Escola Politècnica Superior de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Design of a simulation platform to test next generation of terrestrial DVB

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Sistemes de Telecomunicació

**AUTORS: Carlos Enrique Herrero
Carlos Alberto López Arranz**

**DIRECTOR: Silvia Ruiz Boqué
Luís Alonso Zárate**

DATA: 19 de juliol de 2007

Title: Design of a simulation platform to test next generation of terrestrial DVB

Author: Carlos Alberto López Arranz, Carlos Enrique Herrero

Director: Silvia Ruiz Boqué, Luís Alonso Zárate

Date: July, 19th 2007

Overview

Digital Terrestrial Television Broadcasting (DTTB) is a member of our daily life routine, and nonetheless, according to new users' necessities in the fields of communications and leisure, new challenges are coming up. Moreover, the current Standard is not able to satisfy all the potential requirements.

For that reason, first of all, a review of the current Standard has been performed within this work. Then, it has been identified the needing of developing a new version of the standard, ready to support enhanced services, as for example broadcasting transmissions to moving terminals or High Definition Television (HDTV) transmissions, among others.

The main objective of this project is the design and development of a physical layer simulator of the whole DVB-T standard, including both the complete transmission and reception procedures. The simulator has been developed in *Matlab.* A detailed description of the simulator both from a functional and an architectural point of view is included. The simulator is the base for testing any possible modifications that may be included into the DVB-T2 future standard. In fact, several proposed enhancements have already been carried out and their performance has been evaluated. Specifically, the use of higher order modulation schemes, and the corresponding modifications in all the system blocks, have been included and evaluated. Furthermore, the simulator will allow testing other enhancements as the use of more efficient encoders and interleavers, MIMO technologies, and so on.

A complete set of numerical results showing the performance of the different parts of the system, are presented in order to validate the correctness of the implementation and to evaluate both the current standard performance and the proposed enhancements.

This work has been performed within the context of a project called FURIA, which is a strategic research project funded by the Spanish Ministry of Industry, Tourism and Commerce. A brief description of this project and its consortium has been also included herein, together with an introduction to the current situation of the DTTB in Spain (called TDT in Spanish).

Título: Diseño de una plataforma de simulación para la nueva generación de DVB terrestre

Autores: Carlos Alberto López Arranz, Carlos Enrique Herrero

Directores: Silvia Ruiz Boqué, Luís Alonso Zárate

Fecha: 19 de julio de 2007

Resumen

La DVB-T es ya una parte integrante de nuestra vida cotidiana, y no obstante, debido a las nuevas necesidades de los usuarios en el ámbito de las comunicaciones y el ocio, surgen nuevos retos que el estándar actual no es capaz de satisfacer. Por ello se hace necesaria una revisión del actual estándar de la DVB-T.

A partir de esta evaluación, se ha identificado la necesidad de desarrollar una nueva versión del estándar capaz de soportar nuevos servicios, como por ejemplo transmisiones a terminales móviles o televisión de alta definición (HD-TV), entre otras opciones.

El objetivo principal de este proyecto es el diseño y desarrollo de un simulador de capa física del estándar DVB-T, incluyendo tanto los bloques de transmisión como los de recepción.

A lo largo del documento se incluye una descripción detallada desde el punto de vista funcional y de arquitectura. El simulador ha sido desarrollado en Matlab y es la base para poder probar cualquier posible modificación que pueda ser incluida en el futuro estándar llamado DVB-T2; de hecho, se han implementado y evaluado algunas mejoras que son susceptibles de ser incluidas en el nuevo estándar. Concretamente se ha evaluado el uso de esquemas de modulación de mayor orden, contemplando las correspondientes modificaciones en todos los bloques del sistema. Además, el simulador permitirá probar nuevas mejoras como el uso de codificadores y entrelazadores más eficientes, técnicas MIMO, etc.

En la última parte del documento se presentan los resultados gráficos obtenidos con el simulador para validar su correcto funcionamiento de todas sus partes, así como para poder evaluar las mejoras introducidas.

Por ultimo, se ha de destacar que el trabajo ha sido desarrollado dentro del contexto del proyecto FURIA, un proyecto estratégico singular de investigación propuesto por el Ministerio de Industria, Turismo y Comercio. Por ello se incluye una breve descripción sobre este consorcio, además de una introducción del estado actual de la Televisión Digital Terrestre en España.

A nuestros padres,

Sin vuestra ayuda no podríamos
haber llegado hasta aquí.

INDEX

TABLE INDEX

FIGURE INDEX

DEFINITIONS, SYMBOLS AND ABBREVIATIONS

Definitions

constraint length: number of delay elements +1 in the convolutional coder.

Symbols

$A(e)$	output vector from inner bit interleaver e
$a_{e,w}$	bit number w of inner bit interleaver output stream e
α	constellation ratio which determines the QAM constellation for the modulation for hierarchical transmission
$B(e)$	input vector to inner bit interleaver e
$b_{e,w}$	bit number w of inner bit interleaver input steam e
$b_{e,do}$	output bit number do of demultiplexed bit stream number e of the inner interleaver demultiplexer
b_i	bit number i of the cell identifier
$c_{m,l,k}$	complex cell for frame m in OFDM symbol l at carrier k
C'_k	complex modulation for a reference signal at carrier k
$C'_{l,k}$	complex modulation for a TPS signal at carrier k in symbol l
C/N	Carrier-to-Noise ratio
Δ	time duration of the guard interval
d_{free}	convolutional code free distance
f_c	centre frequency of the emitted signal
G_1, G_2	convolutional code Generator polynomials
$g(x)$	Reed-Solomon code generator polynomial
$h(x)$	BCH code generator polynomial
$H(q)$	inner symbol interleaver permutation
$H_e(w)$	inner bit interleaver permutation
i	priority stream index
I	Interleaving depth of the outer convolutional interleaver
$I0,I1,I2,I3,I4,I5$	inner Interleavers
j	branch index of the outer interleaver
k	carrier number index in each OFDM symbol
K	number of active carriers in the OFDM symbol
K_{min}, K_{max}	carrier number of the lower and largest active carrier respectively in the OFDM signal
l	OFDM symbol number index in an OFDM frame
m	OFDM frame number index
m'	OFDM super-frame number index
M	convolutional interleaver branch depth for j = 1, M = N/I
n	transport stream sync byte number
N	length of error protected packet in byte
N_{max}	inner symbol interleaver block size
p	scattered pilot insertion index
$p(x)$	RS code field generator polynomial

$P_k(f)$	Power spectral density for carrier k
$P(n)$	interleaving Pattern of the inner symbol interleaver
r_i	code rate for priority level i
s_i	TPS bit index
t	number of bytes which can be corrected by the Reed-Solomon decoder
T	elementary Time period
T_S	duration of an OFDM symbol
T_F	Time duration of a frame
T_U	Time duration of the useful (orthogonal) part of a symbol, without the guard interval
u	bit numbering index
v	number of bits per modulation symbol
w_k	value of reference PRBS sequence applicable to carrier k
x_{di}	input bit number di to the inner interleaver demultiplexer
x'_{di}	high priority input bit number di to the inner interleaver demultiplexer
x''_{di}	low priority input bit number di to the inner interleaver demultiplexer
Y	output vector from inner symbol interleaver
Y'	intermediate vector of inner symbol interleaver
y_q	bit number q of output from inner symbol interleaver
y'_q	bit number q of intermediate vector of inner symbol interleaver
z	complex modulation symbol
*	complex conjugate

Abbreviations

ACI	*Adjacent Channel Interference*
ADSL	*Assymetric Digital Subscriber Line*
BCH	*Bose - Chaudhuri - Hocquenghem code*
BER	*Bit Error Rate*
CCI	*Co-Channel Interference*
CD3	*Coded Decision Directed Demodulation*
C/F SS	*Coarse / Fine Symbol Synchronization*
CSI	*Channel State Information*
DAB	*Digital Audio Broadcasting*
DBPSK	*Differential Binary Phase Shift Keying*
DFT	*Discrete Fourier Transform*
DMB	*Digital Multimedia Broadcasting*
DOA	*Directon Of Arrival*
DVB	*Digital Video Broadcasting*
DVB-H	*Digital Video Broadcasting- Handheld*
DVB-T	*Digital Video Broadcasting-Terrestrial*
DVB-S	*Digital Video Broadcasting-Satellite*
DTTB	*Digital Terrestrial Television Broadcasting*
EDTV	*Enhanced Definition TeleVision*
FFT/IFFT	*Fast Fourier Transform / Inverse Fast Fourier Transform*

FIFO	*First-In, First-Out shift register*
HDTV	*High-Definition Television*
HIPERLAN	*High Performance Radio Local Area Network*
HP	*High Priority bit stream*
HSDPA	*High Speed Downlink Packet Access*
ISI	*Inter Symbol Interference*
LDPC	*Low Density Parity Check*
LDTV	*Limited Definition TeleVision*
LOS	*Line Of Sight*
LP	*Low Priority bit stream*
MC-CDMA	*Multi Carrier – Code Division Multiple Access*
MIMO	*Multiple Input Multiple Output*
MPEG	*Moving Picture Experts Group*
MQAM	*Mary Quadrature Amplitude Modulation*
MRC	*Maximal-Ratio Combining*
MSB	*Most Significant Bit*
MUX	*Multiplex*
NICAM	*Near-Instantaneous Companded Audio Multiplex*
OFDM	*Orthogonal Frequency Division Modulation*
PAL	*Phase Alternating Line*
PAPR	*Peak to Average Power Ratio*
PRBS	*Pseudo-Random Binary Sequence*
QAM	*Quadrature Amplitude Modulation*
QEF	*Quasi Error Free*
QPSK	*Quaternary Phase Shift Keying*
RF	*Radio Frequency*
RS	*Red Solomon*
SCCC	*Serial Concatenated Convolutional Codes*
SDVD	*Soft Decision Viterbi Decoder*
SDTV	*Standard Definition TeleVision*
SECAM	*Système Sequentiel Couleur A Mémoire*
SFN	*Single Frequency Networks*
STBC/SFBC	*Space Time Block Codes / Space Frequency Block Codes*
STTC	*Space Time Trellis Codes*
TC	*Turbo Codes*
TPC	*Turbo Product Codes*
TPS	*Transmission Parameter Signalling*
TV	*TeleVision*
UHF	*Ultra-High Frequency*
V-BLAST	*Vertical Bell Laboratories layered Space Time*
VHF	*Very-High Frequency*
WLAN	*Wireless Local Area Network*

CHAPTER 0: INTRODUCTION

0.1. Main goal of this project

During the last years, the mobile telephone has revolutionized society habits and there are also coming up new concepts: terminal associated to people instead of terminals associated to places, multimedia entertainment while people is waiting, streaming on demand; in other words personalised solutions.

Actually mobile phone operators offer different multimedia services, for example: music video-clips, web mail access, TV programs, etc. by means of third generation networks (3G). However the inefficiency of these networks in transmitting the same content to different users, puts a limit in the maximum number of active users supported by the system, and also determines the high cost of these services.

All these factors make necessary a convergence/cooperation between telecommunications and broadcasting networks, but also imply broadcast community to reconsider its technical approach in terms of deciding how to manage new services and opportunities.

In this scenario, and knowing that these requirements are not satisfied by the actual services, it becomes really necessary the deployment of new standards including specifications for the new mobile broadcast scenarios, and with the capability to support the actual and future market demand.

DVB-T standard has been developed more than 10 years ago, and in all of this years new radio communications systems have been appearing using different OFDM modulation techniques, so it’s easy to think that this systems are employing coders, interleavers, modulation techniques, pilots, transmission diversity, etc. offering better features than DVB-T standard does. The incorporation of this techniques requires a detailed analysis of the system impact, and a comparative between classic DVB-T, following the objective that new DVB-T2 standard must support HDTV and must offer mobility and good indoor coverage. In section 1.5 we will define which are the requirements for DVB-T2 model).

The main goal of this project is to deploy the new blocks that should make that all of DVB-T2 wanted specs can be realized. To reach this goal the DVB organization has established some different marks that must be contemplated in this new standard. Now we are going to explain which these conditions are and in ***Chapter 6*** explaining DVB-T2 modifications we will also discuss if this objective is accomplished.

This project has been developed under a more ambitious national project named FURIA [1]. FURIA is a SSP project (Strategically Singular Project) in the field of Network Audiovisual Technologies, which main objective is **to develop and validate the integration of emergent technologies for the spreading of**

audiovisual contents in fixed and mobile devices. Joining forces from the different national organisations (companies, technological centres and universities) with the final purpose of hold on the national high technological level.

FURIA consortium has the capability to finish investigation and development stages in the new contents of broadcasting audiovisual technologies, and will realise valuable contributions to the main standardization bodies in an industrial forum context, to elaborate technical proposals for Digital Video Broadcasting future.

Is expected generation of research results that won't be immediately applied, but they will be necessary in future stages of the project development, it means generate pre-industrial results.

Other objectives of the project are:

- The establishment of relationships with other national and European projects, which will allow the enrichment of Spanish technological level, and the generation of:
- Contributions to forums and European standards, to grow up Spanish technological consortium acknowledgment and to have influence in the standardisation section.

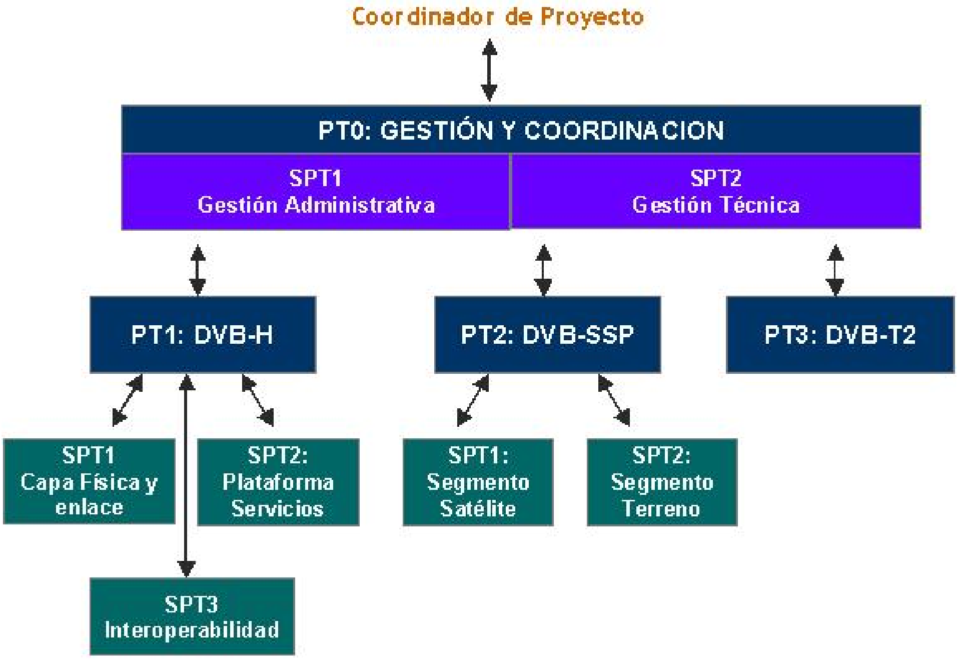


Fig. 0.1: General Scheme of FURIA council.

It is worth highlighting that FURIA project is developed in the framework of the technological Spanish platform e-NEM, national mirror of its European homonymic NEM (Networked and Electronic Media).

The technological Spanish platform eNEM is an open forum promoted by the industry of Audiovisual Technologies Network, with the main objective of obtaining a mass critical investigation in fundamental issues for the Spanish development in this area. It's constituted by groups or excellence poles and a coordination of technical-scientist, composed by relevant actors in the Network Audiovisual Technologies area.

Fig. 0.2 Participating entities.

0.2. Structure of the written report

Within this project framework, Radio Communications Research Group (together with the Fundació i2CAT) participates in the PT3, realising studies, simulations and the solution campaigns that are pretty needed to manage in an optimal way the deployment.

Knowing that the definition guidelines for the new DVB-T2 where not published when this project started, our main task during the 6 months of project has been the implementation of an adapted simulator to the DVB-T standard, with the objective of having a efficient software platform over which future improvements can be added, tested and compared with the previous standard. The second part of our project has been the study of new possible codification, interleaved, modulation schemes, pilots' structure, introduction of MIMO techniques, etc.

In this memory all the work realised during these six working months is summarized, in a theoretical and practical framework in the simulator development. It's remarkable that this simulator has been a collective work, specially mentioning our partner and friend Gabriel Martorell Lliteres, who has been responsible of the OFDM modulator and channel model development. The integration task has been difficult and time consuming, as well as the number of test running to validate that everything was working properly. These are tasks that usually are not reflected at the written report, but are one of the critical tasks in big engineering projects.

Our task in this project has been the deployment of transmission and receiver blocks for the system. Along this memory we are will give a detailed explanation of the global system (including modulation and channel), as the simulator is composed by all these fields.

The project is composed by three significant blocks. First, the chapters' one and two introduce the reader in the DVB-T context, with a short description of the DVB-T in Spain and specifying the initial scenarios. In the second block (chapters 3, 4 and 5), the implemented simulator is described at both, theoretical and programmer level. Finally, in chapters 6, 7 and 8, we expose the results of the simulator and the future modifications for the T2 mode, just as the conclusions.

CHAPTER 1: INTRODUCTION TO DVB

In this section we are going to make an introduction to Digital Video Broadcast systems (DVB), focussing on the Spanish version *"Televisión Digital Terrestre"* (TDT). Also in this section, the technical bases of these systems will be analysed to know which have been the reasons to choose this standard as well as its advantages and drawbacks.

The first question about which standard was going to be used, with its associated technology, was well defined in the seminar about digital television performed by *DigiTAG (Digital Terrestrian Television Action Group).* One of the main targets of this seminar was to establish the following rule: the only competence between operators shall not be a technological contest. With this in mind the development of open standards became really necessary to ensure the acceptation of these systems by users and for the technology development. So now, emergent markets have the responsibility to promote these open standards, making easier the competence between providers and diversifying services between operators.

Thanks to the great efforts from different standardization entities, four international DTTB standards have been ratified during the past ten years and they are:

- USA-based ATSC (Advanced Television System Committee) standard,
- Europe-based DVB-T (Digital Video Broadcasting-Terrestrial) standard, and
- Japan-based ISDB-T (Integrated Services Digital Broadcasting-Terrestrial) standard.
- Chinese-based DTMB (Digital Terrestrial/Television Multimedia Broadcasting) standard.

It's interesting to make a few comments about USA and Chinese standards, because some of the new proposals for the DVB-T2 mode arise from those standards. However, in chapter 6 when we talk about new coding techniques we will analyze both in more detail.

USA-based ATSC is based on the 8-VSB modulation. It's a Vestigial Side Band system (as analogical systems), based on an 8-QAM modulation witch is extended to 64-QAM with a Trellis coding structure.

Chinese-based DTMB is more similar to DVB-T standard, and uses the same modulation schemes. The main differences are coding techniques (LDPC instead of Trellis, BCH instead of Red-Solomon), and OFDM transmission without using pilots.

1.1. Digital TV in front Analogical TV

The main problem in analogical television is that it isn't efficient talking in electromagnetic spectral parameters. This is cause it doesn't take benefit of the low signal changing between pixels, and also cause it generates a lot of redundancy information (it contains more information than the human eye needs to appreciate correctly an image).With digital compression techniques, like MPEG-2, we can send only the image changes, so fewer data is needed to image refresh.

Also thanks to MPEG-2 compression all the radio electrical DTV channels have the same bandwidth (8 MHz), but with the improvement that more than one TV program can be transmitted (5 with similar analogical definition quality) or may be one with High Definition quality.

As DTV uses terrestrial network, it provides a low cost deployment (it takes profit of the receivers, antennas etc. now installed), with the benefit that it requires lower transmission power and allowing mobile reception. Talking about services it allows different TV programs in the same multiplex offering 16:9 quality, and Multilanguage audio services of each program. Also with MHP (Multi Home Platform) it allows television became a multimedia terminal offering a lot of kinds of applications. Finally as we commented in last section the employment of single frequency networks (SFN) reduces deployment complexity so it allows more equilibrated development.

1.2. Modulation schemes: Isofrequential networks

Orthogonal Frequency Division Modulation (OFDM) is well known modulation technique that has been used on the last radio broadcast applications (DAB), ADSL modems or in other standards like *IEEE 802.11* [2].

In one hand the main advantage of OFDM modulation is to transform a frequency selective channel in to a group of different non selective sub-channels (narrower channels). This feature makes OFDM be the most spectral efficient system, witch is really required, and it also provides a strong immunity to multipath. In an OFDM system, data is transmitted in parallel in a group of subcarriers providing natural diversity and frequency interleaving. These subcarriers were orthogonal in time domain (consecutive pilots spaced by the inverse of symbol time), tolerating spectrum overlap in frequency domain so it warranties an efficient spectrum use. OFDM signals can be easy generated and demodulated using Fourier Fast Transform (FFT and IFFT). It's immunity to multipath allows the implementation of easy equalizers and channel estimators, and makes them more tolerant to synchronization errors in broadcast applications. Another advantage of OFDM is that adaptative coding and modulation techniques can be used, and also the OFDM symbol windowing allows implementing spectrum conformation filters.

In the other hand, main disadvantages of OFDM modulation are basically two: high peak power and average power relationship, witch it forces the use of lineal filters and amplifiers. And the other one implies that system is so sensitive to frequency offsets (sampling frequency errors), generated by synchronization errors and Doppler effects, witch make orthogonal loses between subcarriers.

OFDM also allows isofrequency networks (SFN, Single Frequency Networks), where different transmitters send the same signal at the same time inside one carrier. At receiver signals from different transmitters can be constructive combined to generate a diversity gain. The fact that bandwidth is subdivided in different sub channels, and all of them are modulated at low transmission velocity, makes symbol delay large enough to erase delay spread effects. The cyclic prefix (also denominated guard period) between consecutive OFDM symbols reduces intersymbolic interference (ISI) effects.

However is possible, that signal generated by different delays between different transmitters, could not be eliminated.

An easy way to solve this problem is to consider that the different signals arriving from different transmitters are transmitted in independent paths that can be isolated knowing the arrival angle to de receptor (DOA. Direction of Arrival). Using multiple antennas at the receiver, a MIMO (Multiple-Input Multiple-Output) would be performed.

Two modes of operation, a "2K mode" and an "8K mode", are defined for DVB-T and DVB-H transmissions. The "2K mode" is suitable for single transmitter operation and for small SFN networks with limited transmitter distances. The "8K mode" can be used both for single transmitter operation and for small and large SFN networks.

Exclusively for use in DVB-H systems, a third transmission mode the "4K mode" is defined in annex F, addressing the specific needs of Handheld terminals. The "4K mode" aims to offer an additional trade-off between transmission cell size and mobile reception capabilities, providing an additional degree of flexibility for DVB-H network planning.

1.3. DTTB in Spain

In Spain, the December 30th of 2004, the minister announced a plan to impulse DTTB, advancing the end of analogical terrestrial television broadcasting to April 2010.and defining the scenario after this turn off.

1.3.1. Digital Transition

During the writing of this Project we are in the Digital transition period. **(Fig. 1.1)** shows the DTTB coverage map in Spain today.

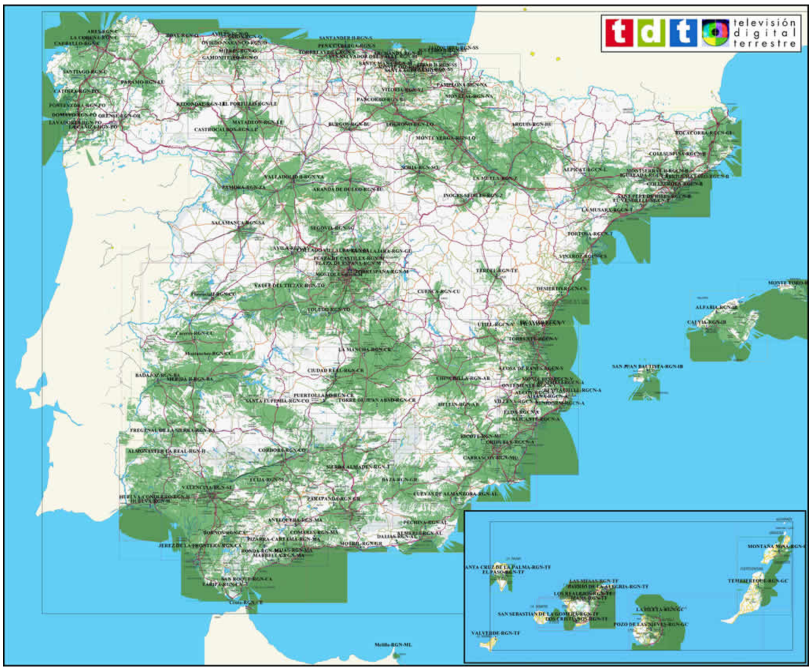


Fig. 1.1 DTTB coverage map in year 2007 [3].

In this transition period, it has been established that all the companies providing analogical terrestrial television services, must cease progressively analogical emissions in a coordinated way so consumers shouldn't experience any change in the TV service.

Finishing this short review of the BTTD television in Spain we are going to expose the main characteristics of the national plan.

Frequency Band:

a) 470 to 758 MHz (channels 21 to 56).
b) 758 to 830 MHz (channels 57 to 65).
c) 830 to 862 MHz (channels 66 to 69).

Multiplex:

a) Digital multiplex in the band between 830 to 862 MHz, are destined to establish national frequency networks.
b) Digital multiplex in the band between 758 to 830 MHz, are destined to establish provincial and autonomic networks.
c) Digital multiplex in the band between 470 to 758 MHz, are destined to establish digital television networks.

Covering targets:

a) Each of channels 66, 67, 68 and 69 will form four digital multiplex in single frequency networks.

b) Radio electrical channel 57, 58, 59, 60, 61, 62, 63, 64 and 65 will form a digital multiplex with capabilities to perform autonomic disconnections.

Covering phases:

a) 80% of inhabitants before December 31st of 2005.
b) 90% of inhabitants before December 31st of 2008.
c) 98% of inhabitants before April 3rd of 2010.

Table 1.1 Details of the parameters defined to exploit digital television in Spain.

Organization	TV Channels	Radio Channels	Multiplex	Frequency
Televisión Española	La Primera	Radio 1	MFN	
	La 2	Radio Clásica	MFN	
	Canal 24 Horas	Radio 3	MFN	
	Clan TVE		MFN	
	Teledeporte		SFN (66)	834 MHz
Antena 3	Antena 3	Onda Cero	SFN (69)	858 MHz
	Antena.Neox	Onda Melodía	SFN (69)	858 MHz
	Antena. Nova	Europa FM	SFN (69)	858 MHz
Sogecable	Cuatro		SFN (67)	842 MHz
	CNN +		SFN (67)	842 MHz
	40 Latino		SFN (67)	842 MHz
Telecinco	Telecinco		SFN (68)	850 MHz
	Telecinco Estrellas		SFN (68)	850 MHz
	Telecinco Sport		SFN (68)	850 MHz
Veo TV	Veo TV		SFN (66)	834 MHz
	SET en Veo		SFN (66)	834 MHz
	Tienda en Veo		SFN (66)	834 MHz
Net TV	Net TV		SFN (66)	834 MHz
	Fly Music	Punto Radio	SFN (68)	850 MHz
La Sexta	laSexta		SFN (67)	842 MHz
	TeleHit (Señal mexicana)		SFN (69)	858 MHz

1.4. European DTTB system

The European system is based on DVB-T (Digital Video Broadcast-Terrestrial), developed by *ETSI (European Telecommunications Standards Institute),* and collected in the document *ETSI 300 744* on March 1997 [4]. This standard has been adopted by European countries, and also by other countries like Australia, Brazil or India.

In the following paragraph a short description about technical aspects from the DVB-T, and its application in DTTB systems.

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of over 260 broadcasters, manufacturers, network operators, software developers, regulatory bodies and others in over 35 countries committed to designing global standards for the global delivery of digital television and data services. This supposes to know real necessities of electronic markets, economical situations and also broadcast television industry. Considering some of these factors, different coding and modulation schemes are defined for the following terrestrial services: LDTV (Limited Definition Television), SDTV (Standard Definition Television), EDTV (Enhanced Definition Television) and HDTV (High Definition Television).

One of the main characteristics from the DVB-T is the use of MPEG-2 packets; it implies that every digital information (video, audio, etc…) is transportable. Also, it’s specified a return for users to interoperate with the different services received.

1.5. DVB-T2 objectives

The main goal of this project is to deploy the new blocks that should make that all of DVB-T2 wanted specs can be realized. To reach this goal the DVB organization has established some different marks [5] that must be contemplated in this new standard. Now we are going to explain with are this conditions and in ***Chapter 5*** explaining DVB-T2 modifications we will also discuss if this objective is accomplished.

Requirements:

1. The DVB-T2 specification shall be designed for stationary reception. However, it shall be possible to design DVB-T2 networks for all three receiving conditions, fixed, portable and mobile.

2. Transmissions using the DVB-T2 specification shall meet the interference levels and spectrum mask requirements as defined by GE06 and not cause more interference than DVB-T would do.

3. The DVB-T2 specification should target the maximum increase in net payload capacity over DVB-T with similar or better robustness than DVB-T under similar conditions.

4. The DVB-T2 specification shall provide a **minimum** increase in net payload capacity of 30% greater than DVB-T for any given channel profile under similar conditions.

5. The characteristics of the DVB-T2 specification shall not impair the ability to perform, or efficiency of, statistical multiplexing of DTV Services.

6. The DVB-T2 specification should offer improved robustness against interference from other transmitters, compared to DVB-T, potentially improving frequency reuse.

7. The DVB-T2 specification shall offer a choice of various robustness and protection levels to be applied equally on all data of a transport stream carried by a DVB-T2 signal in a particular channel.

8. The DVB-T2 specification should offer a choice of various robustness and protection levels for each service separately, within a transport stream carried by a DVB-T2 signal in a particular channel. When more than one transport stream is carried by a DVB-T2 signal in a particular channel the DVB-T2 specification should offer a choice of various robustness and protection levels for each transport stream separately.

9. The DVB-T2 specification shall provide a quality of service across the whole channel that approximates to no more than one corrupted event (to any audio, video or data services) per hour for HDTV and SDTV services.

10. Impulsive noise performance of DVB-T2 shall be no worse than the DVB-T performance and should be substantially improved from that of DVB-T.

11. The DVB-T2 specification shall enable changes in modulation mode to be detected automatically within 0,5s. However, the receiver may not be capable of performing seamless changeover.

12. The DVB-T2 specification shall not introduce any more than 0,3s of additional delay in receiver channel changing and service selection times compared to DVB-T.

13. The DVB-T2 specification shall be able to provide at least the minimum specified increase in payload capacity over DVB-T using existing transmitter sites and masts broadcasting to existing DVB-T domestic antenna and cable installations.

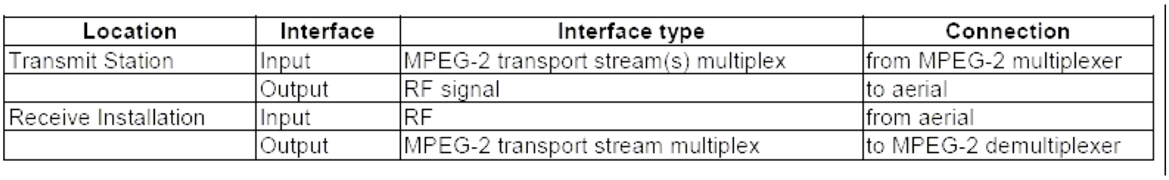
CHAPTER 2: SIMULATION ENVIRONMENT DESCRIPTION

2.1. Initial scenario

Along this chapter we will describe the scenario required to develop the simulator as well as the implied channel models and the election of the simulator platform.

In first place and based on the directives marked by the FURIA council defined in the introduction chapter, the proposed simulator will be orientated to the physical layer to bit level of DVB-T. To make possible those requirements, we implemented a generic transmitter defined at ETSI standard *EN 300_744 Digital Video Broadcasting.* The receiver block needs to be able to reconstruct the transmission signal, following the guidelines proposed by the transmitter. In the same way, the channels models used by the simulator are the described in the standard.

Table 2.1 Interfaces for the Baseline System.



Location	Interface	Interface type	Connection
Transmit Station	Input	MPEG-2 transport stream(s) multiplex	from MPEG-2 multiplexer
	Output	RF signal	to aerial
Receive Installation	Input	RF	from aerial
	Output	MPEG-2 transport stream multiplex	to MPEG-2 demultiplexer

2.2. Baseline system and general considerations

The system is defined as the functional block of equipment performing the adaptation of the base band TV signals from the output of the MPEG-2 transport multiplexer, to the terrestrial channel characteristics. The following processes shall be applied to the data stream (see [6]):

- transport multiplex adaptation and randomization for energy dispersal;
- outer coding (i.e. Reed-Solomon code);
- outer interleaving (i.e. convolutional interleaving);
- inner coding (i.e. punctured convolutional code);
- inner interleaving (either native or in-depth);
- mapping and modulation;
- Orthogonal Frequency Division Multiplexing (OFDM) transmission.

The system is directly compatible with MPEG-2 coded TV signals ISO/IEC 13818 [7].

Since the system is being designed for digital terrestrial television services to operate within the existing VHF and UHF (8MHz, 7MHz and 6MHz channel spacing) spectrum allocation for analogue transmissions, it is required that the system provides sufficient protection against high levels of Co-Channel Interference (CCI) and Adjacent Channel Interference (ACI) emanating from existing PAL/SECAM/NTSC services. It is also a requirement that the System allows the maximum spectrum efficiency when used within the VHF and UHF bands; this requirement can be achieved by utilizing Single Frequency Network (SFN) operation.

To achieve these requirements an OFDM system with concatenated error correcting coding is being specified. To maximize commonality with the Satellite baseline specification (see *EN 300 421* [8]) and Cable baseline specifications (see *EN 300 429* [9]) the outer coding and outer interleaving are common, and the inner coding is common with the Satellite baseline specification. To allow optimal trade off between network topology and frequency efficiency, a flexible guard interval is specified. This will enable the system to support different network configurations, such as large area SFN and single transmitter, while keeping maximum frequency efficiency.

The system allows different levels of QAM modulation and different inner code rates to be used to trade bit rate versus ruggedness. The system also allows two level hierarchical channel coding and modulation, including uniform and multi-resolution constellation. In this case the functional block diagram of the system shall be expanded to include the modules shown dashed in figure **(Fig. 2.1)**. Two independent MPEG transport streams, referred to as the high-priority and the low-priority stream are mapped onto the signal constellation by the Mapper and the Modulator which therefore has a corresponding number of inputs.

To guarantee that the signals emitted by such hierarchical systems may be received by a simple receiver the hierarchical nature is restricted to hierarchical channel coding and modulation without the use of hierarchical source coding.

A programme service can thus be "simulcast" as a low-bit-rate, rugged version and another version of higher bit rate and lesser ruggedness. Alternatively, entirely different programmes can be transmitted on the separate streams with different ruggedness. In either case, the receiver requires only one set of the inverse elements: inner de-interleaver, inner decoder, outer de-interleaver, outer decoder and multiplex adaptation. The only additional requirement thus placed on the receiver is the ability for the demodulator/de-mapper to produce one stream selected from those mapped at the sending end.

The price for this receiver economy is that reception can not switch from one layer to another (e.g. to select the more rugged layer in the event of reception becoming degraded) while continuously decoding and presenting pictures and sound. A pause is necessary (e.g. video freeze frame for approximately 0,5 seconds, audio interruption for approximately 0,2 seconds) while the inner decoder and the various source decoders are suitably reconfigured and reacquire lock. In our simulator, we focused our work into the non-hierarchical mode, the most common and used transmission mode in the commercial receivers.

