become more reliable, and nearly

stable after a certain number of iteration. The $Q_k^{(i)}$ denotes the table of the marginal probabilities $Q_{k,n}(x)$ at the *i*-th iteration. Analogously, $P_k^{(i)}$ denotes the vector of the extrinsic information at iteration *i*. The SISO decoder performs the calculation of the extrinsic information. If the user code is a trellis terminated convolutional code, the SISO decoding can be implemented by the forward-backward BCJR algorithm. The complexity of the BP algorithm is approximately $O(a^K + a^k)$ operations per bit per iteration, where k is the constraint length of the FEC code and a the transmitted symbols cardinality. Therefore the BP algorithm is less complex than the $O(a^{kK})$ of the optimal receiver, although it still has an exponential complexity in the number of users, that makes it appropriate only for a system with a small number of users.

Performance simulations in [23] and [25] show a threshold type behavior: above the threshold E_b/N_o, the performance rapidly approaches single user performance, below the threshold the channel is practically unusable. Different choices of codes, limited to convolutional codes, do not have any influence on the threshold and Moher conjectures that the threshold is capacity related. Analogous numerical results are observed in [26] for convolutional codes. However, assuming an optimum power allocation, theoretical considerations, supported by numerical results, show that the threshold depends on the user code alone, and can be improved by choosing more powerful codes, i.e. turbo codes. Later on we will analyze in detail the problem of optimum power allocation that plays a key role in the performance of overloaded systems with joint iterative decoders.

The scientific community has devoted many efforts to develop iterative joint decoders with lower complexity than the BP decoder. A wide list of references can be found in [24]. All of them share a common structure. They consists of:

- An Interference Cancellation-Multiuser Detection (IC-MUD) able to provide soft information to the following bank of SISO decoders. It is fed with the observed vector signal $\mathbf{Y}^n$ and the extrinsic information $P_k^{(i)}(k = 1, \mathbf{K}, K)$ (some iterative decoders use directly the APP in the feedback instead of the EXT information).
- A bank of *K* SISO decoders able to provide extrinsic information for the feedback and APP for the final detection.

As pointed out in [24], the use of the APP probability in the feedback for the interference cancellation at the multiuser detector violates the basic principles of the sum-product

algorithm. As a consequence, the residual SNIR at the IC-MUD output is biased and the bias reduces the useful signal component. Therefore, in the following we consider only iterative decoders that use the EXT information as interference cancelling in the MUD. The IC-MUD in an iterative joint decoder always consists of two stages:

- In the first stage the observed signal is enforced to be zero mean conditionally to the knowledge of the extrinsic information by subtracting the conditional mean

$$\hat{x}_{k,n} = \sum_{x \in A} x \, P_{k,n}(x) \tag{6.27}$$

to the received vector $\mathbf{Y}^n$, at the *n*-th symbol. The $\hat{x}_{k,n}$ are the estimates of the sent coded symbols and must be subtracted to $\mathbf{Y}^n$ weighted with the appropriate channel matrix coefficient h_{ij}. Therefore we compute the vector of the estimated interference $\hat{\mathbf{Y}}_{int}^n$:

$$\hat{\mathbf{Y}}_{int}^n = \begin{bmatrix} \sum_{j \neq 1} h_{1j} \, \hat{x}_{j,n} \\ \sum_{j \neq 2} h_{2j} \, \hat{x}_{j,n} \\ \vdots \\ \sum_{j \neq K} h_{Kj} \, \hat{x}_{j,n} \end{bmatrix} \tag{6.28}$$

- The zero mean observation $\mathbf{Y}^n - \hat{\mathbf{Y}}_{int}^n$ is then filtered with a $K \times K$ filter matrix $\mathbf{C}_n$ defined according to some optimality criterion. We can use here any multiuser detector able to provide soft information. By selecting different filter matrixes, we obtain different complexity and performance algorithms.

6.4.2. CMMSE-IC and UMMSE-IC

The so called *Conditional* MMSE-Interference Cancellation (CMMSE-IC) algorithm can be obtained as follows:

- Calculate the variance $x_{k,n}$ of the previous estimated sent symbols $\hat{x}_{k,n}$ by:

$$x_{k,n} = \sum_{x \in A} | x - \hat{x}_{k,n} |^2 P_{k,n}(x) \tag{6.29}$$

- Calculate the following $K \times K$ covariance matrix for each channel *k*, each symbol *n* and each iteration:

$$\tilde{\boldsymbol{\Sigma}}_{k,n} = \sum_{j \neq k} \mathbf{H}^j (\mathbf{H}^j)^H \frac{x_{j,n}}{s^2} \tag{6.30}$$

where $\mathbf{H}^j$ is the *j*-th column of the channel matrix $\mathbf{H}$ and s^2 is the noise variance (to be estimated at the receiver).

- It can be demonstrated that the filter vector $\mathbf{c}_{k,n}$ such that its output $z_{k,n} = \mathbf{c}_{k,n}^T (\mathbf{Y}^n - \hat{\mathbf{Y}}_{int}^n)$ is the MMSE for the user *k* conditioned on the knowledge of $P_{j,n}(x)$ is given by:

$$\mathbf{c}_{k,n} = \frac{\tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \mathbf{H}^k}{b_{k,n}} \tag{6.31}$$

where the constant $b_{k,n} = \frac{(\mathbf{H}^k)^H \tilde{\boldsymbol{\Sigma}}_{k,n}^{-1} \mathbf{H}^k}{s^2}$ is introduced for meeting the unbiasedness constraint $\mathbf{c}_{k,n}^H \mathbf{H}^k = 1$.

For the sake of clarity, the IC-MUD block in Figure 6.5 specialized for the CMMSE-IC algorithm is reported in Figure 6.6:

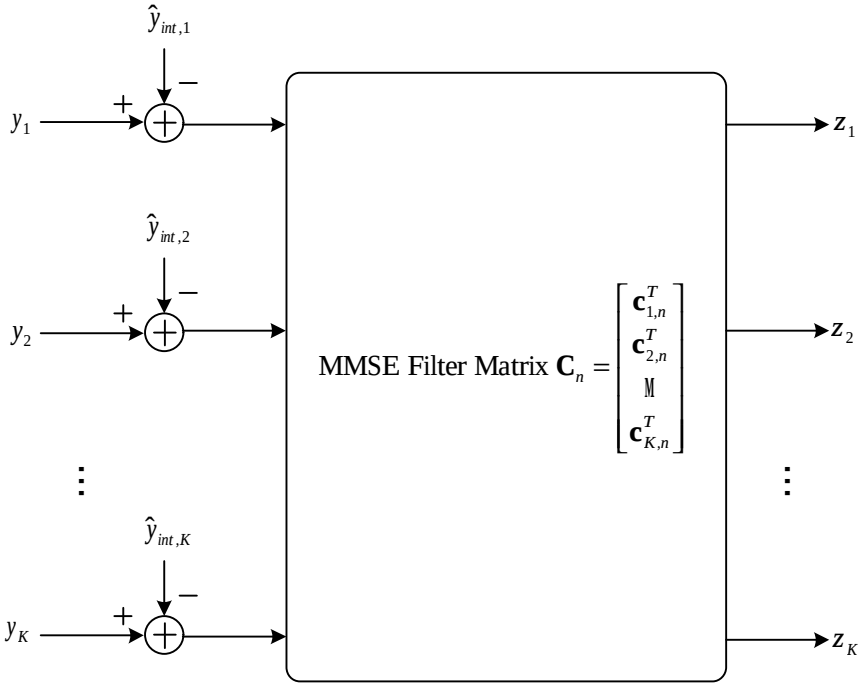


Figure 6.6: MMSE-IC interference cancellation principle.

The CMMSE-IC detector requires the computation of the K filters (6.31) for each user, each symbol and each decoder iteration. A complexity reduction can be obtained by the *Unconditional* MMSE-IC (UMMSE-IC), introduced in [27]. In this case the variance $x_{k,n}$ is not conditioned on $P_{j,n}(x)$ and in practical implementations its conditional mean can be replaced by the empirical mean $x_k = \frac{1}{N}\sum_{n=1}^{N} x_{k,n}$. In this case the inversion of the interference matrix is required only once per user per iteration ($\mathbf{C}_n = \mathbf{C}$ for any n). To take the computational advantages of the UMMSE-IC it is implicitly required constant fading during the transmission of a codeword.

Therefore the two MMSE-IC variants have very different calculation complexity: CMMSE-IC requires the computation of a matrix inverse per symbol per iteration (with complexity per information bit $O(N K^2)$), while the UMMSE-IC requires a matrix inverse per iteration (with complexity per information bit $O(K^2)$). This is clearly an advantage as N grows, with an acceptable performance loss of the UMMSE-IC with respect to the CMMSE-IC case.

6.4.3. MMSE SUMF-IC

A great simplification is obtained by replacing the MMSE filter matrix **C** with the identity matrix. The resulting SUMF-Interference Cancellation (SUMF-IC) algorithm is quite easier than the CMMSE-IC or UMMSE-IC approaches, but it has a considerable loss, as it will be clear next. The SUMF-IC can be seen as an iterative soft-PIC algorithm and it can also be associated with a preliminary MMSE filtering as shown in Figure 6.7:

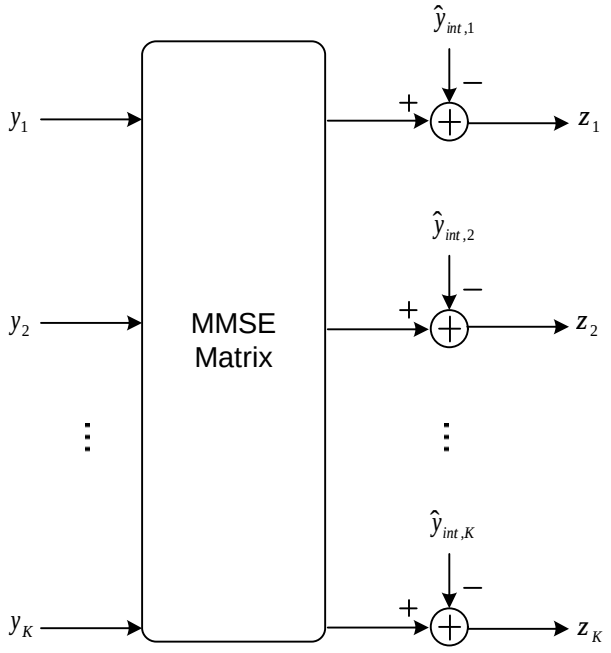


Figure 6.7: MMSE SUMF-IC principle.

6.4.4. MMSE Soft-SIC

The SUMF scheme can also be changed in a serial fashion, by iterating the SIC algorithm already presented in paragraph 6.3 with a soft interference cancellation instead of the coarse hard-SIC. The resulting circuit, reported in Figure 6.8, has a slightly better performance than the MMSE SUMF-IC, because the errors in the interference estimation propagate in a different way. Anyway this algorithm, that has the same complexity as the MMSE SUMF-IC, has a more difficult operations scheduling and for this reason it is not usually implemented.

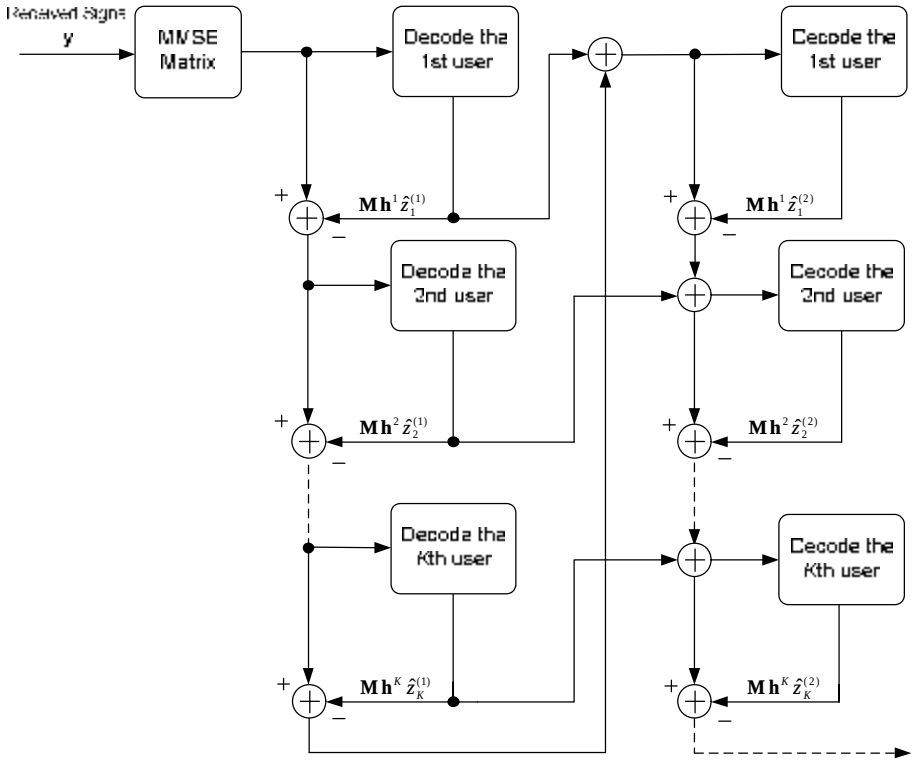


Figure 6.8: MMSE soft-SIC principle.