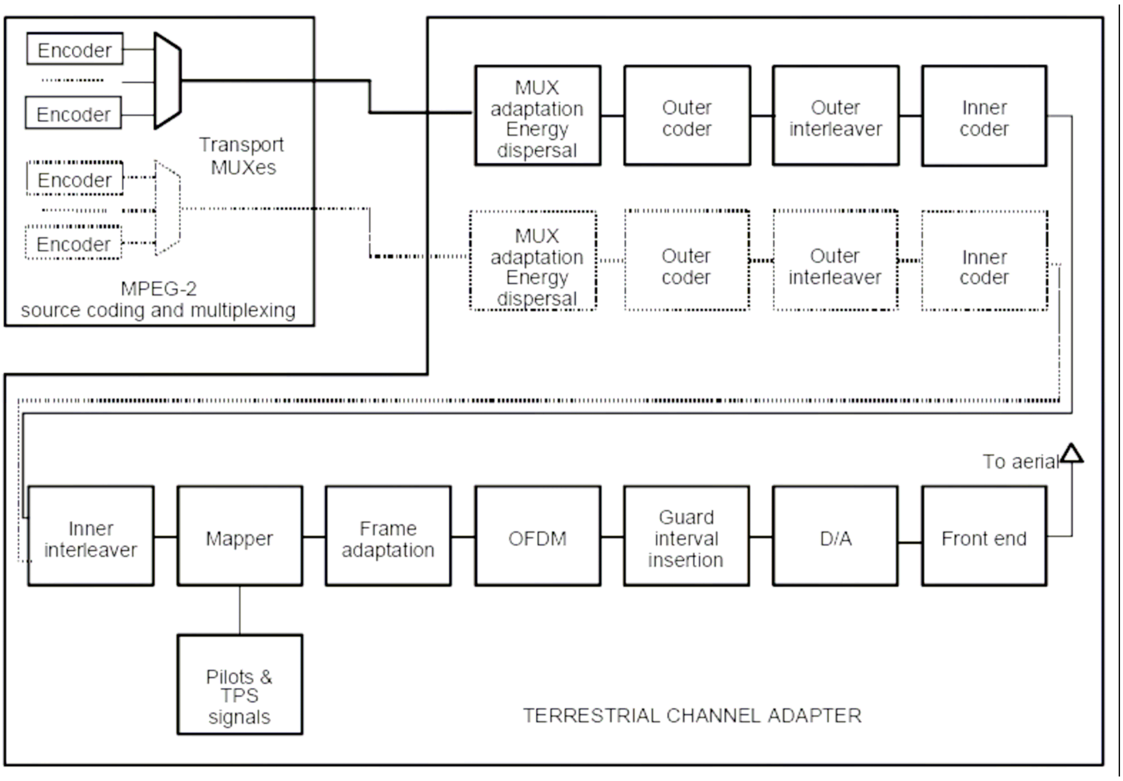


Fig. 2.1 Functional block diagram of the system

2.3. Channel models

The standard *ETSI 300 744* [4] estimates three different scenarios at the reception of the DVB-T TV signals, the reception from a Gaussian channel (AWGN), the reception from a Rician (F1) channel, and the reception through a Rayleigh (P1) channel. These models have been used in our system using the channel models described at *ETSI 300 744*, and defined in **Table 2.2.** This table specifies the delays, amplitudes and phases of each one of the components of the received signal. In our simulator we decided to implement two different channel models, both of them proposed by the standard. In the first one, we implemented the so called Gaussian channel (AGWN), where only the thermal noise affects the signal. Secondly, the next channel model is the so called F1, which is a Rician channel, where 20 copies of the signal are received, each one with different constant delay, power and relative phase. After that we are going to make a deeply description of these channel models, together with other possible channels that could be included in the simulator in future research.

2.3.1. Standard proposed models

The two channel models for fixed reception F1 and portable reception P1 have been generated from the following equations where x(t) and y(t) are input and output signals respectively:

- Fixed reception F1:

$$y(t) = \frac{\rho_0 x(t) + \sum_{i=1}^{N} \rho_i e^{-j\theta_i} x(t-\tau_i)}{\sqrt{\sum_{i=0}^{N} \rho_i^2}} \tag{2.1}$$

Where:

- the first term before the sum represents the line of sight ray;
- N is the number of echoes equals to 20;
- θi is the phase shift from scattering of the i'th path - listed in table **(Table 2.2)**;
- ρi is the attenuation of the i'th path - listed in table **(Table 2.2)**;
- τi is the relative delay of the i'th path - listed in table **(Table 2.2)**;

The Ricean factor K (the ratio of the power of the direct path (the line of sight ray) to the reflected paths) is given as:

$$K = \frac{\rho_0^2}{\sum_{i=1}^{N} \rho_i^2} \tag{2.2}$$

In the simulations a Ricean factor K = 10 dB has been used. In this case:

$$\rho_o = \sqrt{10 \sum_{i=1}^{N} \rho_i^2} \tag{2.3}$$

- Portable reception, Rayleigh fading (P1):

$$y(t) = k \sum_{i=1}^{N} \rho_i e^{-j\theta_i} x(t-\tau_i) \quad \text{where} \quad k = \frac{1}{\sqrt{\sum_{i=1}^{N} \rho_i^2}} \qquad \textbf{(2.4)}$$

θi, ρi and τi are given in table **(Table 2.2)**.

Table 2.2 Amplitudes, delays and phases to the multipath components of the radio channel to DVB-T

Table Attenuation, Phase and Delay Values for F1 and P1			
i	**ρ_i**	**ζ_i[us]**	**σ_i[rad]**
1	0.057662	1.003019	4.855121
2	0.176809	5.422091	3.419109
3	0.407163	0.518650	5.864470
4	0.303585	2.751772	2.215894
5	0.258782	0.602895	3.758058
6	0.061831	1.016585	5.430202
7	0.150340	0.143556	3.952093
8	0.051534	0.153832	1.093586
9	0.185074	3.324866	5.775198
10	0.400967	1.935570	0.154459
11	0.295723	0.429948	5.928383
12	0.350825	3.228872	3.053023
13	0.262909	0.848831	0.628578
14	0.225894	0.073883	2.128544
15	0.170996	0.203952	1.099463
16	0.149723	0.194207	3.462951
17	0.240140	0.924450	3.664773
18	0.116587	1.381320	2.833799
19	0.221155	0.640512	3.334290
20	0.259730	1.368671	0.393889

2.3.2. TU6 model

In 1989, the EU-COST207 project (1984–1988) deeply studied channel propagation models to be used for mobile communications. The Typical Urban 6-paths model (TU6) depicted in **Table 2.3** [10], proven to be representative for the typical mobile reception with Doppler frequency above 10 Hz. Assessment of mobile reception performance requires setting up a reproducible environment. The TU6 has been heavily used both for simulation and for laboratory test (using a channel simulator), and results from numerous field trials highly correlate with the obtained results. Nevertheless, concerns remain in regard to the TU6 suitability for reception with Doppler frequency below 10 Hz (i.e., the pedestrian and indoor reception) suggesting further modelling work.

Table 2.3 Proposed TU6 channel model for mobile communications.

Tap Number	Delay (µs)	Power (lin)	Power (dB)	Doppler Spectrum
1	0,0	0,5	-3,0	Rayleigh
2	0,2	1,0	0,0	Rayleigh
3	0,5	0,63	-2,0	Rayleigh
4	1,6	0,25	-6,0	Rayleigh
5	2,3	0,16	-8,0	Rayleigh
6	5,0	0,1	-10,0	Rayleigh

2.4. C/N calculation for DVB-T Simulator

In this section we are going to explain which type of algorithm has been used for the Carrier to Noise ratio.

For this process we have made the supposition that there aren't pilots inside of each symbol, that is, all the sub carriers are data transporters (use of OFDM training symbols). So first of all we are going to calculate the C/N relationship from the $\left.\frac{Eb}{N_0}\right|_{data\ carrier}$ function **(4.5)**:

$$\left.\frac{Eb}{N_0}\right|_{data\ carrier} = \frac{\overline{s^2(t)}\cdot T_b\cdot\frac{1}{N_c}}{N_0} = \frac{\frac{N_c}{2}\cdot T_b\cdot\frac{1}{N_c}}{\sigma^2 T_m} = \frac{\frac{N_c}{2}\cdot\frac{T_c}{\log_2 M}\cdot\frac{1}{N_c}}{\sigma^2\frac{T_c}{N_t}} = \frac{N_T}{2\sigma^2\log_2 M} \qquad \textbf{(4.5)}$$

Where No is noise power. E bits bit energy, Nc is the sub carrier number (1512 or 6048), N_T total number of carriers (2048, 8192) T_b and T_c are bit and sub carrier duration and σ^2 is the noise spectral density. Now we can calculate the $\left.\frac{C}{N}\right|_{Total}$ **(4.6)** ratio, as:

$$\frac{C}{N}_{Total} = \frac{\overline{s^2(t)}}{N_0 \dfrac{N}{T}} = \frac{\dfrac{N_u}{2}T}{N_0 N_T} = \frac{N_u T}{2\sigma^2 T_m N_T} = \frac{N_u T}{2\sigma^2 \dfrac{T}{N_T} N_T} = \frac{N_u}{2\sigma^2} \qquad \textbf{(4.6)}$$

Where $N_u = N_c + (1.3)^2 N_{pilot}$ for both 2K and 8K modes.

To make the calculation **(4.7)**, we need that the average signals power, in the time domain, being unitary. Knowing that the average received power in the time domain is $C_{rx} = \dfrac{N_u}{N_T}$, we will need to normalize it with $f_{norm} = \dfrac{N_T}{N_u}$, so finally we obtain:

$$N_0 = \frac{c_{rx} f_{norm}}{\dfrac{Nu}{N_T}\dfrac{Eb}{N_0}\log_2 M} = \frac{c_{rx}\dfrac{N_T}{N_u}}{\dfrac{Nu}{N_T}\dfrac{Eb}{N_0}\log_2 M} = \frac{1}{\dfrac{Nu}{N_T}\dfrac{Eb}{N_0}\log_2 M} \qquad \textbf{(4.7)}$$

2.5. Election of the simulation platform

Exits an extensive fan of proposes to improvement in the different subsystems to the transmitter/receiver chain inside the FURIA council. However, the objective of is this work during that project was not to make a unique simulator product of the integration efforts of participants members. The final objective was to offer a technical debate forum where the different strategies brought to the project can be analyzed and compared. Whatever is well known that one of the principal problems found by the expert forums and the standardization institutions is that frequently is very difficult to compare lending, when you don’t know the details of the simulation parameters and the platforms are different. Trying to reduce this dispersion, it has been token the decision of improve simulations with Matlab, tool that makes close module tests easier (without the code) in the different simulators brought by the participants.

Matlab (Simullink, Communications ToolBox etc.) counts with complete simulators of DVB-T with accessibility to the different blocks, we prefer to programme most of them. That decision is done looking for a better control to the system, and to improve the simulator execution velocity

The designed DVB-T block scheme is shown in the figure below **(Fig. 2.3)**.

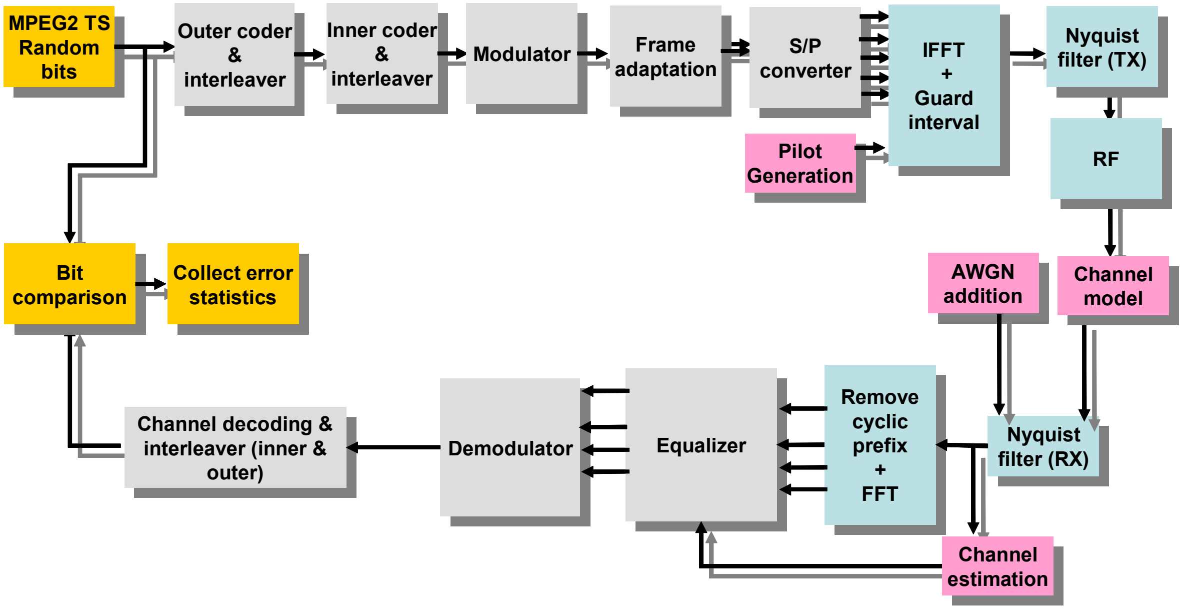


Fig 2.2 TX/RX chain DVB-T system scheme.

The Simulator is designed to make possible changes in anyone of his parts in the analysis proposed to the DVB-T2 mode. These changes will be made with the minimum effort of programmation possible.

In that way, at least all the codifiers and decodifiers, the type and depth of the interleavers, the modulators, the bit assignations mechanisms to the carriers, the carriers potency assignation, the pilot structure and the equalizing channel mechanisms will be reconfigurable.

CHAPTER 3: TRANSMITTER

3.1. Signal Input

As we described in ***Chapter 2*,** the system will use MPEG-2 files as input information. In this first simulation approach we are not going to use real video compressed files, because it's unnecessary for programming purposes. Knowing that each *MPEG-2* [6] file consists on a blocks of 187 bytes, we generate random sequences (with equal probability for 1's and 0's) of this dimension to simulate this behaviour.

According to the standard specifications we have decided to generate the information corresponding to 8 Transport Streams, that is, sequences of 11968 bits (187 bytes · 8 bits/byte · 8 Transport Streams). This decision was taken because every 8 Transport Streams the energy dispersal block is re-started so this is the only way to perform the signal randomization correctly. On other hand, we have developed a function called **gen_TS** to generate these transport streams, which will be described in next section.

3.1.1. Designed functions for signal input

· gen_entrada;

This function generates the random sequence input from *Matlab* library functions *round* and *rand*. *Rand* function generates a uniform distribution of pseudo randomly numbers between 0 and 1. *Round* function rounds each number generated to the nearest integer.

- Input parameters:
    - None.

- Output parameters:
    - fich: vector of 11968 positions of binary values 0 and 1.

3.2. Adaptation and energy dispersion

Base band data is organized in constant length packets, constituting the transport multiplex (TS, Transport Stream) at the MPEG-2 compressor output. Every TS has a constant length of 188 bytes, where the first one, is the synchronism word ("01000111", 47 hex). Every TS is always processed starting by the most significant bit (MSB) from the synchronism byte (bit "0").

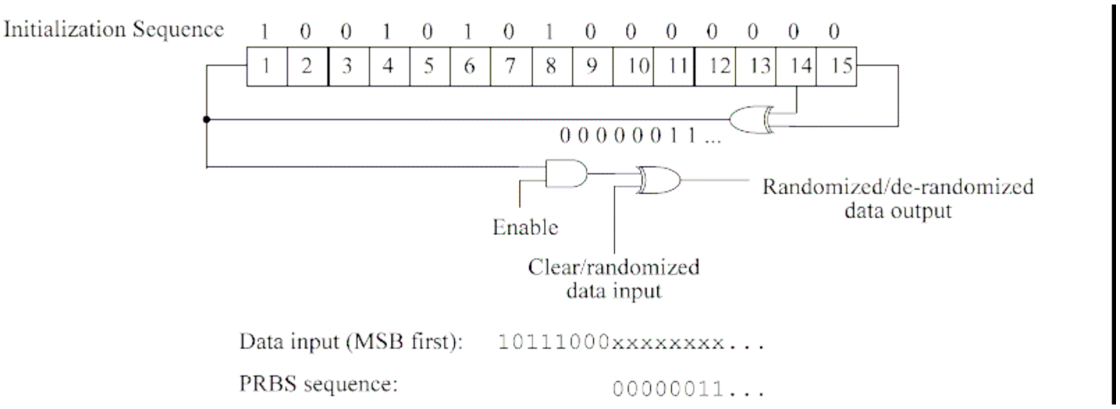

Initialization Sequence
1 0 0 1 0 1 0 1 0 0 0 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 0 1 1 ...
Enable
Clear/randomized data input
Randomized/de-randomized data output
Data input (MSB first): 10111000xxxxxxxx...
PRBS sequence: 00000011...

Fig. 3.1 Scrambler/Discrambler schematic diagram

The main objective of this block is to perform similar number of transitions ("0" and "1"), for this reason all the data bytes are randomized thanks to a pseudo random sequence (PRBS, Pseudo Random Binary Sequence) in a process called energy dispersal. The PRBS sequence is generated by means of a slicing feedback register **(Fig.3.1)**, with the following polynomial generator:

$$1+x^{14}+x^{15} \qquad \textbf{(3.1)}$$

Where the initializing sequence is:"100101010000000", and it should be re-initialized every 8 Transport Streams. As it is shown in figure **(Fig. 3.1)** there are two logical OR-ex (module 2 adding) and an AND door.

A special action is required to inform the des-randomizer that a randomized sequence is incoming. For that reason every 8 TS the synchronism sequence is inverted (the sequence transmitted is "10111000", B8 hex, in front of "01000111", 47 hex). This process is known as "transport flux adaptation" (a $SYNC$ sequence becomes a $\overline{SYNC}$. The first bit after the PRBS output is applied to the first bit (MSB) after the synchronism byte (synchronism byte is never randomized). With the following 7 transport packets, when the synchronism byte arrives the PRBS computation continues working, but it isn't multiplied by the synchronism byte. In consequence the PRBS signal period is (188·8 – 1 = 1503 bytes).

This process must be always active although there isn't any input TS, or if the input data doesn't satisfies the MPEG-2 standard.

3.2.1. Designed functions for adaptation and energy dispersal

In the general system, after the signal input block, the first step is the generation of the Transport Streams. As we have explained before they are composed by 188 bytes packages (the first byte with the synchronization sequence and the 187 of data bytes), where every 187 data bytes the energy dispersion is applied.

· energy_dispersal;

We generate the energy dispersal sequence with a total length of 12032 bits (8 TS), employing XOR logical operators and PRBS (pseudo randomly bit sequence) register.

- Input Parameters:
    - None.

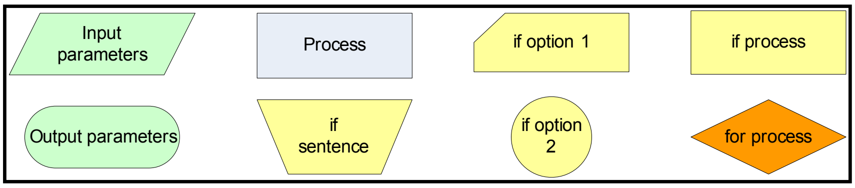
- Output Parameters:
    - w: 12032 bits vector, with the energy dispersal sequence ready to be used with data input.

· gen_TS;

This function has two main goals as we can see in **Fig. 3.2**. On one hand it has to insert synchronism sequences inside the Transport Streams. On the other hand it has to apply the energy dispersal sequence to all the data input without randomizing synchronism sequences.

Both objectives are arranged in one iteration cycle employing *xor (x,y)* function defined in *Matlab* library and combined with the capability of being applied to portions from two different vectors, (as obviously they must have the same length). So with this scheme, we can execute in just one instruction the XOR logical operation between different vector positions, which is really suitable to apply energy dispersion to all the 11968 bits.

Flux diagram key to icons



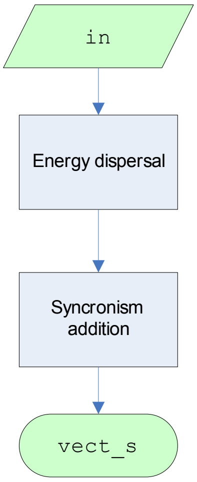


Fig. 3.2 Flux diagram gen_TS.

- Input Parameters:
  - in: 11968 bits input vector.

- Output Parameters:
  - fich: 12032 bits (8TS) vector, with the energy dispersion performed.

3.3. Reed-Solomon Codification

DVB-T system employs an external Reed-Solomon (RS) coder, and an external convolutional interleaver to break the error bursts introduced by the channel (a big number of consecutive errors, that makes packets to don't be able to be recovered. Whatever the synchronism byte is inverted or not, codification is always applied to the data input.

The RS code (204, 188, t=8) is a shortened code obtained from the systematic code RS (255, 239, t=8). It means that data input is a 188 bytes packet, and the output is a 204 bytes, it implies that 16 redundancy bytes are added, and allows 8 bits correction in each packet. In this context the Reed-Solomon code is generated by the following Code Generator Polynomial:

$$g(x) = (x+\lambda^0)(x+\lambda^1)(x+\lambda^2)\ldots(x+\lambda^{15}),\ where\ \lambda = 02_{HEX} \qquad \textbf{(3.2)}$$

And the Field Generator Polynomial:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \qquad \textbf{(3.3)}$$

To implement the shortened code, 51 null bytes are added in the beginning of the 188 information bytes (this makes codeword arrive to the 239 bytes required for systematic coder without modifying the final result). At the coder output, 255 bytes are received, where the first 51 bytes are the null bytes (inherit before), so we finally obtain the 204 bytes transport packets.

3.3.1. Designed functions for Reed-Solomon Codification

Once each Transport Stream is generated, and after the energy dispersal is applied, the first coding scheme is applied, using a Reed-Solomon (204,188).

· codif_RS;

First of all, the 12032 bits input vector is converted in to a 1504 byte vector. This must be performed because Reed-Solomon codes use *Galois Field* to code the data information, so the change to 2^8 base is mandatory. Once the input vector is in bytes, we return to convert it in a *Galois Field* vector, employing the defined in *Matlab* library function *gf (x,m).*

To implement de RS coder, we use the *Matlab* defined function *rsenc (msg, n,k)*. This function allows to use the shortened Reed-Solomon (204,188, t=8)), generated by the mother code RS (255,239). Once the input stream is coded, the data elements are in *Galois Field* representation, so that it must be re-converted to a byte vector for the outer interleaver by means of the *code.x(y)* instruction **(Fig. 3.3)**. That sentence takes out the elements in a *Galois* array to pass them to decimal (bytes).

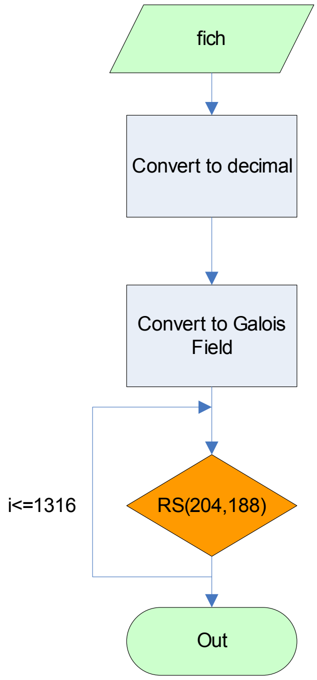


Fig. 3.3 Flux diagram codif_RS.

- Input parameters:
    - fich: 12302 bits vector.

- Output parameters:
    - out: 1632 byte vector codified with RS (204,188).

3.4. Outer Interleaver

The outer interleaver employed is called ‘byte wise convolutional interleaver’. The interleaved data bytes belonging to the Transport Streams are delimited by synchronism bytes (inverted or not) that don’t suffer any alteration. For that reason the transport packet length (204 bytes) at the interleaver output is kept unchanged. In **Fig. 3.4** a block scheme of this procedure is shown:

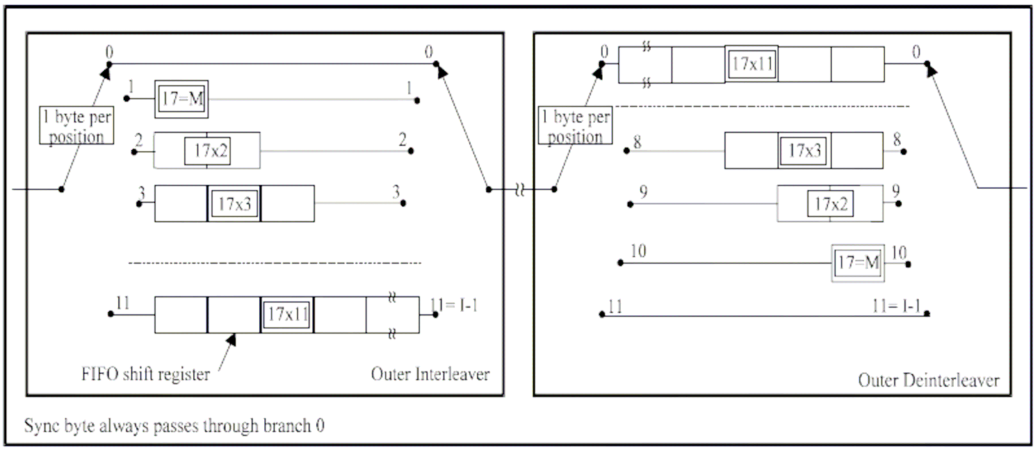
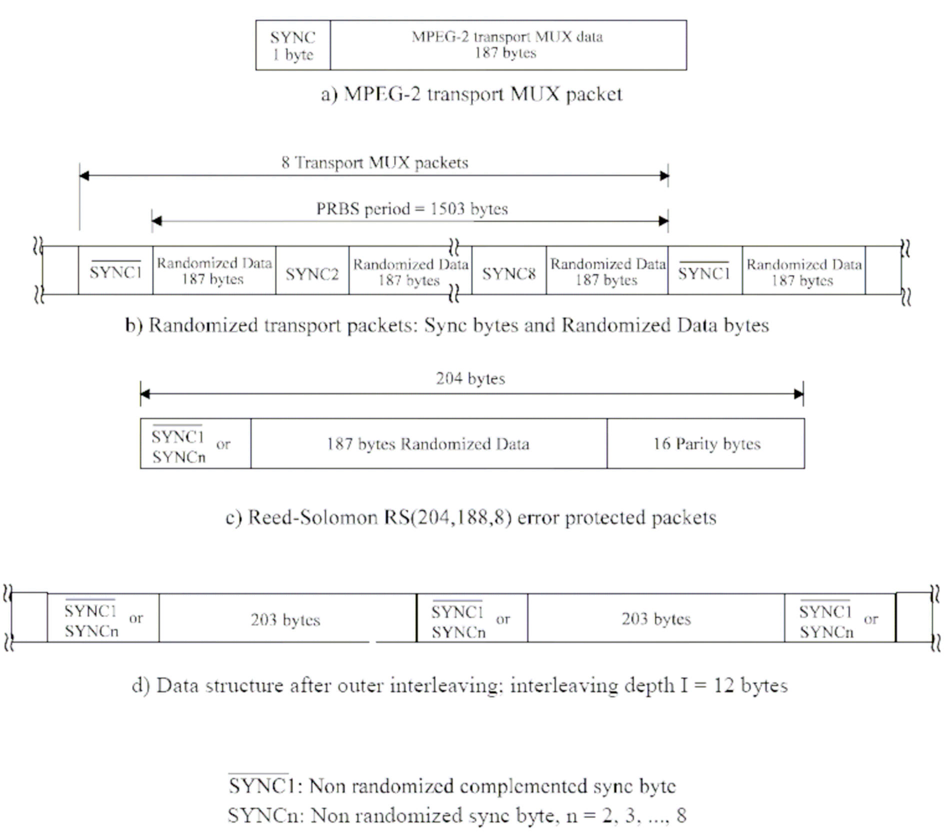


Fig. 3.4 Conceptual Diagram of the outer interleaver and deinterleaver.

The interleaver is composed of I=12 branches, cyclically connected to the input byte stream by the input switch, so consecutive bytes are written in different branches. The input and output switchers are synchronized allowing that when a branch is written, data from this branch is also extracted. In each branch there is a slicing register FIFO of a different depth. So in the j-th branch the depth (or number of cells) is composed by j·M bytes, where M=17=N/I with N=204 and I=12. Every FIFO’s cell contains one byte. The synchronizing byte (inverted or not) is always in the j=0 interleaver branch (it corresponds to no delay).

To completely understand how it works we are going to make a short description of the outer de-interleaver. The outer de-interleaver follows the same FIFO philosophy, however the branch indexes are in the other way around, that is, in the first branch, j=0, the delay is maximum, in particular (17–j)·M. The de-interleaver synchronization is performed routing the first recognized synchronism byte to the branch j=0.

As a result of this process each TS byte gets delayed j·17 positions, so the original TS bytes are distributed between two consecutive packets. At the de-interleaver every byte is delayed (1–j)·17 positions, which makes a total transmission-reception delay of (j+11–j)·17=187. Since this value is constant for all the bytes, it allows retrieving the original order at the receiver. In **Fig. 3.5** the representation of the different data structures and TS at the different described blocks output is shown.


SYNC
1 byte
MPEG-2 transport MUX data
187 bytes
8 Transport MUX packets
PRBS period = 1503 bytes
$\overline{\text{SYNC1}}$
Randomized Data
187 bytes
SYNC2
Randomized Data
187 bytes
SYNC8
Randomized Data
187 bytes
$\overline{\text{SYNC1}}$
Randomized Data
187 bytes
204 bytes
$\overline{\text{SYNC1}}$ or SYNCn
187 bytes Randomized Data
16 Parity bytes
$\overline{\text{SYNC1}}$ or SYNCn
203 bytes
$\overline{\text{SYNC1}}$ or SYNCn
203 bytes
$\overline{\text{SYNC1}}$ or SYNCn

a) MPEG-2 transport MUX packet

b) Randomized transport packets: Sync bytes and Randomized Data bytes

c) Reed-Solomon RS(204,188,8) error protected packets

d) Data structure after outer interleaving: interleaving depth I = 12 bytes

$\overline{\text{SYNC1}}$: Non randomized complemented sync byte
SYNCn: Non randomized sync byte, n = 2, 3, ..., 8

Fig. 3.5 Steps in the process of the different blocks by the DVB-T transmitter.

3.4.1. Designed functions for outer interleaver

As previously described, the outer interleaver block does not employ a static matrix. Instead, a dynamical matrix is used, which is composed by 12 rows with up to 187 variable number of columns; this is the main consideration for the deployment of this function:

· out_interleaver;

This function is composed by an external counter inherited in order to have an initialization index. This index allows initiating all the meaningful variables of the function, like memory matrix or internal counters. The memory matrix is composed by twelve independent vectors, each of them with a different length **(see Fig. 3.4)**, corresponding to the different interleaving registers.

The first time the function is executed, the counter of each vector is initialized to 1 and the memory matrix is padded with zeros. The main body of the interleaver is composed by a loop associated to the input vector and it's also nested to another vector which makes vector rounds. While performing this process the first input element will be moved to the first position of the first vector. The second input element will be moved to the first position of second vector, and so on, that is, the rest of iterations follow this pattern. When one element arrives to a vector that has all its positions with valid values, this element makes all the vector positions get moved one position right-hand, and then the first element inherited in the matrix is sent.

At the end of the first iteration of the function, there are many vectors without all of their positions filled, so when the second call to the function is performed, all the internal counters from the registers must be initialized with the corresponding value that grants the correct operation of the function. For third and subsequent function executions, all vectors are completely filled from the beginning, so the interleaving matrix is initialized with the same parameters that in the last executed iteration. Also each individual counter is initialized with the maximum length value of its corresponding vector **(Fig. 3.6)**.

Each time the block of 8 TS has passed the interleaver, we save the final state of each register in a dynamic matrix for future function executions. The output vector is converted to bits (see ***Chapter 3.3.***). With this procedure we obtain the bit stream required for the next transmission system block.

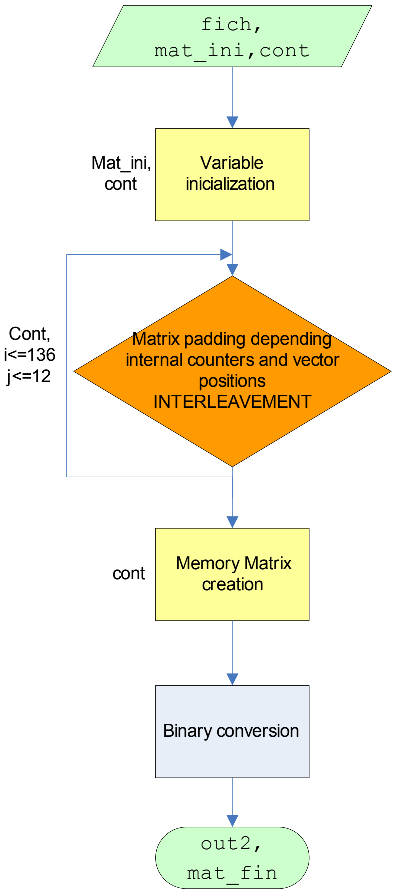


Fig. 3.6 Flux diagram out_interleaver.

- Input parameters:
    - fich: 1632 bytes vector.
    - mat_ini: matrix which contains the memory registers of the previous iterations.

- cont: external counter. It specifies the number of executions of the function.

- Output parameters:
    - out2: 13056 bits vector (8TS codified).
    - mat_fin: register states at the end of each function execution.

3.5. Inner Coder

For the inner coder, punctured convolutional codes are employed. These codes are based in a mother convolutional code of rate 1/2 with 64 states (K=7). This will allow selection of the most appropriate level of error correction for a given service or a data rate in either non-hierarchical or hierarchical transmission mode (see ***chapter 2.2.***).

We can see in **Fig. 3.7** the structure of the coder. It is oriented to bit level and it has two outputs: X and Y. They are obtained by combining (module 2 addition) data input with the branches allocated behind a group of slicing registers. In the figure scheme **(Fig. 3.7)** we can appreciate the generator polynomials from each of the two coder outputs (1111001, 171 octal for X, and 1011011, 133 octal for Y). At the beginning of each super-frame, the MSB from the synchronism codeword must be allocated at the coder input.

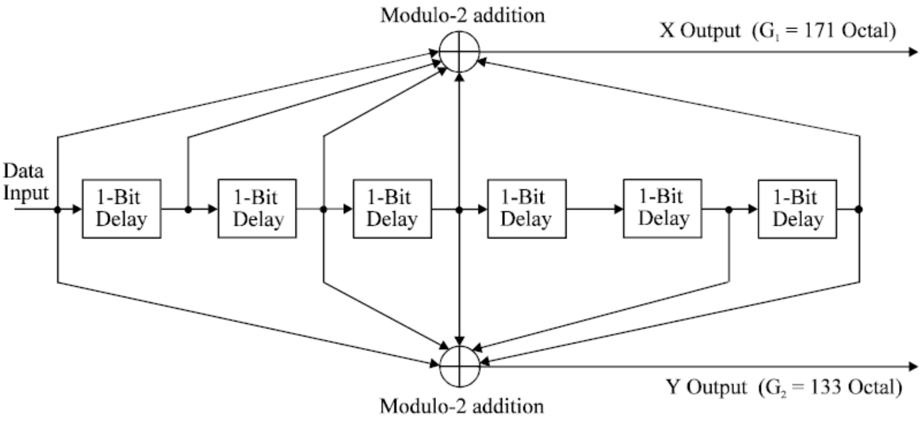


Fig. 3.7 Convolutional coder scheme.

The puncturing pattern with the different options is indicated in **Table 3.1**. As the coding tax grows up, the error protection is smaller and channel capacity is greater (less redundancy is transmitted). To establish which is the most suitable value for each transmission, a definition of the covering area for a certain power is required. If a two level hierarchical transmission is performed, each of both parallel channel coders can employ different codes.

Table 3.1 Puncturing Patterns and Transmitted sequences.

Code Rates	Puncturing Pattern	Transmitted sequence
1/2	X:1 Y:1	X_1 Y_1
2/3	X:10 Y:11	X_1 Y_1 Y_2
3/4	X:101 Y:110	X_1 Y_1 Y_2 X_3
5/6	X:10101 Y:11010	X_1 Y_1 Y_2 X_3 Y_4 X_5
7/8	X:1000101 Y:1111010	X_1 Y_1 Y_2 Y_3 Y_4 X_5 Y_6 X_7

3.5.1. Designed functions for Inner Coder

Following the coder explanation, this function develops a convolutional coder with the following possible different code rates: 1/2, 2/3, 3/4, 5/6 and 7/8. That block is implemented in one function, explained below.

· inner_coder2;

At the beginning, all the coder puncturing patterns are defined. After this, the next step is to generate the Trellis polynomial for the convolutional coder. To do so, we employ the *poly2trellis (ConstraintLength, CodeGenerator)* function defined in the *Matlab* library. Once the Trellis polynomial is generated and the state diagram is defined, the output from the outer interleaver is coded with the function *convenc (msg,trellis,puncpatt,init_state)* **(Fig. 3.8)**. This function allows setting the initial convolutional coder state, also allowing its use to code an input vector in different iterations without initializing the coder in every iteration.

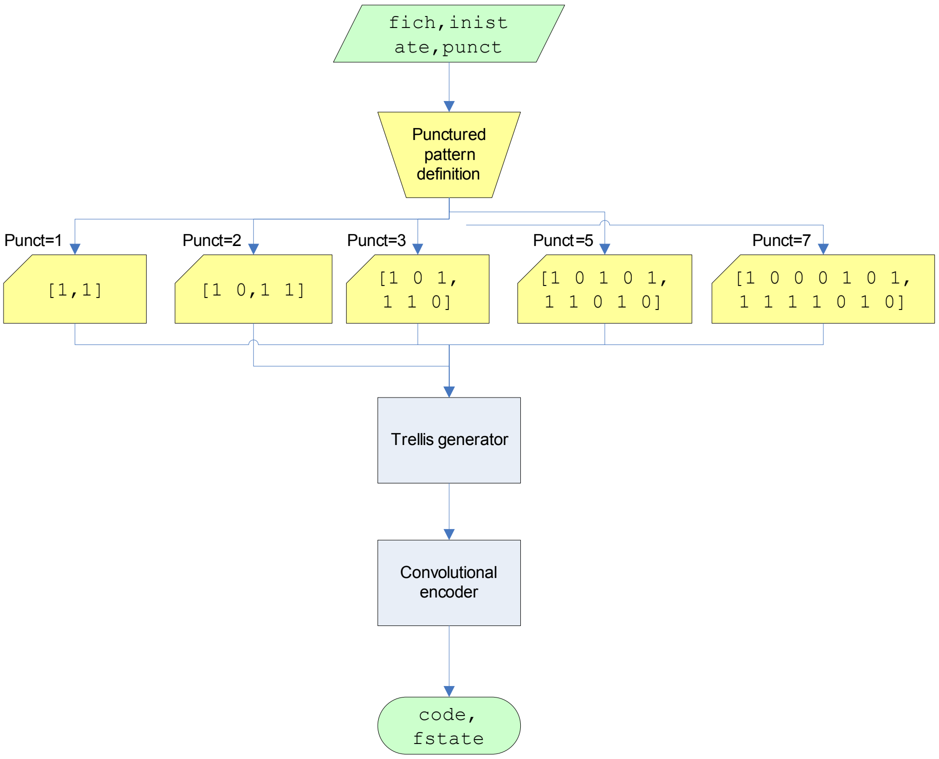


Fig. 3.8 Flux diagram inner_coder2.