6.4.5. Performance Comparison

Figure 6.9 compares the performance, in terms of spectral efficiency, of the presented detectors and is taken from [26]. Let us notice that the effect of spatial diversity is not considered in this analysis because the previous works on MUD are basically focused on CDMA systems instead of spatial diversity channels scenarios. Figure 6.9 shows that the UMMSE-IC yields spectral efficiency very close to the BP decoder with much smaller complexity compared to both BP and CMMSE-IC. This makes the UMMSE-IC decoder a good candidate for high-performance low-complexity iterative MUD.

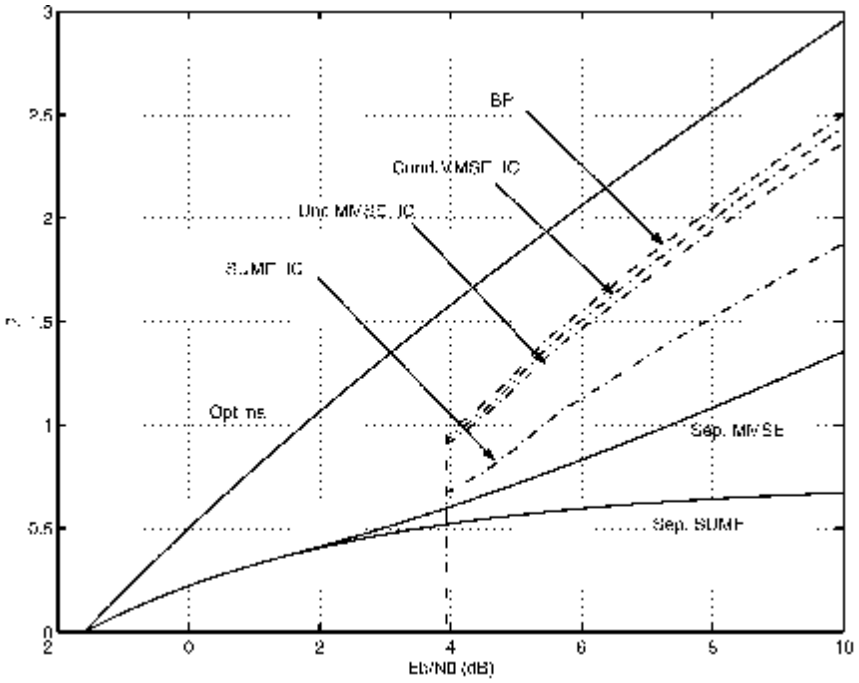


Figure 6.9: Spectral efficiency versus $(E_b/N_o)_{system}$ at $BER \le 10^{-5}$ for convolutionally coded CDMA with rate $1/3$, optimized received SNR distribution and different iterative decoding algorithms. Curves for joint optimal decoding and usual separated SUMF and MMSE detection are reported for the sake of comparison.

Reminding that for an usual single coded channel with code rate *R*, roll-off *r* and modulation symbols cardinality *M*, the spectral efficiency is:

$$r_{\text{single}} = \frac{R \log_2 M}{1+r} \quad [(\text{bit}/\text{s})/\text{Hz}] \tag{6.32}$$

and introducing the *channel load* $a = K / L$, with K the number of CDMA overlapped randomly spread channels and L the chips per symbol (sufficiently high for having good cross-correlation property among the channels), the spectral efficiency r for the whole CDMA system is then equal to:

$$r = \frac{K R \log_2 M}{L(1+r)} = a\ r_{\text{single}} \quad [(\text{bit}/\text{s})/\text{Hz}] \tag{6.33}$$

Therefore Figure 6.9 shows that, fixed s_{single}, for increasing E_b/N_o more channels can be packed in the system (keeping L constant) when using iterative MUD methodologies, while for the single, separated detection, s remains nearly constant, i.e. other channels cannot be added to the system. In the usual CDMA systems $a \leq 2$, but with iterative MUD this number can be considerably higher. For example referring to Figure 6.9, with E_b/N_o = 10 dB, the separated SUMF has a spectral efficiency of 0.7 (bit/s)/Hz, while the UMMSE-IC has 2.3 (bit/s)/Hz, with an increase of the 228%.

Finally, an example of interference cancellation for the BER curve with constant $a = 1$ is reported in Figure 6.10, that leads to the same guidelines as Figure 6.9.

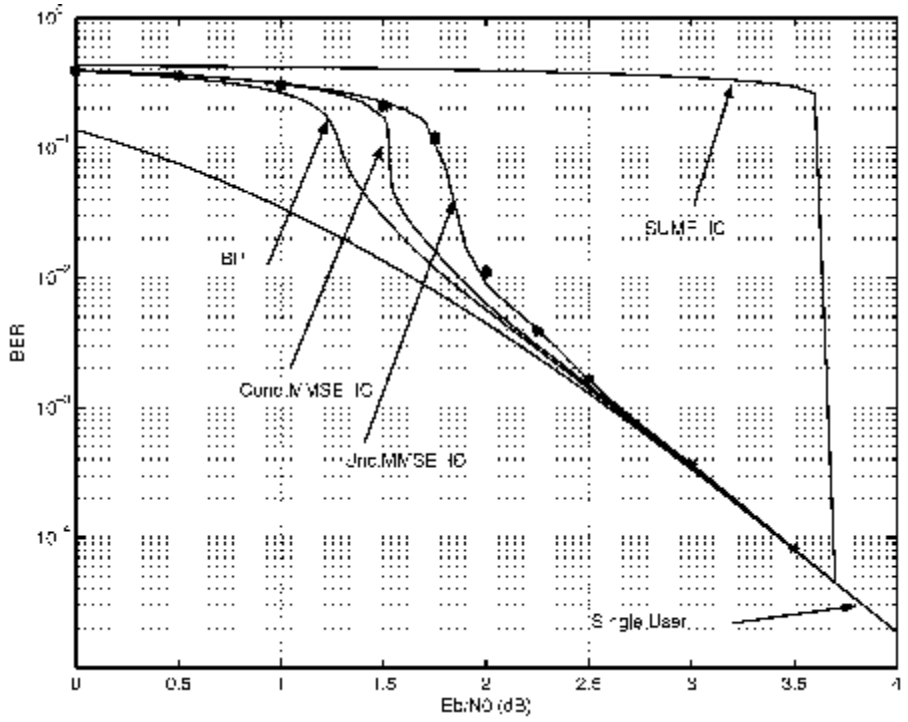


Figure 6.10: BER versus E_b/N_o, constant received SNR, convolutional code with rate 1/2, and different iterative decoding algorithms.

6.4.6. Power Allocation

We must now consider the issue of power allocation, that plays a central role in joint decoding for overloaded systems. In conventional system the concept of power control is

widely spread: it is a mechanism that tends to maintain a given target SNR at the receiver for a given channel, irrespectively of what happens in the other interfering links. In overloaded systems, the aggregate throughput can be maximized, subject to some per link quality of service constraints, by choosing an optimal power allocation strategy that in general implies different SNR levels for different users at the receiver input. The optimal power allocation has been studied in [26] for CDMA systems with random spreading sequences and without spatial diversity. It has been shown that:

- For large $(E_b/N_o)_{system}$, the iteratively decoded systems with optimized SNR distribution are not interference limited, in the sense that their spectral efficiency increases with $(E_b/N_o)_{system}$. For the BP and the CMMSE-IC or UMMSE-IC decoders the slope of spectral efficiency at large $(E_b/N_o)_{system}$ is close to optimum.
- CDMA systems with equal received SNR for all users are basically interference limited and iterative joint decoding provides a significant gain with respect to the conventional separate multiuser detection and single user decoding only for small $(E_b/N_o)_{system}$.

Figure 6.11 shows the improvements in terms of spectral efficiency that can be obtained by optimum power allocation (note that the optimal SNR distribution curves are the same as Figure 6.9). Therefore, for iterative joint decoded systems a mechanism for optimal power allocation has to be implemented. For system without spatial diversity it requires the solution of a linear optimization problem at the gateway to determine the optimal power distribution of the received signals. The optimization problem is solved by approximating the power pdf with a probability mass function (pmf). Numerical results in [26] show that a very low number of the power levels is needed to achieve optimum performance. No protocol modification is required for implementing the optimum power allocation since the usual signaling mechanisms for power control can be used.

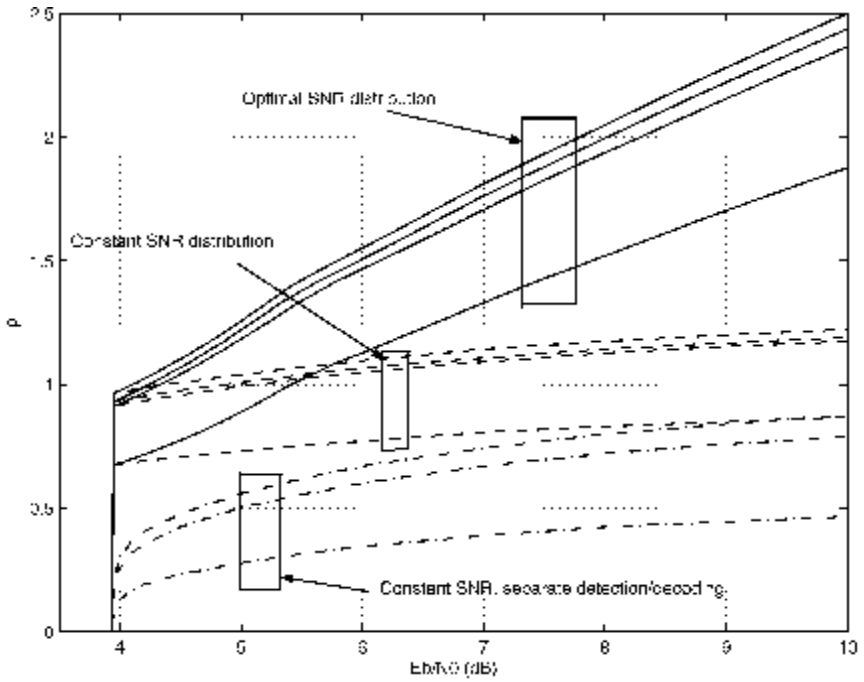


Figure 6.11: Spectral efficiency versus $(E_b/N_o)_{system}$ at $BER \leq 10^{-5}$ for convolutionally coded CDMA with rate $1/3$, constant received SNR distribution and different iterative decoding algorithms. Curves for joint optimal decoding and usual separated SUMF and MMSE detection are reported for the sake of comparison.

These results do not have to surprise, since we already noted that the iterative MUD algorithms work well with different power for the channels, so that the strongest one can be perfectly decoded and subtracted to the other channels. Nevertheless, in our scenario of spatial diversity with independent ACM modulation for each channel, we have a natural not constant SNR distribution for the *K* channels and therefore we can expect an efficiency close to the optimal SNR distribution case without implementing a particular SNR distribution mechanism.

6.5. CONCLUSIONS

In this Chapter, an exhaustive overview of several iterative and non-iterative MUD techniques has been presented. The performance can be summarized in Table 6.1, where *k* is the code constraint length, *K* the number of users and *N* the symbols in the burst. A BPSK modulation is supposed (i.e. constellation cardinality equal to 2).

Iterative MUD technique	Complexity	Interference mitigation
Optimum decoder	$O(2^K + 2^k)$	Very high
Belief propagation	$O(2^{kK})$	high
CMMSE-IC	$O(N K^2)$	high
UMMSE-IC	$O(K^2)$	high
SUMF-IC	$O(K)$	medium

Table 6.1: Iterative MUD techniques comparison.

For the AMPIST project, the best compromise between complexity and performance has led to the SUMF-IC algorithm selection, with a spatial MMSE pre-processing. The simulation results for the MMSE SUMF-IC are reported in the next Chapter 7.

7. MULTIUSER DECODING RESULTS

The simulated scenarios employ a MMSE filtering followed by an iterative MUD (also called turbo-MUD) demodulation. The general scheme is shown in Figure 7.1, where y_i are the input channels from the *K* antennas: the EXT (or APP) information is re-modulated and fed back at every iteration in order to clean up the MMSE filtered signals. The SUMF-IC algorithm is adopted for the CCI cancellation, i.e. the feedback signals are subtracted with appropriate coefficients on every channel. We will study the MMSE only correction and the MMSE plus SUMF-IC correction.

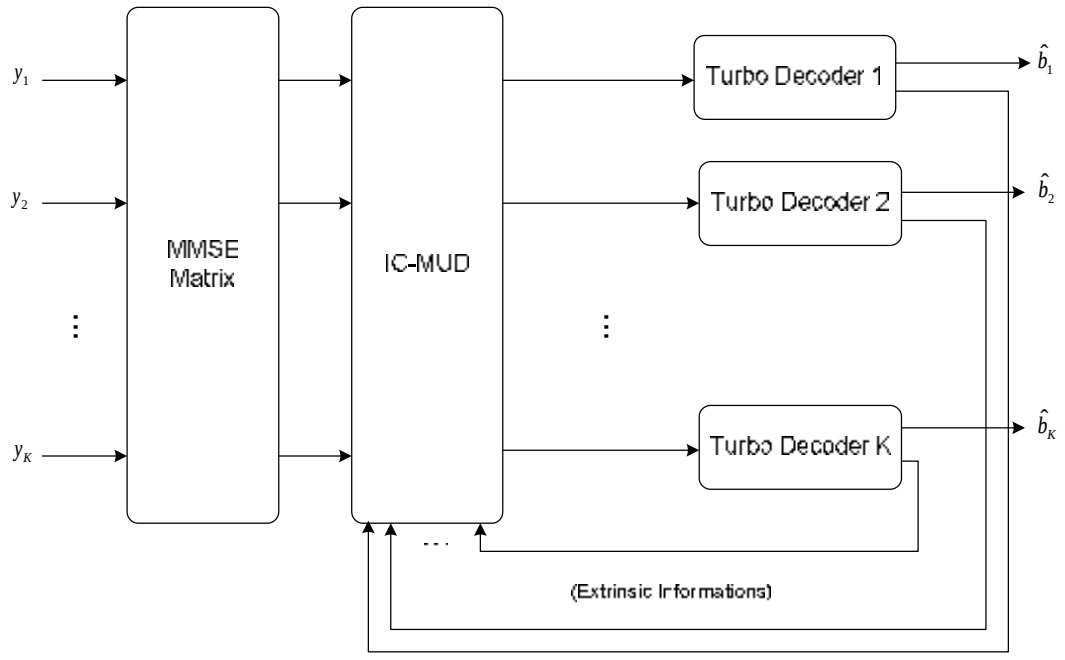


Figure 7.1: Adopted turbo-MUD decoder.

7.1. SIMULATED SCENARIOS

Multiuser scenarios were simulated for four users, two different channel matrixes and two different coding rates. They are organized as follows:

- *Scenario 1*
  Modulation: QPSK 1/2
  Channel matrix: $|\mathbf{H}| = \begin{bmatrix} 3.28 & 0.93 & 1.9 & 1.78 \\ 1.07 & 3.29 & 0.38 & 2.06 \\ 2.19 & 0.26 & 4.11 & 0.73 \\ 3.11 & 2.99 & 0.88 & 4.07 \end{bmatrix}$

  Therefore the four channels have the following different C/Is:

  Channel 1 C/I = 1.47 dB
  Channel 2 C/I = 2.9 dB
  Channel 3 C/I = 4.93 dB
  Channel 4 C/I = -0.69 dB

  The channel matrix is relative to the users spatial position shown in Figure 7.2 and has been carefully selected in order to represent the general case, with very different users condition. As usual, the QPSK 1/2 is chosen because it is the most hostile AMPIST operative mode and the other modes are expected to exhibit a better behaviour. Only for this scenario the usual EXT information feedback cancellation will be compared to the APP feedback cancellation.

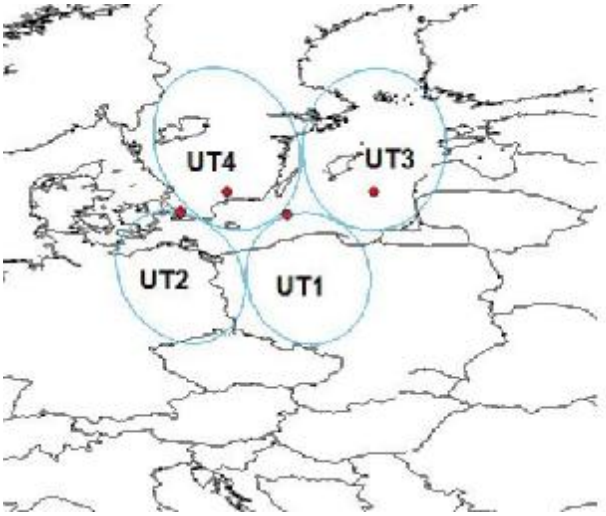


Figure 7.2: Scenario 1 users position.

- *Scenario 2*

  Modulation: QPSK 1/2

  Symmetrical channel matrix: $|\mathbf{H}| = \begin{bmatrix} 1 & a & a & a \\ a & 1 & a & a \\ a & a & 1 & a \\ a & a & a & 1 \end{bmatrix}$

  A symmetrical channel matrix leads to the same C/I for all the four channels and is not a practical case, but it is useful for evaluating the turbo-MUD behaviour. The adopted C/I ratios were: 5 dB, 3 dB and 0 dB, so as to represent a wide C/I range.

- *Scenario 3*

  Modulation: QPSK 2/3

  Symmetrical channel matrix: $|\mathbf{H}| = \begin{bmatrix} 1 & a & a & a \\ a & 1 & a & a \\ a & a & 1 & a \\ a & a & a & 1 \end{bmatrix}$

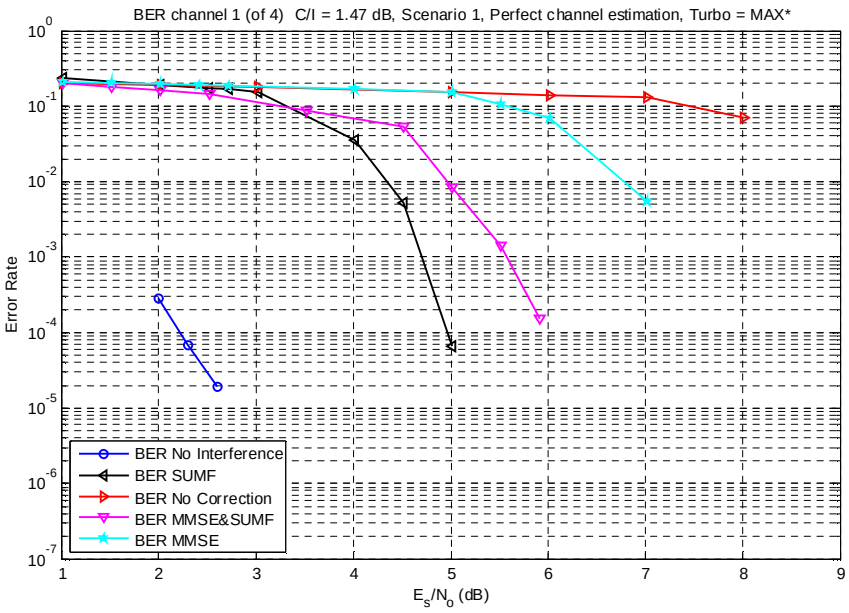
In this last scenario, the code rate 2/3 is used in order to analyze the differences with respect to the scenario 2 code rate 1/2. The adopted C/I ratios were: 5 dB, 3 dB and 0 dB.

In all the three scenarios, the EXT information is used for the feedback cancellation. The scenarios have been tested using both the MAX* and MAX algorithms (already discussed in paragraph 5.2.3) for the turbo decoding. The results are presented in the following paragraphs.

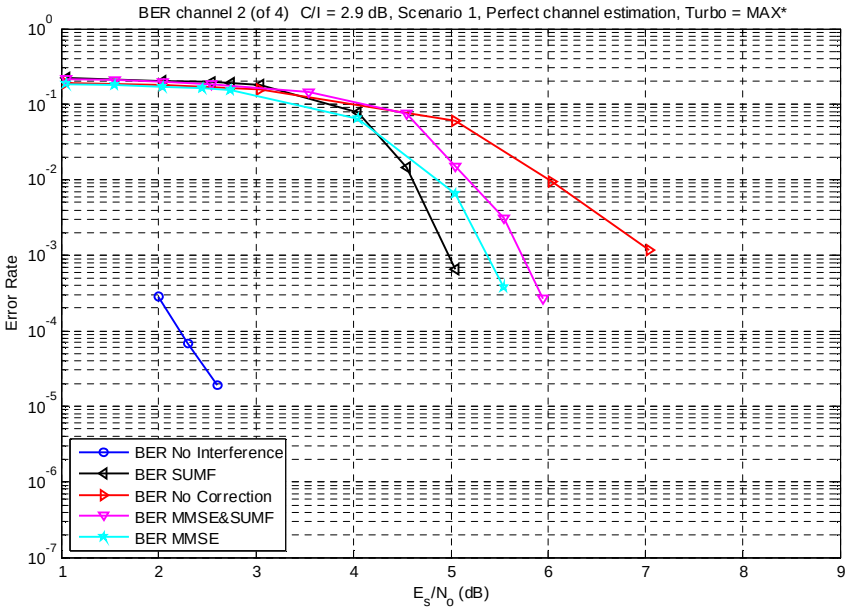
7.2. MAX* TURBO DECODING

7.2.1. Scenario 1

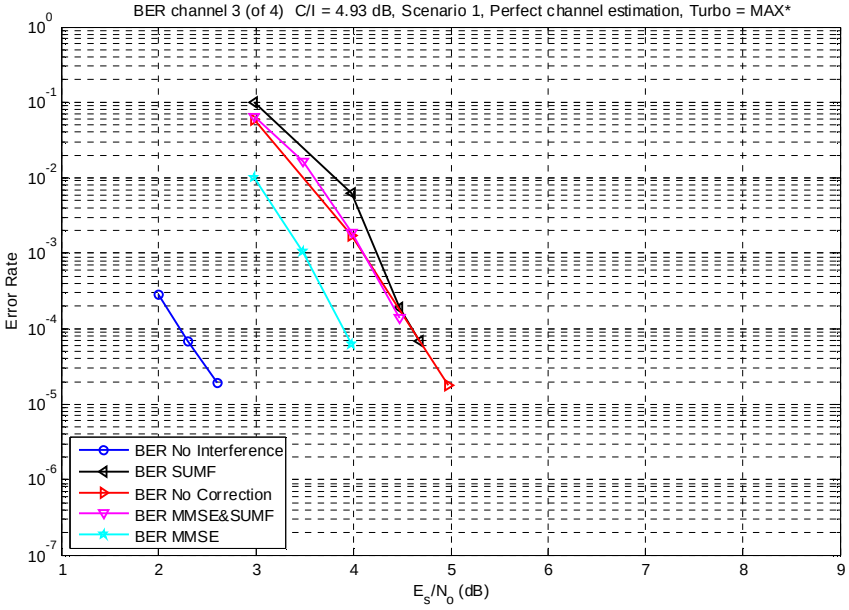
The simulation results for the scenario 1, with perfect channel estimation (i.e. the channel matrix is supposed known at the receiver side), are summarized in the next Figure 7.3:



(*a*)


BER channel 2 (of 4) C/I = 2.9 dB, Scenario 1, Perfect channel estimation, Turbo = MAX*
Error Rate
E_s/N_o (dB)
BER No Interference
BER SUMF
BER No Correction
BER MMSE&SUMF
BER MMSE

(*b*)


BER channel 3 (of 4) C/I = 4.93 dB, Scenario 1, Perfect channel estimation, Turbo = MAX*
Error Rate
E_s/N_o (dB)
BER No Interference
BER SUMF
BER No Correction
BER MMSE&SUMF
BER MMSE

(*c*)