- Input Parameters:
  - fich: 13056 bits vector.
  - inistate: initialises convolutional coder in a certain state
  - punct: shows the punctured pattern employed by the coder.

- Output Parameters:
  - code: variable length vector, it depends on the code rate employed but it corresponds to the 8 TS coded.
  - fstate: final state from the convolutional coder. Is used to store the next instate value.

3.6. Inner interleaving

At the inner interleaver input we find sequences of bits in series. This interleaving is composed by a bit-wise interleaver followed by a symbol interleaver both based in blocks[1] **(Fig. 3.9)**.

[1] These blocks are the conjunction of two different subblocks. We treated both in two different sections.

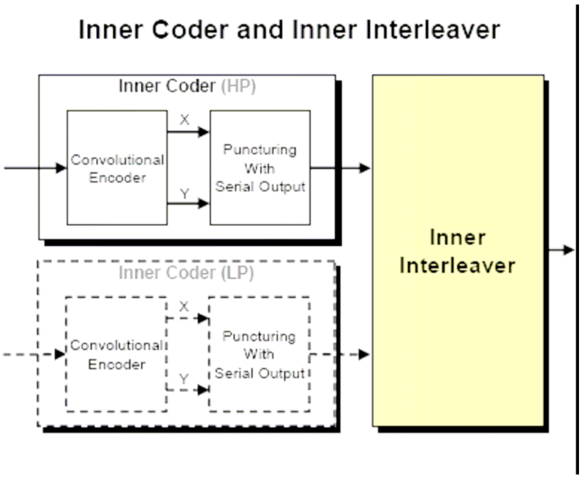


Fig. 3.9 Inner coding and interleaver for hierarchical and non-hierarchical mode.

3.6.1. Bit wise interleaving

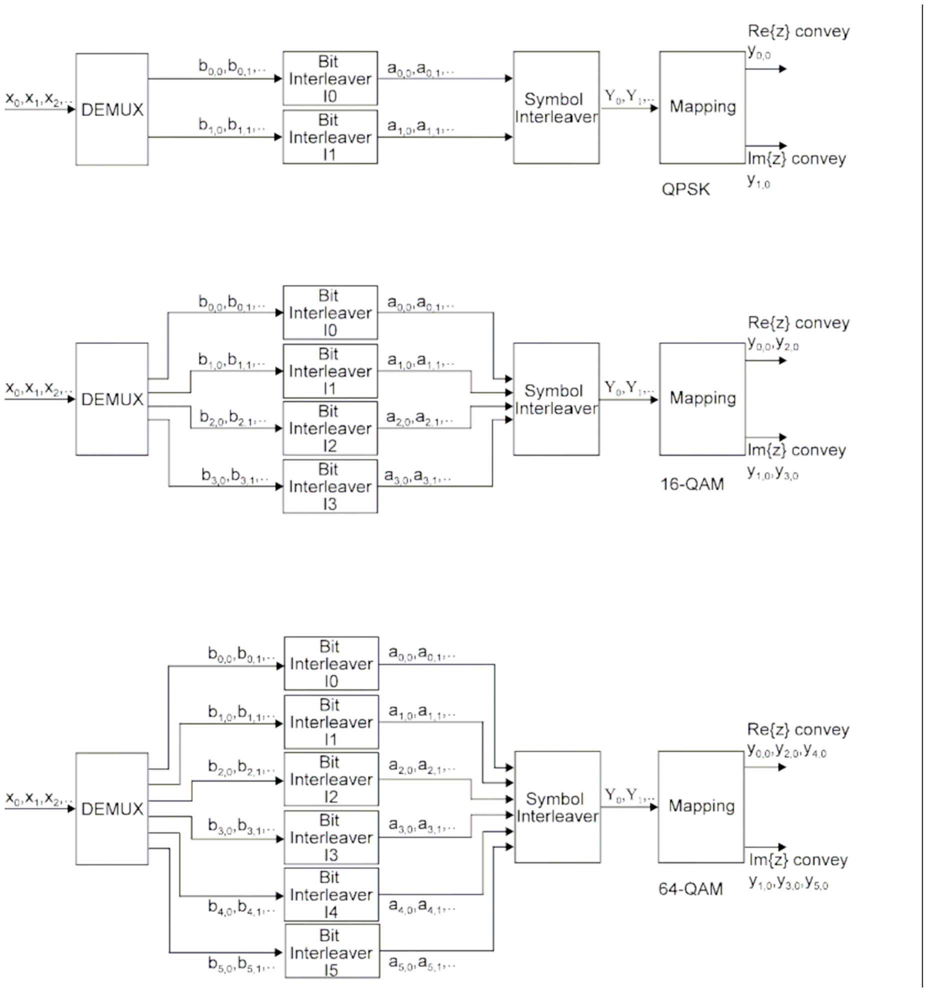
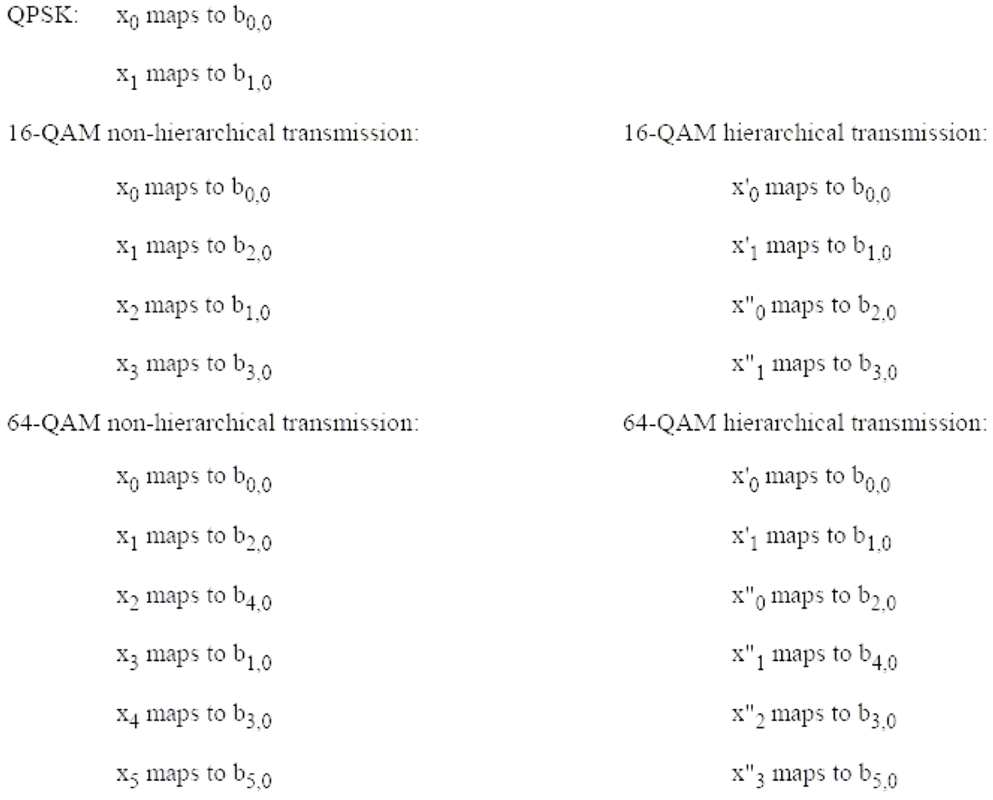


Fig. 3.10 Mapping of input bits onto output modulation symbols, non-hierarchical transmission.

Figure **(Fig. 3.10)** schematizes how the interleavement for the different modulation patterns in the non-hierarchical mode is distributed. The hierarchical mode scheme is also defined in ETSI technical documentation [4], however, as commented in ***Chapter 2,*** to delve deep into the hierarchical mode goes out of the scope of this project.

The interleaver is oriented to symbol size, which is defined by the constellation employed. Input is composed by a maximum of two bit sequences, only one for the non hierarchical mode. In this latter mode, input bits are divided into v sequences, where v is the number of bits/symbol of each modulation (v=2 for QPSK, 4 for 16 QAM and 6 for 64 QAM). In figure **(Fig. 3.11),** the mapping of input bits to the different interleaver inputs is shown.



QPSK: x_0 maps to $b_{0,0}$

x_1 maps to $b_{1,0}$

16-QAM non-hierarchical transmission:

x_0 maps to $b_{0,0}$

x_1 maps to $b_{2,0}$

x_2 maps to $b_{1,0}$

x_3 maps to $b_{3,0}$

16-QAM hierarchical transmission:

x'_0 maps to $b_{0,0}$

x'_1 maps to $b_{1,0}$

x''_0 maps to $b_{2,0}$

x''_1 maps to $b_{3,0}$

64-QAM non-hierarchical transmission:

x_0 maps to $b_{0,0}$

x_1 maps to $b_{2,0}$

x_2 maps to $b_{4,0}$

x_3 maps to $b_{1,0}$

x_4 maps to $b_{3,0}$

x_5 maps to $b_{5,0}$

64-QAM hierarchical transmission:

x'_0 maps to $b_{0,0}$

x'_1 maps to $b_{1,0}$

x''_0 maps to $b_{2,0}$

x''_1 maps to $b_{4,0}$

x''_2 maps to $b_{3,0}$

x''_3 maps to $b_{5,0}$

Fig. 3.11 Bit mapping to interleaver.

After this phase, each sub-stream from the demultiplexer is processed by a separate bit interleaver. Therefore, there are up to six interleavers (I0 to I5) depending on v, and always starting by I0. I0 and I1 are used for QPSK, I0 to I3 for 16 QAM and I0 to I5 for 64 QAM.

The interleaving block size (126 bits) is always the same for all the interleaving blocks; however the interleaving sequence is different in each block. The size of the blocks implies that this process is repeated a certain number of times for each OFDM symbol, as illustrated in **Table 3.2**. Specifically, the process is repeated 12 times for every OFDM symbol in 2K mode (12·126=1512), and 48 times in 8K mode (48·126=6048).

Table 3.2 Interleaving bit data.

	2K	8K
Number of data carriers	1512	6048
Total bits in a OFDM symbol	v x 1512	v x 6048
Number of bit interleavers	v	v
Interleaver repeat (number of iterations by symbol)	v x 1512 / v x 126 = 12	v x 6048/v x 146 = 48

The bit interleaver outputs are grouped to form the symbols, making that each v-bits symbol is composed by one of each interleaving blocks, being the I0 output the MSB.

For each bit interleaver, the input bit vector is defined by:

$$B(e) = (b_{e,o}, b_{e,1}, b_{e,2}, ..., b_{e,125}) \quad \textbf{(3.4)}$$

Where e ranges from 0 to v–1.

The interleaved output vector $A(e) = (a_{e,0}, a_{e,1}, a_{e,2}, ..., a_{e,125})$ is defined by:

$$a_{e,w} = b_{e,He(w)} \qquad w = 0,1,2,...,125 \quad \textbf{(3.5)}$$

Where H_e(w) is a permutation function which is different for each interleaver. For each interleaver, H_e(w) is defined as follows:

I0:	H_0(w)=w	
I1:	H_1(w)= (w+63)	mod 126
I2:	H_2(w)= (w+105)	mod 126
I3:	H_3(w)= (w+42)	mod 126
I4:	H_4(w)= (w+21)	mod 126
I5:	H_5(w)= (w+84)	mod 126

Finally, the overall output vector is:

$$y'_w = (a_{0,w}, a_{1,w}, ..., a_{v-1,w}) \quad \textbf{(3.6)}$$

3.6.1.1. Designed functions for Bit wise interleaver

· inner_interleaver;

In the inner interleaver, we have to define the six possible branches (as we explained in ***Chapter 3.6.1.***). Depending on which modulation scheme is being performed, only 2, 4 or the 6 de-multiplexer branches will be actually used. In each branch we must introduce 126 bit length data packets as defined by ETSI for this block **(Fig 3.12)**.

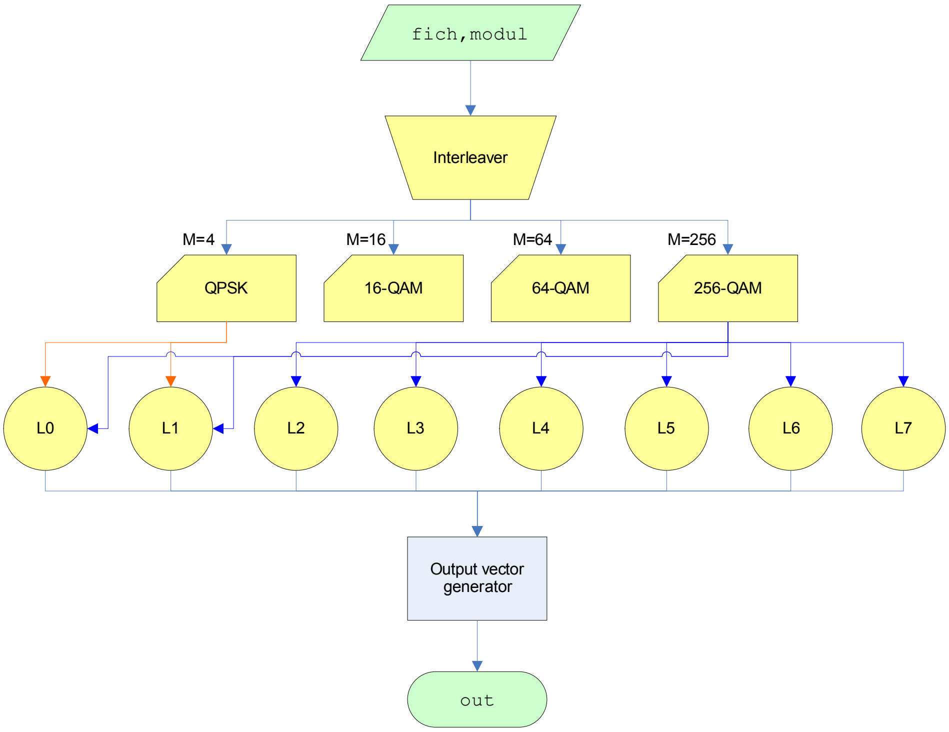


Fig. 3.12 Flux diagram inner_interleaver.

- Input Parameters:
    - fich: variable length vector; depending on the code rate used in convolutional coder.
    - modul: modulation.
    - memoria: previous memory from the input vector

- Output Parameters:
    - out: variable length vector; always 126 multiple.
    - mem_out: memory where the rest of the 126 “out” bits are saved.

3.6.2. Symbol interleaver

The objective of the symbol interleaver is to map the v bit words, that words would be regrouped in blocks at the output of the interleaver. This block size is calculated to assure that the data gets uniformly spread along the 1512 (2K) or 6048 (8K) active carriers from each OFDM symbol.

In mode 2K there are 12 groups of 126 words at the interleaver output; they are read sequentially generating a vector Y'= (y'0, y'1, y'2, …, y'1511). The same applies for 8K mode although this time Y'=(y'0, y'1, y'2, …, y'6047), which is assembled into 48 groups of 126 words.

The interleaved vector Y = (y_o, y_1, y_2, …, y_{Nmax-1}) is defined by:

$$\begin{aligned} y_{H(q)} &= y'_q \ \textit{for even symbols for } q = 0,\dots,N_{\max}-1 \\ y_q &= y'_{H(q)} \ \textit{for odd symbols for } q = 0,\dots,N_{\max}-1 \end{aligned} \quad \textbf{(3.7)}$$

Where N_{max}=1512 in the 2K mode and N_{max}=6048 in the 8K mode.

H (q) is a permutation function defined by the following:

A word R'$_i$ containing (Nr – 1) bits is defined, with $N_r = \log_2 M_{\max}$, where $M_{\max} = 2048$ in the 2K mode and $M_{\max} = 8192$ in the 8K mode. R'$_i$ may take the following values:

$$\begin{array}{ll} i = 0,1: & R'_i[N_r-2, N_r-3,\dots,1,0] = 0,0,\dots,0 \\ i = 2: & R'_i[N_r-2, N_r-3,\dots,1,0] = 0,0,\dots,1 \\ 2 < i < M_{\max} & \{ R'_i[N_r-3, N_r-4,\dots,1,0] = R'_{i-1}[N_r-2, N_r-3,\dots,2,1]; \\ & \textit{in the } 2K \bmod e: R'_i[9] = R'_{i-1}[0] \oplus R'_{i-1}[3] \\ & \textit{in the } 8K \bmod e: R'_i[11] = R'_{i-1}[0] \oplus R'_{i-1}[1] \oplus R'_{i-1}[4] \oplus R'_{i-1}[6] \} \end{array} \quad \textbf{(3.8)}$$

A vector R$_i$ is derived from the vector R'$_l$ by the bit permutations given in tables **(Table 3.3a and Table 3.3b)**.

Table 3.3a Bit permutations for the 2K mode.

R'$_i$ bit positions	9	8	7	6	5	4	3	2	1	0
R$_i$ bit positions	0	7	5	1	8	2	6	9	3	4

Table 3.3b Bit permutations for the 8K mode.

R'$_i$ bit positions	11	10	9	8	7	6	5	4	3	2	1	0
R$_i$ bit positions	5	11	3	0	10	8	6	9	2	4	1	7

A schematic block diagram of the algorithm used to generate the permutation function is represented in **Fig. 3.13** for the 2K mode and **Fig. 3.14** for 8K mode.

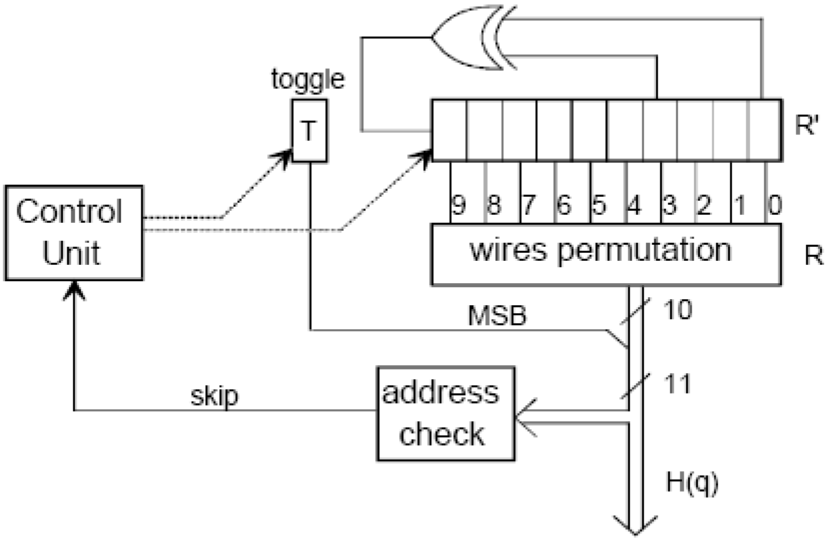


Fig. 3.13 Symbol interleaver address generation scheme for the 2K mode.

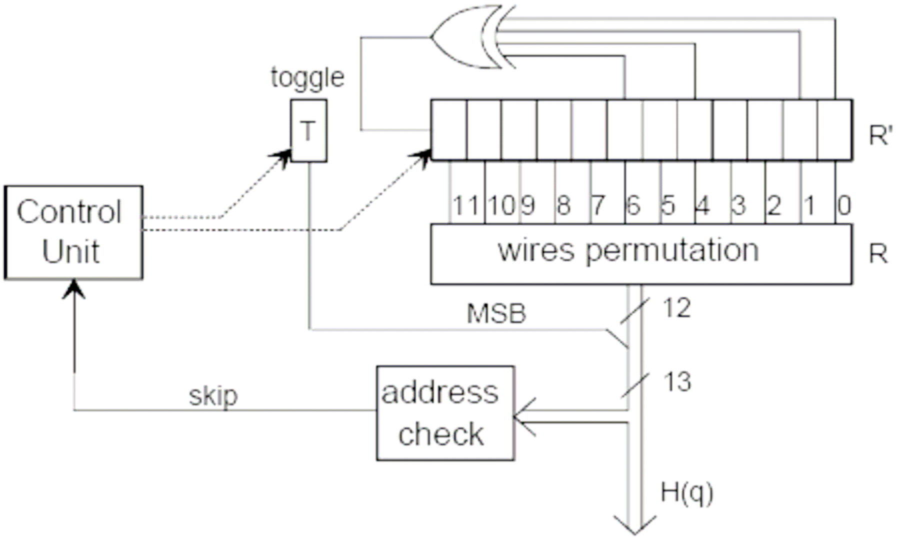


Fig. 3.14 Symbol interleaver address generation scheme for the 8K mode.

In a similar way, y is made up of v bits, having:

$$y_{q'} = \left(y_{0,q'}, y_{1,q'}, \dots, y_{v-1,q'}\right) \quad \textbf{(3.9)}$$

Where q' is the symbol number at the output of the symbol interleaver, we can see an example in **Table 3.4**.

Table 3.4 Example for symbol interleaver.

Input (y')	0	1	2	3	4	5	6	7	8	9
H(q)	3	4	0	6	9	8	1	7	2	5
Output (y)	2	3	8	9	0	1	4	7	6	5
Even symbols					Odd symbols					

3.6.2.1. Designed function for Symbol interleaver

The symbol interleaver employees a displacement register vector, previously defined as H(q). Hereafter, the functioning of this displacement is described.

The displacement makes even positions from the input vector get into other even positions from the output vector, but randomly ordered following the swapping function H(q). In the same way, odd positions get into other odd positions from the output vector, but randomly ordered following the same H(q).

· H_generator;

This function generates the positions' vector H(q) **(Fig. 3.15)**. This vector is generated following the model described in section ***3.6.2.***.

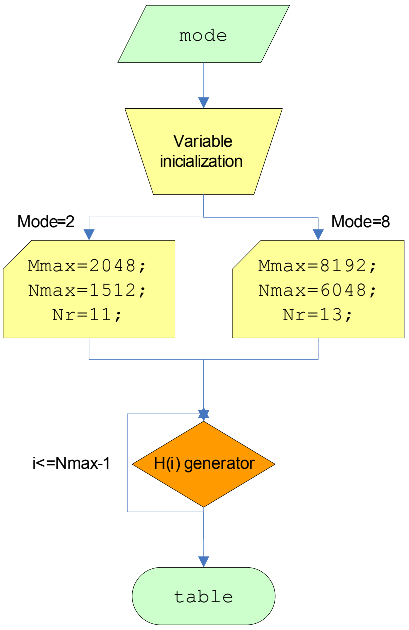


Fig. 3.15 Flux diagram H_generator.

- Input parameters:
  - mode: system mode (2K or 8K).

- Output parameters:

- table: 1512 vector positions, corresponding to the displacement register, H(q).

· symbol_interleaver;

The symbol interleavement converts the input vector (formed by bits) into a symbol vector (decimal figures), which values depend on the modulation used (f.e. symbols between 0 and 3 corresponding to the four symbols of the QPSK modulation). Immediately afterwards, symbols get interleaved by employing the H(q) function provided by `H_generator` **(Fig 3.15)**.

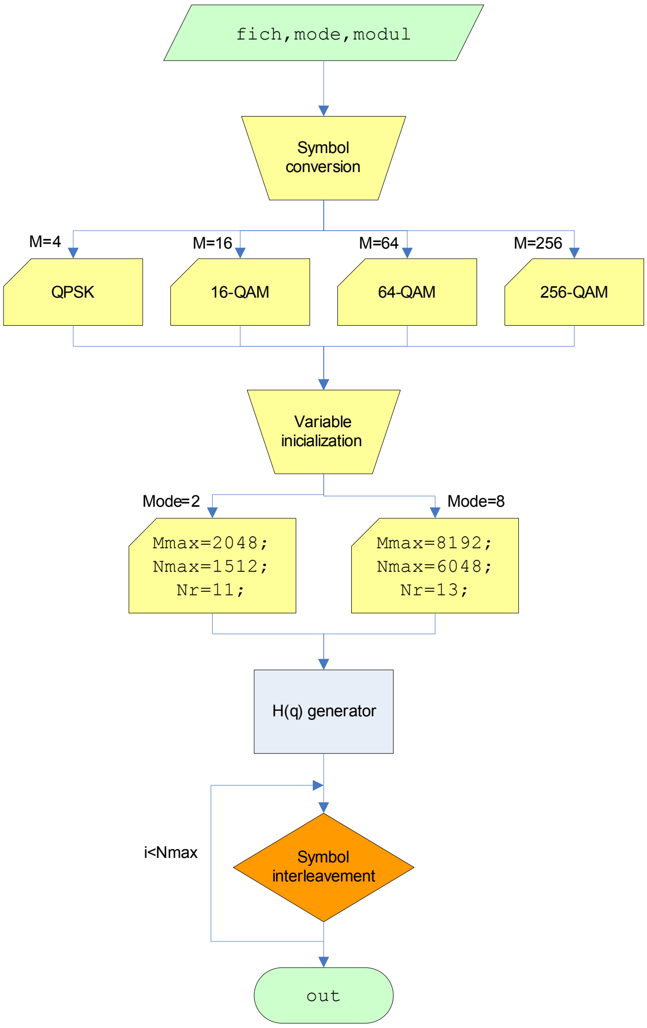


Fig. 3.16 Flux diagram symbol_interleaver.

- Input parameters:
  - fich: input file.
  - mode: system mode (2K or 8K).
  - modul: modulation employed

- Output parameters:
  - out: variable length symbol vector, with the correct format for the mapping block.

CHAPTER 4: MODULATION AND CHANNEL

In this chapter we are going to explain all the relative process of modulation and transmission aspects, as for example: the mapped symbols, the OFDM modulation, the used channels and their corresponding channel estimations, and finally the OFDM frame structure.

4.1. Constellation and mapping

The v bits from the q' word modulates the q' carrier from the 1512 (2K) or 6048 (8K) active carriers of every OFDM symbol. To perform it the data stream of the symbol interleaver is mapped in two signals I/Q. For example 2 bytes (16 bits) mapped in a QPSK constellation, became eight symbols. The serial/parallel converter sends bits to the I/Q branches, so the even bits go to the *I* branch and odd bits go to the Q branch, so bit rate at each branch is half the bit rate before parallel conversion.

In the hierarchical mode, the first two bits have the greatest priority; while the rest are low priority bits. If constellation is decoded by using QPSK, only high priority bits are going to be retrieved. If all bits shall be retrieved is very important to correctly identify the constellation. The distance between different constellation points is characterised by the modulation parameter α (quotient of the length of two constellation points in adjacent quadrants and distance between two points of the same quadrant). The standard allows three possible values for this parameter: 1 (uniform modulations), 2 or 4 (non uniform modulations) **(**see **Fig. 4.2** and **Fig. 4.3)**.

In figure **(Fig. 4.1)** it's represented the constellations for α=1.

At the inner interleaver output v bits words are obtained, which are mapped in a complex number z {n +jm} where *n* and *m* are:

QPSK
n{-1,1}, m{-1,1}

16-QAM (non-hierarchical and hierarchical with α=1)
n{-3,-1,1,3}, m{-3,-1,1,3}

Non uniform 16-QAM with α=2
n{-4,-2,2,4}, m {-4,-2,2,4}

Non-uniform 16-QAM with α=4
n{-6,-4,4,6}, m {-6,-4,4,6}

64QAM(non hierarchical and hierarchical with α=1)
n{-7,-5,-3,-1,1,3,5,7}, m {-7,-5,-3,-1,1,3,5,7}

Non-uniform 64-QAM with α=2
n{-8,-6,-4,-2,2,4,6,8}

Non-uniform 64-QAM with α=4
n{-10,-8,-6,-4,4,6,8,10}, m {-10,-8,-6,-4,4,6,8,10}

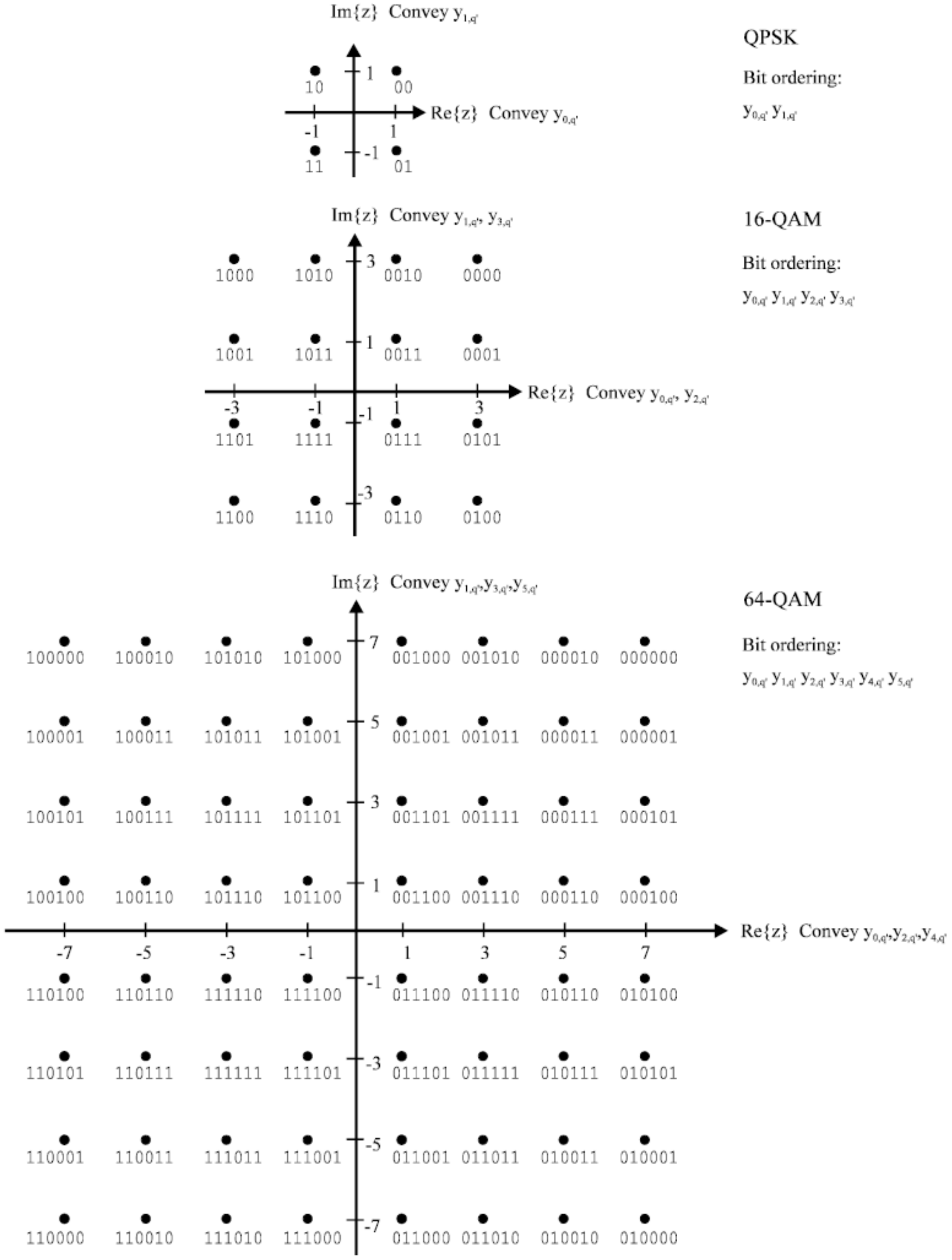


Fig. 4.1 Bit to symbol mapping for the different modulations (α=1 for hierarchical and no-hierarchical transmission mode).

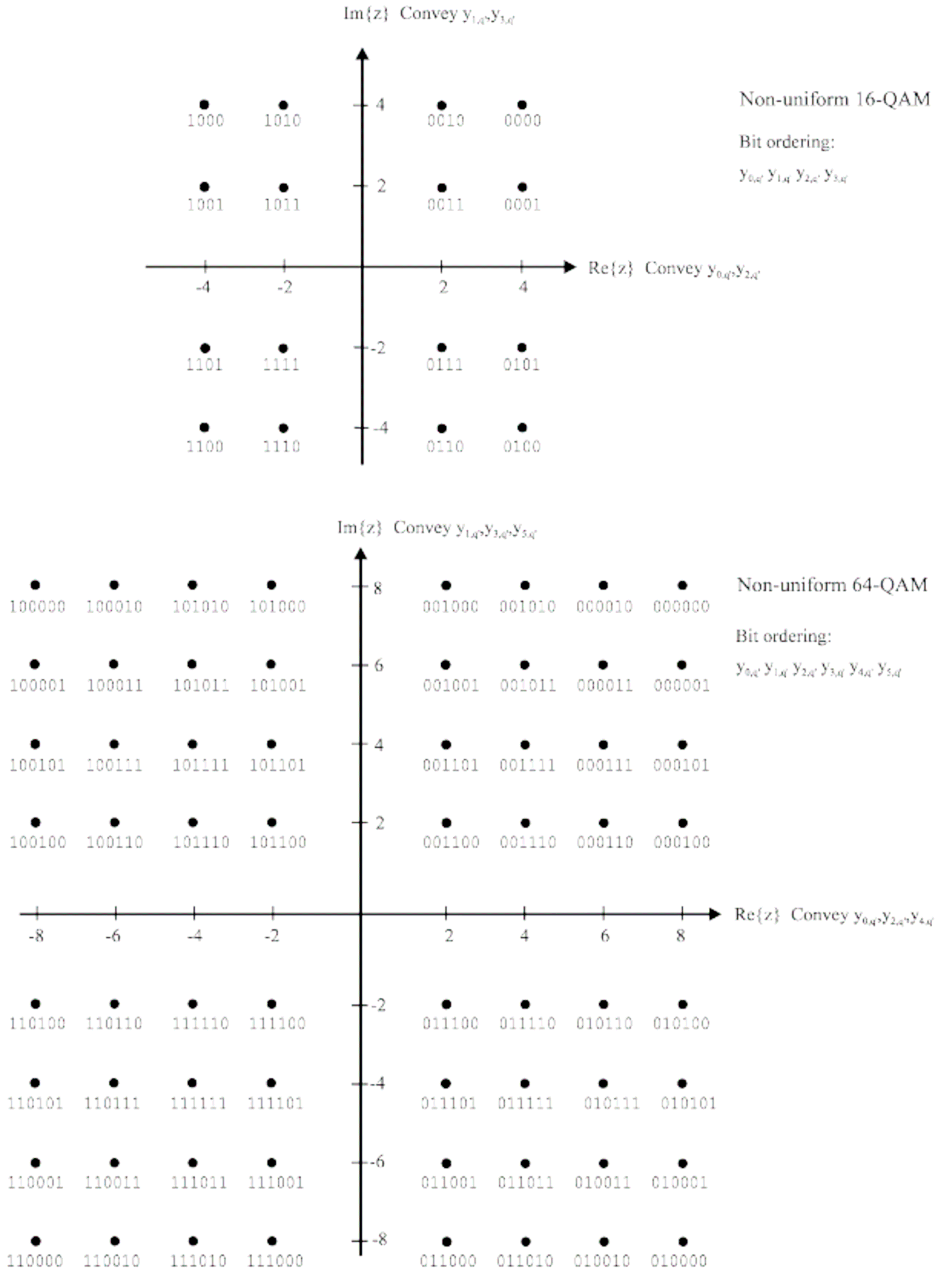


Fig. 4.2 Non-uniform 16-QAM and 64-QAM mappings with α = 2.