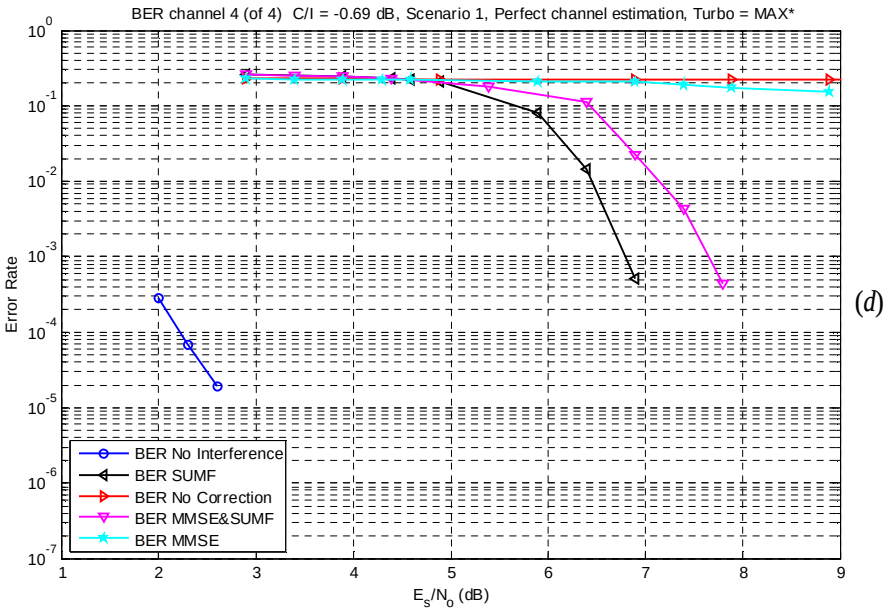
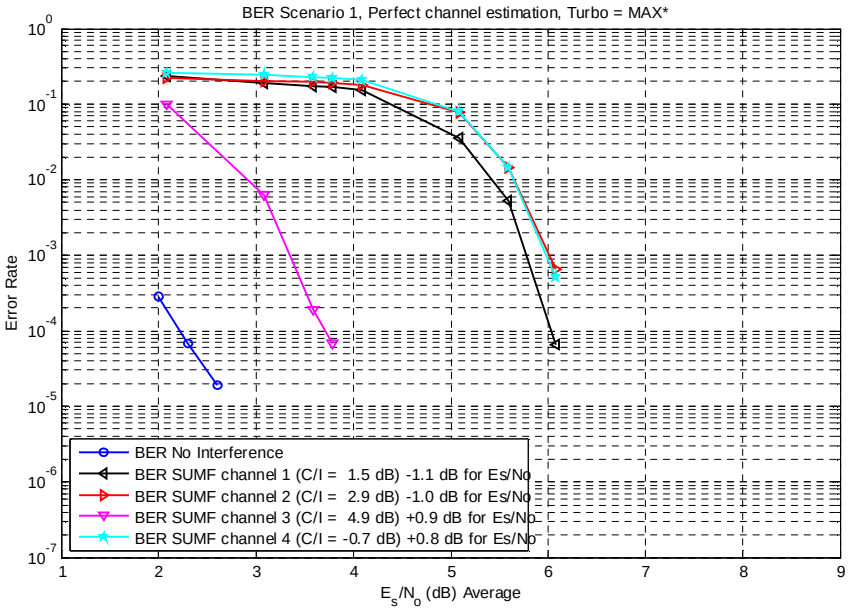
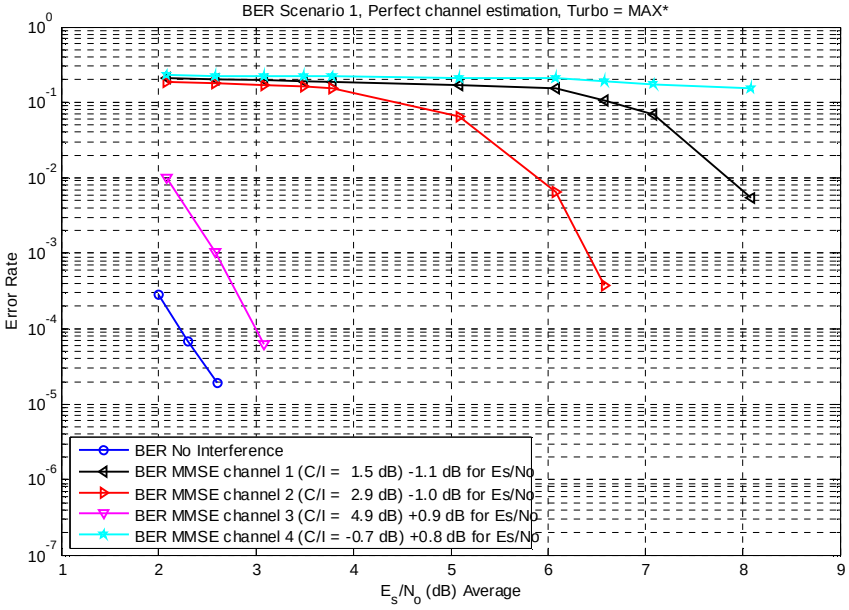


Figure 7.3: Scenario 1, perfect channel estimation. (*a*) C/I = 1.47 dB. (*b*) C/I = 2.9 dB. (*c*) C/I = 4.93 dB. (*d*) C/I = -0.69 dB.

The simulations show a clear interference mitigation and this improvement is stronger as the interference grows. Anyway the loss with respect to the no interference case can be significant and depends on the particular channel matrix. Note that the MMSE SUMF-IC usually performs worse than the SUMF-IC only, but this can be explained by observing that the enhancement of the noise floor typical of the MMSE transformation (although the SNIR is maximized) can not be deleted by the iterative SUMF-IC cancellation, thus leading to the worse behaviour with respect to the SUMF-IC only correction. The same simulations are reported as a function of the average E_s/N_o in Figure 7.4, so that the contemporaneous behaviours of the four channels can be observed.


BER Scenario 1, Perfect channel estimation, Turbo = MAX*
Error Rate
Es/No (dB) Average
BER No Interference
BER SUMF channel 1 (C/I = 1.5 dB) -1.1 dB for Es/No
BER SUMF channel 2 (C/I = 2.9 dB) -1.0 dB for Es/No
BER SUMF channel 3 (C/I = 4.9 dB) +0.9 dB for Es/No
BER SUMF channel 4 (C/I = -0.7 dB) +0.8 dB for Es/No

(*a*)


BER Scenario 1, Perfect channel estimation, Turbo = MAX*
Error Rate
Es/No (dB) Average
BER No Interference
BER MMSE channel 1 (C/I = 1.5 dB) -1.1 dB for Es/No
BER MMSE channel 2 (C/I = 2.9 dB) -1.0 dB for Es/No
BER MMSE channel 3 (C/I = 4.9 dB) +0.9 dB for Es/No
BER MMSE channel 4 (C/I = -0.7 dB) +0.8 dB for Es/No

(*b*)

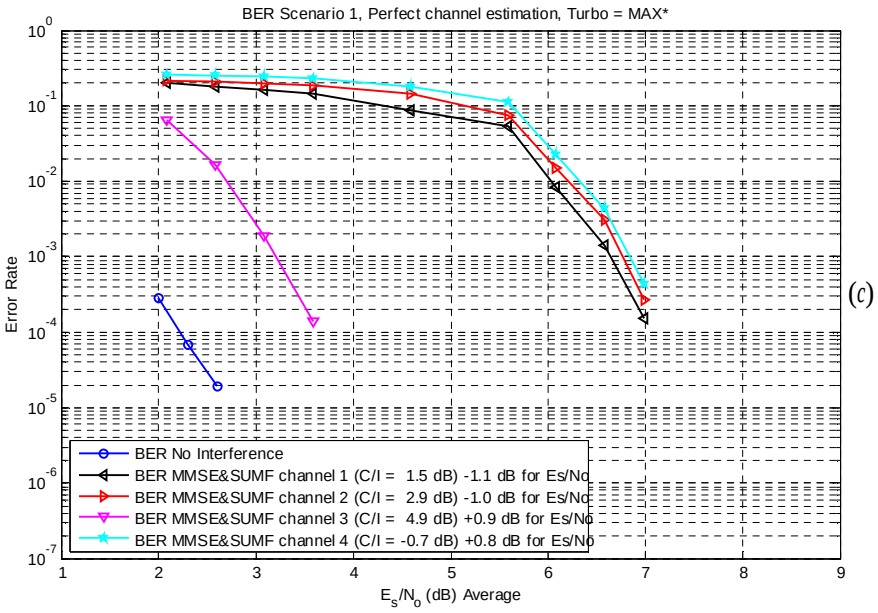


Figure 7.4: Scenario 1, perfect channel estimation, average E_s/N_o. (*a*) SUMF-IC correction. (*b*) MMSE correction. (*c*) MMSE SUMF-IC correction.

Figure 7.4 shows that, a part the less disturbed channel 3, all the other channels converge together when the SUMF-IC algorithm is activated. This is because as soon as a channel is well decoded, also its interference on the other channels is well cancelled and there is an overall improvement.

As already studied in paragraph 6.2, the channel matrix must be estimated by means of algorithms that rely on the known UW sequence of the traffic bursts. In particular, the value of 40 for the UW length, coherently with the single user decoding guidelines summarized in Table 4.2, can be considered sufficient, leading to a less than 0.5 dB loss, as shown in Figure 7.5. The loss is similar for all the four channels, therefore only channel 1 is reported for the sake of brevity. Only the SUMF-IC case is analyzed, since it is the best correction scheme.

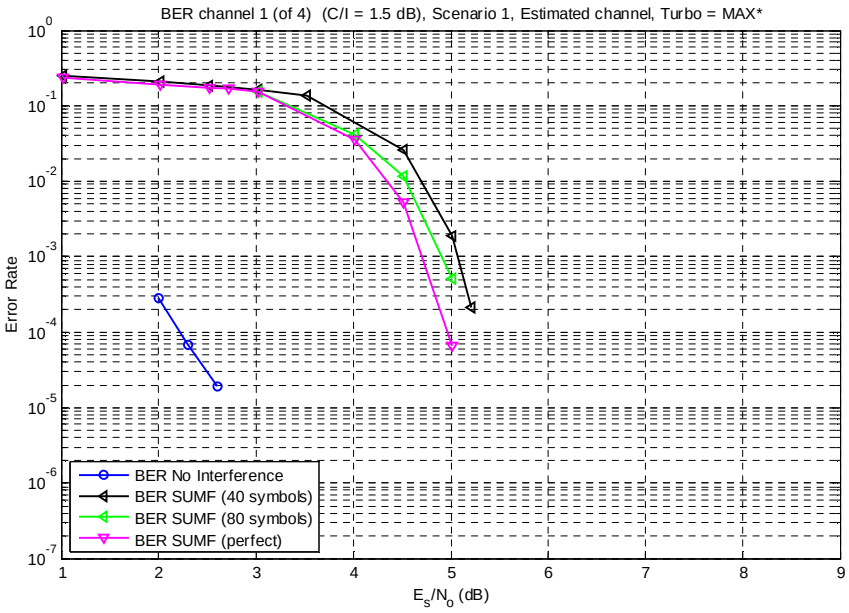


Figure 7.5: Scenario 1, estimated channel, SUMF-IC correction on channel 1.

Finally the APP information has been adopted instead of the EXT for the feedback cancellation, but no relevant differences have been observed, as shown in Figure 7.6 (see Figure 7.4 (*a*) for reference). Also the decoding improvement with the iteration number is similar between EXT and APP and shows that no further practical cancellation can be achieved after 15÷20 iterations. A typical curve is reported in Figure 7.7.

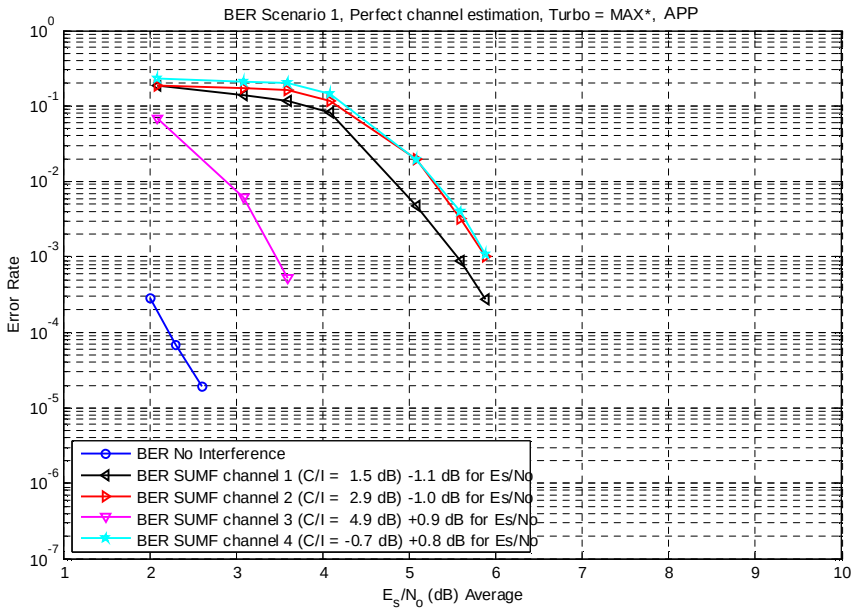


Figure 7.6: Scenario 1, perfect channel estimation, APP feedback cancellation, average E_s/N_o, SUMF-IC correction.