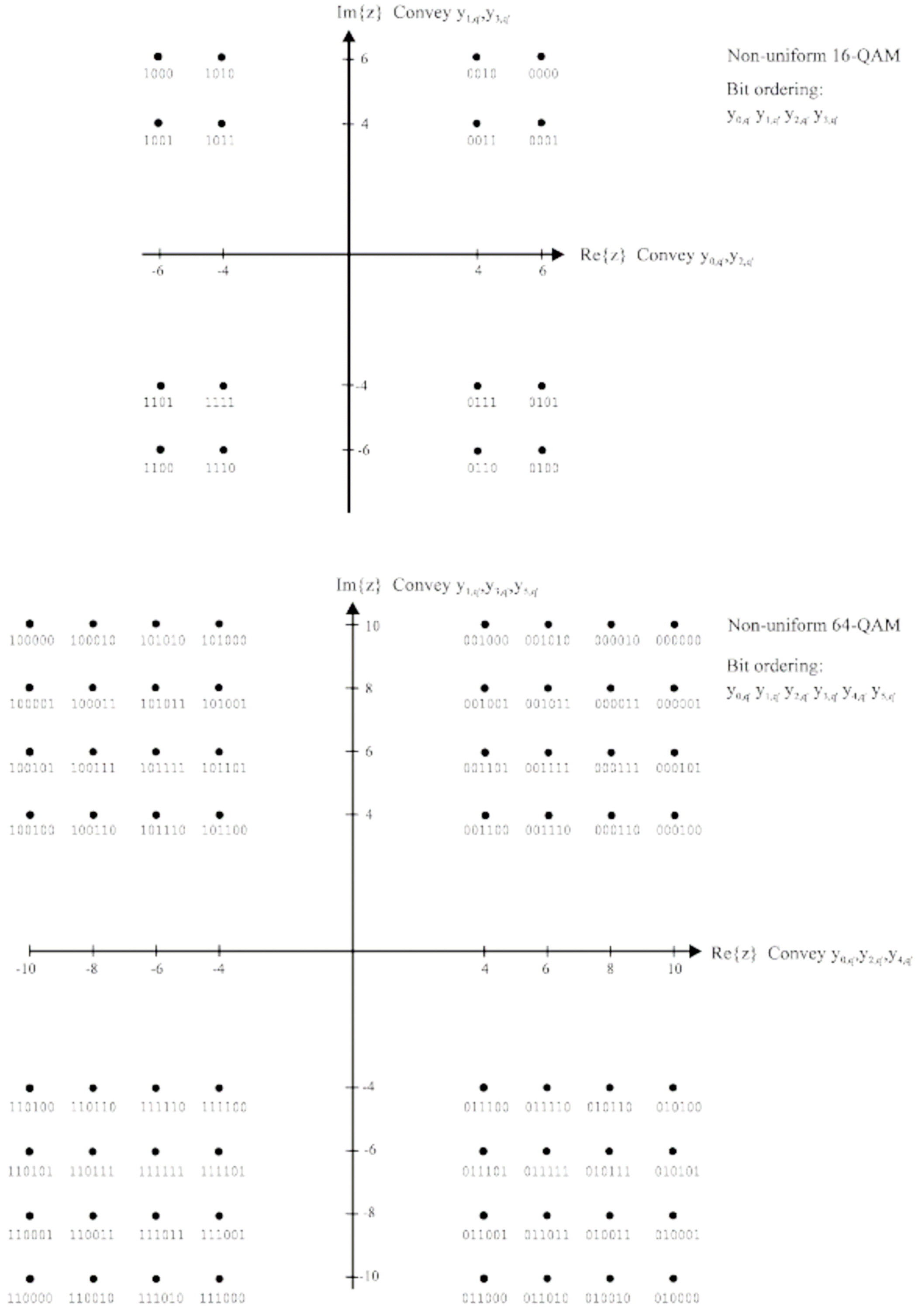


Fig. 4.3 Non-uniform 16-QAM and 64-QAM mappings with α = 3.

4.1.1. Designed functions of constellation and mapping

4.1.1.1. Transmission

· **`determineConstellation;`**

This function determines the actual constellation and then generates the vector with the constellation values **(Fig. 4.4)**. To perform it, first of all, all the values are ordered following the standard specifications with Matlab function *quamod(x,M)* that returns the phase and amplitude values of every element, as can be appreciated in figure **(Fig. 4.1)**. Also a normalization factor is applied on every constellation element, depending on the modulation used to guarantee that the mean envelope power of the modulated signal is equal to one.

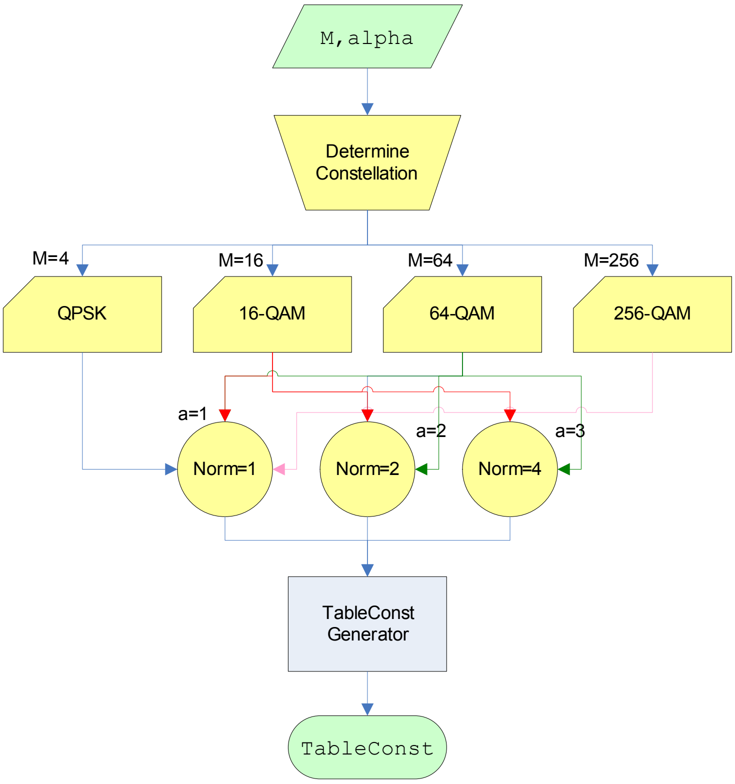


Fig. 4.4 Flux diagram determineConstellation.

- Input parameters:
  - M: modulation
  - alpha: in the hierarchical mode, parameter for constellation tuning.

- Output parameters:
  - tableConst: vector containing the constellation values depending on M and alpha (starting in the zero symbol of the constellation).

4.1.1.2. Reception

· **`Detector;`**

The objective of this function is returning the value of the different transmitted symbols. Initially it has to be determined in which quadrant is the symbol located, then it is compared with the constellation, choosing the one that is at close distance from the received ones **(see Fig. 4.5)**.

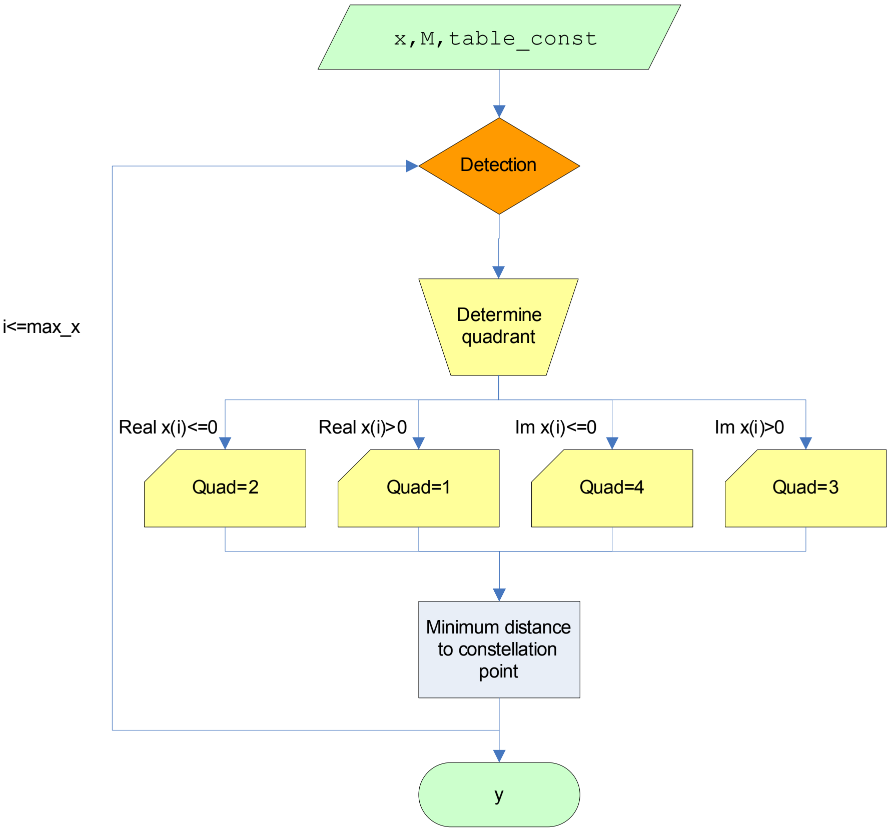


Fig. 4.5 Flux diagram detector.

- Input parameters:
  - x: symbol to symbol detected vector.
  - M: used modulation.
  - table_const: in the hierarchical mode, parameter for constellation tuning.

- Output parameters:
  - y: vector containing the symbols sent.

4.2. OFDM Modulation

The main objective of this part is to distribute the binary information stream between an elevated number of carriers, making that each of them manage a low data bit rate in comparison to the total rate (initial bit rate is divided by the number of carriers employed). The symbol duration *Tu* grows up comparing to the case of modulating only one carrier. With this, the signal becomes stronger in front of multipath interferences because the eco delay is smaller compared with the symbol duration.

Separation between adjacent carriers is *1/Tu* so the orthogonally is granted between carriers. This means that the demodulator of one carrier isn't affected by the signal of the others, so there is no ISI (Inter Symbol Interference) between carriers, although there isn't an explicit filter and spectrums are overlapped.

To make stronger, even more, the transmitted signal in front of the eco delays, time duration is expanded adding a guard interval, Δ, so the total symbol duration is $Ts=Tu+\Delta$. Guard interval is a cyclical continuation from the useful part of the symbol and it's inherited before it. The system allows four different guard period values, with a duration defined by useful time fractions Δ/Tu=1/4, 1/8, 1/16 and 1/32 **(Table 4.1 and Table 4.2)**.

If signal is received from two different ways with a relative delay between them, and if the delay is smaller than the guard interval, both can be constructively added. Receiver ignores the received signal along the guard period (also known as cyclic prefix). However this guard interval represents a channel transmission capacity loose.

Table 4.1 Numerical values for the OFDM parameters for the 8K and 2K modes for 8 MHz channels.

Parameter	8K mode	2K mode
Number of carriers K	6817	1705
Value of carrier number K_{min}	0	0
Value of carrier number K_{max}	6816	1704
Duration Tu	896 µs	224 µs
Carrier Spacing 1/Tu	1116 Hz	4464 Hz
Spacing between carriers $K_{min\ and}$ K_{max} (K-1)/Tu	7.61 MHz	7.61 MHz

Table 4.2 Duration of symbol part for the allowed guard intervals for 8 MHz channels.

Mode	8K mode				2K mode			
Guard interval Δ/Tu	1/4	1/4	1/16	1/32	1/4	1/8	1/16	1/32
Duration of symbol part Tu	8192T 896μs				8192T 896μs			
Duration of guard interval Δ	2048T 224μs	2048T 224μs	512T 56μs	256T 28μs	512T 56μs	256T 28μs	128T 14μs	64T 7μs
Symbol duration Ts=Δ+Tu	10240T 1120μs	10240T 1120μs	8704T 952μs	8448T 924μs	2560T 280μs	2304T 252μs	2176T 238μs	2112T 231μs

In the table shown above **(Table 4.3.)**, T represents the elementary time period T=7/64 *μs* for 8 MHz.

To have an idea of what sort of distances represent the different guard intervals, in the following tables the associated delays are translated in to km. It means that a system with a guard interval of 1/4 allows delays that represent a difference between propagation ways of about 67 *km*. While this guard interval gets smaller, distances are smaller too.

Table 4.3 Delay in front km.

D/Tu	μsec	Km
1/4	224	67
1/8	112	34
1/16	56	17
1/32	28	8

Every symbol is composed by K=6817 carriers in the 8K mode and 1705 in the 2K mode, and they are transmitted in a Ts length duration. Not all the carriers are modulated by data coming from the channel coder. Describing the symbol mapping we have seen that only 1512 (2K) and 6048 (8K) are useful.

Calling *Ft* to the total binary flux transported by the data useful carriers, it is calculated as:

$$Ft = fs \cdot v \cdot L \; (bits/\sec) \qquad \textbf{(4.1)}$$

Where:

fs=1/Ts symbols frequency (symbols/sec)
Ts= symbol duration
V= number of bits per second or bits per carrier
L= active carriers for data information.

The useful binary flux or channel capacity *Fu* is calculated disregarding the binary flux from the redundancies introduced by the internal encoding (convolutional coder code rate) and the Red Solomon coder redundancy.

$$Fu = Ft \cdot r \cdot 188 / 204 \ (bits / \sec) \qquad \textbf{(4.2)}$$

For example: 8K mode transmission, coder tax of 2/3, guard interval of 1/4, and 64 QAM modulations for 8 MHz channels. We obtain:

Symbol duration: Ts= 1120 µsec.
Symbol frequency: fs= 1/Ts=892,857 symbols/sec
Bits/carrier: v=6
Number of active carriers: 6048
Total binary flux: Ft= 32,4Mbps
Coding rate: 2/3
Channel capacity: Fu= 19,90588 Mbps

The symbol length is calculated taking into consideration that in the 8K mode, there should be 7168 carriers. If the bandwidth is 8 MHz, the separation between carriers is 8 MHz/7168=1116 Hz, *Tu* inverse, so *Tu*=896 *µs*. If the guard period is 1/4, the symbol duration is *Ts*=*Tu*/4 + *Tu*=1120 *µs*. The other parameters can be obtained easily applying all the formulas explained before **(see Table 4.4)**.

Table 4.4 Useful bitrate (Mbit/s) for non hierarchical systems for 8 MHz channels.

Modulation	Code rate	Guard interval			
		1/4	1/8	1/16	1/32
QPSK	1/2	4,98	5,53	5.85	6.03
	2/3	6.64	7.37	7.81	8.04
	3/4	7.46	8.29	8.78	9.05
	5/6	8.29	9.22	9.76	10.05
	7/8	8.71	9.68	10.25	10.56
16-QAM	1/2	9.95	11.06	11.71	12.06
	2/3	13.27	14.75	15.61	16.09
	3/4	14.93	16.59	17.56	18.10
	5/6	16.59	18.43	19.52	20.11
	7/8	17.42	19.35	20.49	21.11
64-QAM	1/2	14.93	16.59	17.56	18.10
	2/3	19.91	22.12	23.42	24.13
	3/4	22.39	24.88	26.35	27.14
	5/6	24.88	27.65	29.27	30.16
	7/8	26.13	29.03	30.74	31.67

4.2.1. Designed functions of OFDM modulation

- **`TX_Canal_RX;`**

The objective of the main function from the channel and modulation block is to transmit and receive a complete OFDM symbol. First of all the mode variables are initialized (number of total carriers, total power of each OFDM symbol and symbol number). Immediately afterwards, we go on carriers modulation with the function **`modulador_OFDM`**.

In the transmission block, we add zeros until to reach the correct symbol length (2048 in 2K mode, 8192 in the 8K mode). We apply the IFFT (Inverse Fast Fourier Transform) to the signal, using the Matlab library defined function *ifft(x)*; so actually the signal is in the time domain and we add the guard period at the beginning of the symbol.

Now the carriers are ready for their transmission, so we make them pass through the channel. We can choose between two different model channels: Gaussian or F11 (Ricean). After the channel, we add the previous symbol tail to the actual symbol. In a parallel mode we also perform the calculation of the actual symbol to add it to the next symbol.

At the receiver, the first step is to erase the tails and the guard period from the actual symbol to just take the OFDM symbol. The next step is the channel estimation. In that case, we can also choose between two modes; the ideal estimation and the average estimation, both defined in the following functions. After the channel estimation, an AWGN (Additive White Gaussian Noise) is added to simulate channel noise. Finishing the symbol reception, we apply the FFT (Fast Fourier Transform), *fft(x)* in Matlab library, to the entire signal and then recover the original sent signal in the frequency domain. Then we make the erasure of the added zeros to obtain the transmitted symbol samples (1705 to 2048 carriers in the 2K mode).

In the demodulator block, the first step is the channel estimation due the needing to counteract channel effects. To finish all the process we make the detection of all the sent symbols, comparing them with the symbols in the constellation **(Fig. 4.6)** with the **`detector`** function.

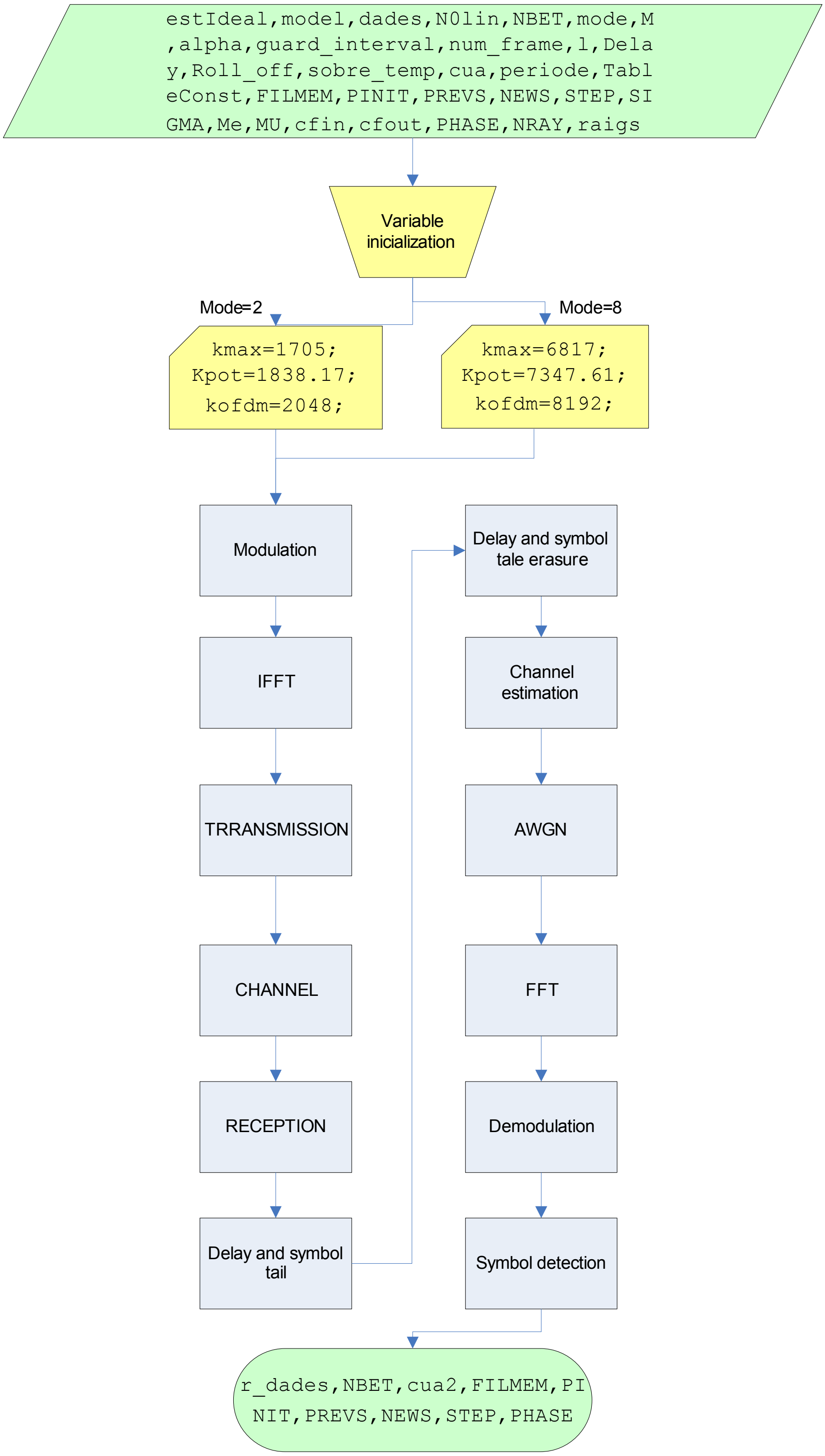

estIdeal,model,dades,N0lin,NBET,mode,M,alpha,guard_interval,num_frame,l,Delay,Roll_off,sobre_temp,cua,periode,TableConst,FILMEM,PINIT,PREVS,NEWS,STEP,SIGMA,Me,MU,cfin,cfout,PHASE,NRAY,raigs
Variable inicialization
Mode=2
Mode=8
kmax=1705; Kpot=1838.17; kofdm=2048;
kmax=6817; Kpot=7347.61; kofdm=8192;
Modulation
IFFT
TRRANSMISSION
CHANNEL
RECEPTION
Delay and symbol tail
Delay and symbol tale erasure
Channel estimation
AWGN
FFT
Demodulation
Symbol detection
r_dades,NBET,cua2,FILMEM,PINIT,PREVS,NEWS,STEP,PHASE

Fig. 4.6 Flux diagram TX_Canal_RX.

- Input parameters:
  - estIdeal: channel estimation mode (1 for ideal estimation, else for average estimation).

- model: channel mode used.
- dades: input file, in this case it corresponds with the symbol interleaver block outputs.
- N0lin: noise spectral density in linear.
- mode: used mode.
- M: modulation.
- NBET: number of error bits until the last iteration.
- alpha: in hierarchical mode, is a parameter for the constellation tuning.
- guard_interval: length fraction of the guard interval against the OFDM symbol (1/4, 1/8, 1/16, 1/32).
- num_frame: frame number.
- l: total number of symbols transmitted.
- delay: number of zeros until the D/A main sample.
- roll_off: roll-off parameter from the raised cosine of the D/A conversor.
- sobre_temp: number of samples added between two consecutive samples.
- cua: samples out of the OFDM symbols and that are useful for the next symbol.
- periode: time between samples in the time domain.
- tableConst: vector containing the values of the constellation points depending on M and alpha (constellation shall start by zero symbol).
- FILMEM, PINIT, PREVS, NEWS, STEP, SIGMA, Me, MU, cfin, cfout, PHASE, NRAY, raigs: channel filter parameters.

- Output parameters:
  - r_dades: Received data, one OFDM symbol in reception.
  - NBET: number of erroneous bits until the last iteration.
  - cua2: samples that are out the OFDM symbol; useful for the following symbol.
  - FILMEM, PINIT, PREVS, NEWS, STEP, PHASE: channel filter parameters.

· **`idealChannelEstimator;`**

The idealChannel Estimator, as its own name shows, makes ideal channel estimation. That procedure is made with a division of the knowing signal transmitted between the received signals.

- Input parameters:
  - indScateredPilots: symbol scattered pilots.
  - OFDM_sym: OFDM transmitted symbol.
  - r_OFDM_sym: OFDM received symbol.
  - kmax: number of carriers of the OFDM symbol.

- Output parameters:
  - channelEst: vector containing the channel estimation.

· **`averageChannelEstimator;`**

This function is responsible of the channel estimation, due this we realise a lineal estimation between two consecutive continue pilots. This lineal estimation is realised applying the convolution process (*conv(x,y)*) of two pilots and an over sampled triangular signal **(Fig. 4.7 Channel Estimation)**. With this process a lineal approximation of the sent signals between the two pilots is obtained.

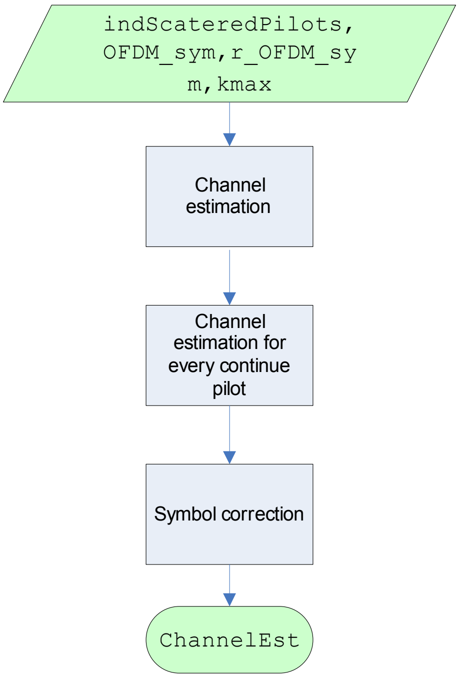

indScateredPilots, OFDM_sym,r_OFDM_sym,kmax
Channel estimation
Channel estimation for every continue pilot
Symbol correction
ChannelEst

Fig. 4.7 Flux diagram averageChannelEstimator.

- Input parameters:
  - indScateredPilots: symbol scattered pilots.
  - OFDM_sym: OFDM transmitted symbol.
  - r_OFDM_sym: OFDM received symbol.
  - kmax: number of carriers of the OFDM symbol.

- Output parameters:
  - channelEst: vector containing the channel estimation.

· **`modulador_OFDM;`**

The objective of this function is to return an OFDM symbol of 1705 (2K) or 6816 (8K) carriers from the input symbols. The first step is to generate a generator polynomial from the reference carriers using the **`PRBS_generator`** function. The next step is the layout of the continues pilots and scattered pilots. Both follow a pattern which is repeated along all the symbols. So once we know which are the positions that will be occupied, we pass to modulate them inside the OFDM symbol. The following carriers to be modulated are the TPS carriers; to perform it, first of all we load them from the two text files containing the carriers positions and we determine which symbols shall be transmitted inside them employing **`gen_mod_TPS`** `function`. Then these carriers are added to the OFDM symbol, which has all the required parameters except the information to be transmitted. To inherit the information, first of all the input symbols are modulated the *TableConst* input parameter, which defines the constellation points **(Fig. 4.8)**. Free carriers are determined from the pilots, then the mapping is performed and finally the data symbols are normalised at the free carriers in the OFDM symbol.

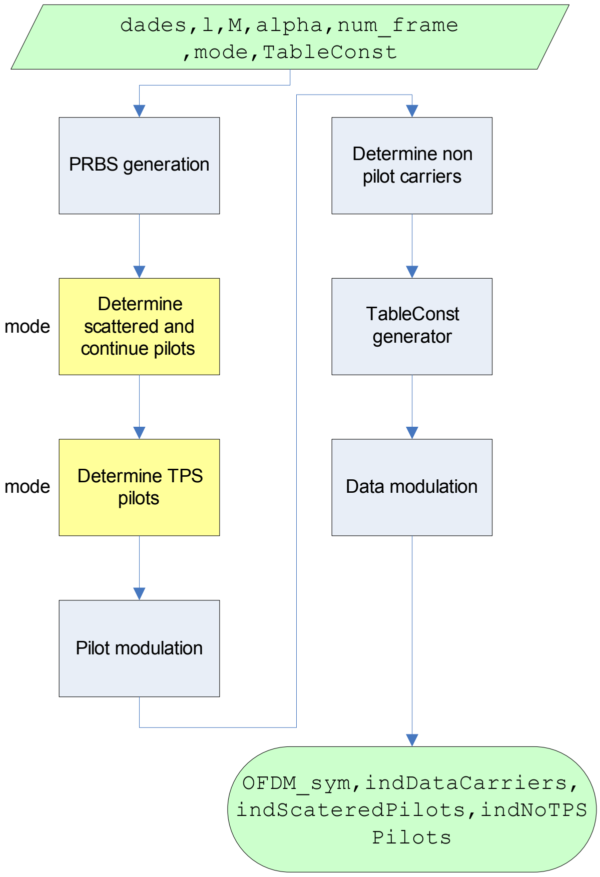

dades,l,M,alpha,num_frame ,mode,TableConst
PRBS generation
mode
Determine scattered and continue pilots
mode
Determine TPS pilots
Pilot modulation
Determine non pilot carriers
TableConst generator
Data modulation
OFDM_sym,indDataCarriers, indScateredPilots,indNoTPS Pilots

Fig. 4.8 Flux diagram modulador_OFDM.

- Input parameters:
  - dades: input symbols file.
  - l: total number of transmitted symbols.
  - M: modulation
  - alpha: in the hierarchical mode, parameter for constellation tuning.
  - num_frame: frame number of the OFDM symbol.
  - mode: mode used.
  - TableConst: vector containing the constellation values depending on M and alpha (starting in the zero symbol of the constellation).

- Output parameters:
  - OFDM_sym: vector containing the OFDM symbol to be transmitted.
  - indDataCarriers: indicates the data carriers.
  - indScateredPilots: indicates scattered pilots.
  - indNoTPSPilots: indicates TPS pilots.

4.3. Pilot carriers and OFDM frame structures

As we have commented in the last section, an OFDM symbol is a group of K elements or cells, where each of them correspond to a carrier (1702 in 2K, 6817 in 8K). Data precedent from the channel encoder just modulate 1512 (2K) or 6048 (8K) carriers, the rest of them simply are carriers or cells transmitting pilots **(Table 4.5)**. Pilots can be:

- Dispersal carrier pilots: channel regeneration in amplitude and phase.
- Continue carrier pilots: receiver synchronisation in frequency and phase.
- TPS (Transmission Parameter Signalling): information about the transmitted mode.

Table 4.5 Pilot Distribution.

	2K	8K
Continue	45	177
Dispersal	131	524
TPS	17	68
Data	1512	6048
Total	1705	6817

For this process is necessary distribute signals in frames **(see Table 4.6)**.

- One frame has a T_F length of 68 OFDM symbols numbered from 0 to 67: $T_{F=}$ 68 Ts.
- A super frame is composed by 4 frames.
- A mega frame is composed by 32 frames (8 super frames) in the 2K mode and 8 frames (2 super frames) in the 8K mode.

Table 4.6 Number of Reed-Solomon packets per OFDM super-frame.

Code rate	QPSK		16-QAM		64-QAM	
	2K mode	8K mode	2Kmode	8K mode	2K mode	8K mode
1/2	252	1008	504	2016	756	3024
2/3	336	1344	672	2688	1008	4032
3/4	378	1512	756	3024	1134	4536
5/6	420	1680	840	3360	1260	5040
7/8	441	1764	882	3528	1323	5292

All the data transmitted carriers are transmitted with normalised power level E[c x c*]=1 where c is *cm, l, k* points from the constellation of *k* carrier (from 0 to 6816 in 8K), of the *l* symbol (from 0 to 67) and from the frame number *m*.

In the **Table 4.7** the different normalization values for the *z* points of different constellations (see constellations figure).

Table 4.7 Normalization factors for data symbols.

Modulation Scheme		Normalization factor
QPSK		c=z/√2
16-QAM	α =1	c=z/√10
	α =2	c=z/√20
	α =4	c=z/√52
64-QAM	α =1	c=z/√42
	α =2	c=z/√60
	α =4	c=z/√108

All the cells containing pilots, whenever the modulation scheme is BPSK (dispersal and continues)(null imaginary part), are transmitted with a power level reinforced E[c x c*]=16/9. The TPS carriers are also transmitted in BPSK, but normalised to 1, as data.

With these values the pilot carriers and the TPS will occupy the positions of the real edge from the constellation:

- pilots ±4/3 x Z/C (4/3 x (42)^1/2=±8,64 in 64QAM with α=1).
- TPS ±Z/C ((42)^1/2=±6.48 in 64QAM with α=1).

4.3.1. Reference signals

Every continue pilot coincides with a scattered pilot every 4 symbols. The number of carriers of useful data keeps constant symbol to symbol. The pilots (scattered or continues) are modulated following a pseudo randomly bit sequence (PRBS) wk that corresponds to its *k* index from the carrier. This sequence keeps control of the initial phase from the TPS information. The PRBS sequence is initialized in such a way that the first output bit fro the PRBS coincides with the first active carrier. Then for every used carrier a new value is generated (whatever if it's or it isn't a pilot).

The polynomial generator is $x^{11} + x^2 + 1$ and the starting sequence is 11111111111.

4.3.2. Scattered pilots

The reference information is transmitted in the scattered pilot cell in every symbol. The corresponding modulation is:

Re{cm,l,k} = 4/3 x 2 (1/2 – wk)
Im{cm,l,k} = 0

m is the frame indicator, *k* the frequency indicator from the carriers and *l* is the temporal indicator from symbols. For the symbol indexed by *l* (since 0 to 67), the carriers for where the index k pertains to the subgroup {k=K_{min} + 3x(l mod 4) + 12 p | p integer, p ≥ 0, k E [K_{min};K_{max}] corresponding to pilots, where *p* in an integer defined for all possible positive values, in such a way the resulting *k* values don't exceed the valid margin [K_{min}; K_{max}] (K_{max}= 1704 for the 2K and 6816 for the 8K, K_{min} = 0) **(Fig. 4.9)**.

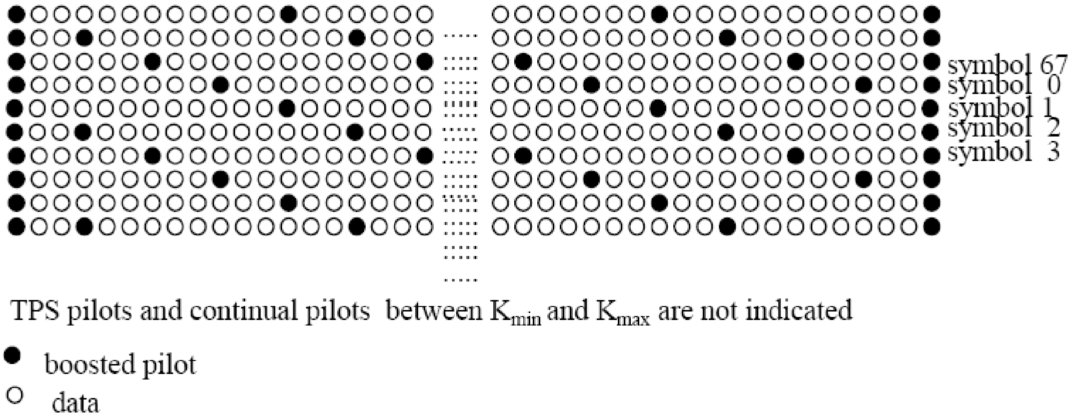

symbol 67
symbol 0
symbol 1
symbol 2
symbol 3
TPS pilots and continual pilots between K_{min} and K_{max} are not indicated
● boosted pilot
○ data

Fig. 4.9 Frame structure [4].

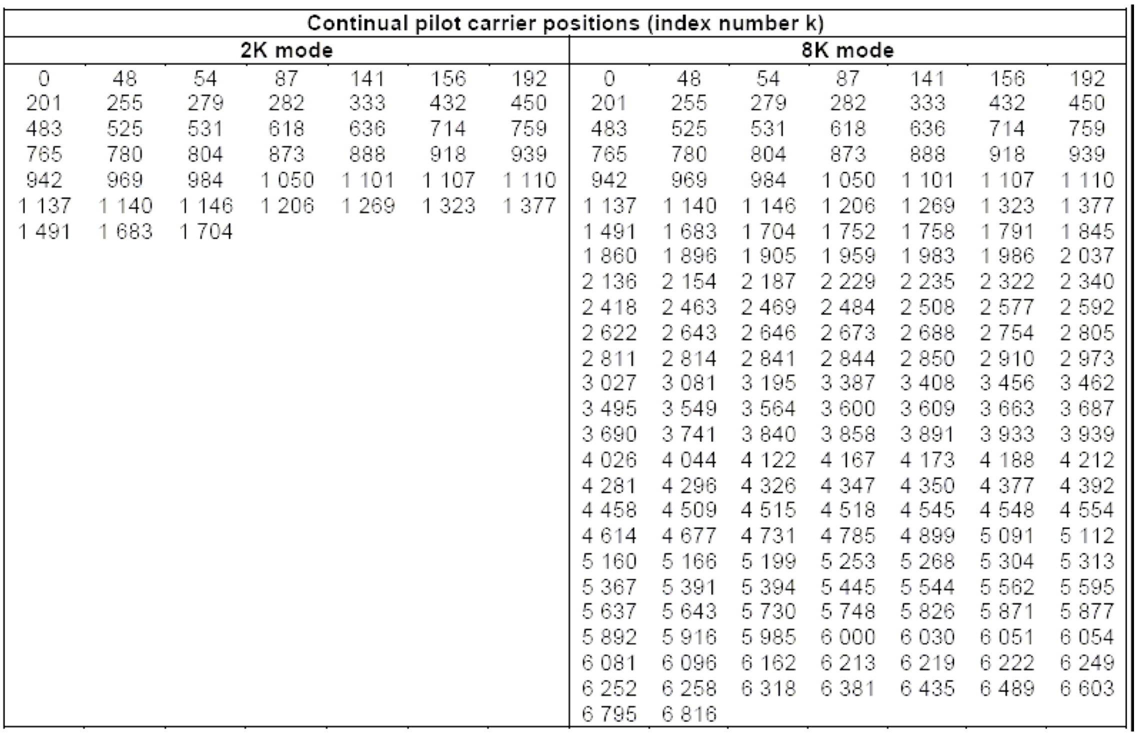
4.3.3. Continue pilots

Furthermore the distributed pilots, there are 177 continue pilots (it means that it happen in all the symbols) in the 8K and 45 in the 2K, that are inherit following the table values **(Table 4.8)**.

All the continue pilots are modulated following the reference sequence defined previously, and its corresponding modulation is:

Re {cm,l,k} = 4/3 x 2 (1/2 – wk)
Im {cm,l,k} = 0

Table 4.8 Carrier indices for continual pilot carriers.