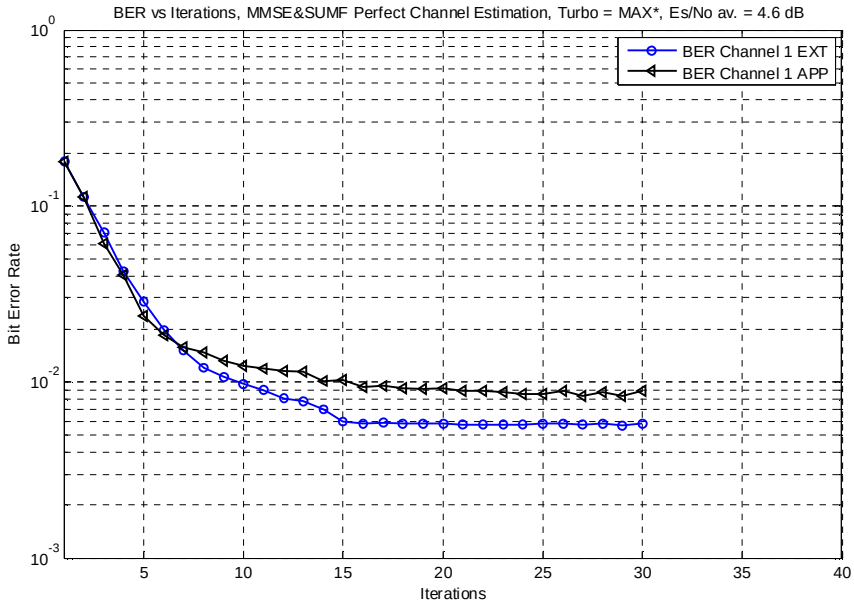
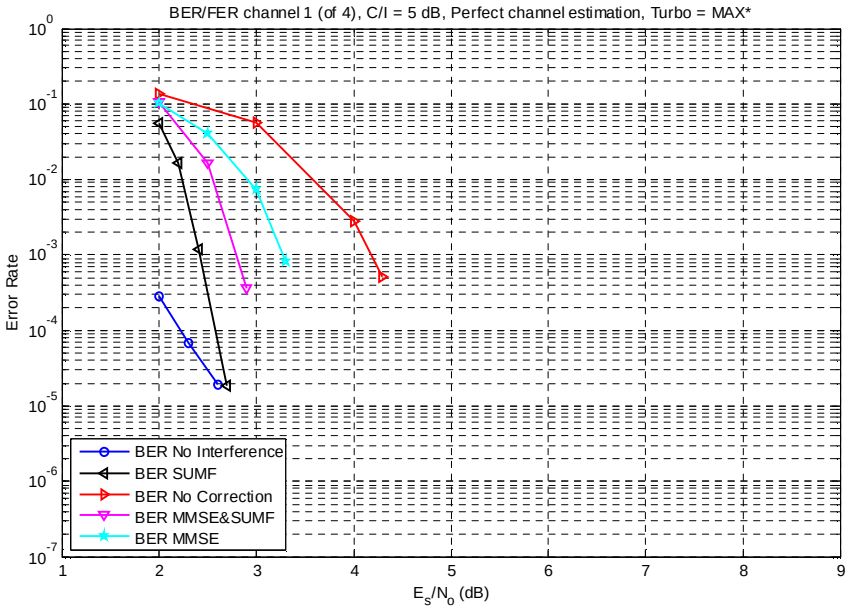


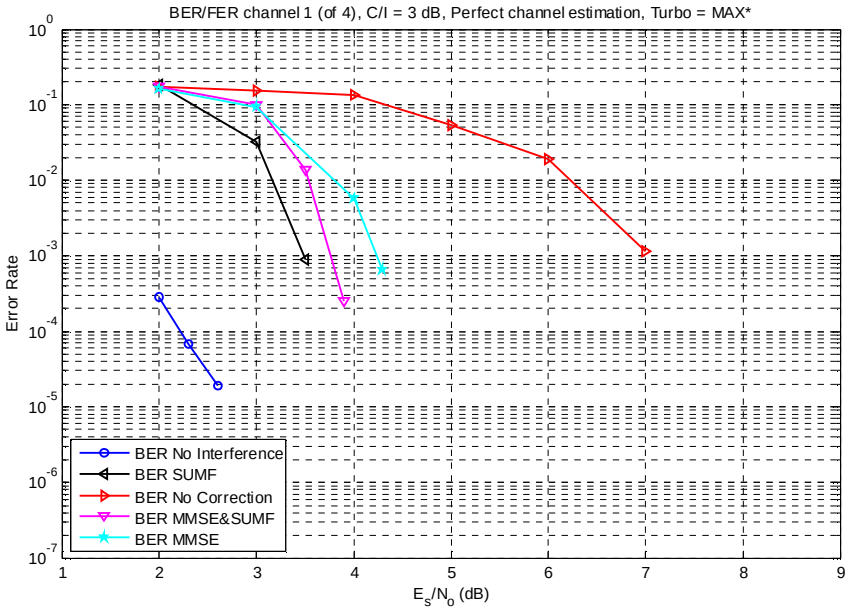
Figure 7.7: Example of BER versus turbo iterations.

7.2.2. Scenario 2

In this scenario the symmetrical channel matrix is adopted. The results are reported in the next Figure 7.8:



(*a*)



(*b*)

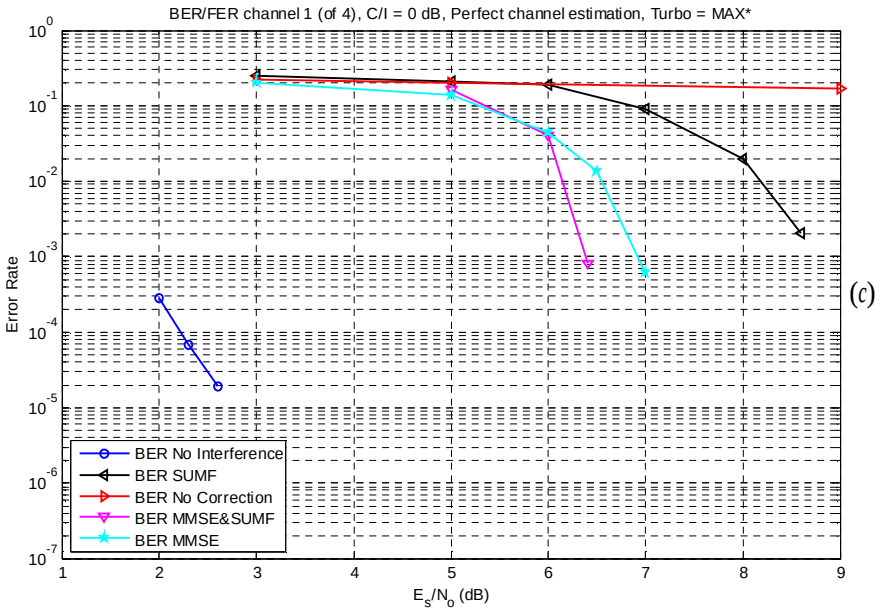
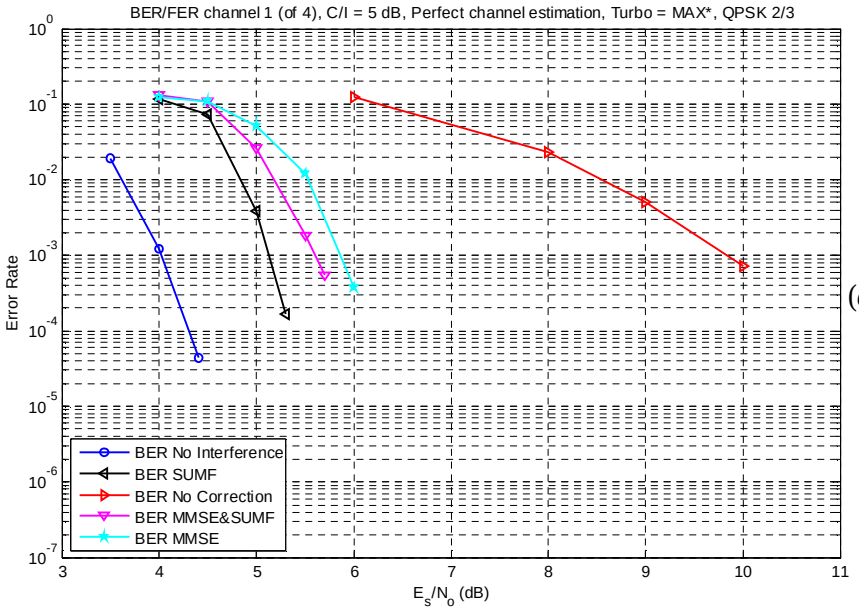


Figure 7.8: Scenario 2, perfect channel estimation. (*a*) C/I = 5 dB. (*b*) C/I = 3 dB. (*c*) C/I = 0 dB.

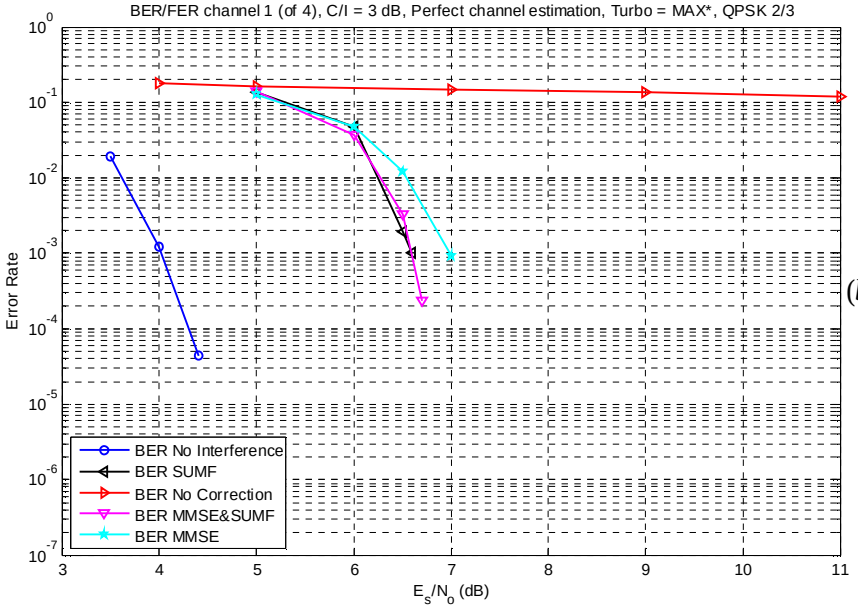
It can be observed that with the same user C/I, the symmetrical matrix leads to a better interference cancellation with respect to the scenario 1. This is because all the channels converge at the same E_s/N_o. Actual estimation of the channel matrix leads to a loss similar to the one observed in the scenario 1, showing the typical degradation less than 0.5 dB for the SUMF-IC case.

7.2.3. Scenario 3

The last scenario is the same as the scenario 2, but with rate 2/3 instead of 1/2. The results are reported in Figure 7.9:


BER/FER channel 1 (of 4), C/I = 5 dB, Perfect channel estimation, Turbo = MAX*, QPSK 2/3
Error Rate
E_s/N_o (dB)
BER No Interference
BER SUMF
BER No Correction
BER MMSE&SUMF
BER MMSE

(*a*)


BER/FER channel 1 (of 4), C/I = 3 dB, Perfect channel estimation, Turbo = MAX*, QPSK 2/3
Error Rate
E_s/N_o (dB)
BER No Interference
BER SUMF
BER No Correction
BER MMSE&SUMF
BER MMSE

(*b*)

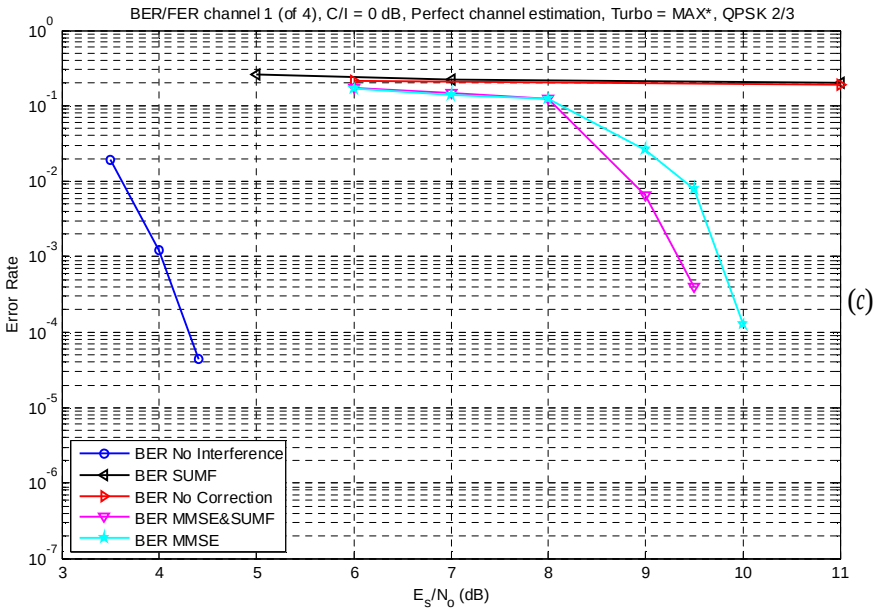
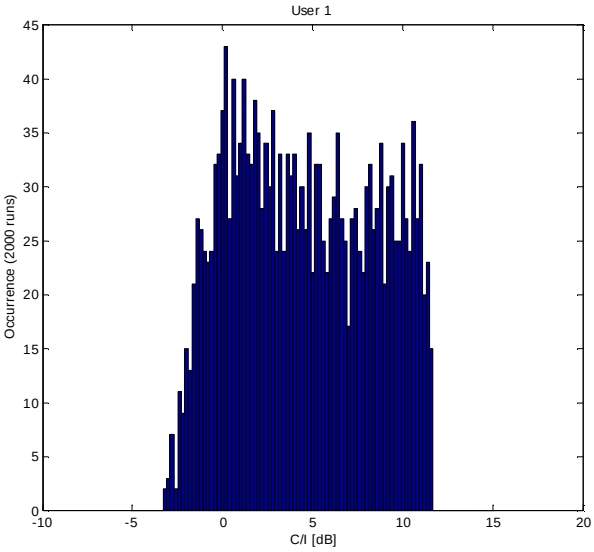