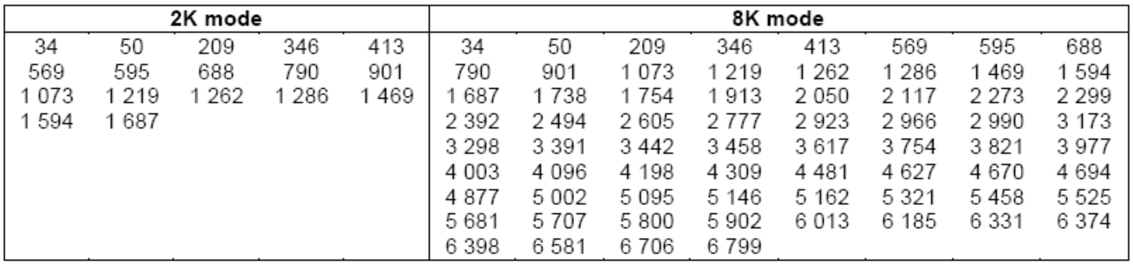


Continual pilot carrier positions (index number k)													
2K mode							8K mode						
0	48	54	87	141	156	192	0	48	54	87	141	156	192
201	255	279	282	333	432	450	201	255	279	282	333	432	450
483	525	531	618	636	714	759	483	525	531	618	636	714	759
765	780	804	873	888	918	939	765	780	804	873	888	918	939
942	969	984	1 050	1 101	1 107	1 110	942	969	984	1 050	1 101	1 107	1 110
1 137	1 140	1 146	1 206	1 269	1 323	1 377	1 137	1 140	1 146	1 206	1 269	1 323	1 377
1 491	1 683	1 704					1 491	1 683	1 704	1 752	1 758	1 791	1 845
							1 860	1 896	1 905	1 959	1 983	1 986	2 037
							2 136	2 154	2 187	2 229	2 235	2 322	2 340
							2 418	2 463	2 469	2 484	2 508	2 577	2 592
							2 622	2 643	2 646	2 673	2 688	2 754	2 805
							2 811	2 814	2 841	2 844	2 850	2 910	2 973
							3 027	3 081	3 195	3 387	3 408	3 456	3 462
							3 495	3 549	3 564	3 600	3 609	3 663	3 687
							3 690	3 741	3 840	3 858	3 891	3 933	3 939
							4 026	4 044	4 122	4 167	4 173	4 188	4 212
							4 281	4 296	4 326	4 347	4 350	4 377	4 392
							4 458	4 509	4 515	4 518	4 545	4 548	4 554
							4 614	4 677	4 731	4 785	4 899	5 091	5 112
							5 160	5 166	5 199	5 253	5 268	5 304	5 313
							5 367	5 391	5 394	5 445	5 544	5 562	5 595
							5 637	5 643	5 730	5 748	5 826	5 871	5 877
							5 892	5 916	5 985	6 000	6 030	6 051	6 054
							6 081	6 096	6 162	6 213	6 219	6 222	6 249
							6 252	6 258	6 318	6 381	6 435	6 489	6 603
							6 795	6 816					

4.3.4. TPS: Transmission parameter signalling

The TPS carriers are used for signalling parameters referring to the transmission system (channel encoding and modulation) are parallel transmitted in 17 TPS in the 2K mode and in 68 carriers in the 8K mode **(see Table 4.9)**. Every TPS carrier takes the same information bits differentially encoded, in the same symbol.

Table 4.9 Carrier indices for TPS carriers.



2K mode					8K mode							
34	50	209	346	413	34	50	209	346	413	569	595	688
569	595	688	790	901	790	901	1 073	1 219	1 262	1 286	1 469	1 594
1 073	1 219	1 262	1 286	1 469	1 687	1 738	1 754	1 913	2 050	2 117	2 273	2 299
1 594	1 687				2 392	2 494	2 605	2 777	2 923	2 966	2 990	3 173
					3 298	3 391	3 442	3 458	3 617	3 754	3 821	3 977
					4 003	4 096	4 198	4 309	4 481	4 627	4 670	4 694
					4 877	5 002	5 095	5 146	5 162	5 321	5 458	5 525
					5 681	5 707	5 800	5 902	6 013	6 185	6 331	6 374
					6 398	6 581	6 706	6 799				

The TPS carriers convey information on:

- Modulation including the α value of the QAM constellation pattern (see note);
- Hierarchy information;

- Guard interval (not for initial acquisition but for supporting initial response of the receiver in case of reconfiguration);
- Inner code rates;
- Transmission mode (2K or 8K, not for the initial acquisition but for supporting initial response of the receiver in case of reconfiguration);
- Frame number in a super-frame;
- Cell identification.

The TPS is defined over 68 consecutive OFDM symbols (a frame). The reference sequence corresponding to TPS carriers of the first symbol of every OFDM frame are used to initialize the TPS modulation on each TPS carrier. Each OFDM symbol conveys one TPS bit. Each TPS block (corresponding to one OFDM frame) contains 68 bits, defined as follows: 1 initialization bit (signalling), 16 for synchronisation, 37 for information (31 are used; the remaining 6 bits shall be set to zero) and 14 redundancy bits for error protection **(Table 4.10)**. The first bit sent is the bit is the leftmost bit.

Table 4.10 TPS signalling information.

Bit number	Purpose/Content
s0	Initialization
s_1 to s_{16}	Synchronization word
s_{17} to s_{22}	Length indicator
s_{23}, s_{24}	Frame number
s_{25}, s_{26}	Constellation
s_{27}, s_{28}, s_{29}	Hierarchy information
S_{30}, s_{31}, s_{32}	Code rate. HP stream
s_{33}, s_{34}, s_{35}	Code rate. LP stream
s_{36}, s_{37}	Guard interval
s_{38}, s_{39}	Transmission mode
s_{40} to s_{47}	Cell identifier
s_{48} to s_{53}	See annex F [4]
s_{54} to s_{67}	Error protection

- The first bit, s0, is an initialization bit for the differential 2-PSK modulation. The modulation of the TPS initialization bit is derived from the PRBS sequence previously defined.
- s1 – 16: of the TPS constitute a synchronization word: the first and the third TPS block in each super-frame have the following s1 – s16 word: s1 - s16 = 0011010111101110. And second and fourth TPS block have: s1 - s16 = 1100101000010001.
- The cell identifying information is an optional requirement. The length indicator has a “010111” if no cell information is transmitted (23 used bits) and “011111” if it’s transmitted (31 TPS used bits).
- There are 4 frames so two bits are needed to indicate each frame number. The TPS length indicator indicates the number of bits used at the TPS.

- Two bits to indicate the constellation (“00” QPSK, “01” 16 QAM, “10” 64 QAM, “11” reserved).
- The three following bits specify if it’s a hierarchical transmission and if its true the α value.
- The two following bits specify the guard period.
- The two others the transmission mode (2K and 8K).
- The nest 8 bits perform the cell identification. The most significant byte from the cell-id (b15-b8) is transmitted in the first and third frames, and the less significant byte (b7-b0) in the second and fourth frames.
- The 53 bits are spread with a error protection code, so that finally are 14 bits added, BCH (67, 53, t = 2) shortened code, derived from the systematic BCH (127, 113, t=2) code. This code can be implemented adding 60 null bits before the information bits arrive to the systematically code input. At the output these null bits are discarded, having a final word of 67 bits.

The TPS is transmitted with the normalised normal power, with an energy equal to all the restating data cells average, so that E[c x c*] = 1. Every TPS carrier is DBPSK modulated and carries the same message. The DBPSK is initialised at every TPS starting block.

The modulation rule is the following:

If sl = 0, Re{cm,l,k} = Re{cm,l-1,k}; Im{cm,l,k} = 0;
If sl = 1, Re{cm,l,k} = -Re{cm,l-1,k}; Im{cm,l,k} = 0;

4.3.5. Designed functions for pilot carriers

· **`PRBS_generator;`**

This function generates a random sequence from the generator polynomial *$p(x)=x^{11}+x^2+1$* defined in the standard for the correct modulation of the reference carriers in every iteration. To do it, an **`energy_dispersal`** very similar function is used, generating a slicing vector by XOR logical doors.

- Input parameters:
  - Mode: used mode (2K or 8K)
  - Seq_inicialitzacio: initialization input sequence, default: the parameter is all ones initialised.

- Output parameters:
  - W: output random sequence.

· **`gen_mod_TPS;`**

For the TPS carriers modulation, the function starts establishing its values as defined by the standard. To obtain its last values, the vector is encoded with a BCH shortened code, employing the Matlab defined function *bchenc(msg,n,k)*. As it happened in the Reed Solomon code at section of ***Chapter 3.3.*** First of all the input vector must be expressed in Galois Field terms **(Fig. 4.10)**.

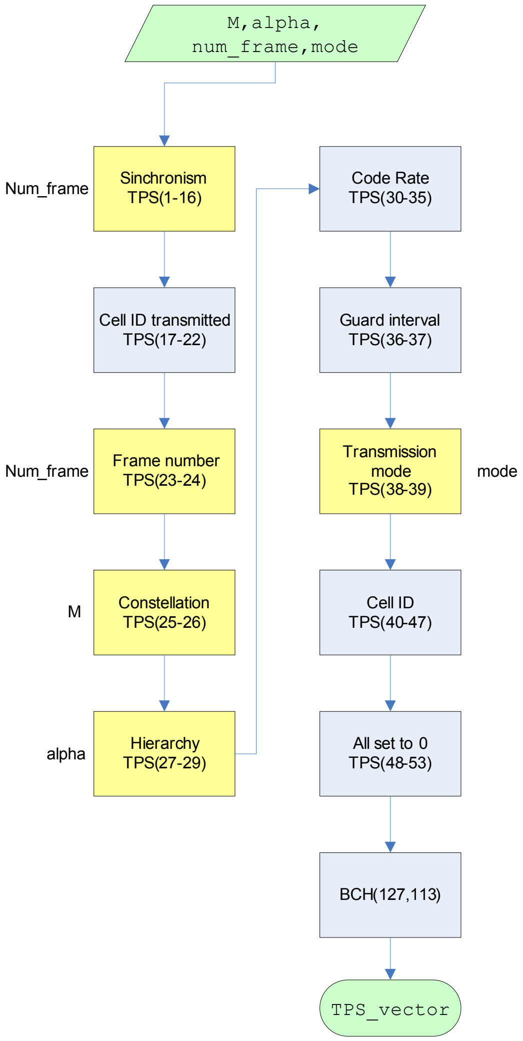


Fig. 4.10 Flux diagram gen_mod_TPS.

- Input parameters:
  - M: modulation
  - alpha: in the hierarchical mode, parameter for constellation tuning.
  - num_frame: frame number of the OFDM symbol used.
  - mode: mode used.

- Output parameters.
  - TPS_vector: 127 position vector which contains the TPS carriers values.

CHAPTER 5: RECEIVER

Along this chapter we will analyse all the different blocks of the receiver block, getting deep into the functions used in each case and in the main programming aspects of each block.

5.1. Symbol De-Interleaver

Symbol de-interleaver follows the same procedure as it's homologue symbol interleaver in the transmitter, but in the reverse way: odd symbols from output vector correspond to odd random symbols following the permutation function H(q) in the input vector, and in a similar way, even position symbols from the input vector correspond to even positions as defined by H(q) at the output vector.

5.1.1. Designed functions for symbol de-interleaver

· **`symbol_deinterleaver;`**

Inside symbol interleaver function the same scheme previously defined is followed to re-order input vector **(Fig. 5.1)**. At the end of this process, output vector is re-constructed in a bit vector for the next block (internal de-interleaver ***chapter 5.2.***).

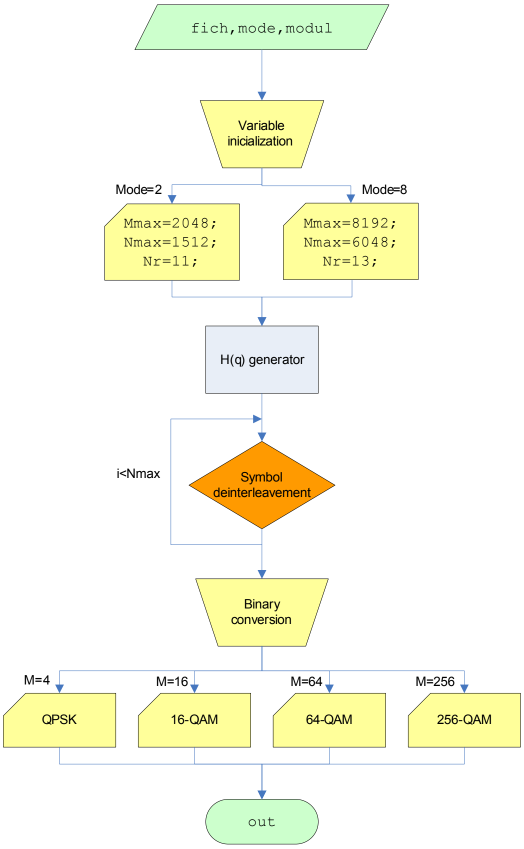


Fig. 5.1 Flux diagram symbol_deinterleaver.

- Input parameters:
    - fich: symbol input file.
    - mode: system mode (2K or 8K).
    - modul: modulation performed.

- Output parameters:
    - out: variable length output vector.

· **`H_generator;`**

The same function used in the symbol interleaver is executed to obtain the same ordering sequence as in the transmitter block, and then to generate the reverse effect.

5.2. Inner De-Interleaver

Every modulation employs its corresponding modules as defined in ***Chapter 2*** within the inner interleaver.

5.2.1. Designed functions for inner de-interleaver

· **`inner_deinterleaver;`**

First of all, we segment the input file in 126 bits length packets. Then they are de-multiplexed and re-ordered following the modulation scheme employed. Every vector is multiplexed again in order to get the correct bit ordering. In this case, this function does not use any memory element because in the last block we have granted that all vectors' length is multiple of 126 bits, so there is no problem when de-interleaving these vectors **(Fig. 5.2)**.

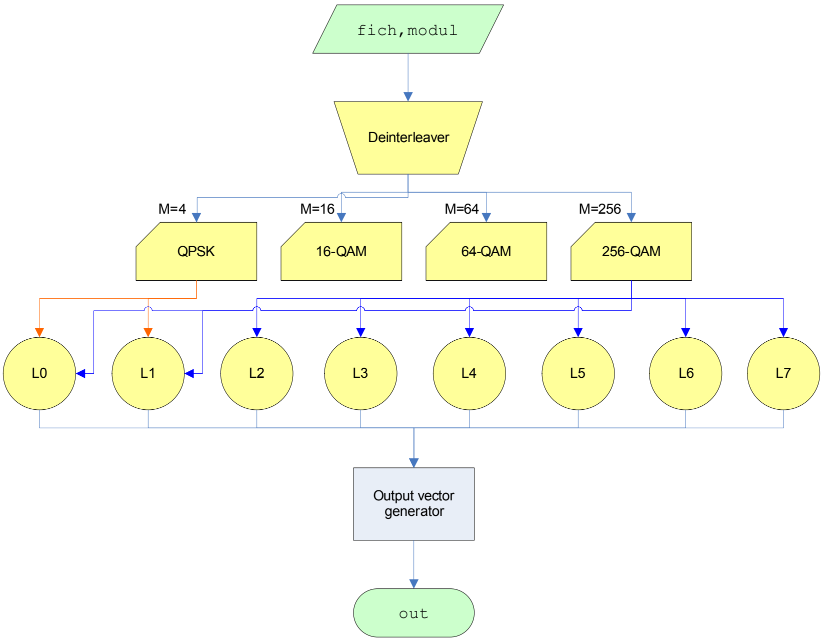


Fig. 5.2 Flux diagram inner_deinterleaver.

- Input parameters:
  - fich: multiple length files in bits with a multiple length of 126.
  - modul: modulation employed.
- Output parameters:
  - out: bit vector with the same length as the input vector.

5.3. Viterbi Decoder

A soft-decision Viterbi decoder will be employed to decode the information from the convolutional encoder.

5.3.1. Designed functions for Viterbi decoder

- **inner_decoder2;**

Analogously to the encoding function, the first steps in the function development have been the definition of the punctured patterns to generate an adequate encoder Trellis polynomial using the same function poly2trellis employed at the transmitter, see **(Fig. 5.3)**.

For the decoding process, we take profit of the implemented Matlab function *vitdec*(*code,trellis,tblen,opmode,dectype*), where *code* is the input file, *trellis* is the generator polynomial generated in the poly2trellis function, *tbleng* specifies the interleaver depth, *opmode* initialises the decoder and *dectype* chooses the decision type, in our case will be hard-decision.

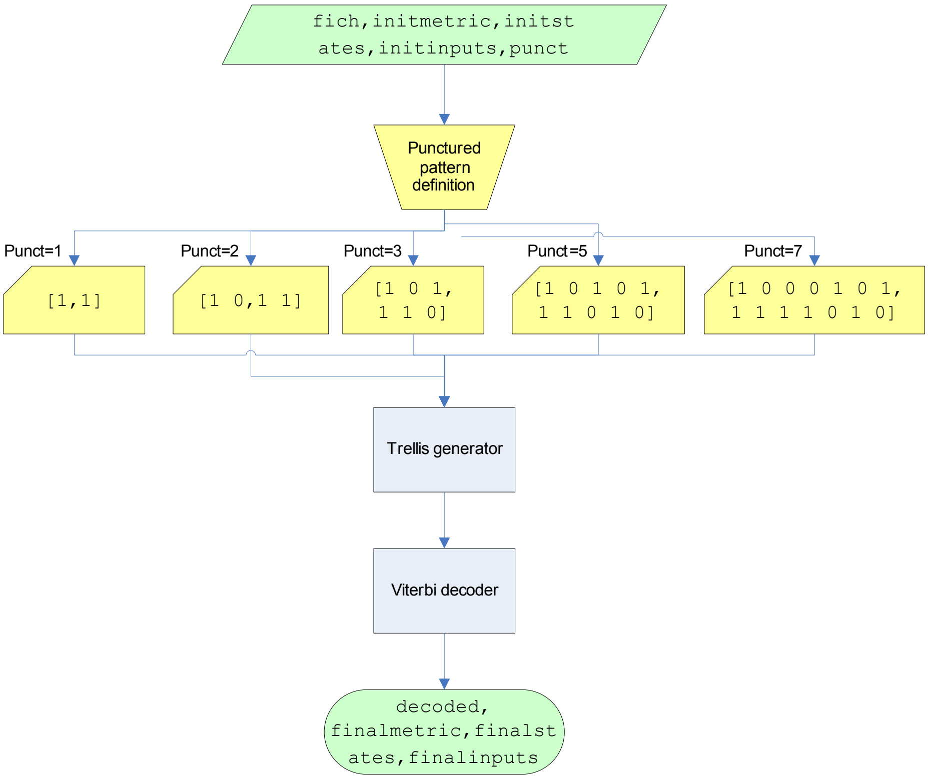


Fig. 5.3 Flux diagram inner_decoder2.

- Input parameters:
  - fich: 126 bit multiple input file
  - initmetric: represents the starting state metric of the corresponding state.
  - initstate and initinputs: represent the starting state metric of the corresponding state.
  - punct: defines the code rate used in the system.

- Output parameters:
  - decoded: vector bits decoded in soft-decision mode.
  - finalmetric: return the state metrics.
  - finalstates: return the traceback states.
  - finalinputs: return the traceback inputs.

5.4. Outer Interleaver

The outer interleaver block is the responsible of generating the biggest delay inside the system. This is due to the two interleaver matrixes that are previously initialised to zero, so when the transmitter receiver block is completed, a delay of two times filling the interleaver matrix is suffered. This delay corresponds to 16456 bytes (8228 bytes per matrix), which will be represented in an output system delay.

5.4.1. Designed functions for outer interleaver

· **`out_deinterleaver;`**

The outer interleaver has a dynamic memory matrix that varies depending on the iteration number, and also an external counter that counts the number of times the function has been executed. As it also happens at the interleaver block, the registers matrix is all-zero and each vector counters are also initialised to zero. Once the first execution is completed the matrix is not completely filled, so every internal counter is filled with the final reached position inside the matrix in last execution. It will be at third execution of the de-interleaving when the process will become finally iterative, with the advantage of automatically reinitialising counters to always the same value as described the figure below **(Fig. 5.4)**.

To perform this de-interleavement, input vector shall be in decimal bytes format. The conversion to bytes will be also helpful for the next decoder block as it will be seen hereafter in this chapter.

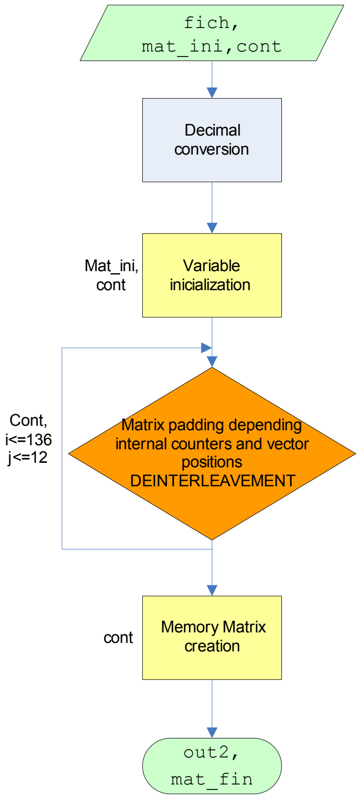


Fig. 5.4 Flux diagram out_deinterleaver.

- Input parameters:
  - fich: 13056 bits (8TS) vector.
  - mat_ini: matrix with memory registers from previous iterations.
  - cont: external counter, indicating the executions performed by the function.

- Output parameters:
  - out2: 1632 bytes (8TS) vector.
  - mat_fin: register status at the end of every function execution.

5.5. Reed-Solomon Decoder

In this block we decode input signal employing a shortened Reed-Solomon (204,188) code, the same as used at the transmitter.

5.5.1. Designed functions for Reed-Solomon Decoder

· **decodif_RS;**

At the RS decoder, we make a segmentation of the 1632 bytes input vector in to 8 packets of 204 bytes each. Every packet is transformed in a Galois field format, for its next decoding, performed by the function *rsdec(code,n,k)*. Where *code* is the 204 bytes segment, while *n* and *k* are the defining parameters of the Reed-Solomon code. Once is decoded, each segment is re-converted into a decimal number and transformed into a binary vector. Finally the total vector length is 12032, corresponding to 8 TS of the system **(Fig. 5.5)**.

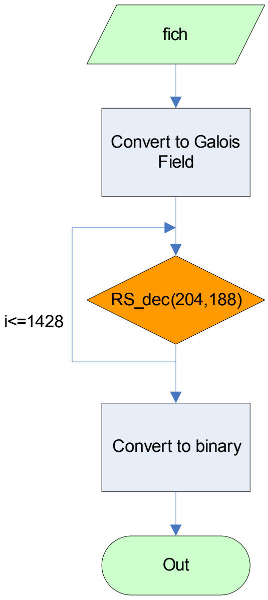


Fig. 5.5 Flux diagram decodif_RS.

- Input parameters:
    - fich: 1632 bytes vector.

- Output parameters:
    - out: 12032 bits (8TS) vector.

5.6. Recovering initial signal

In the last system block the objective is to proceed with the final reconstruction of the initial transmitted signal. To perform this, the first step is to undo the energy dispersal process and erase all the synchronisation sequences inserted at the transmitter.

5.6.1. Designed functions for recovering initial signal

- **`energy_dispersal;`**

Thanks to the *xor* logical door properties, we can undo the energy dispersal by just making the same process as in the transmitter, so undergoing the received vector with the sequence defined in transmitter block we obtain the same sequence as in the transmitter block. For this reason we have to regenerate the complete sequence using the energy_dispersal function.

- **degen_TS2;**

Unexpectedly, this block does not keep any symmetry with its homologue at the transmitter. This is caused by the delay caused by the outer interleaver. The first useful bit we find is at the second data packet of 8 TS sent by the transmitter, so we have to apply the same energy dispersal sequence to the same bit, in order to 'des-randomize' all data packets correctly. For this fact, at the first function execution energy dispersal is not applied since the entire received bits are delayed bits caused by the interleaver. At second iteration, the interleaver delay continues affecting until the 4521-th bit (8 first bits correspond to the synchronisation sequence) of the second iteration when the logical sequence of the energy dispersal starts to apply. As performed by the transmitter, when we apply the des-randomizing to the input file, we also perform the erasure of synchronisation sequences **(see Fig. 5.6)**.

- Input parameters:
  - fich: 12032 bits vector
  - cont: function execution counter.

- Output parameters:
  - vect_s: 11968 bits vector, bits that shall be the same as generated in function gen_entrada.

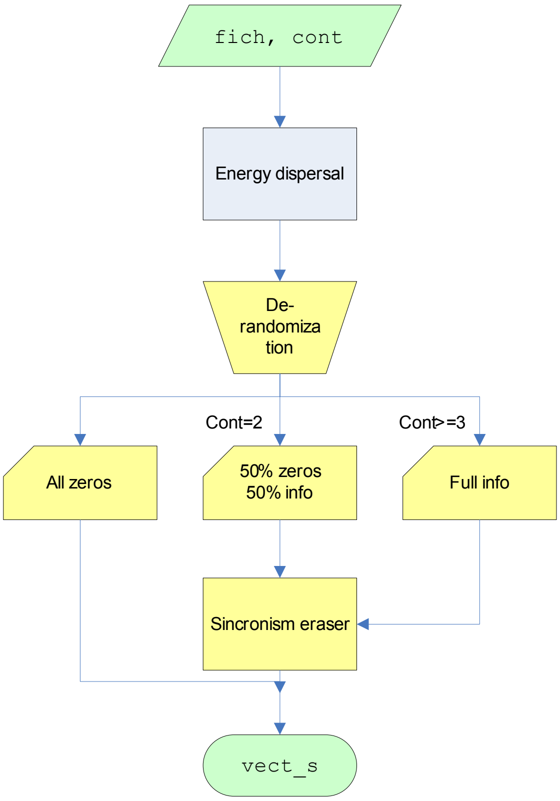


Fig. 5.6 Flux diagram degen_TS2.

5.7. Auxiliary functions

During the simulator development, it has appeared the needing to create some support functions for some of the system blocks. These auxiliary functions are used in several parts of the system. A clear example is the function required to convert a binary vector into a byte vector. Below is shown a list of all of these auxiliary functions.

5.7.1. Sequence creation

- energy_dispersal;
- H_generator;
- PRBS_generator;

These functions have already been defined in each of their corresponding blocks. They have the particularity to generate necessary sequences for the implementation of master blocks.

5.7.2. Convert2bin series

- convert2bin1:
- convert2bin2;
- convert2bin3;
- convert2bin4;

Functions composing convert2bin series are designed to make a decimal vector become a binary vector. The three first functions convert a vector into a symbol and they are used for the following modulations: QPSK, 16QAM, 64QAM, respectively. The convert2bin4 has a twofold purpose, convert a vector expressed into a binary vector expressed in bits and convert a 256-QAM modulation symbols into a binary (bits).

5.7.3. Convert2dec series

- convert2dec1;
- convert2dec2;
- convert2dec3;
- convert2dec4;

Together with the convert2bin series we can find the convert2dec series whose function is to convert binary vectors to decimal vectors, whatever we are employing modulation symbols or bytes. The conversion sequence follows the same pattern as in the previous case, where the three first functions are useful to do the bit to symbol conversion and the last one allows doing the bit to byte and also the bit to symbol conversion for the 256 QAM[2].

[2] Note: This modulation is not defined in the standard but it is one of the new solutions proposed for the DVB-T2 standard, you can find more information in *Chapter 6*.

CHAPTER 6: TECHNOLOGY AND STUDY FIELDS ANALYSIS DEVELOPED FOR DVB-T2

Since the defined DVB-T standard was been developed 10 years ago, and since there have been appearing a big number of radio communication systems employing the OFDM as multiplexing techniques, it seems to be reasonable assume that some of these systems are employing coders, interleavers, modulation techniques, pilot methods or synchronism sequences, transmission diversity techniques, MIMO schemes and other advanced mechanisms that offer better features than as defined in current DVB-T standard. In order to incorporate these techniques, a detailed analysis about their system impact is required. This analysis will be performed comparing new features with the classical DVB-T, trying to achieve the objective of performing a DVB-T2 standard capable to support High Definition Television (HDTV) transmission and also a good indoor coverage.

It is important to mention that any of the subtasks inside the FURIA project are not finished yet. For that reason most of these techniques have not been tested with simulations, so at this moment it is not possible to determine which the best solutions are.

As there is not any detailed information of the possible new features to implement specifically for the DVB-T2 standard, an exhaustive search for all the different systems employing OFDM has been performed. In this search a configuration study has been carried out with the objective of identifying possible feasible improvements and testing them with the developed simulator. In this scenario the following issues have been chosen to be studied (although the study is not closed and new proposals can be included while others can be discarded):

- New Modulation Schemes
- MIMO Techniques
- New pilot pattern for channel estimation
- New encoding algorithms for error protection

6.1. New Modulation Schemes

One of the main parameters that are put at stake for the DVB-T2 development and was one of the first ideas for the development of a new standard has been the inclusion of new modulation schemes. This allows to the system get higher transmission rates, ready to support HDTV. The FURIA consortium gave specific instructions that one of these improvements had to be the inclusion of the 256-QAM modulation and also probably the 512-QAM modulation.

For that reason, one of the first tasks carried out was the adaptation of the system to these new specifications, including the 256-QAM modulation scheme into the system block chain. The greatest part of the transmitter/receiver didn't suppose a big change. However in the bit-wise interleaver since it strongly depends on the employed modulation, a completely block re-organization was required.

The bit interleaving model proposed by the standard centres the interleavement pattern between module 126 and multiples of number 21, to generate the interleaver sequences. It can be seen (***Chapter 3)***, that having six interleaver blocks (L0-L5), 126/6=21, this block satisfies the interleaver necessities when the modulation sets up to 6 bits per symbol. The problem is that when we plan to include a 256 QAM modulation, we need eight blocks corresponding to the pertinent bits of each symbol. This situation forces to change the defined modules. Another option is include two new sequences for the two final blocks (L6 and L7) corresponding to the new modulation. Following we make a short a description of this proposals:

6.1.1. Proposed models for the improvement of bit-wise interleaver

As defined in ***chapter 3***, actually the standard defines the following block scheme for the internal interleaver in the following modulations:

L0=w mod 126. QPSK
L1=w+63 mod 126.
L2=w+105 mod 126. 16-QAM
L3=w+42 mod 126. 64-QAM
L4=w+21 mod 126.
L5=w+84 mod 126.

6.1.1.1. Proposal 1

The first model defined for this modification of bit-wise interleaver is based in the no-repetition of any of the previous blocks. So, knowing that the 126 model, for the blocks of less complex modulations, restricts the module number to 21 multiples, we decided to try the idea of using the half length of the blocks that would have to be repeated. So the module scheme is as it follows

L0=w mod 126.
L1=w+63 mod 126.
L2=w+105 mod 126.
L3=w+42 mod 126.
L4=w+21 mod 126.
L5=w+84 mod 126.
L6=w+94 mod 126.
L7=w+31 mod 126.

Whit this method, we keep the continuity with the sign permutation (+/-) between consecutive offsets, that is to say, with the increasing or decreasing interleavers order offset (+,+,-,-,+,+,-).

6.1.1.2. *Proposal 2*

In the second model, blocks employed for the QPSK modulation are repeated, adding them to the 64-QAM block scheme to generate the 256-QAM interleaver. The final scheme is:

```
L0=w mod 126.
L1=w+63 mod 126.
L2=w+105 mod 126.
L3=w+42 mod 126.
L4=w+21 mod 126.
L5=w+84 mod 126.
L6=w mod 126.
L7=w+63 mod 126.
```

6.1.1.3. *Proposal 3*

Finally, in this model we change all the previous blocks (only when the 256-QAM modulation is employed, for the other modulations all the blocks keep unchanged). Knowing that with 6 interleavers we obtained number 21 multiples block lengths, for the 8 blocks the same process is used having an offset difference of 126/8=15.75, which rounded up to its immediately higher integer number we obtain that employing a 16 multiple length a more homogenised sequence is found. So the final scheme for the 256-QAM is as it follows:

L0=w mod 126.
L1=w+64 mod 126.
L2=w+112 mod 126.
L3=w+48 mod 126.
L4=w+16 mod 126.
L5=w+80 mod 126.
L6=w+96 mod 126.
L7=w+32 mod 126.

In ***Chapter 7*** the evaluation of which is the best solution will be discussed according to the simulation results.

6.2. MIMO Techniques

In the following section, an enumeration of the different study fields over MIMO techniques is included. This study is realised because it seems to be clear that the standardization process is going in this line.

However, it shouldn’t be forgotten that, from a commercial point of view for the standardisation commission, the employment of MIMO techniques can suppose a so elevated increment to the cost of the equipments for the final user. Duplicate the number of antennas and the extra installation wiring clearly is a disadvantage of some of the most popular MIMO techniques.

- System evaluation using MIMO techniques (Multiple Input Multiple Output antennas), which are oriented to provide:

  - Spatial diversity in transmission, with the objective of reducing the bit error probability (BER) thanks to the diversity gain that partially makes up for the fading effects. These techniques include different strategies as the STBC codes (Space Time Block Codes) or SFBC (Space Frequency Block Codes), in the same concept as STBC but reinterpreting time for frequency. And the STTC (Space Time Trellis Codes). Initially, it seems that STBC codes are the most suitable ones, because they don't increase the decoder complexity when the modulator order also increases. These codes are well described in [11]. In figure **(Fig. 6.1),** a representation of a simplified scheme of the transmitted signal from both antennas and the signal received is shown. Developing the signal estimation at the decoder output we can see that:

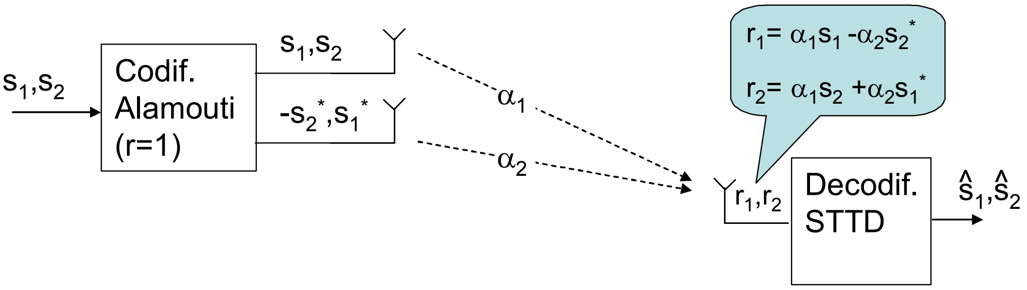


Fig. 6.1 Simplified scheme of a spatial diversity transmitter/receiver as described in [11]

$$\begin{aligned}\hat{s}_1&= \alpha_1^* r_1 + \alpha_2 r_2^* = |\alpha_1|^2 s_1 \cancel{-\alpha_2\alpha_1^* s_2^*} \cancel{+\alpha_2\alpha_1^* s_2^*} +|\alpha_2|^2 s_1 = (|\alpha_1|^2+|\alpha_2|^2)\, s_1\\ \hat{s}_2&= -\alpha_2 r_1^* + \alpha_1^* r_2 = \cancel{-\alpha_2\alpha_1^* s_1^*} +|\alpha_2|^2 s_2 +|\alpha_1|^2 s_2 \cancel{+\alpha_1^*\alpha_2 s_1^*} = (|\alpha_1|^2+|\alpha_2|^2)\, s_2\end{aligned} \qquad \textbf{(6.1)}$$

Therefore the estimated signals ($\hat{s}_1$ and $\hat{s}_2$) are proportional to the transmitted signals (s_1 and s_2), so a feedback is unnecessary, since this scheme works in open loop and it only needs to estimate the channel (CSI) at the receiver.

In figure **(Fig. 6.2)** the system that should be implemented when it's incorporated in to an OFDM system is schematically represented. As it can be seen, except for the Space Time Encoder (usually called MIMO Encoder), the rest of the parts of the system block are the same as implemented in the classical DVB-T simulator. Some schemes consider the addition of a preamble in each slot (a slot composed by a certain number of OFDM symbols) to make the synchronism recovering, easier. At the receiver, after the radiofrequency stage, the first performed step is the strong frequency-temporal synchronisation. Then the preamble and the cyclic prefix are erased and the OFDM signal gets demodulated applying the FFT. After this, a thin frequency-temporal synchronisation is performed to end this block. The channel estimator must perform a channel estimation for each antenna (or every two antennas if there is more than one at the receiver) and, because of this, usually this estimation is described as the channel impulsional response matrix.

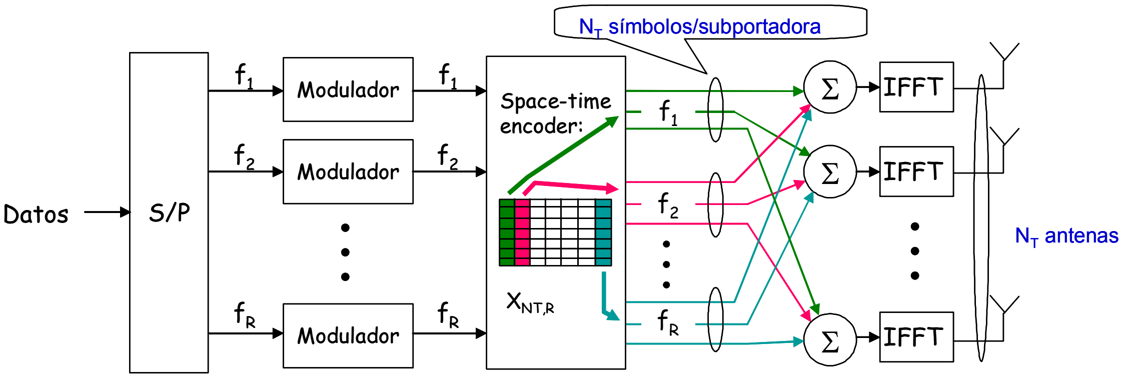


Fig. 6.2 Scheme of the configuration of MIMO-OFDM Space Block codes to implement

- Space multiplexing with 2X2 structure increasing channel capacity (bits/s/Hz) due to a *throughput* increase represented by an independent transmission in every antenna. Between these techniques some of them are discarded, for example the so called Waterfiling, because it requires to know the CSI (Channel State Information) both in the transmitter and the receiver. One of the most attractive techniques is V-BLAST (Vertical Bell Laboratories Layered Space Time) [12]. In general it offers better features than diversity techniques in transmission, but it has the inconvenient that gets faster degradation if the channel spatial correlation is not null (for example it doesn’t work well in LOS scenarios). And it is also very sensible to the estimation of channel errors. In figure **(Fig. 6.3)** the scheme of the BLAST is applied to a generic communications system. In an OFDM application it should exist a Mapping block (map to M-QAM symbols), and also it should perform the OFDM modulation. (IFFT and pilot and cyclic prefixes introduction). Equivalently in the receiver an OFDM demodulation block must be added.