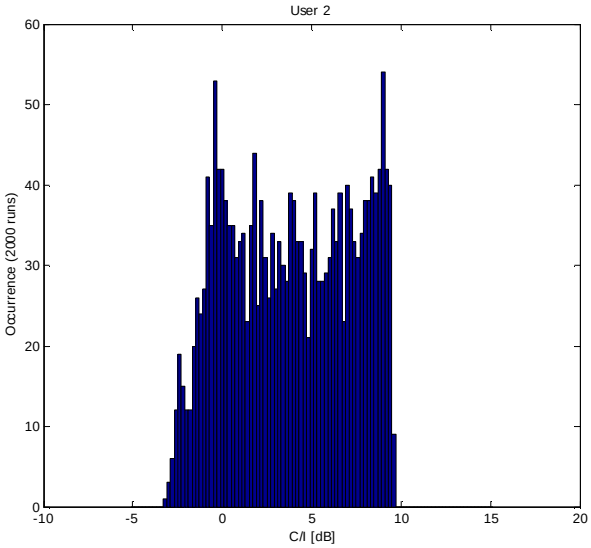
Figure 7.9: Scenario 3, perfect channel estimation. (*a*) C/I = 5 dB. (*b*) C/I = 3 dB. (*c*) C/I = 0 dB.

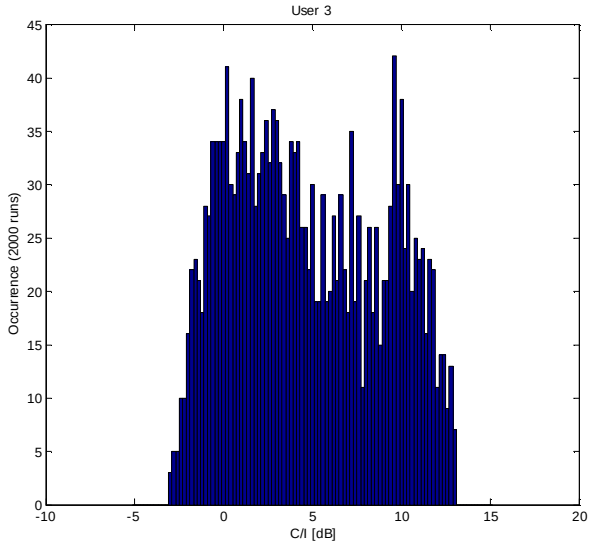
It can be observed that the loss with respect to the no interference case is higher than in the scenario 2, although the interference mitigation is considerably stronger.

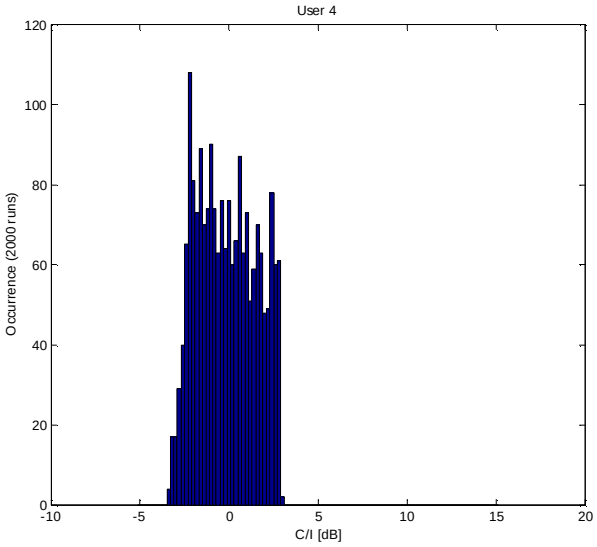
7.2.4. Carrier to Interference Distribution

An extensive simulation of 8 users located in 8 different beams has been performed. The users were randomly located in the beams and 2000 iterations were run, therefore 2000 different channel matrixes were generated. The 2000 different C/Is for the 8 users are shown as histograms in the next Figure 7.10:


User 1
Occurrence (2000 runs)
C/I [dB]


User 2
Occurrence (2000 runs)
C/I [dB]


User 3
Occurrence (2000 runs)
C/I [dB]


User 4
Occurrence (2000 runs)
C/I [dB]

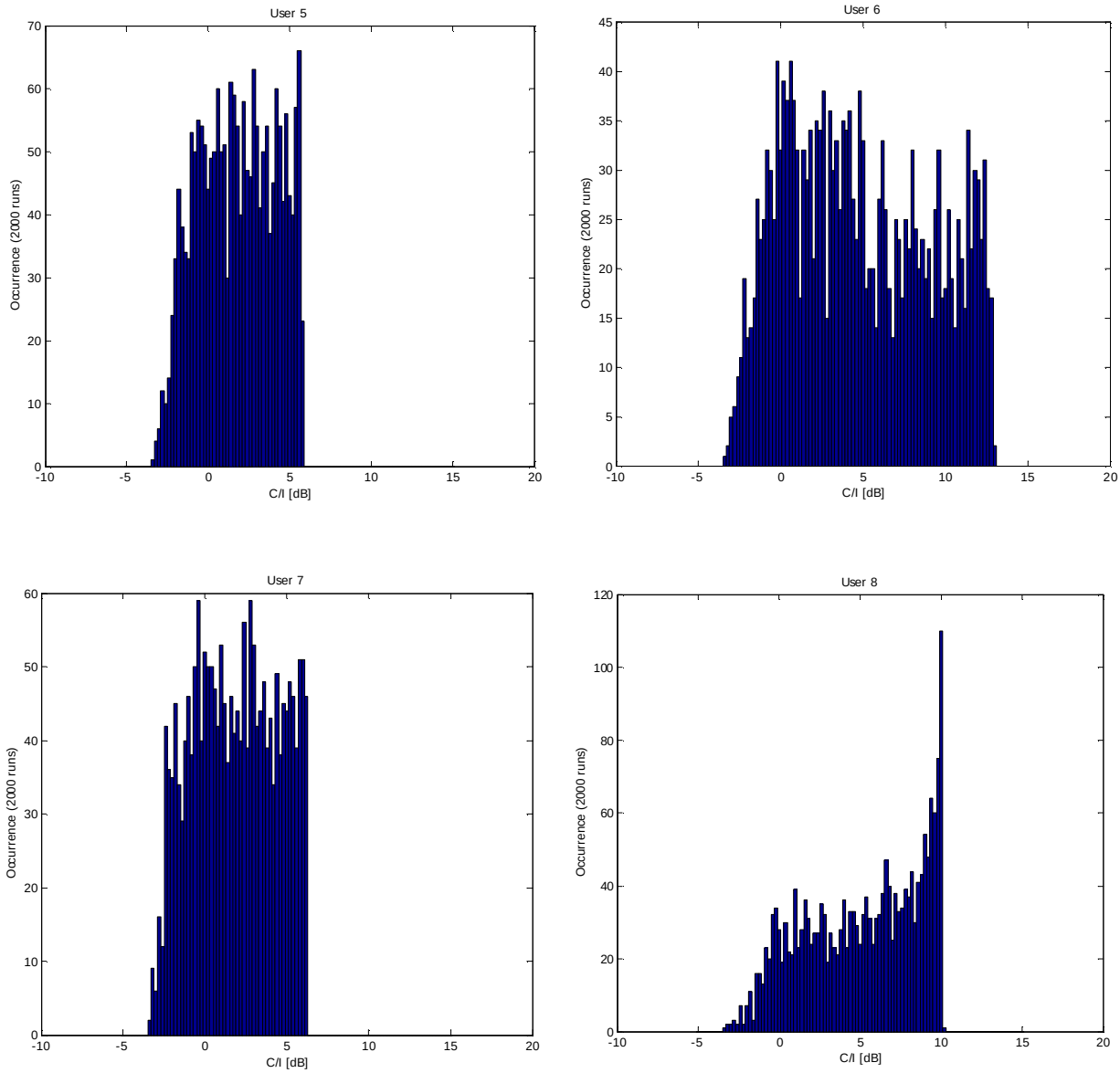


Figure 7.10: C/I distributions for 8 random users.

Note that the C/I distributions are different since they depend on the beam position in the subset of beams jointly decoded by the same gateway (e.g. the border beams show a better C/I because they have less interference, see Figure 3.2). Anyway the worst C/I for any user is about -3 dB, therefore the scenario 1 already simulated has been changed so as to obtain C/I = -2.5 dB for the channel 4 (while in the previous simulation it was -0.7 dB), keeping

unchanged the other C/Is. The resulting channel 4 performance, with the SUMF-IC correction scheme, is reported in Figure 7.11:

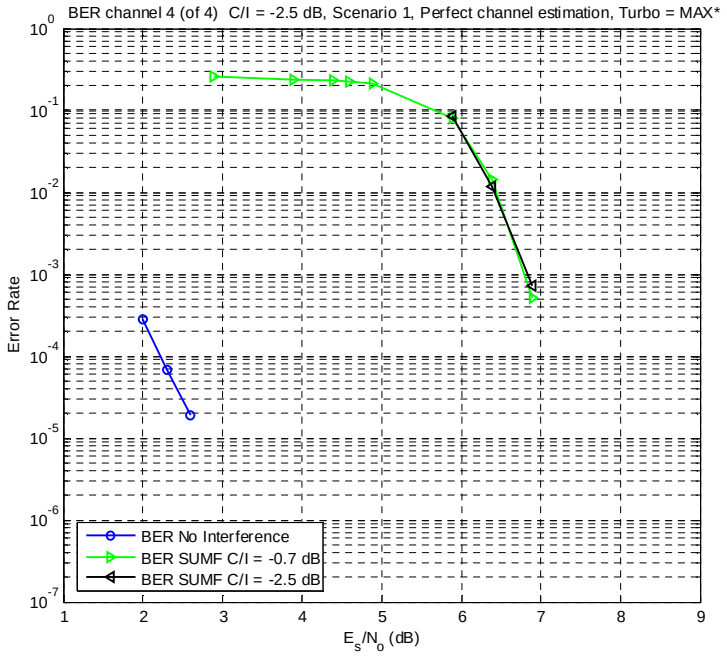


Figure 7.11: Scenario 1, perfect channel estimation, C/I = -2.5 dB (worst expected C/I).

It can be noted that with C/I = -2.5 dB, the SUMF-IC algorithm mitigates the interference like the C/I = -0.7 dB case. This happens because as soon as the other channels are well demodulated, their interference on channel 4 is deleted exactly in the same way for the two possible C/Is.

7.3. MAX TURBO DECODING

In this second set of simulations, the MAX* algorithm has been replaced by the approximated MAX algorithm in order to speed up the simulations and reach lower BER values. In fact, using the MAX* in the turbo decoder is by far more time consuming than adopting the simpler MAX. It can be showed that the loss associated to the usage of the MAX instead of the MAX*, in the single user case, is less than 0.2 dB.

The aim of this paragraph is to investigate the system behaviour in the real case, i.e. time/frequency recovery activated, users not synchronous and estimated channel matrix. Only the SUMF-IC case will be analyzed, because the SUMF-IC was elected the best mitigation scheme in the previous paragraph 7.2.

7.3.1. Reference Case for Synchronous Users

In this paragraph, the reference scenario 1 BER curves for synchronous users (i.e. the bursts on the four channels are aligned) adopting the MAX algorithm, with perfect time/frequency recovery and channel estimation, have been computed. These BER curves will be also compared to the MAX* BER curves calculated in paragraph 7.2 in order to validate the MAX algorithm for the multiuser scenario.

The weights associated with the MAX algorithm have been tuned in a standard way and the system performance is reported in Figure 7.12. The comparison with Figure 7.4 (*a*) shows a loss of about 0.2 dB, that is exactly the expected BER degradation for the single user case. The system has been tested at higher E_s/N_o than the previous simulations, in order to verify that no noise floor was introduced. The simulations have confirmed that the noise floor is not visible at $BER = 10^{-5} \div 10^{-6}$. Note that channel 3 is not showed because at these E_s/N_o ratios it is always perfectly decoded.

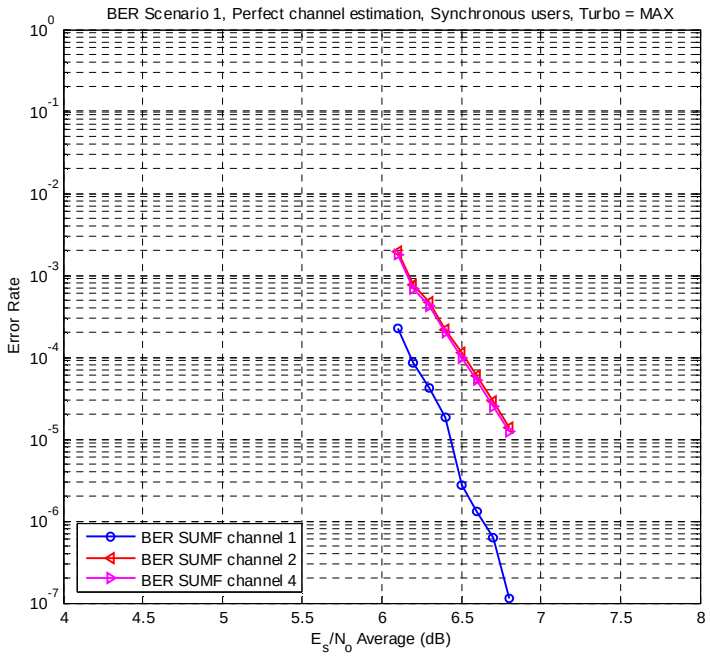


Figure 7.12: Scenario 1, synchronous users, MAX algorithm for turbo, perfect channel estimation, perfect time/frequency recovery, SUMF-IC correction.

7.3.2. Reference Case for Asynchronous Users

In this paragraph, the reference scenario 1 BER curves for asynchronous users (i.e. the bursts on the four channels are not aligned) adopting the MAX algorithm, with perfect time/frequency recovery and channel estimation, have been computed. The bursts can have a maximum shift of a few symbols and the minimum admitted shift is equal to a sample (4 samples/symbol are used). It is expected that in this case the interference suffered by channels whose peaks are not aligned will be lower than in the aligned case. The adopted delay pattern is the following (expressed in samples, 0 is considered the reference sample):

Delay 1 = 7

Delay 2 = 5

Delay 3 = 0

Delay 4 = 11

That is channel 1 and channel 4 are symbol-aligned, therefore they are expected to have worse performance than the other channels.

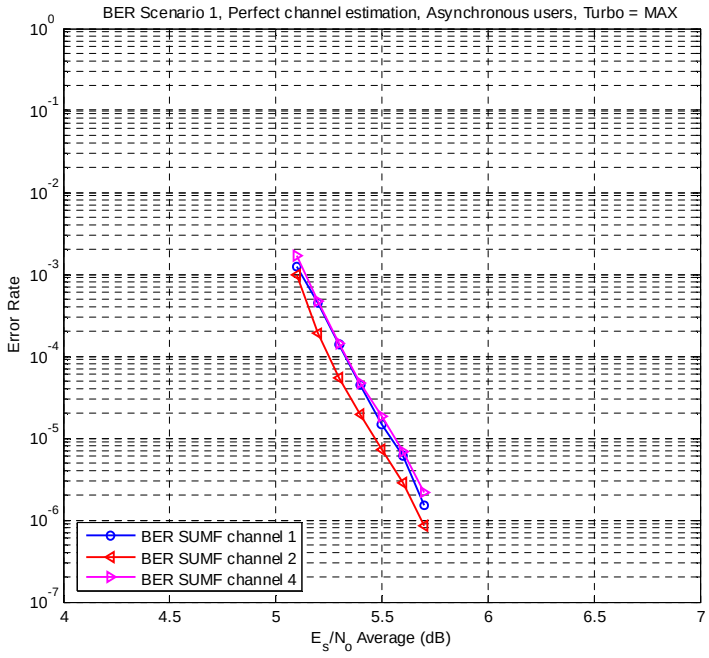


Figure 7.13: Scenario 1, asynchronous users, MAX algorithm for turbo, perfect channel estimation, perfect time/frequency recovery, SUMF-IC correction.

Comparing Figure 7.13 with Figure 7.12, one can note that a considerably lower E_s/N_o is required to reach the same BER values: for asynchronous users the interference peaks are not aligned, with a much better system behaviour than in the synchronous case. In addition, as expected, channel 1 and channel 4 have a stronger interference than channel 2.

7.3.3. Time and Frequency Recovery

In this set of simulations, time and frequency shifts are added to the system. The frequency error (normalized to the symbol rate) applied to each channel is 1.5% (i.e. the AMPIST maximum allowed frequency error) and the following assumptions are made for the bursts (similar to the single user assumptions summarized in Table 4.2):

- 6 guard symbols (3 at the beginning, 3 at the end).

- 20 symbols truncated gold code preamble.
- 20 symbols truncate gold code postamble.
- 40 scrambled pilots.

The following recovery algorithms have been used:

- Time recovery : O&M.
- Frequency recovery: M&M.
- Phase recovery: Sliding window (window length = 504 symbols). Note that the required window length is longer than in the single user case (229 symbols) because of the mutual interference impairment.

The simulation results for the synchronous users case are depicted in Figure 7.14:

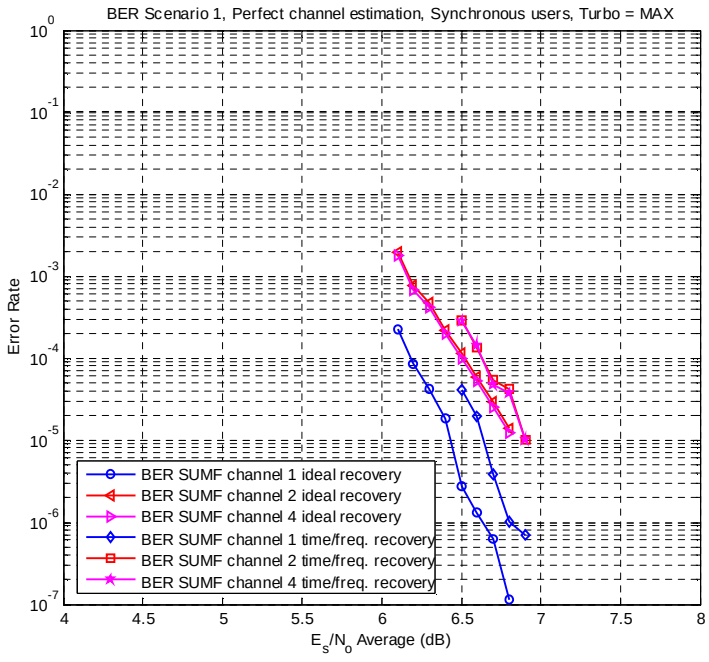


Figure 7.14: Scenario 1, synchronous users, perfect channel estimation, time/frequency recovery, SUMF-IC correction.

The simulation shows a very small loss occurred for the time/frequency recovery (less than 0.2 dB) due to the high operative E_s/N_o (compare the 0.35 dB single user decoding loss in Figure 4.16). Also note that the curve slopes remain the same. Instead simulations made using not aligned bursts show a different behaviour, as reported in Figure 7.15:

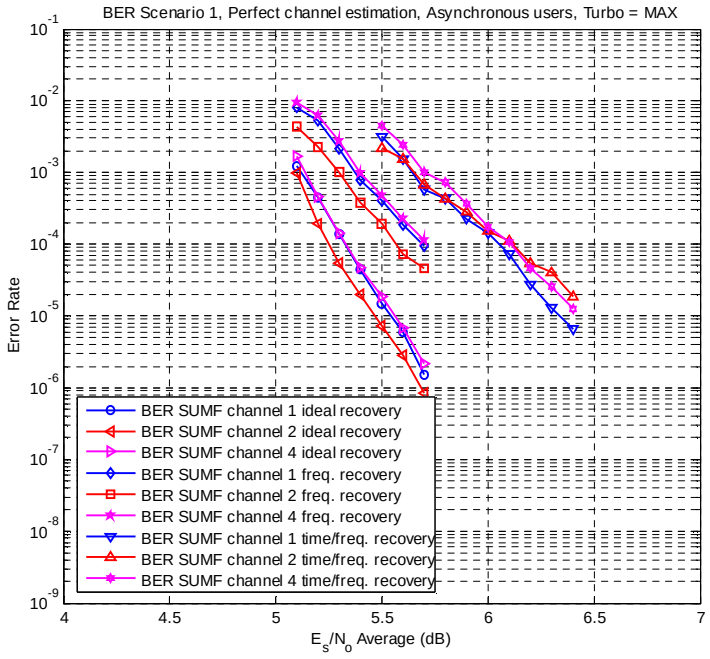


Figure 7.15: Scenario 1, asynchronous users, perfect channel estimation, time/frequency recovery, SUMF-IC correction.

For not aligned bursts, the loss due to the time/frequency recovery is much higher than for aligned bursts, therefore, for a better investigation, the simulation has been separated in two cases: only frequency recovery active (with no time shift) and time/frequency recovery active. It can be observed that the time recovery is more difficult for asynchronous users because the interfering delayed channels deviate the recovered exact timing.

The two contemporaneous recovery algorithms lead to about 0.7 dB degradation for $BER = 10^{-4}$ (and no channel matrix estimation is made yet). It is also to be noted the slope variation of the BER curves. Nevertheless, the E_s/N_o values necessary for reaching the same BER are lower than the ones of the aligned case.

7.3.4. Channel Estimation

Final simulations have been run in order to show the degradation introduced by the channel estimation (always with the burst format of paragraph 7.3.3). Channel estimation has been applied both on synchronous and asynchronous users, and the results are reported in the following Figure 7.16 and Figure 7.17. Note that these curves also represent the synchronous and asynchronous "real cases", i.e. all the recovery algorithms are activated together like in the actual AMPIST demodulator.

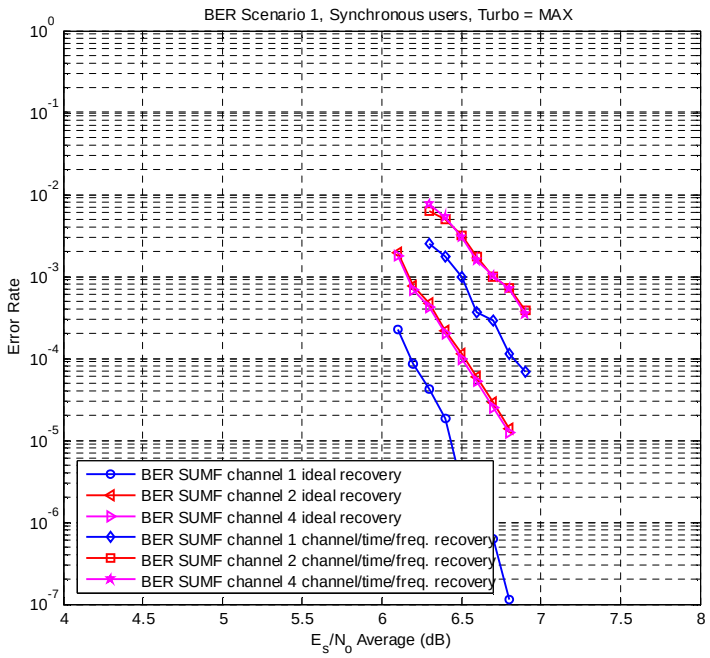


Figure 7.16: Scenario 1, synchronous users, channel estimation, time/frequency recovery, SUMF-IC correction.

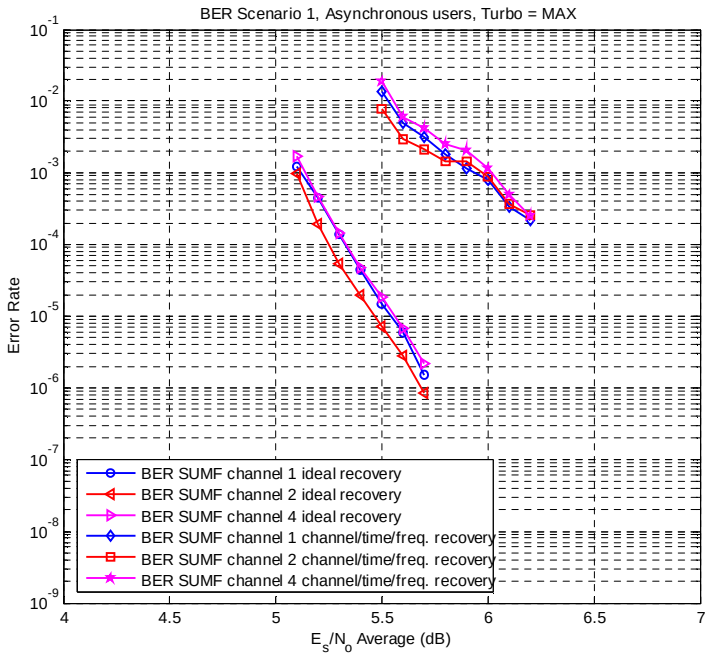


Figure 7.17: Scenario 1, asynchronous users, channel estimation, time/frequency recovery, SUMF-IC correction.