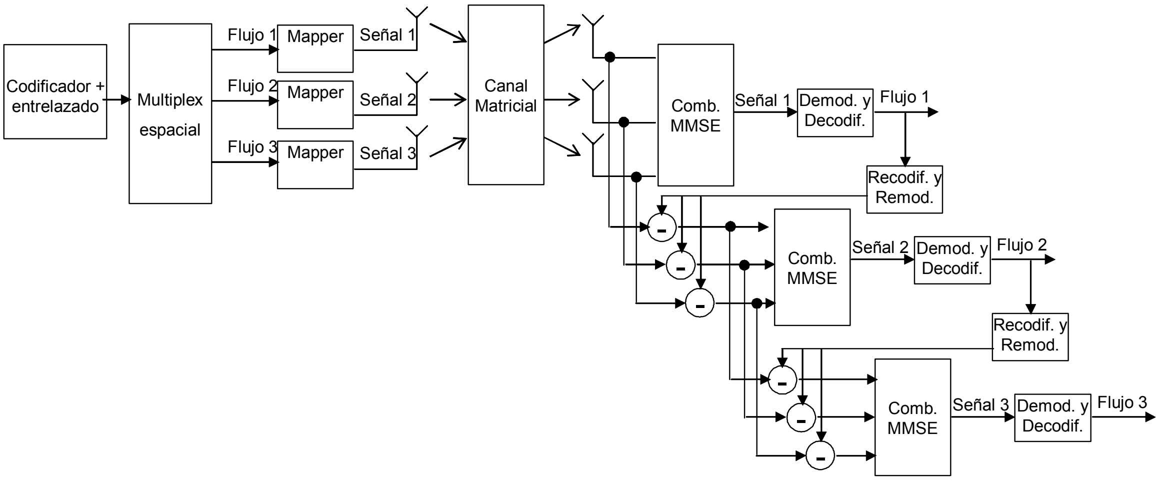


Fig. 6.3 Transmitter/Receiver generic scheme for BLAST.

- A combination of both principles can be also contemplated.

- Another possibility is to obtain a diversity gain with the use of a MIMO macro-structure in a SFN network (where each antenna corresponds to each different transmission station). This idea experiments a series of practical difficulties and has not been exploded in the scientific literature, except for just a contribution in the *IEEE Transactions on Broadcasting* that has made an interesting proposal currently under study [13].

Whatever is so important to study and to analyse the potential improvements of the stated configurations, in the presence of real channels where it exists a certain correlation between received signals. The main reason for this is that in the greatest part of publications referred to these issues just the ideal case is analysed. This fact could restrict the MIMO advantages in several environments (differentiate between urban areas where for example cell size is not so big, etc).

6.3. New pilot pattern for channel estimation

These improvements are oriented to flexibly the channel estimation techniques (employing pilots), studying the effects of a reduction in the number and type of the used pilots. For example: orthogonal, distributed, and as it can be, the use of completely different sequence techniques as the transmission of sequences known by the receiver. Taking advantage of the guard period [14] or the channel estimation based on the precedent OFDM symbol (CD3 or Coded Decision Directed Demodulation)[15], C/F SS (coarse/fine Symbols Synchronisation) [16] or may be a combined mix of these different techniques.

6.4. New encoding algorithms for the error protection

6.4.1. Encoder and interleaver schemes

Some of the relevant aspects that incentives the standard evolution to the DVB-T2 standard, is the bit rate increase due to the necessity to transmit high definition video. Also the introduction of better improvements for the system robustness against Doppler effects makes that possible changes in the encoding and interleavement techniques that currently are in use, must be considered.

For this reason, a revision of the interleaver and encoder schemes employed in the different world DTTB standards will be initially processed. Lately, the other wireless communications standards that have evolved to newer standards, offering more capacity and robustness will be also analysed. Some examples of these systems are WLAN and WWAN networks where the introduction of the MIMO technologies has played a fundamental paper. Taking care of all these arguments, special attention will be payed to the interleavers and coders that are currently working in standards like WLAN 802.11n and WMAN 802.11e.

Talking about error protection, the DVB-S2 standard gave a radical change from the previous standard DVB-S. DVB-S2 employs a powerful FEC system based on the concatenation of an external encoder BCH (Bose-Chaudhuri-Hocquenghem) with an internal LDPC (Low Density Parity Check). Instead of the traditional concatenation of Reed-Solomon and Viterbi used in the DVB-S standard. Therefore, it is reliable to think that the DVB-T2 will add similar changes, using LDPCs, or any other kind of turbo codes as the DVB-SH performs.

6.4.1.1. Study of Existing interleaver and encoding schemes

The different standards developed in Europe, China, Japan and EEUU for the DTTB are summarized in the following table **(Table 6.1)**:

Table 6.1 DTTB worldwide standards.

	Standard	Geography
Mobile	DVB-H	EMEA, America, APAC, Africa, ME
Mobile	ISDB-T	Japan
Mobile	DTMB	China
Fixed	DVB-T	EMEA, S. America, APAC, Africa, ME
Fixed	ATSC	N. America
Fixed	ISDB-T	Japan
Fixed	DTMB	China

6.4.2. Interleaver and encoding techniques in European DTTB standards

6.4.2.1. DVB-T Standard

The European standard DVB-T [4] uses two encoders and two interleavers, both inner and outer. The outer encoding is a shortened Reed-Solomon code RS (204,199, t=8), and the outer interleaver is a bit wise interleaver with a depth of: I=12.

The inner encoder is a convolutional code composed by a code with a 1/2 base rate, and some puncturing strategies, completing the rest of the standard rates:2/3,3/4,5/6 and 7/8.

The inner interleaver consists in a bit-wise interleaver followed by a symbol interleaver. Within the inner interleaver the modulation type must be specified: QPSK, 16QAM and 64-QAM, hierarchical or not hierarchical in the modes: 2K and 8K cause the variation of some parameters.

6.4.2.2. DVB-H Standard

In the DVB-H standard [17] the same encoders and the same interleavers as in DVB-T are used. The only difference is that there is new mode; beside the 2K and 8K mode, a 4K mode is defined. This implies some modifications on the inner interleaver. In modes 2K and 4K is possible to have a symbol interleaver as used in DVB-T with the corresponding parameters for each mode or a in depth symbol interleaver for the 6048 symbol length blocks for every mode.

6.4.3. Interleaver and encoding techniques in the United States DTTB standards

In the United States standard [18] two services are allowed, the ATSC 8-VSB which is the main used and the ATSC E8-VSB which is the enhanced service used to improve robustness.

6.4.3.1. ATSC 8-VSB

First of all, the block encoding is a Reed Solomon encoder RS (207, 187, t=10) followed by a byte-wise convolutional interleaver with a depth I=52, N=208. Finally, a Trellis encoder with code rate 2/3 is employed. An input bit is encoded in two output bits employing a convolutional encoder with 1/2 rate and the other bit is pre-encoded as shown in figure **(Fig. 6.4)**. The Trellis encoder is inside a symbol interleaver/de-interleaver with twelve pre-codifiers and twelve Trellis encoders.

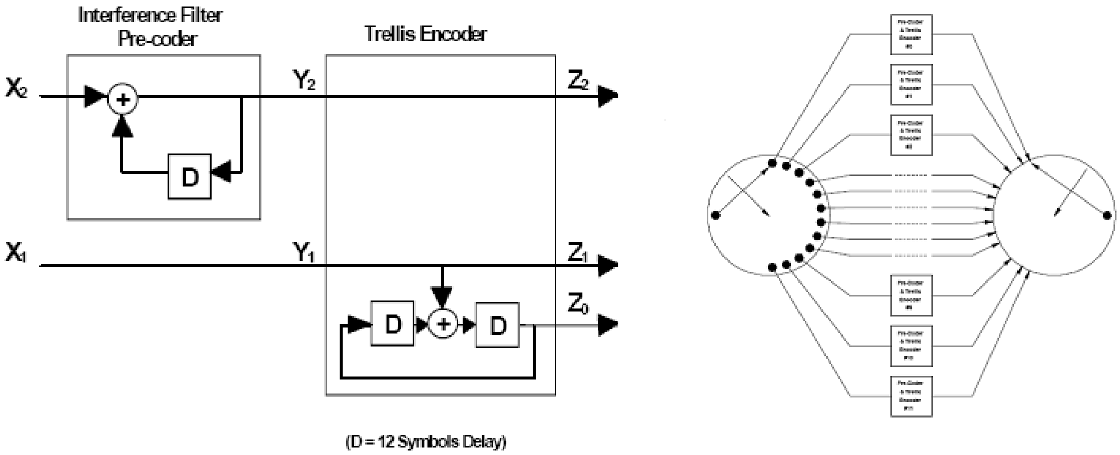


Fig. 6.4 ATSC8-VSB encoding scheme and Trellis encoder.

6.4.3.2 ATSC E8-VSB

The enhanced service, in addition to employing the encoders and the interleavers from the ATSC 8-VSB, it previously uses a Reed-Solomon encoder RS (184,164, t=10), followed by a bit-wise convolutional interleaver with depth I=46, N=184. After the interleaver there is a convolutional encoder compounded by another interleaver/de-interleaver block as shown in figure **(Fig. 6.4).** And another Trellis encoder that processes the input bytes in 2 bits nibbles **(Fig. 6.5)**. After the convolutional encoder the convolutional de-interleavement and the Reed-Solomon decoding are performed.

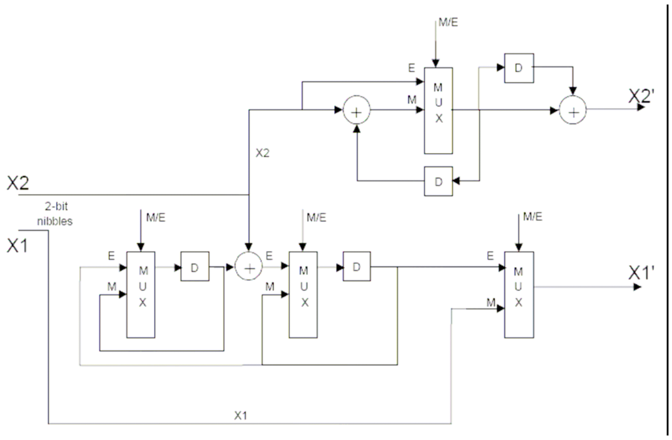


Fig. 6.5 ATSC E8-VSB Encoding scheme

6.4.4. Interleaver and encoding techniques in Japan DTTB standards

6.4.4.1. ISDB-T

The Japanese standard [19] employees a Reed-Solomon external encoder, RS (204, 188, t=8), then a bit-wise external interleaver with depth I=12 and a internal convolutional encoder with a base code with code rate 1/2 and puncturing patterns to obtain the other code rates: 2/3, 3/4, 5/6 y 7/8.

Inside of the modulation process there are three interleavers, bit interleaver that makes a serial sequence become a parallel sequence and it also introduces a delay. Symbol-wise time interleaver which introduces a delay for every symbol and a frequency interleaver.

6.4.5. Interleaver and encoding techniques in Chinese DTTB standards

6.4.5.1. DTMB

Inside the Chinese standard [20] there are two modes: multi carrier mode or single carrier mode. The encoding technique is the same for both modes, although the interleaver presents some variations.

For the error correction (FEC) a Bose-Chaudhuri-Hocquenghem external encoder is used, BCH (762, 752) derived from BCH (1023, 1013), and an internal LDPC (Low Density Parity Check) encoding with the following code rates: 0.4, 0.6 y 0.8.

As interleaver, it employees a convolutional interleaver defined in the time domain with depth I=52 and interleaver defined in frequency domain inside of each frame for the multi-carrier mode.

6.4.6. Interleaver and encoding techniques of MIMO WLAN and WWAN standards

6.4.6.1. WLAN 802.11n

The encoding and interleaver structure employed in [21] is shown at figure **(Fig. 6.6)**.

In the case the number of spatial streams is Nss= 1 or 2, only one FEC (Forward Error Correction) encoder is employed. If Nss= 3 or 4, two FEC encoders are employed. In the case that two FEC are employed, a Parsing encoder sends data bits to the different FEC inputs, following the algorithm:

$$x_i^{(j)} = b_{N_{ES} i - j} \quad ; \quad 0 \le j \le N_{ES} - 1 \qquad \textbf{(6.2)}$$

Where N_{ES} is de FEC index. After the Parsing operation, six zeros are added at the end of the message to reset the convolutional encoder to the zero state. Then is employed at FEC block, a convolutional encoder with a code rate of 1/2 and a puncturing pattern to obtain the rest of the rates: 2/3, 3/4, 5/6 y 7/8.

The data interleavement is composed by two parts, the first part called stream parser takes consecutive bit blocks that are assigned to the different space chains employing the round robin planning. This is performed in order to carry out the frequency interleavement in a second part. The frequency interleaver has different parameters attending to the channel bandwidth. This interleavement consists of three permutations. The first permutation ensures that adjacent encoded bits are alternatively mapped in LSB and MSB from the constellation. The third permutation is a frequency rotation where we obtain a function with spatial chain index.

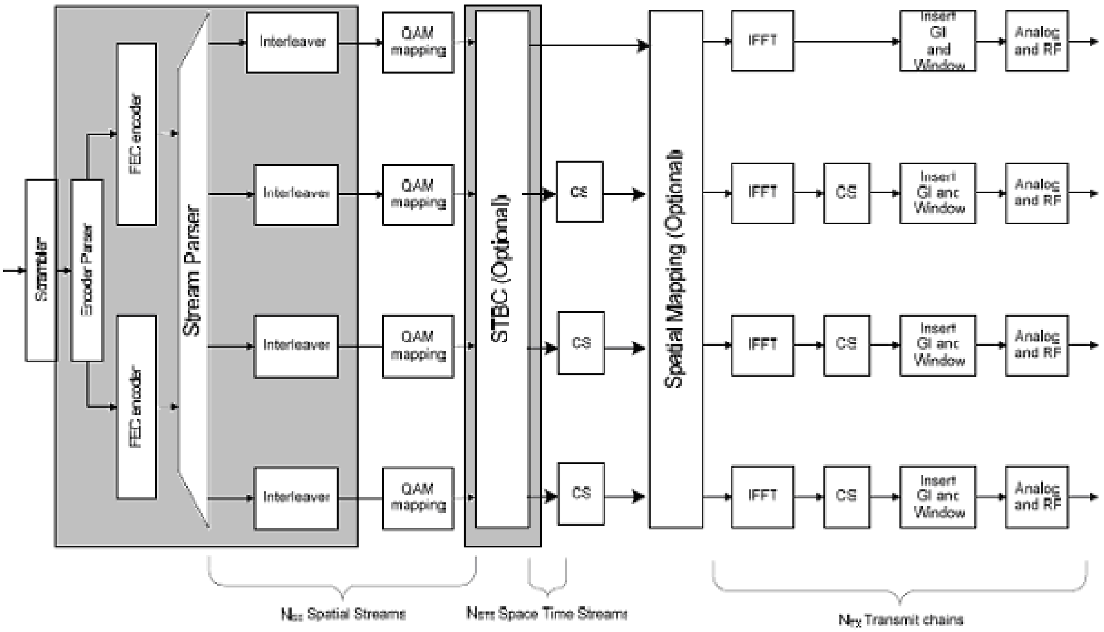


Fig. 6.6 WLAN 802.11n. interleaver and encoder schemes.

After mapping, we have an optional encoding block called Space-Time-Block-Coding (STBC) which is applied to generate robustness transmission code rates. N_{sts} is bigger than the spatial streams, N_{ss}. And sometimes a hybrid of STBC and SDM (Spatial Division Multiplexing) is also employed.

Table 6.2 WLAN 802.11n. parameters.

Parameter	20MHz	40MHz
N_{COL}	13	18
N_{ROW}	$4N_{BPSC}$	$6N_{BPSC}$
N_{ROT}	11	29

N_{col}: Number of column bits.
N_{row} Number of row bits.
N_{rot} Frequency rotation parameter.
N_{bpsc} Number of encoded bits.
For a single carrier

6.4.6.2. WMAN 802.16e

Inside the standard [22], the utilization of MIMO technology for the WMAN-OFDMA is contemplated. In the blocks diagrams shown in figure **(Fig. 6.7)** we can see the interleaver and control error block shadowed.

A FEC encoder is employed with a convolutional code with a base code rate 1/2. It is optional the implementation of a HARQ, in where is allowed to perform a puncturing with the following rates: 2/3, 3/4 and 5/6 and other two encoding modes: the turbo block encoding and the Convolutional Turbo Coding (CTC).

The Turbo block encoding consists of the multiplication of two simple different components. The codes employed are extended Hamming binary codes or parity check. Also the convolutional turbo coding employs a round double binary systematic recursive convolutional code and a CTC interleaver performed in two steps employing the CTC parameters

The interleaver employed is a bit interleaver with two permutations, the same as the two first used in the WLAN interleaver.

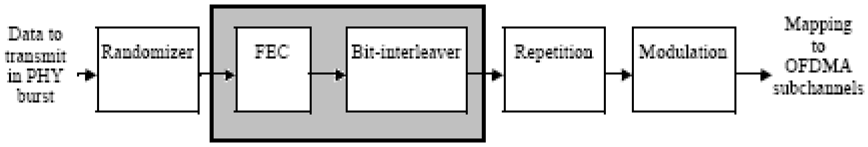


Fig. 6.7 WMAN 802.16e. interleaver and encoding schemes.

After the interleaver and the different convolutional encoder possibilities, turbo block and CTC, another optional encoding is proposed: the space time coding for the different MIMO configurations with two, three and four antennas.

6.4.7. Possible interleaver and encoder schemes for DVB-T2

The introduction of the Turbo Codes [23] and the LDPC (Low -Density Parity-Check) [24] codes has become one of the most important advances talking about encoding techniques during the last years. Supposing that the data block length is big enough, both schemes are close to the Shannon limit. However, paying attention to practical situations, turbo codes represent a low encoding complexity, while their decoding complexity is really high. Otherwise the LDPC codes represent a low decoding complexity against a higher complexity at the encoder. Precisely, in response to this last complexity in the LDPC encoding, by means of including a low density matrix at the encoder, new codes were born LDGM (Low-Density Generation-Matrix) [25]. In this context, we also have to mention ZigZag codes [26] [27]. And even though they are far away of the results obtained with the already mentioned encoding schemes, they offer a low complexity in the encoder and decoder blocks.

6.4.7.1. Encoding schemes

The encoding schemes that should be suitable for be considered to be included in DVB-T2 are:

- Employ LDPC codes (Low Density Parity Check), are easy implemented codes, and faster than TC (Turbo Codes). In some documents is proposed their combination with MIMO-OFDM techniques. Some examples are WLAN and fourth generation mobile communication systems (4G) [28].

- TPC (Turbo Product Code) codes: these codes have quite good features (near Shannon theoretical capacity) and they don't represent the inconvenient of the TC (Turbo Codes) while inhering big delays due to the interleavers dimensions.

- SCCC (Serial Concatenated Convolutional Codes) codes are in some issues similar to Turbo Codes. They have already been used in MC-CDMA (Multi Carrier- CDMA) systems, which are using OFDM as transmission techniques to obtain high transmission bit rates in mobile scenarios (Rayleigh channel) [29] with a low bit error probability.

6.4.8. BCH fundamentals

BCH codes are multiple error correcting, cyclic codes, independently discovered by Bose, Chaudari, and Hocquenghem. They form part of a large class of cyclic error correcting codes that allow correcting errors up to the designated bound. RS codes were discovered independently by Reed and Solomon, in 1960. Primitive BCH codes are given, by the following limits called as the BCH bound. A BCH code exists for a t error correcting code in GF (2^m), where:

$$\begin{aligned} &Block\ Length:\ n = 2^m - 1 \\ &Parity\ Check\ Bits: n - k \leq m \cdot t \\ &Minimum\ d: d_{\min} \geq 2t + 1 \\ &if \quad m \geq 3, t \leq 2^{m-1} \end{aligned} \tag{6.3}$$

The code words are formed by taking the remainder after dividing a polynomial representing our information bits by a generator polynomial. The generator polynomial is selected to give the code its characteristics. All code words are multiples of the generator polynomial.

The construction of the generator polynomial is actually a combination of several polynomials corresponding to several powers of a primitive element in GF (2^m).

The discoverers of the BCH codes determined that if α is a primitive element of GF (2^m), the generator polynomial is the polynomial of lowest degree over GF (2) with $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ as roots. The length of a codeword is 2^{m-1} and t is the number of correctable errors. Concluding the generator is the least common multiple of the minimal polynomials of each α^i term. A simplification is possible because every even power of a primitive element has the same minimal polynomial as some odd power of the element, halving the number of factors in the polynomial. Then

$g(x) = LCM\{\phi_1(x)\phi_2(x)...\phi_{2t}(x)\}$ g(x) where ϕ_r represents the minimal polynomial of α^i, where $O(\alpha) = 2^m - 1$, and α belongs to GF(2^m).

These BCH codes are called primitive because they are built using a primitive element of GF(2^m). BCH codes can be built using non primitive elements, too, but the block length is typically less than 2^m -1.

For our encoding purposes the first idea is to implement the BCH encoder used in Chinese DTTB [20] standard to obtain a first round of results, and make a comparison with actually DVB-T Reed-Solomon.

The Chinese BCH encoder is a BCH (762,752) derived from BCH (1023, 1013). Applying the formulas showed in **(6.3)** we can determine the number of correctable errors:

$$\begin{gathered} n = 2^m - 1 \rightarrow 1023 = 2^m - 1 \rightarrow m = 10 \\ n - k \leq m \cdot t \rightarrow 1023 - 1013 \leq 10t \rightarrow t \geq 1 \end{gathered} \quad \textbf{(6.4)}$$

After performing this operation we can appreciate that this particular BCH encoder will just be able to correct one error in front the Reed-Solomon code which it can correct up to eight errors. This worst error correction capacity will be rectified in the next block (LDPC encoder) which will improve better error correction than Viterbi encoder.

The benefit that offers this BCH encoder is that we can improve faster code rates than in RS. BCH encoder adds 10 redundancy bytes every 4 MPEG-2 data packets and RS encoder adds 16 redundancy bytes every MPEG-2 data packet. So totally we have tax rate gain of a 7% (in BCH for 100 bytes we send 7 information bytes more than RS). Also the BCH encoders offer another important benefit in front the RS, because they perform lower complexity operations than Reed-Solomon.

6.4.9. LDPC Fundamentals

LDPC codes [24] are linear block codes with sparse parity check matrices $H_{(n-k)xn}$. As an example, an LDPC code of length N=10 and rate 1/2 can be specified by the following parity check matrix **(Fig. 6.8)**.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Fig 6.8 Example of a parity check matrix.

The purpose of this decoder is to determine the transmitted values of the bits. Bit nodes check nodes communicate with each other to accomplish that. The decoding starts by assigning the channel values to the outgoing edges from bit nodes to check nodes. Upon receiving that, the check nodes make use of the parity check equations to update the bit node information and send it back. Each bit node then performs a soft of majority vote among the information reaching him. At this point, if the hard decisions on the bits satisfy all of the parity check equations, it means a valid codeword has been found and the process stops **(Fig. 6.9)**. Otherwise bit nodes go on sending the result of their soft majority votes to the check nodes.

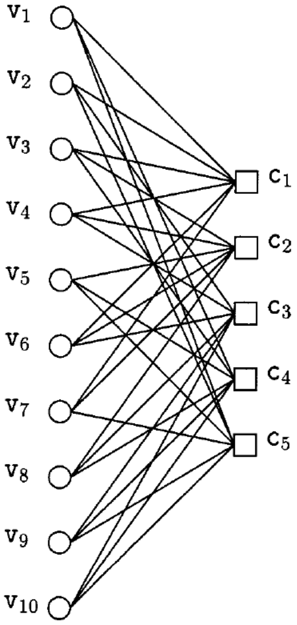


Fig 6.9 Bipartite graph of an LDPC code. [30]

LDPC codes possess several distinct advantages over BCH and Reed-Solomon codes. First (belief-propagation) decoding for LDPC codes is fully parallelizable and can potentially be accomplished at significantly greater speeds. Second, very low complexity decoders that closely approximate belief propagation in performance may be (and have been) designed for these codes. Third, LDPC decoding is verifiable in the sense that decoding to a correct world is a detectable event.

One practical objection to LDPC codes has been that their encoding complexity is high. However, recently studies have proposed different solutions that reduce this complexity. One way to get around this problem is to slightly modify the construction of codes from bipartite graphs to a cascade of such graphs.

LDPC codes are well represented in bipartite graphs in which one set of nodes, corresponds to elements of the codeword and the other set of nodes, the check nodes, corresponds to the set of parity-check constraints which define the code. In figure **(Fig. 6.9)** we can appreciate the bipartite graph determining such a regular LDPC code. Also in **Fig. 6.10** we can appreciate the initialization of outgoing messages from bit nodes.

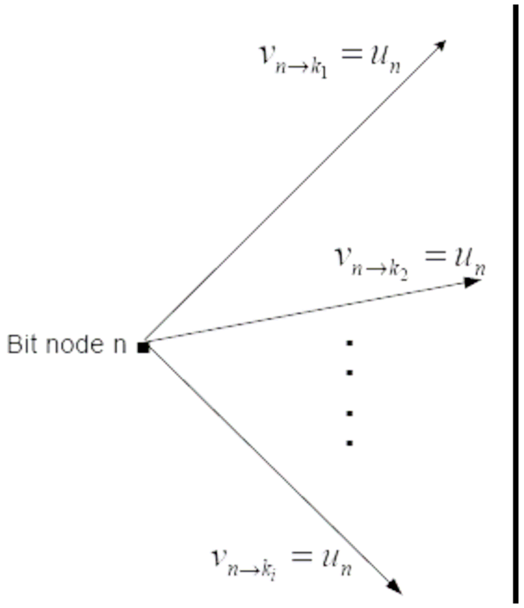


Fig 6.10 Outgoing message scheme.

However before getting deeply in the study of the possible LDPC schemes for lower complexity encoders we will implement the LDPC encoder defined in DVB-S2 standard [31] and see the LDPC effects in the simulator. So at this moment this will be our next investigation line for the DVB-T2 development.

CHAPTER 7: SIMULATION RESULTS

Along this chapter we are going to expose and comment the different results obtained with the simulator. The figures include channel comparative graphical results of the different modulation rates and some comparatives about codification rates and modes.[3]

7.1. Channel Comparative

As previously mentioned in ***Chapter 2*** we have considered two channel models: a Gaussian (AWGN) channel and a Rician/Rayleigh (F1). F1 channel is a static multipath channel with 20 multipath components showing a given delay, amplitude and phase, and it is proposed to test DVB-T system in a realistic environment where there is a direct path (Rice) with a power that is 10dB higher than the sum of the powers of the rest of multipath components. Details of this model can be found in [4]. Along the project, also a mobile multipath channel following the classic Doppler spectrum [32] with variable multipath components was also implemented. We have validated that this channel model is properly implemented but during the integration phase some problems have appeared and due to the time schedule for finishing the project, this channel has been finally not included in the analysis, but it is one of our next tasks in the FURIA project. The representation of the two considered models is shown in **Fig 7.1**.

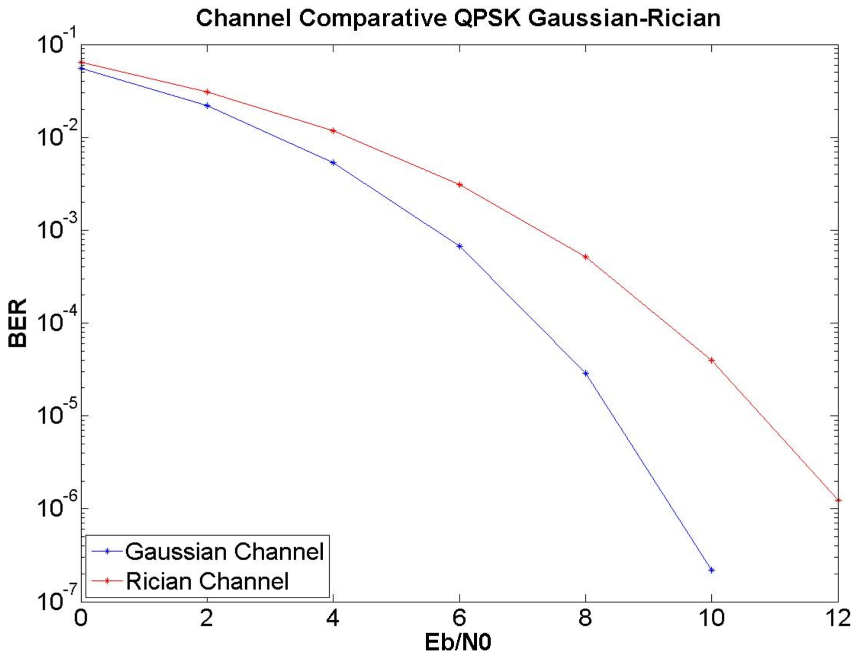


Fig. 7.1 Gaussian and Rician channel for QPSK modulation.

[3] In Annex 1 Section A1.1. all the graphical results are shown.

As we can see in the graphic (**Fig. 7.1)** for low Eb/No values, both channels have similar bit error rates, being clearly worse the F1 channel for high Eb/No values, due to selective fading caused by the multipath components. These are well known values and have been only used to test that the simulator works properly.

Testing the same channel but for different modulation rates it is observed that higher Eb/No values are required to maintain the BER. This is due to the reduction of the decision areas corresponding to each symbol, in the symbol constellation, as the modulation rate increases causing that the same noise level (and therefore Eb/No) produces a higher symbol error rate and therefore a higher bit error rate. This can be appreciated in **Fig. 7.2** and **Fig. 7.3**.

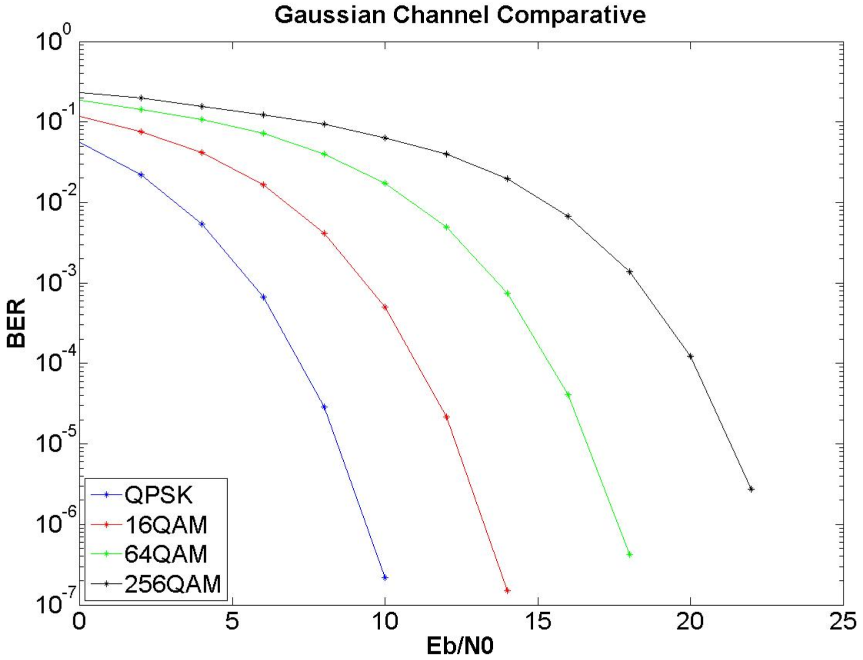


Fig. 7.2 Gaussian channel comparative.

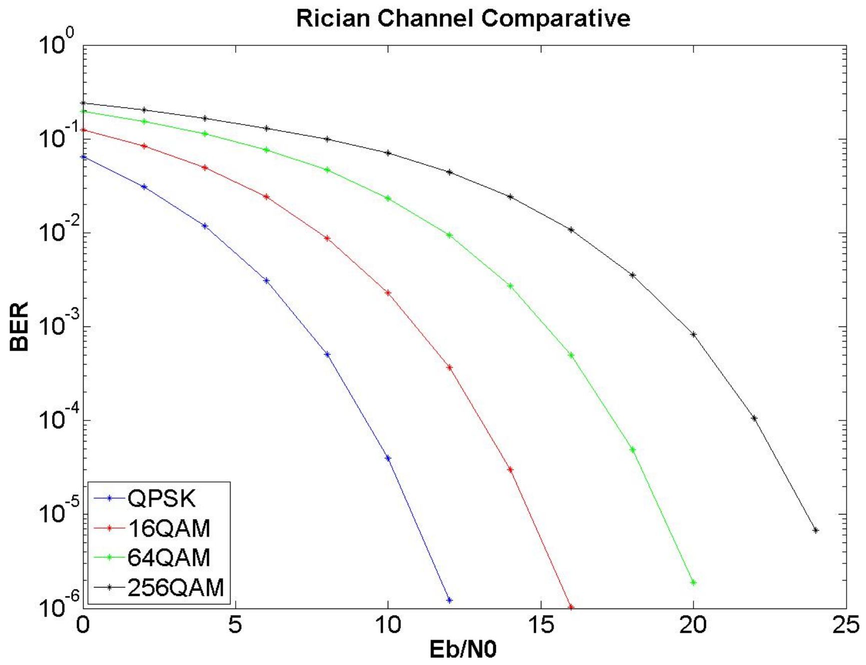


Fig. 7.3 Rician channel comparative.

This Eb/No variation corresponds to an increasing of 4,4 dB for QPSK to 16QAM and 16QAM to 64QAM, and approximately 4,8 dB for the 64QAM to 256QAM modulation **(7.1)** when considering BER values lower than 10^{-3}

$$P_M \leq 4Q\left(\sqrt{\frac{3\log_2 M}{(M-1)}\frac{E_{bav}}{N_o}}\right) \tag{7.1}$$

7.2. Shannon's Channel Capacity

Before showing the graphical results, we are going to explain the channel capacity limit. The Shannon capacity channel limit is the maximum theoretically achievable data rate in a channel with BER close to zero. This limit is defined by the following expression **(7.2)**. In graphical results of both modes we can appreciate that the BER curves have plain behaviour until certain Eb/No relationship threshold. From that point, the curve decreases with the expected response.

$$r_b < C = B_T \log_2\left(1 + SNR_R\right) \tag{7.2}$$

Where r_b *is the binary bit rate*;

B_T *is the channel bandwith and*

$SNR_R = \frac{E_b}{N_o}\frac{r_b}{B_T}$;

If the actual information rate on the channel is lower than the Shannon capacity then it is theoretically possible to achieve error free transmission using error detection and correction.

At **Table 7.1** there is a relationship between all combinations of coding rates and modulation types and the required Eb/No to reach the maximum capacity defined by the Shannon limit. Comparing the obtained QEF results in the graphics and the theoretical Shannon Limit, we can resolve that our limits of QEF with the Viterbi decoder are approximately in average of 3,7 dB below this limit. This threshold is represented at the figures with a black vertical line in the corresponding value of Eb/No.

Table 7.1 Required Eb/No to reach Shannon limit for all combinations of coding rates and modulation types.

Modulation	Code Rate	Eb/No Limit (dB)
QPSK	1/2	-3,01
	2/3	-1,19
	3/4	-0,39
	5/6	0,36
	7/8	0,73
16QAM	1/2	-1,25
	2/3	1,26
	3/4	2,43
	5/6	3,56
	7/8	4,11
64QAM	1/2	0,67
	2/3	3,98
	3/4	5,57
	5/6	7,13
	7/8	7,91
256QAM	1/2	2,73
	2/3	6,92
	3/4	8,96
	5/6	10,99
	7/8	12,00

7.3. 2K Mode simulation results

The **Table 7.2** gives a comparative between the ETSI (EN 300 744 [4]) standard requirements and the simulated performance anticipating "perfect channel estimation and without phase noise" of channel coding and modulation combinations in both cases.

These results are given for the Gaussian channel and the Rician channel (F1), when the centre carrier of the DVB-T signal is positioned at 32/7 MHz.