The channel estimation for both aligned and not aligned bursts adds a maximum loss of about 0.5 dB, as expected for the ideal time/frequency recovery scenario (see Figure 7.5). Therefore a total loss of about 0.7 dB is observable for the synchronous multiuser real case while a 1 dB loss at $BER = 10^{-4}$ is reached for the asynchronous multiuser real case.

7.4. CONCLUSIONS

In conclusion, the multiuser scenario simulations have led to the following guidelines:

- The MMSE combined with the SUMF-IC usually does not produce further improvement and therefore the adopted CCI cancellation strategy will be SUMF-IC only.
- The SUMF-IC algorithm shows a clear mitigation of the interference and this improvement is stronger as the interference grows. Anyway the loss with respect to the no interference case can be significant and depends on the particular channel matrix.

- As the code rate grows, the interference mitigation with respect to the no mitigation case is stronger, but the loss with respect to the no interference case is higher.
- With the same C/I, the scenario 2 symmetrical channel matrix leads to a better interference cancellation with respect to the scenario 1 actual channel matrix. This happens because all the channels converge together at the same E_s/N_o.
- For the channel matrix estimation, a value of 40 for the UW sequence length can be considered sufficient, leading to a less than 0.5 dB loss with respect to the perfect channel estimation.
- EXT or APP feedback cancellations show a similar behaviour. EXT feedback cancellation is used. The number of 8 iterations for the turbo decoder can be considered sufficient.
- The usage of the MAX algorithm for the turbo decoding is advisable because:
    - It is lighter than the MAX^* algorithm in terms of hardware requirements.
    - It shows almost no losses with respect to the MAX^*.
    - Its weights are adjustable. Their tuning can be further investigated.

8. CONCLUSIONS

In this work, techniques for improving the bandwidth utilization of a multibeam satellite scenario RL, adopting an enhanced DVB-RCS standard, have been studied and simulated. In order to achieve an efficient spectrum usage, the same frequency has been reused for every satellite beam, producing a mutual CCI interference. Consequently, severe degradation in performance is experienced unless powerful interference cancellation schemes are implemented, therefore iterative MUD algorithms have been considered, where all the beams are jointly demodulated by the same gateway so as to suppress the CCI interference. This state-of-the-art interference mitigation approach can be applied to any multibeam satellite interactive system (using DVB-RCS or other proprietary standards for the RL) with the aim of increasing the overall efficiency and decreasing the cost per bit delivered to the end users. The RL access was assumed to be low-rate TDMA, with data rates from around one hundred kbit/s to few Mbit/s, as this scheme is currently the most popular in the considered scenario.

The PhD activity was part of a larger research program, named AMPIST, that consisted of developing a complete system laboratory hardware prototype to allow full validation of the physical layer and upper layers for an interactive satellite scenario that implements modified DVB-RCS PHY and MAC layers on the RL together with a DVB-S2 based FL. The PhD developed activities can be divided into two consequent simulations sets: single user simulations and multiuser simulations. The analyzed parameters for the two cases are summarized in Table 8.1 for the sake of clarity.

The single user simulations had the purpose to propose a burst format for the enhanced DVB-RCS used in the AMPIST RL (which adopts the ACM methodology in order to maximize the transmitted bit-rate on the available spectrum), compatible with the time/frequency recovery algorithms for very low SNR signals and the channel matrix estimation, necessary for the MUD techniques. Different burst formats have been compared and the one with the best performance has been selected for the successive multiuser simulations. This burst format exhibits a maximum 0.4 dB loss in the BER curve when the time/frequency recovery algorithms are activated, with a 14% pilot symbols overhead.

	SINGLE USER SIMULATIONS	MULTIUSER SIMULATIONS
Frequency/phase recovery	Yes	Yes
Time recovery	Yes	Yes
Turbo MAX	Yes	Yes
Turbo MAX*	Yes	Yes
Different code rates (for ACM)	Yes	Yes
Different burst formats	Yes	No
Asynchronous users	n/a	Yes
Synchronous users	n/a	Yes
Different real CCI channel matrixes	n/a	Yes
CCI channel matrix estimation	n/a	Yes
MMSE	n/a	Yes
SUMF-IC	n/a	Yes

Table 8.1: Analyzed parameters for the single user and multiuser simulations.

For the multiuser simulations, four single user channels were combined in order to simulate the CCI interference and a novel iterative MUD interference cancellation scheme was adopted, adapting the so called SUMF-IC algorithm presented in literature for CDMA systems. Complete demodulator simulations were performed in general scenarios representative of real cases, with time/frequency recovery, channel matrix estimation and asynchronous users. The multiuser simulations have demonstrated the proposed interference mitigation technique's reasonable complexity and an effective CCI reduction. Therefore the same carrier frequency can be reused in adjacent satellite beams that are jointly demodulated

by the same gateway (at the expense of a certain users power increase). Detailed multiuser scenario guidelines are reported in paragraph 7.4. This work has also been presented in [36].

Finally, multiuser simulations have shown some synchronization criticalities, in particular the time recovery, when the C/I before the interference mitigation is very low (values of C/I lower than 0 dB are not totally unlikely). To solve this problem, iterative time/frequency recovery may be used, i.e. timing, carrier frequency and phase are re-estimated at every interference cancellation iteration. That should lead to better results and such an improved approach could be further investigated in future works.

BIBLIOGRAPHY

[1] T. R. Giallorenzi and S. G. Wilson, "Multiuser ML sequence estimator for convolutionally coded asynchronous DS-CDMA systems", *IEEE Trans. on Communications*, vol. 44, no. 8, pp. 997-1008, August 1996.

[2] S. Verdù, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[3] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system", *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, pp. 796–807, June 1994.

[4] T. C. Yoon, R. Kohno and H. Imai, "A spread-spectrum multi-access system with cochannel interference cancellation for multipath fading channels", *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 7, pp. 1067–1075, September 1993.

[5] M. Varanasi, "Decision feedback multiuser detection: a systematic approach", *IEEE Trans. on Inform. Theory*, vol. 45, no. 1, pp. 219–240, January 1999.

[6] A. Lampe and J. Huber, "On improved multiuser detection with soft decision interference cancellation", in *Proc. Int. Conf. Communications*, Vancouver, BC, Canada, pp. 172–176, June 1999.

[7] A. Hui and K. Ben Letaief, "Successive interference cancellation for multiuser asynchronous DS/CDMA detectors in multipath fading links", *IEEE Trans. on Communications*, vol. 46, no. 3, pp. 384–391, March 1998.

[8] D. Divsalar, M. Simon and D. Raphaeli, "Improved parallel interference cancellation for CDMA", *IEEE Trans. on Communications*, vol. 46, no. 2, pp. 258–268, February 1998.

[9] ETSI EN 301 790 V1.3.1 (2003-03), “Digital Video Broadcasting (DVB); interaction channel for satellite distribution systems”. Available on the DVB organization website (http://www.dvb.org) or on the ETSI website (http://www.etsi.org).

[10] ETSI EN 302 307 V1.1.2 (2006-06), “Digital Video Broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications”. Available on the DVB organization website (http://www.dvb.org) or on the ETSI website (http://www.etsi.org).

[11] R. Rinaldo and R. De Gaudenzi, “Capacity analysis and system optimization for the reverse link of multi-beam satellite broadband systems exploiting adaptive coding and modulation”, *International Journal of Satellite Communications and Network*, vol. 22, no. 4, pp. 425-448, 2004.

[12] U. De Bie, B. Collini-Nocker, G. Fairhurst, A. Ginesi, A. Jahn, R. Rinaldo and O. Del Rio “GSE: DVB-S2 generic stream IP encapsulation protocol”. Available on the ESA web site (http://emits.esa.int/emits-doc/REF6_Annex1.pdf).

[13] Modem for High Order Modulation Schemes: Turbo-Φ for the Reverse Link. ENST Bretagne, TRO contract number 16593/02/NL/EC.

[14] A. J. Viterbi, “An intuitive justification and a simplified implementation of the MAP decoder of a convolutional code”, *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 260-264, February 1998.

[15] J. Vogt and A. Finger, “ Improving the max-log-map turbo decoder”, *IEE Electronics Letters*, vol. 36, no. 23, pp. 1937-1939, November 2000.

[16] J. Cioffi, G. Dudevoir, M. Eyuboglu and G. Forney, “MMSE decision feedback equalizers and coding - Part I: equalization results”, *IEEE Trans. on Communications*, vol. 43, no. 10, pp. 2582–2594, October 1995.

[17] J. Cioffi, G. Dudevoir, M. Eyuboglu and G. Forney, "MMSE decision feedback equalizers and coding - Part II: coding results", *IEEE Trans. on Communications*, vol. 43, no. 10, pp. 2595–2604, October 1995.

[18] M. Varanasi and T. Guess, "Achieving vertices of the capacity region of the synchronous correlated-waveform multiple-access channel with decision-feedback receivers", in *IEEE International Symposium on Information Theory*, p. 270, 1997.

[19] V. Marchenko and L. Pastur, "Distribution of eigenvalues for some sets of random matrices", *Math USSR Sb.*, vol. 1, pp. 457–483, 1967.

[20] C. Berrou, A. Glavieux and P. Titimajshima "Near Shannon limit error-correcting coding and decoding: turbo codes", *Proc. 1993 Int. Conf. on Communications*, vol. 2, pp. 1064-1070, 1993.

[21] J. C. Moreira and P. G. Farrell, *Essential of Error-Control Coding.* New York: John Wiley & Sons, 2006.

[22] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimising symbol error rate", *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284-287, March 1974.

[23] M. Moher, "An iterative multiuser decoder for near–capacity communications," *IEEE Trans. on Communications*, vol. 46, no. 7, pp. 870–880, July 1998.

[24] J. Boutros and G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis", *IEEE Trans. on Information Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.

[25] M. Moher, "Multiuser decoding for multibeam systems", *IEEE Trans. On Vehic. Tech.*, vol. 49, no. 4, pp. 1226–1234, July 2000.

[26] G. Caire, R. R. Müller and T. Tanaka, "Iterative multiuser joint decoding: optimal power allocation and low-complexity implementation", *IEEE Trans. on Information Theory*, vol. 50, no. 9, pp. 1950-1972, September 2004.

[27] G. Caire and R. R. Müller, "The optimal received power distribution for IC–based iterative multiuser joint decoders", in *Proc. of 39th Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL, October 2001.

[28] J. G. Proakis, *Digital Communications (3rd ed.)*. New York: McGraw-Hill, 1995.

[29] S. Benedetto *et al.*, "MHOMS: high speed ACM modem for satellite applications", *IEEE Wireless Communications*, vol. 12, no. 2, pp. 66-76, April 2005.

[30] H. Li, S. M. Betz and H.V. Poor, "Performance analysis of iterative channel estimation and multiuser detection in multipath DS-CDMA channels", *IEEE Trans. on Signal Processing*, vol. 55, no. 5, pp. 1981-1993, May 2007.

[31] X. Wang and H.V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA", *IEEE Trans. on Communications*, vol. 47, no. 7, pp. 1046-1061, July 1999.

[32] M. Kobayashi, J. Boutros and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation", *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 8, pp. 1450-1460, August 2001.

[33] G. Caire, S. Guemghar, A. Roumy and S. Verdù, "Maximizing the spectral efficiency of coded CDMA under successive decoding", *IEEE Trans. on Information Theory*, vol. 50, no. 1, pp. 152-164, January 2004.

[34] B. F. Beidas, H. El Gamal and S. Key, "Iterative interference cancellation for high spectral efficiency satellite communications", *IEEE Trans. on Communications*, vol. 50, no. 1, pp. 31-36, January 2002.

[35] H. El Gamal and E. Geraniotis, "Iterative multiuser detection for coded CDMA signals in AWGN and fading channels", *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 1, pp. 30-41, January 2000.

[36] F. Di Cecca, G. Gallinaro, B. Tacca and A. Vernucci, "Co-channel interference mitigation for the reverse-link of multibeam satellite networks", *14th Ka and Broadband Communications Conference*, Matera, Italy, September 2008.

[37] L. Hanzo, T. H. Liew and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding, for Transmission over Fading Channels*. New York: IEEE Press/Wiley, 2001.

[38] A. J. Viterbi and A. M. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission", *IEEE Trans. on Information Theory*, vol. 29, no. 4, pp. 543-551, July 1983.

[39] M. Oerder and H. Meyr, "Digital filter and square timing recovery", *IEEE Trans. on Communications*, vol. 36, no. 5, pp. 605-612, May 1988.

[40] U. Mengali and M. Morelli, "Data-aided frequency estimation for burst digital transmission", *IEEE Trans. on Communications*, vol. 45, no. 1, pp. 23-25, January 1997.