Table 7.2 Required Eb/No (dB) for non hierarchical transmission to achieve a BER=2×10-4 after the Viterbi decoder for all combinations of coding rates and modulation types at 2K Mode.[4]

		Required Eb/No for BER=2x10^{-4} after Viterbi QEF after Reed-Solomon			
Modulation	Code Rate	Gaussian Channel		Rician Channel (F1)	
		ETSI	Simulated	ETSI	Simulated
QPSK	1/2	0,1	-0,25	0,6	0,25
QPSK	2/3	1,9	1,25	2,7	2,0
QPSK	3/4	2,9	2,25	3,8	3,25
QPSK	5/6	3,9	3,25	5,0	4,75
QPSK	7/8	4,7	4,25	5,7	5,5
16QAM	1/2	2,8	2,75	3,6	3,25
16QAM	2/3	5,1	4,75	5,6	5,5
16QAM	3/4	6,5	5,75	7,0	6,75
16QAM	5/6	7,5	7,0	8,4	8,25
16QAM	7/8	7,9	7,75	9,0	9,0
64QAM	1/2	6,12	6,5	6,92	7,0
64QAM	2/3	8,72	8,75	9,02	9,5
64QAM	3/4	10,22	10,0	10,82	10,75
64QAM	5/6	11,52	11,25	12,22	12,25
64QAM	7/8	12,32	12,0	13,22	13,25

The results obtained by our simulator are close to the ones given in the previous table, with differences lower than 0.3 dB.

7.3.1. QPSK Modulation

7.3.1.1. Code rate: 1/2

For the convolutional encoder, the code rate 1/2 (so without puncturing) is the most robust in front of noise since for each information bit, two bits are sent to the channel. This allows to obtain really good BER performance even with small Eb/No. For example, as we can see in figures **Fig. 7.4 and Fig. 7.5**, for QPSK modulation in a Gaussian channel a Eb/No= 0 dB a quasi error free (QEF) signal reception at the end of the receiver is obtained. Also in the F1 channel with a Eb/No=0,25 dB we obtain that QEF reception.

[4] See Annex 1 Section A1.2. for a graphical description of the results.

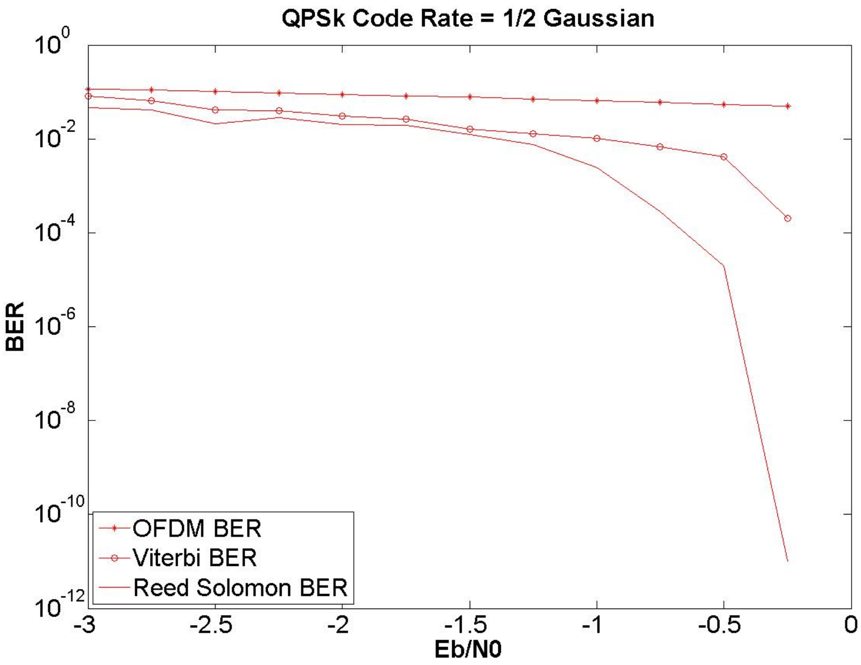


Fig. 7.4 Gaussian 1/2 Code rate.

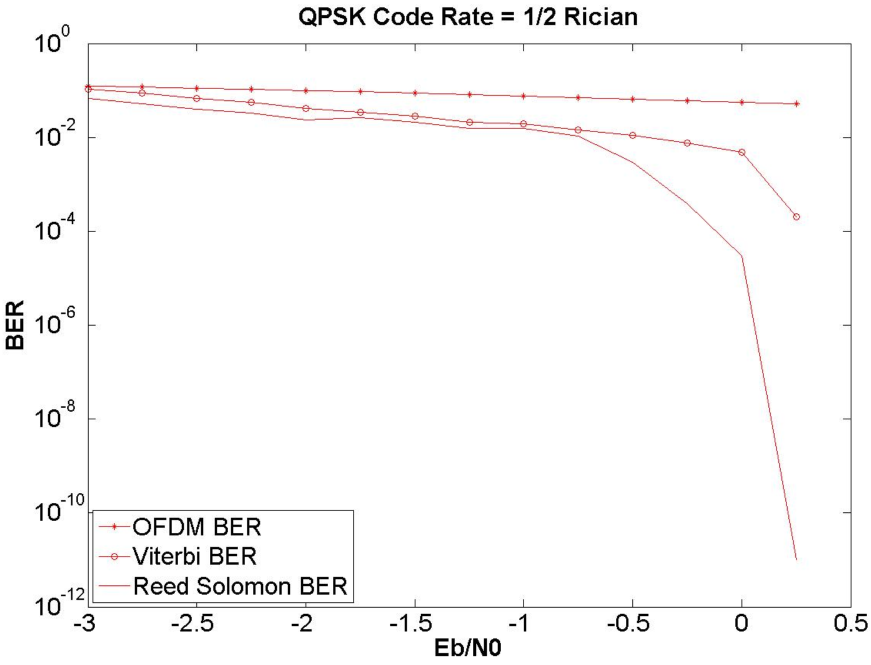


Fig. 7.5 Rician 1/2 Code Rate.

In the figures the three legends mean:

- OFDM BER is the BER calculated without considering the effect of the codes; this is comparing the transmitted bits sequence just before the mapping to QAM with the received bits sequence just after the QAM demodulator.

- Viterbi BER is the BER considering the capacity of the convolutional code, as well as the symbol and bit interleaver to reduce the channel impairments.
- Reed Solomon BER is the complete BER considering the effects of both codes, the interleavers and all the elements of the transmission/reception chain.

We think that is relevant to give these three figures, because then we can observe the degradation introduced by the channel when no error correction is implemented, then the effects of the convolutional code, and finally the complete system. When the BER after the Viterbi decoder is lower than 10^{-4} we can considerer that we are in a Quasi Error Free Channel. This allows us to perform faster simulations, because it is not necessary to simulate all the rank of BER until 10^{-12}.

7.3.1.4. Code rate: 2/3, 3/4, 5/6 and 7/8

As we increase the code rate (increasing the puncturing pattern), the system performance are worst. This is normal because we are sending less redundant bits so the Viterbi decoder becomes more sensitive to errors as can be appreciated in figure **(Fig. 7.7)**. In this way we can check that for higher code rates it is necessary to increase the Eb/No for the correct demodulation; and the QEF margin will be considerably sliced for higher rates **(see Fig. 7.6 and Fig. 7.7)**.

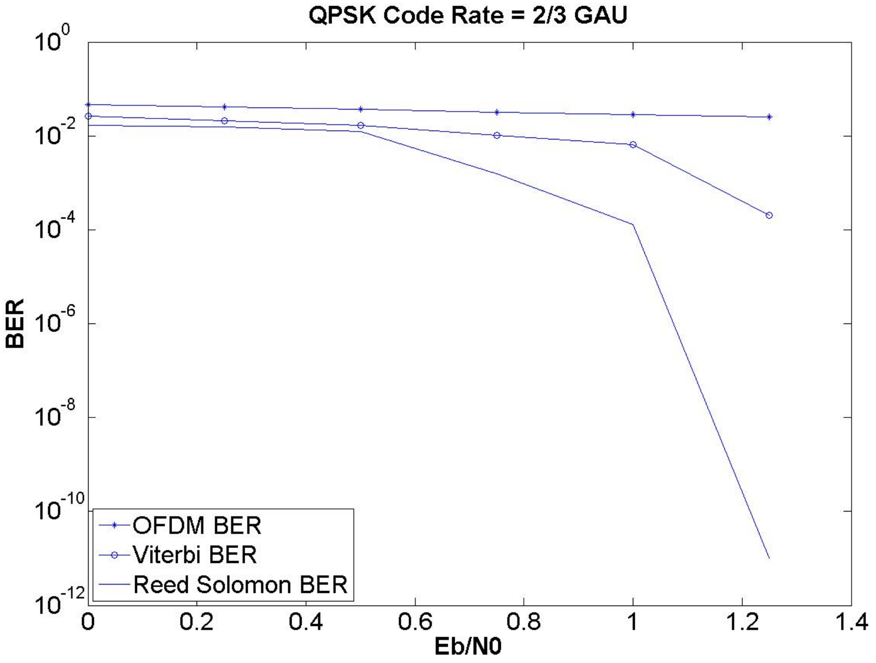


Fig. 7.6 Gaussian 2/3 Code rate.

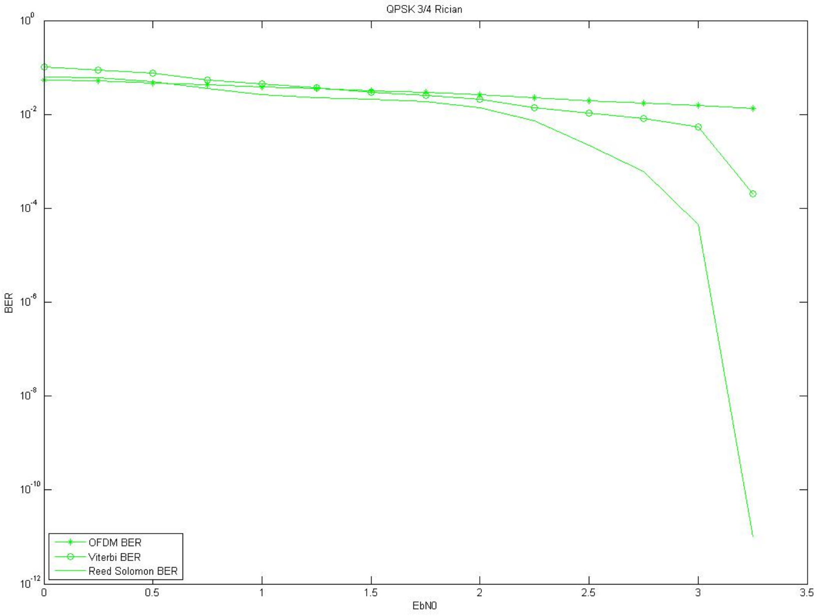


Fig. 7.7 Rician 3/4 Code Rate.

Knowing that when the code rate is higher, information bit rate also increases, better features can be offered. In other hand we can find that in low Eb/No relationships is possible that the system works above the limit of the Shannon channel capacity **(see Table 7.2 at *section 7.3.*)**, then the Viterbi decoder won't be able to correctly decode the transmitted signal, even the decoded signal can contain more errors than the received (through the channel) signal. This scenario can be found below **(Fig. 7.8 and Fig. 7.9)**.

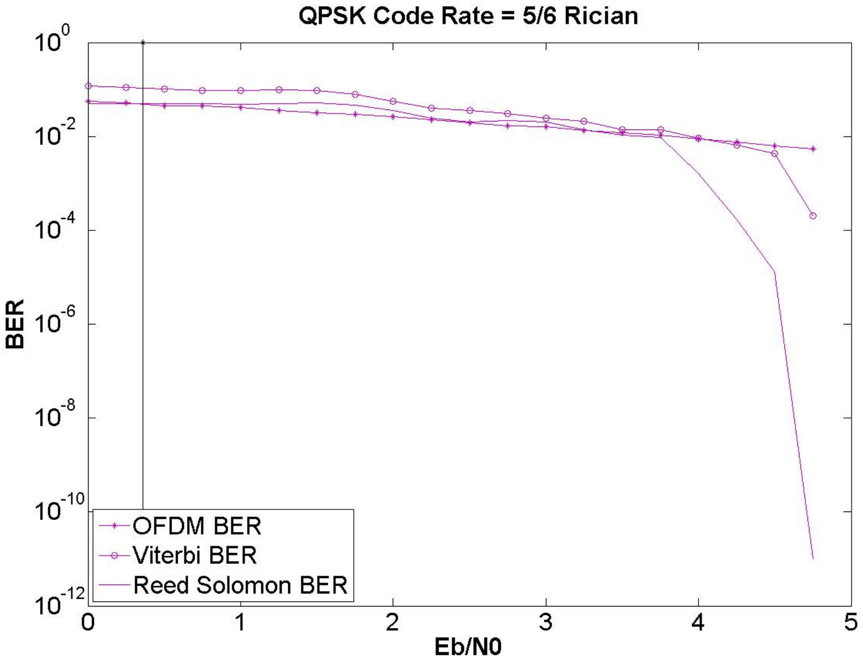


Fig. 7.8 Rician 5/6 Code rate.

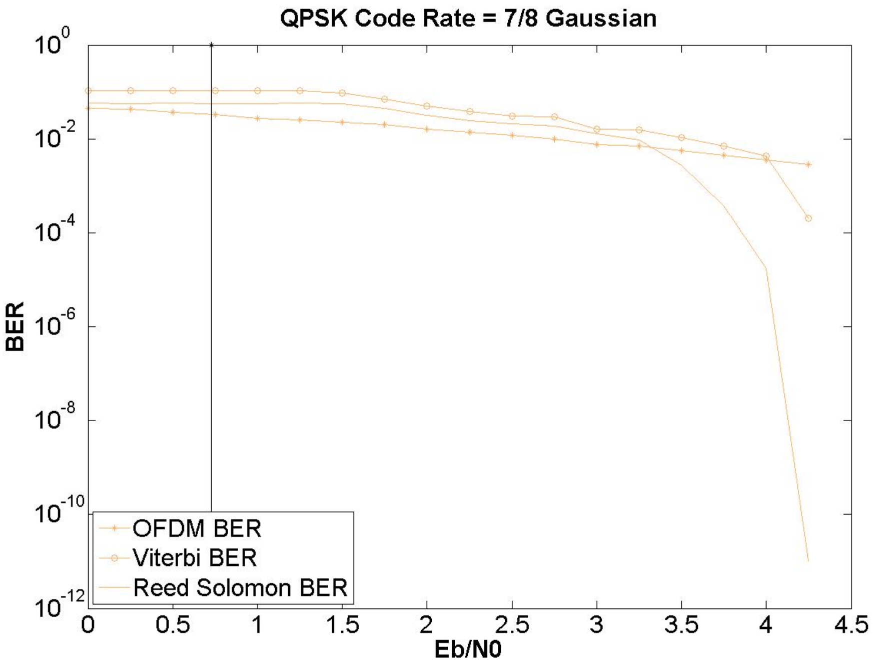


Fig. 7.9 Gaussian 7/8 Code Rate.

7.3.2. 16QAM and 64QAM Modulations

The behaviour found in the QPSK modulation is also extrapolable to higher modulations **(see Fig. 7.10)**, but with worst probabilities when the number of bits per symbol increases. This phenomenon has been explained in the Channel comparative section and it is caused by the bit energy requirements of the different modulation symbols.

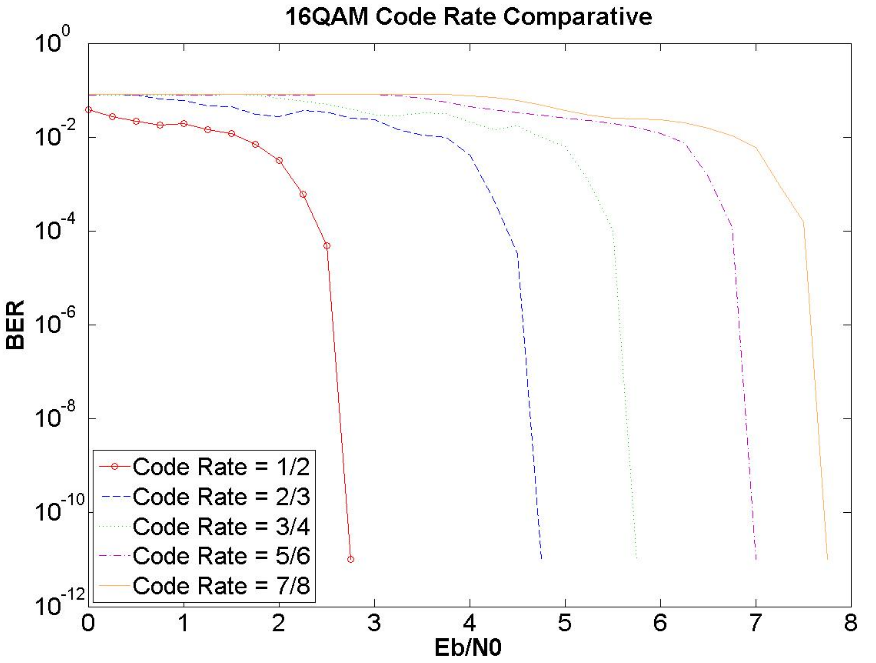


Fig. 7.10 Gaussian 16QAM Rate Comparative.

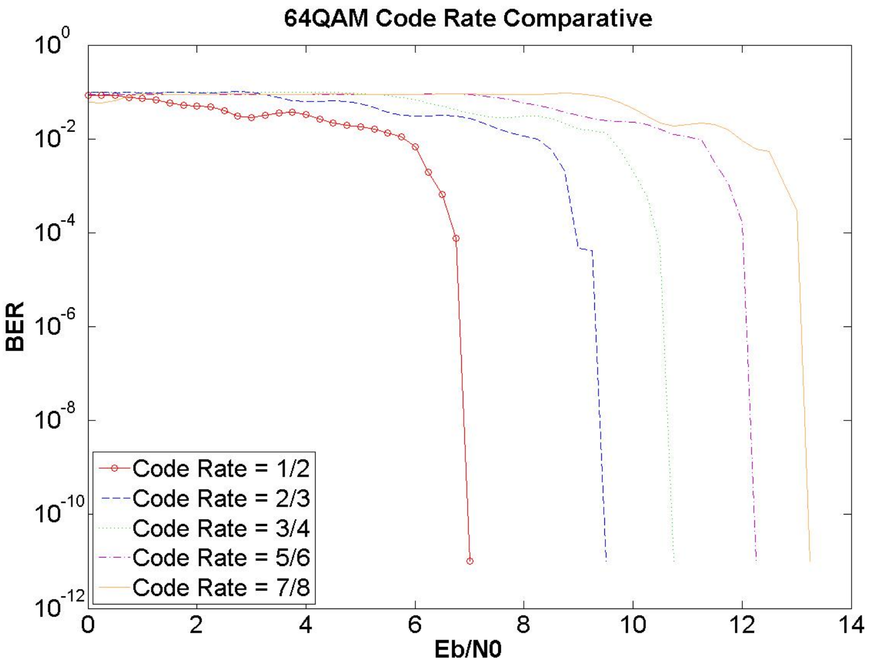


Fig. 7.11 Ricean 64QAM Rate Comparative.

In the 64QAM modulation **(Fig. 7.11)** the required Eb/No takes the higher values to reach the QEF received signal. These values are around 13 dB for the higher punctured pattern, so it is expected that this modulation will work in low noise environments with the actual standard system definitions.

Again we could use these two figures to measure the differences between the different code rates, and then faster the simulation, through the detailed calculation only for one of the code rates (usually 1/2). We could obtain:

- the difference between Gaussian and Rice channels.
- the difference between code rates for the Gaussian channel.
- the difference between code rates for the F1 channel.
- the difference between different modulation schemes.

7.4. QPSK Comparative between 2K and 8K modes

At the ETSI Standard [4] there is no distinction between both modes talking about BER. This happens since the number of sub carriers doesn't make any influence in terms of bit error probabilities ratios (bit energy per symbol will be the same) **(Fig. 7.12 and Fig. 7.13)**.

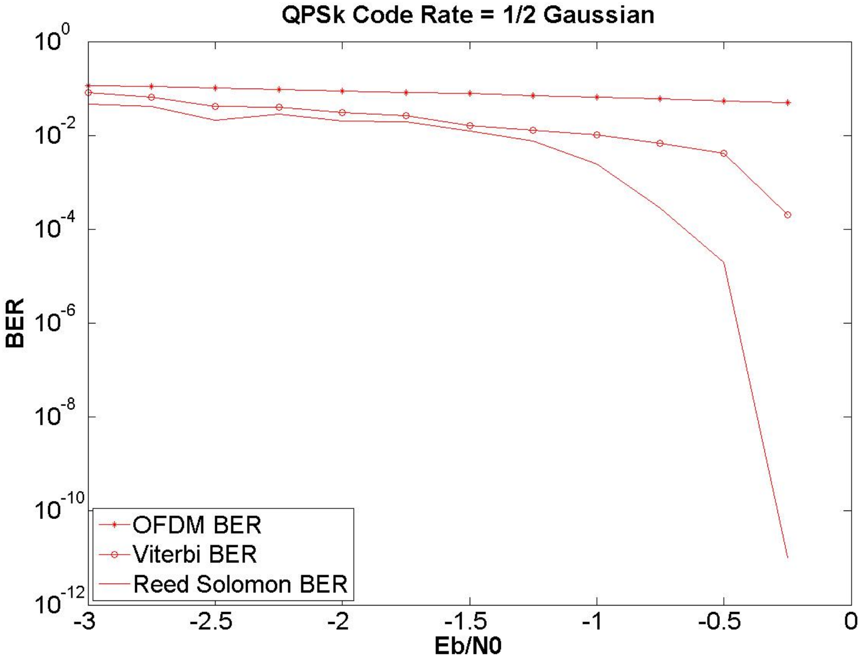


Fig. 7.12 QPSK Gaussian 2K 1/2 Code Rate.

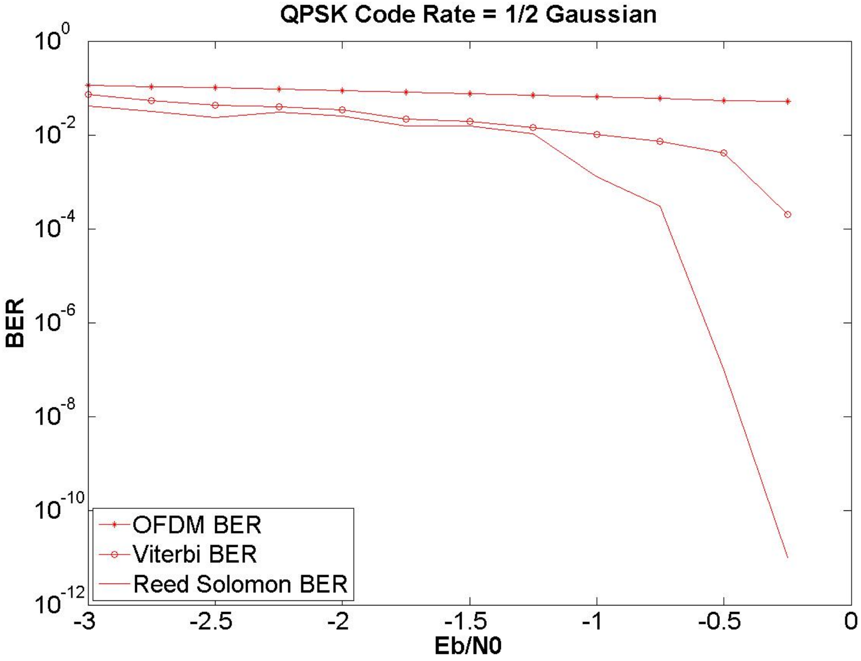


Fig. 7.13 QPSK Gaussian 8K 1/2 Code Rate.

At the results obtained by the simulator **(Fig. 7.14 and Fig. 7.15)** we can check that in the programmed system the 2K and 8K modes the curves are quite similar.

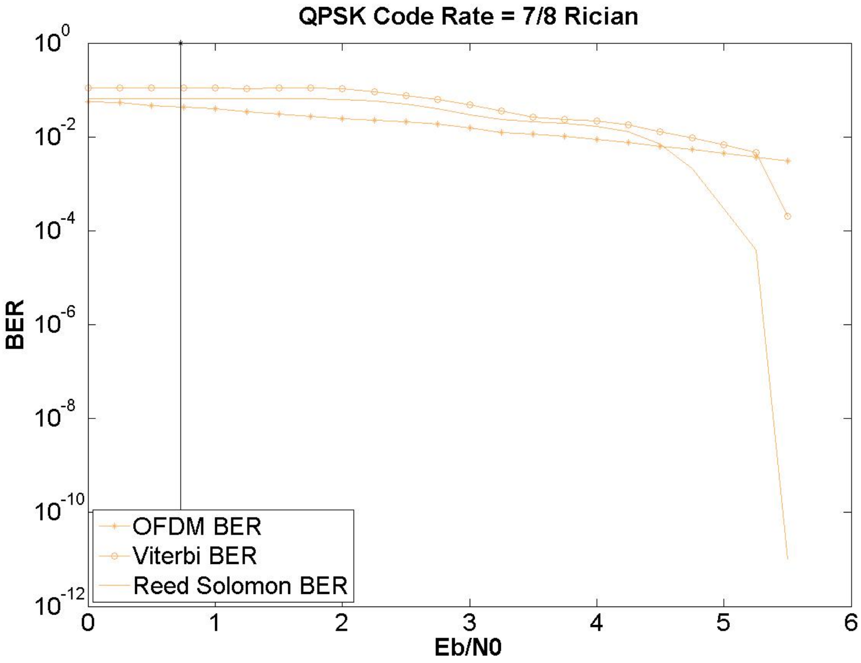


Fig. 7.14 QPSK Rician 2K 7/8 Code Rate.

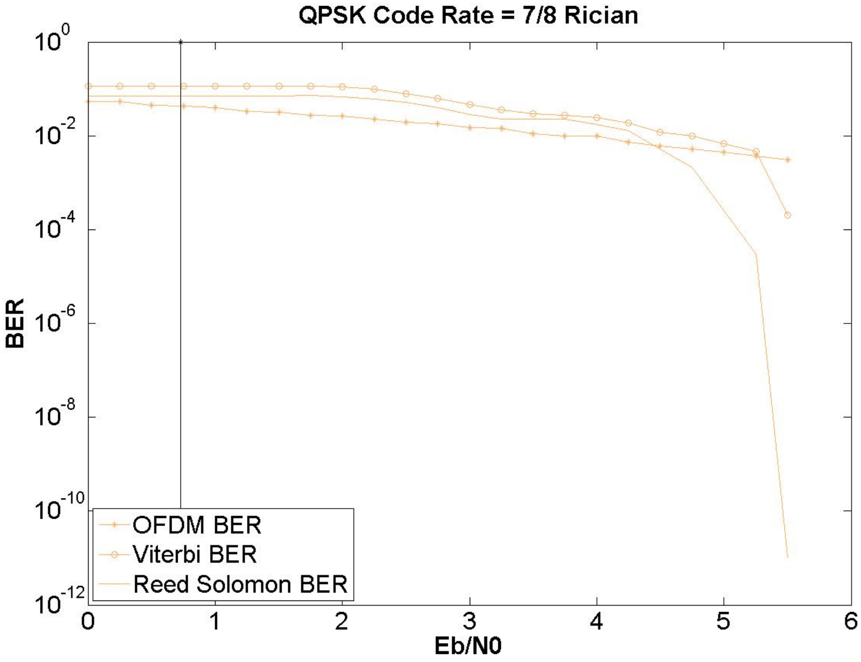


Fig. 7.15 QPSK Rician 8K 7/8 Code Rate.

7.5. New improvements

Until now, the new improvements tested in the simulator are the 256QAM modulation with all the specified code rates and the different interleavers, already described in ***Chapter 6***, programmed for that modulation.

7.5.1. 256QAM Modulation

The 256QAM allows a higher transmission data rate, however it needs higher Eb/No values to accomplish the QEF requirements **(see Table 7.3)**. With the actual system definitions adapted for this modulation the Eb/No relationships are about 16,5 dB for the 7/8 Code Rate in Gaussian Channel and about 17,75 dB for the 7/8 Code Rate in the Rician Channel. As these are high values and one of the requirements of the standard are that the transmission powers will be the same and the cell coverage should be increased, with a distance between transmitters around a 30% higher than with DVB-T, the immediate conclusion is that 256QAM can only be implemented together with other changes. The uses of concatenated codes with BCH and LDPC codes have been proposed. This has also the advantage of using codes of relative easy implementation, with better performance than Reed Solomon and Convolutional codes, and that these codes are the ones used in DVB-H. We have studied and implemented these codes, but we have had no time to calculate the improvement for the different system configurations.

Table 7.3: Required Eb/No (dB) for non hierarchical transmission to achieve a BER=2×10-4 after the Viterbi decoder for 256QAM modulation with all combination of coding rates.

		Required Eb/No for BER=2x10^{-4} after Viterbi QEF after Reed-Solomon	
Modulation	Code Rate	Gaussian Channel	Rician Channel (F1)
256QAM	1/2	10,25	10,75
256QAM	2/3	12,75	13,5
256QAM	3/4	14,25	15,0
256QAM	5/6	15,75	16,75
256QAM	7/8	16,5	17,75

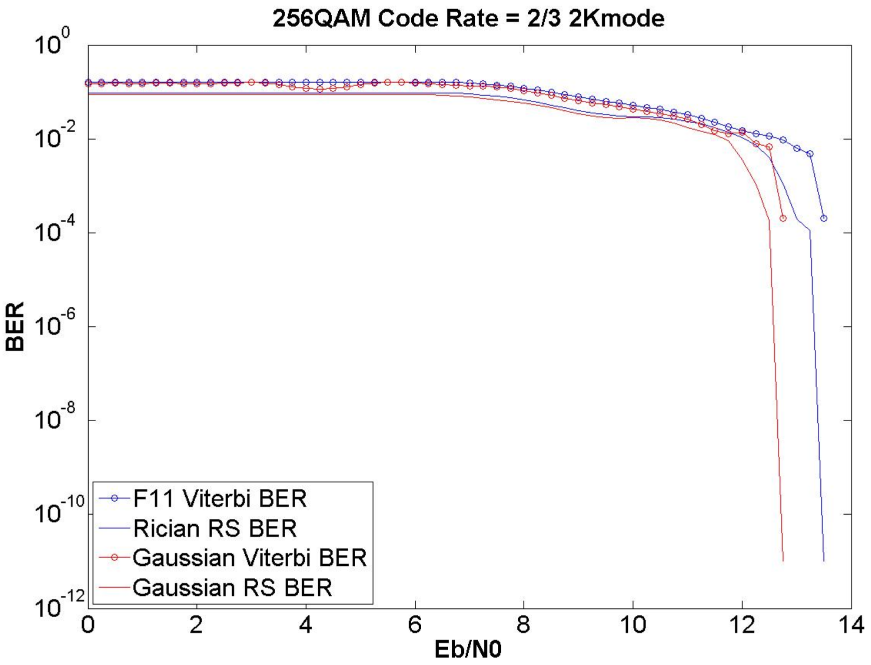


Fig. 7.16 256QAM 2K 2/3 Code Rate Comparative.

In the figures we can appreciate that although it's a new modulation, the system behaviour is the same as described by the standard for the defined modulations. However, and as it was expected, the QEF margins are allocated at higher Eb/No **(see Fig. 7.16 and Fig. 7.17)**.

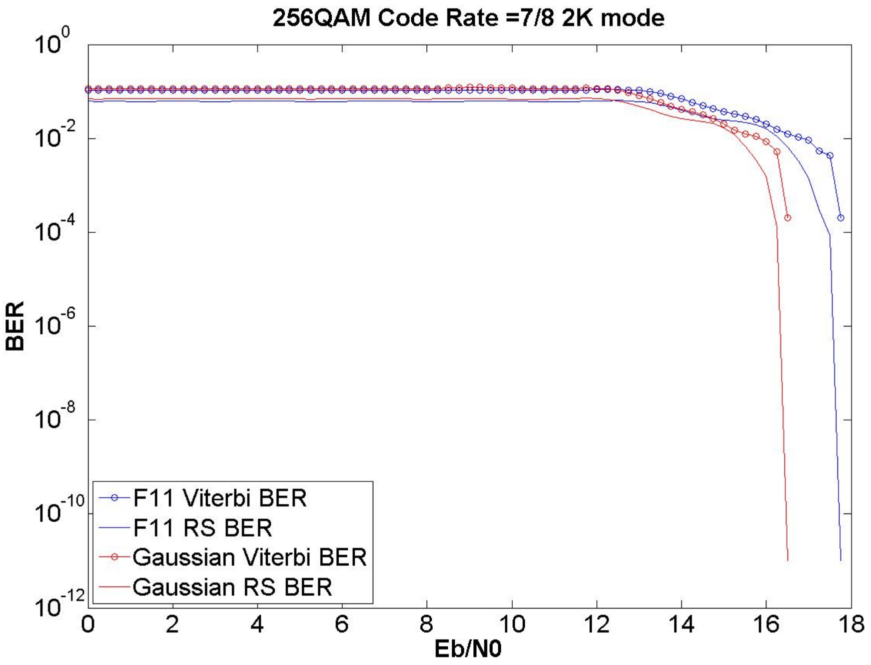


Fig. 7.17 256QAM 7/8 Code Rate Comparative.

7.5.2. 256QAM Interleaver Comparative

This section shows the results obtained by the three proposed interleavers for the 256QAM modulation. In the first proposal, the two added blocks are based in the half length of the blocks that have to be repeated. The second one repeats the two firsts blocks of bit interleavers and the third follows a different scheme based on the equality give out of the bits along all the blocks. In the graphics there is a comparison between the three proposals and the system without inner interleaver block to compare the effects and the gain introduced by this block. These graphics are simulated with a 1/2 code rate **(Fig. 7.18 and 7.19)**.

The results of the proposed interleavers shows that in an AWGN (channels without error bursts), the three interleavers proposed have a similar behaviour because in these kind of channels doesn't make sense re-order the information due to this absence of error bursts. For that reason interleavers don't introduce any gain in that kind of channels **(Fig. 7.22)**.

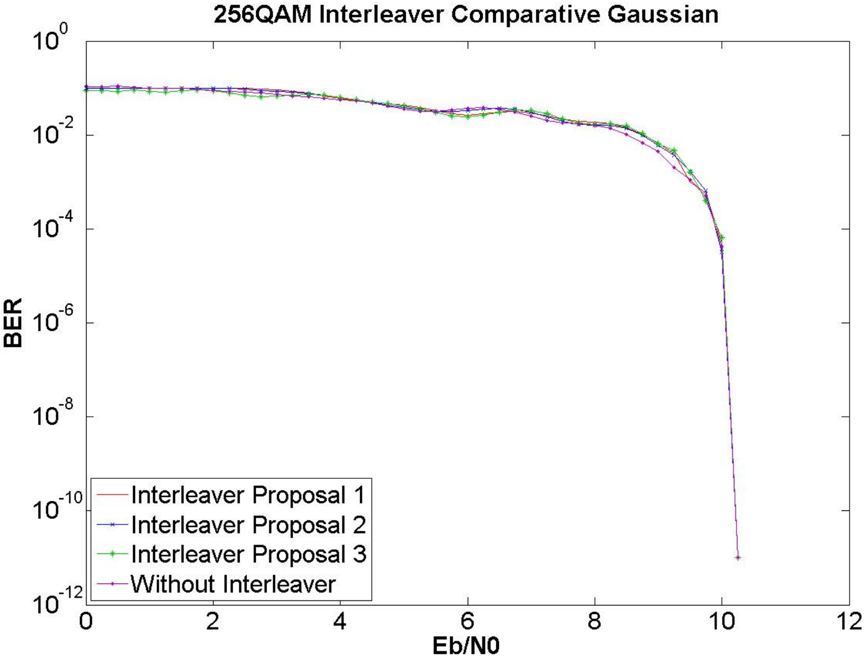


Fig. 7.18 Gaussian 256QAM Interleaver Comparative.

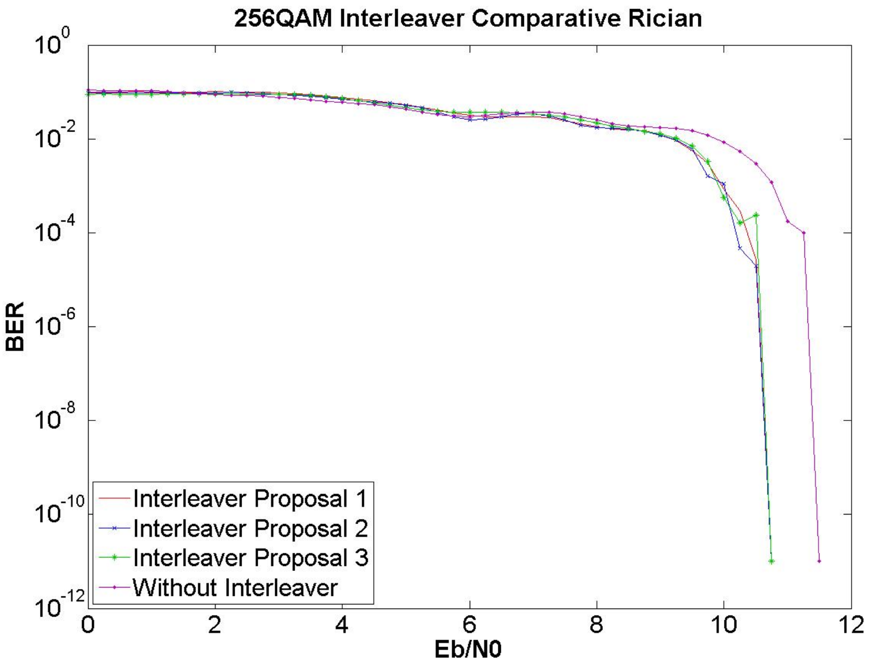


Fig. 7.19 Rician 256QAM Interleaver Comparative.

There is a significant gain when the system is simulated in an error burst scenario (F1 channel model) and there are no interleaver blocks **(Fig. 7.19)**. In this case the gain introduced by the inner interleavers is around 1 dB. There are small differences between the proposed interleavers so each of them could be used indistinctly. Anyway, before deciding which one is the best option, we should test their performance using the mobile channel model, which as we mentioned previously has been programmed and tested but still no integrated with the rest of the simulation blocks.

CHAPTER 8: CONCLUSIONS AND FUTURE RESEARCH LINES

In the development of this project we have studied and implemented the main blocks of almost any radio communication system and specifically the ones defined by the DVB-T standard. All of the work has been performed inside the time schedule established by the FURIA project. We have developed a simulation platform in *MatLab* (was one of the requirements of the project). There is a big difference between studying, through the different subjects along the telecommunication studies, the main principles of coding, interleaving, filtering, the problems associated with real channels, and implementing them. Along the months of execution of the project, many times, we have learned that ideas and concepts that we thought clearly understand, were not so easy, as we initially thought.

Now, we have a platform with the DVB-T standard, and we are in optimum condition to start to analyse several improvements over the standard, testing and comparing them with the actual features.

As we have seen along this report, the simulator has been programmed to work with all the modulations, code rates and modes defined in the actual standard. The complete simulator includes the possibility to do tests with different channel models (Gaussian, F11 and TU6) and different channel estimators (ideal and average). It also includes the possibility to make graphic comparatives between all of these parameters.

Furthermore through the development of the simulation platform, we have deeped our knowledge and studied different encoding and interleavement techniques, OFDM, modulations, pilot's insertion, etc. to implement them in a simulation platform. Also we have had to check all the results obtained by this simulator and compare them with the standard requirements, adjusting always all the parameters to accomplish the standard objectives, and to check that the simulator works properly.

On the other hand, during this period and in parallel with the main core of the project, we have also studied and proposed new improvements of the standard. Those improvements take full advantage of the actual system structure, taking care to not make changes in significant aspects.

The first one has been the 256-QAM modulation. In this improvement we have make three proposals taking care in the restriction that this interleaver should respect the algorithm defined by the standard. The reason of taking this modulation is that it will improve faster transmission speed. However it requires a high C\N ratio and it seems that it only would be available in line of sight (LOS) environments due to its requirements.

After this improvement we have also deployed a new channel definition according to DVB definitions for the DVB-T2 standard. Since one of the main objectives for the DVB-T2 is the correct signal reception for mobile devices we have implemented the TU6 channel. Nowadays this feature is in a debugging period to optimize it. Simulating this channel we expect to see how hostile is it and how necessary are all these new techniques to reach the necessary parameters for a quality transmission for mobile TV.

Finally talking about future investigation lines our next main investigation, as seen in ***Chapter 6***, is the use of transmitter/receiver diversity as well as MIMO techniques and new encoding techniques.

With MIMO techniques we hope to obtain an antenna diversity gain, enough to minimize the C\N ratio, so then we can obtain higher coverage areas, and also higher modulation orders will be able to work in worst channel conditions.

The second future improvement to perform is the implementation of a BCH – LDPC structure (used in new DVB-S2 standard and recent Chinese DTT standard) instead of the classical Reed-Solomon - Viterbi (used in DVB-T and DVB-S standards). With LDPC codes will improve lower calculation complexity comparing with the *Turbo Codes* in the transmitter and in the receiver if the optimal algorithms are employed.

BIBLIOGRAPHY

[1] FURIA Project: www.furiapse.com
Official webpage of FURIA council.

[2] IEEE 802.11: "IEEE Standard for Information Technology-Telecommunications and Information Exchange between systems". 1999 Edition.

[3] Spanish BTTD: www.televisiondigital.es
Official webpage of BTTD of Spanish Ministry of Industry, Tourism and Commerce.

[4] ETSI EN 300 744: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T)". November 2004.

[5] DVB Organization: www.dvb.org
Official webpage of DVB European Organization.

[6] ETR 154: "Digital Video Broadcasting (DVB); Implementation Guidelines for the use of MPEG-2 Systems, Video and Audio in Satellite, Cable and Terrestrial Broadcasting Applications"

[7] ISO/IEC 13818 (Parts 1 to 3): "Information technology – Generic coding of moving pictures and associated audio information".

[8] ETSI EN 300 421: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services".

[9] ETSI EN 300 429: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services".

[10] Gerard Faria, Jukka A. Henriksson, Erik Stare and Pekka Talmola: "DVB-H: Digital Broadcast Services to Handheld Devices" Proceedings of the IEEE. Vol. 99, January 2006.

[11] ", S.M.Alamouti, "A Simple Transmitt diversity technique for wireless communications" IEEE, Journal of Selected Areas in Communications, vol. 16, October 1998.

[12] Golden et al, "Detection algorithm and initial laboratory results using V-BLAST space time communication architecture", Electronic Letters, vol.35, 1999

[13] "Obtaining diversity gain for DTV by using MIMO structure in SFN", L.Zhang et al, IEEE Transactions on Broadcasting, vol.50, March 2004

[14] J.Wang et al, "A general SFN structure with transmitt diversity for TDS-OFDM System", IEEE Transactions on Broadcasting, vol.52, June 2006.

[15] Mignone, V.; Morello, A., "CD3-OFDM: a novel demodulation scheme for fixed and mobile receivers", IEEE Transaction on Communication. Volume 44, Issue 9, Sept. 1996

[16] Peng Liu; Bing-bing Li; Zhao-yang Lu; Feng-kui Gong, “A novel symbol synchronization scheme for OFDM International Conference on Communications”, Circuits and Systems, 2005. Vol.1, May 2005.

[17] ETSI EN 302 304: “Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)”. November 2004.

[18] ATSC Digital Television Standard (A/53) Revision E, with Amendments No 1 and 2. December 2005.

[19] ITU-R 205/11. Channel coding, frame structure and modulation scheme for terrestrial integrated services digital broadcasting. (ISDB-T). March 1999.

[20] Jiang Song. Technical Review of the Chinese Digital Terrestrial Television Broadcasting Standard (DTMB). December, 2006.

[21] EWC HT PHY Specification. December 2005.

[22] IEEE Standard for Local and metropolitan area networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. December 2005.

[23] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near Shannon limit error correcting coding and encoding: Turbo-codes (1). Pages 1064 – 1070, 1993.

[24] R. G. Gallager. Low-density parity-check codes. IEEE Transactions on Information Theory, 8(1):21 – 28, 1962.

[25] J.-F. Cheng and R.J. McEliece. Some high-rate near capacity codecs for the Gaussian channel. Proceedings. Thirty-Fourth Annual Allerton Conference on Communication, Control, and Computing, pages 494 – 503, 1996//.

[26] L. Ping, X. Huang, and N. Phamdo. Zigzag codes and concatenated zigzag codes. IEEE Transactions on Information Theory, 47(2):800 – 807, 2001.

[27] Xiaofu Wu, Yingjian Xue, and Haige Xiang. On concatenated zigzag codes and their decoding schemes. IEEE Communications Letters, 8(1):54 – 56, 2004.

[28] S.Yoon et al, “Transaction Level analysis of NoC based coded MIMO-OFDM Receiver”, WCNC 2006

[29] Kafle, P.L.: “On the performance of MC-CDMA with interleaved concatenated coding and interference cancellation for high-rate data transmission” IEEE International Conf. on Communications, ICC 2002.

[30] Thomas J. Richardson, M. Amin Shokrollahi*, Member, IEEE*, and Rüdiger L. Urbanke: ”Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes”. IEEE Transactions on information theory, vol. 47, no. 2, FEBRUARY 2001.

[31] ETSI EN 302 307: “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for

Broadcasting, Interactive Services, News Gathering and other broadband satellite applications”.

[32] W.C. Jakes, Ed., “Mircrowave Mobile Communications”. New York: Wiley, 1974.

Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANNEX 1

TFC TITTLE: Design of a simulation platform to test next generation of terrestrial DVB

DEGREE: Technical Communications Engineering, speciality Telecommunications Systems

AUTHORS: Carlos Enrique Herrero
Carlos Alberto López Arranz

DIRECTORS: Silvia Ruiz Boqué
Luís Alonso Zárate

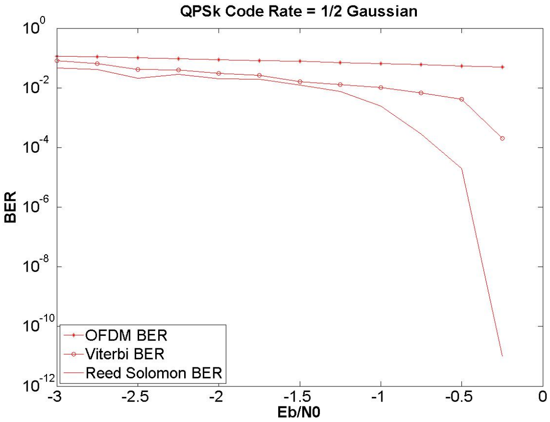
DATE: July, 19th of 2007

ANNEX 1

A1.1. Simulation Results

A1.1.1. QPSK Modulation

A1.1.1.1. 1/2 Code Rate



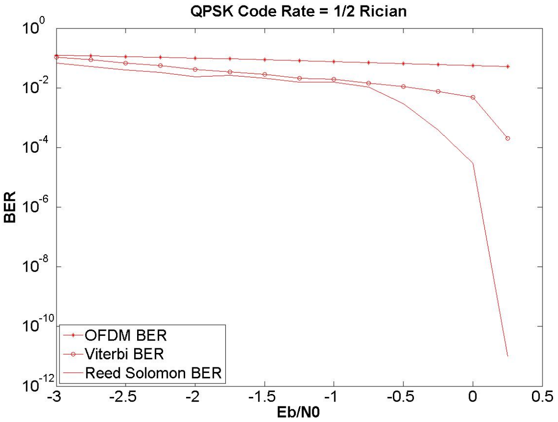
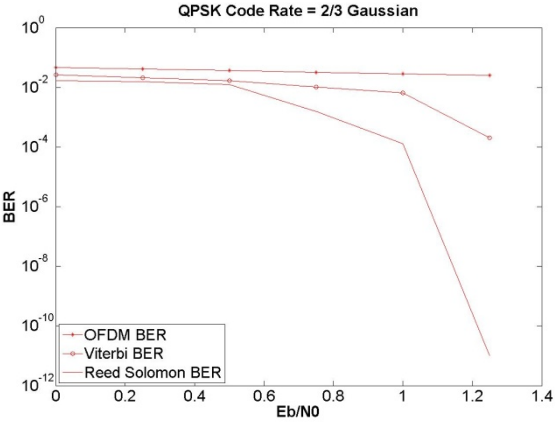


Fig. A1.1.1 Gaussian 1/2 Code rate. **Fig. A1.1.2** Rician 1/2 Code rate.

A1.1.1.2. 2/3 Code Rate



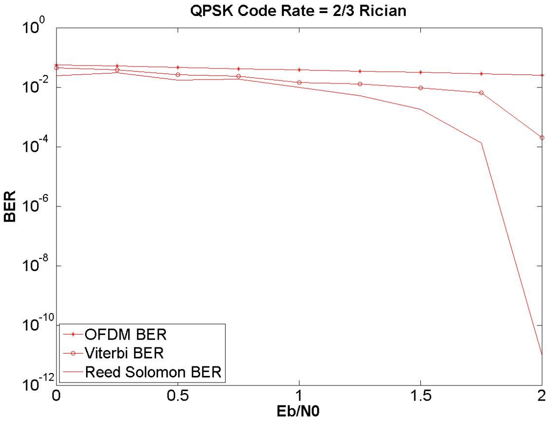
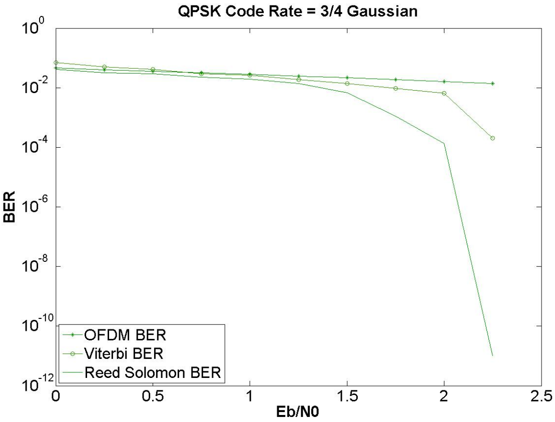


Fig. A1.1.3 Gaussian 2/3 Code rate. **Fig. A1.1.4** Rician 2/3 Code rate.

A1.1.1.3. 3/4 Code Rate



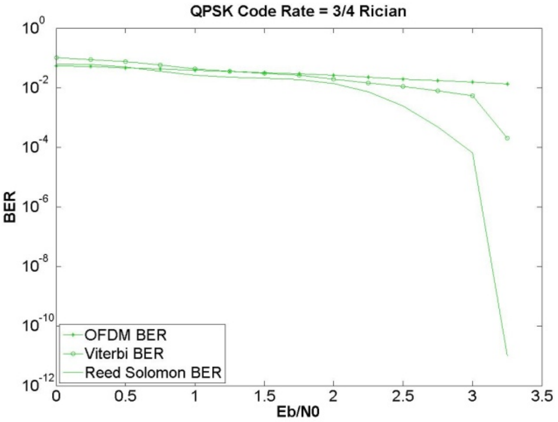
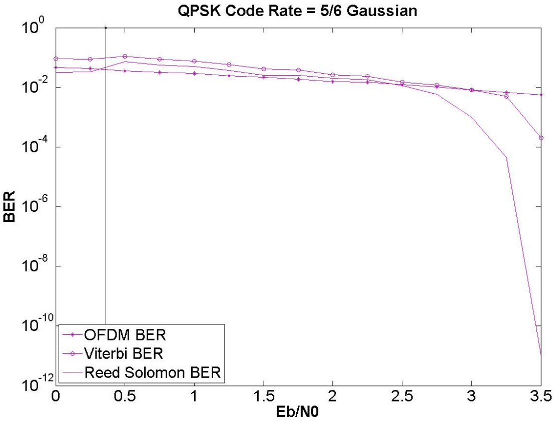


Fig. A1.1.5 Gaussian 3/4 Code rate.

Fig. A1.1.6 Rician 3/4 Code rate.

A1.1.1.4. 5/6 Code Rate



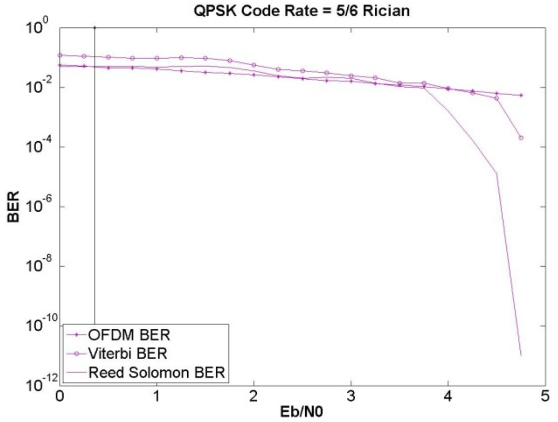
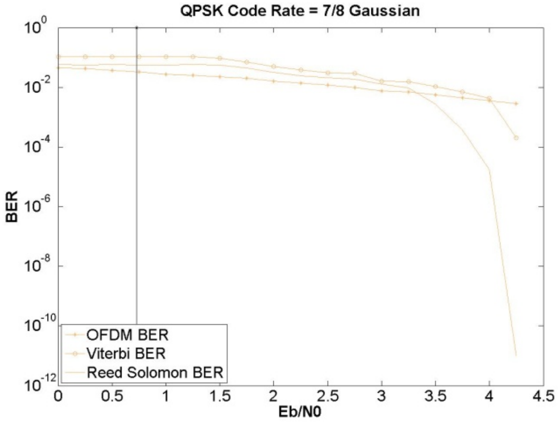


Fig. A1.1.7 Gaussian 5/6 Code rate.

Fig. A1.1.8 Rician 5/6 Code rate.

A1.1.1.5. 7/8 Code Rate



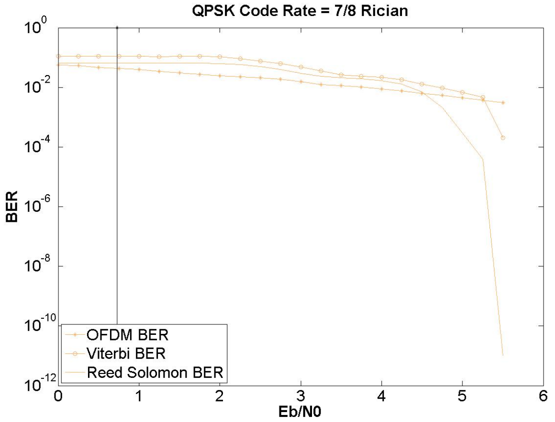
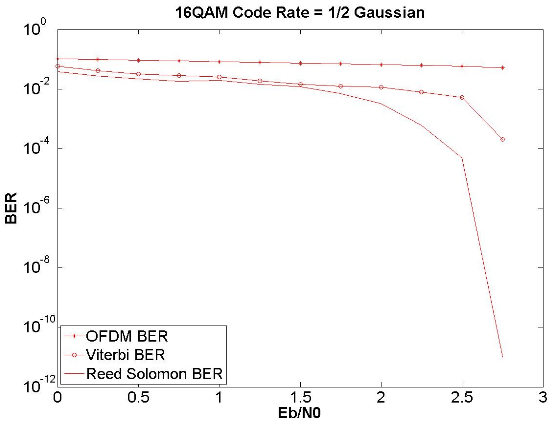


Fig. A1.1.9 Gaussian 7/8 Code rate.

Fig. A1.1.10 Rician 7/8 Code rate.

A1.1.2. 16QAM Modulation

A1.1.2.1 1/2 Code Rate



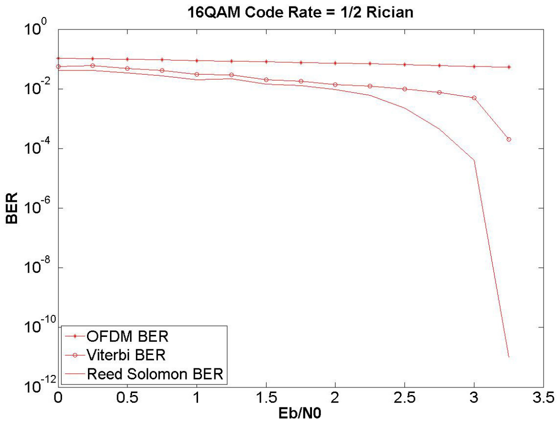
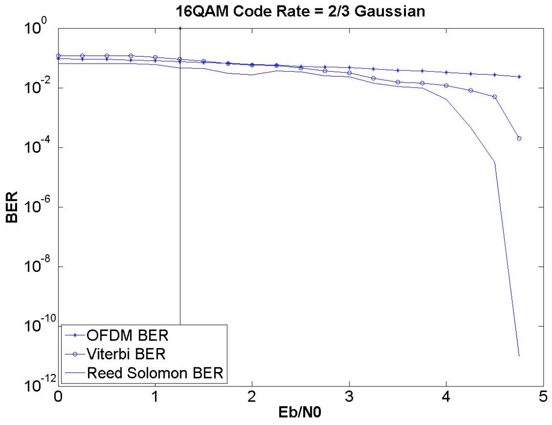


Fig. A1.1.11 Gaussian 1/2 Code rate. **Fig. A1.1.12** Rician 1/2 Code rate.

A1.1.2.2. 2/3 Code Rate



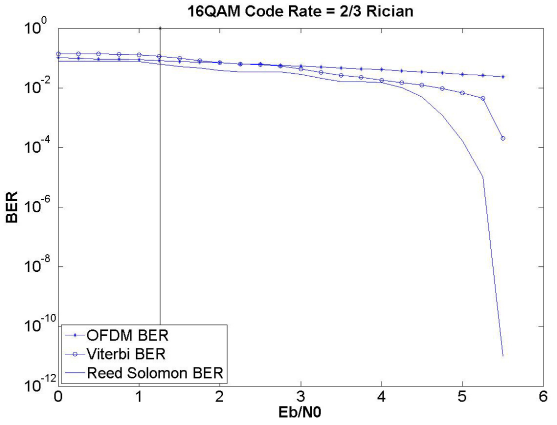
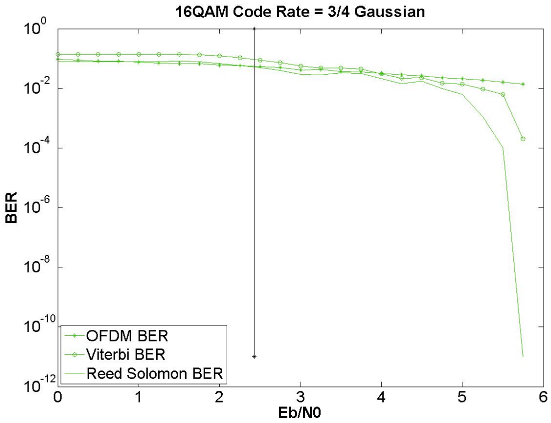


Fig. A.1.1.13 Gaussian 2/3 Code rate. **Fig. A1.1.14** Rician 2/3 Code rate.

A1.1.2.3. 3/4 Code Rate



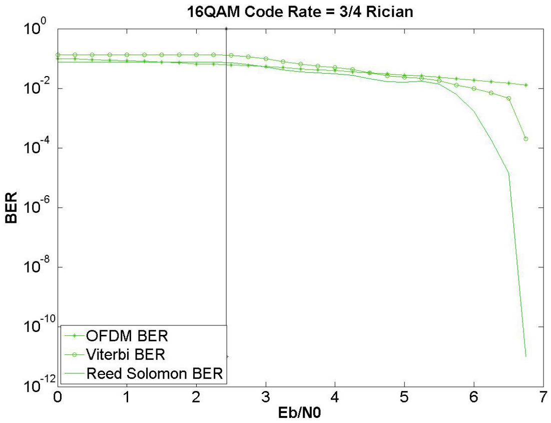
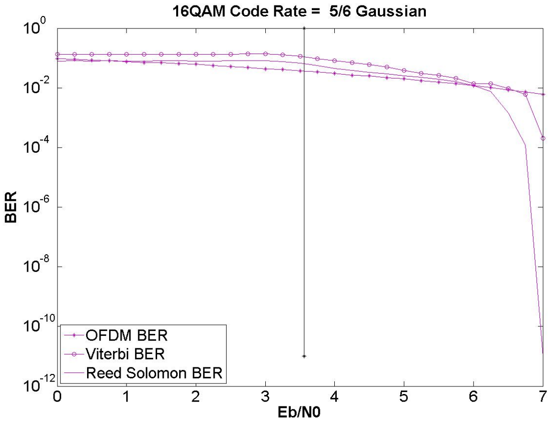


Fig. A1.1.15 Gaussian 3/4 Code rate. **Fig. A1.1.16** Rician 3/4 Code rate.

A1.1.2.4. 5/6 Code Rate



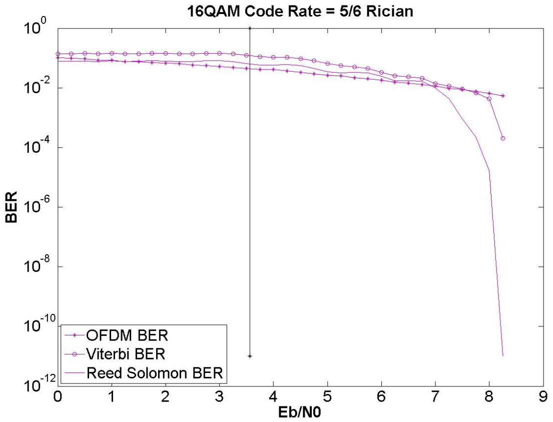
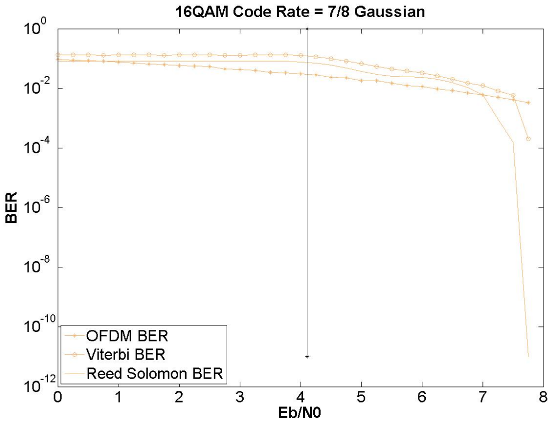


Fig. A1.1.17 Gaussian 5/6 Code rate. **Fig. A1.1.18** Rician 5/6 Code rate.

A1.1.2.5. 7/8 Code Rate



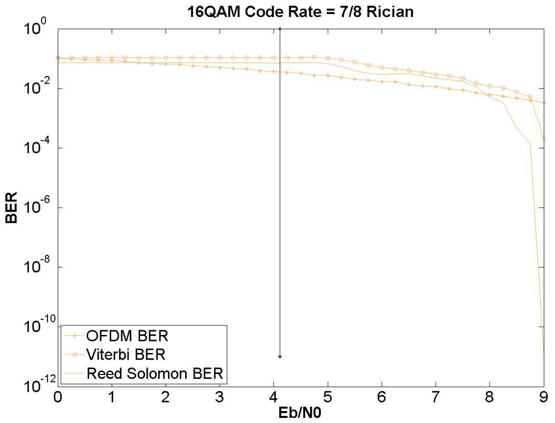
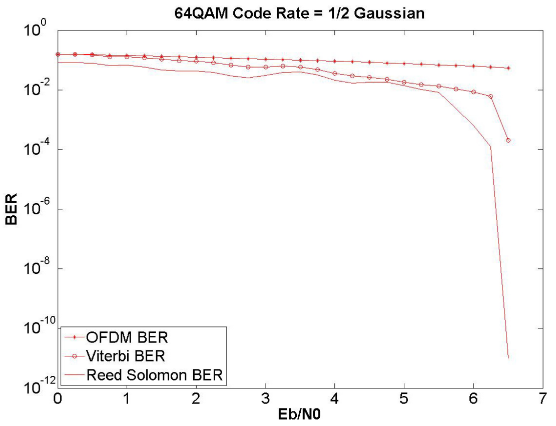


Fig. A1.1.19 Gaussian 7/8 Code rate. **Fig. A1.1.20** Rician 7/8 Code rate

A1.1.3. 64QAM Modulation

A1.1.3.1. 1/2 Code Rate



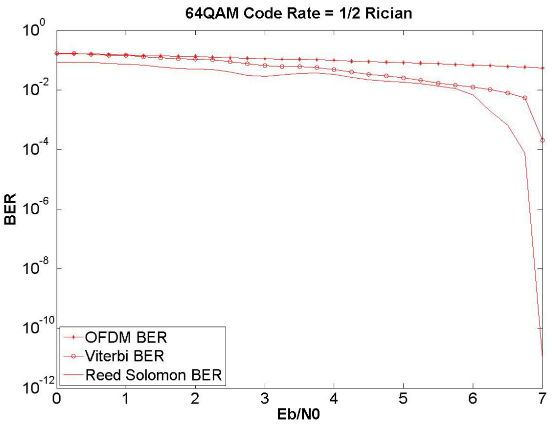
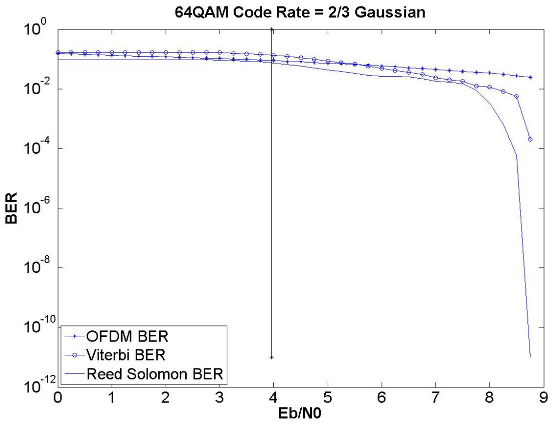


Fig. A1.1.21 Gaussian 1/2 Code rate **Fig. A1.1.22** Rician 1/2 Code rate.

A1.1.3.2. 2/3 Code Rate



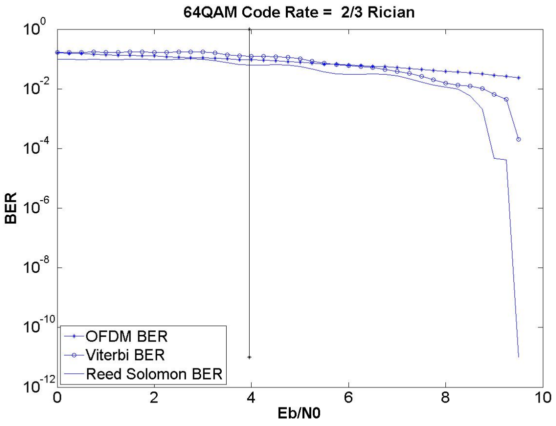


Fig. A1.1.23 Gaussian 2/3 Code rate. **Fig. A1.1.24** Rician 2/3 Code rate.

A1.1.3.3. 3/4 Code Rate

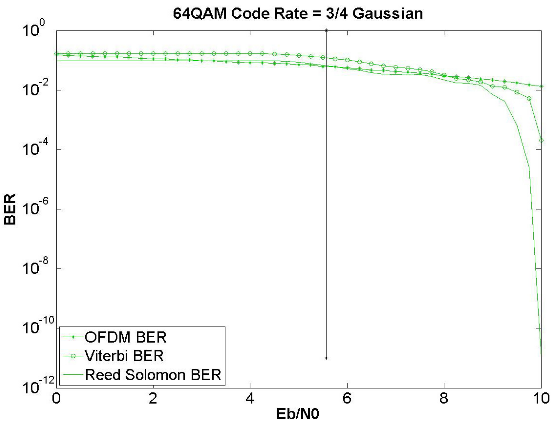


Fig. A1.1.25 Gaussian 3/4 Code rate.

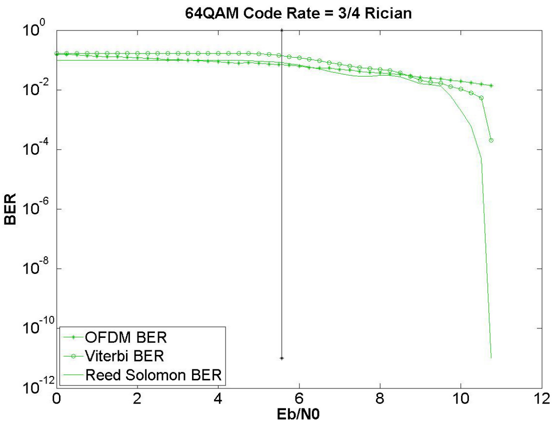


Fig. A1.1.26 Rician 3/4 Code rate.

A1.1.3.4. 5/6 Code Rate

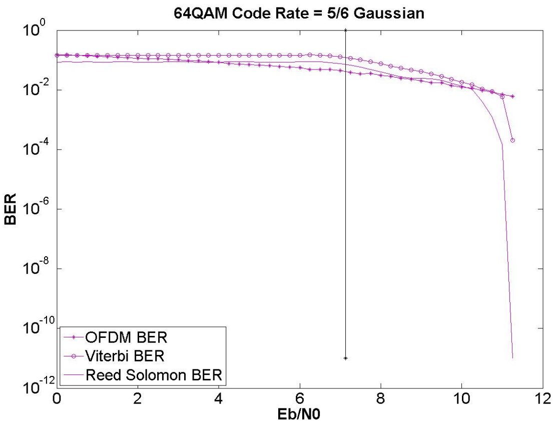


Fig. A1.1.27 Gaussian 5/6 Code rate.

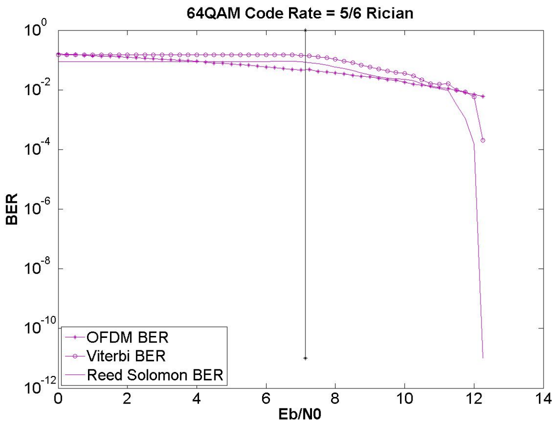


Fig. A1.1.28 Rician 5/6 Code rate.

A1.1.3.5. 7/8 Code Rate

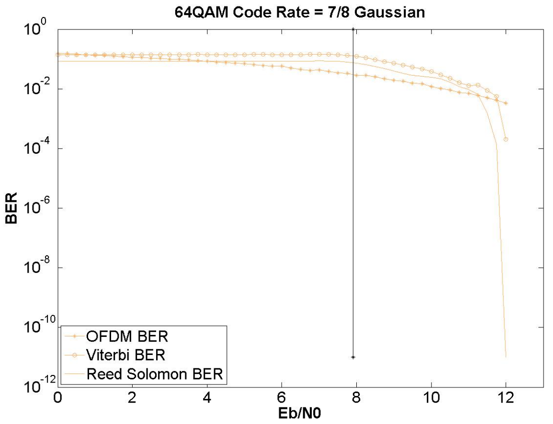


Fig. A1.1.29 Gaussian 7/8 Code rate.

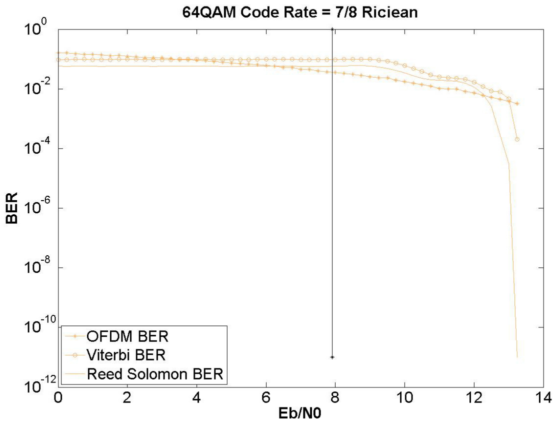
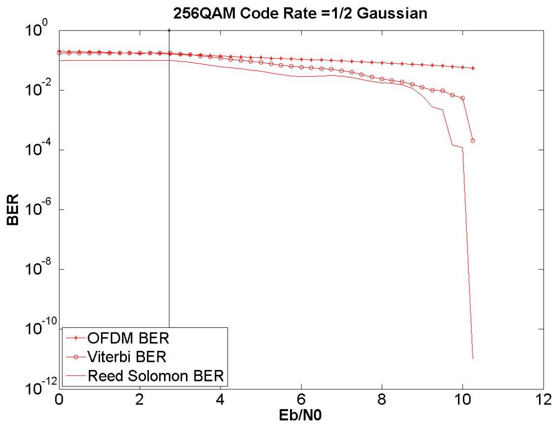


Fig. A1.1.30 Rician 7/8 Code rate.

A1.1.4. 256QAM Modulation

A1.1.4.1. 1/2 Code Rate



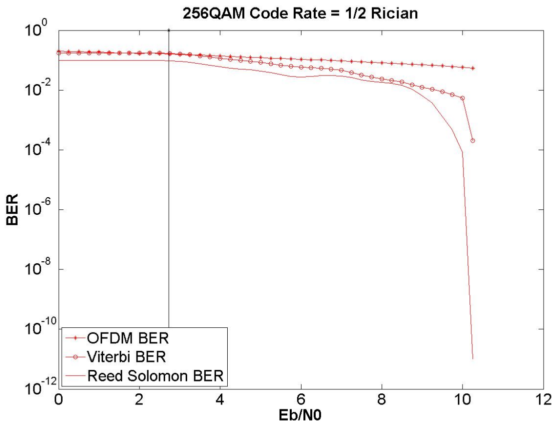
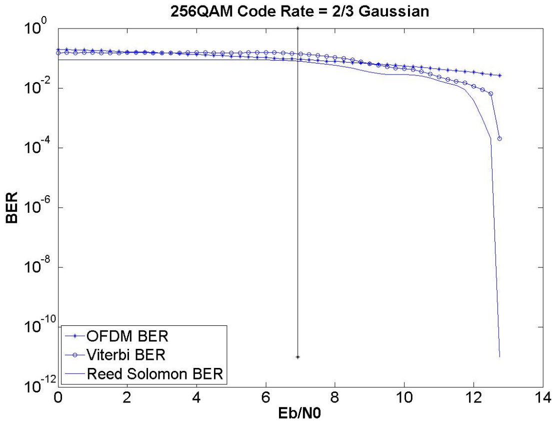


Fig. A1.1.31 Gaussian 1/2 Code rate. **Fig. A1.1.32** Rician 1/2 Code rate.

A1.1.4.2. 2/3 Code Rate



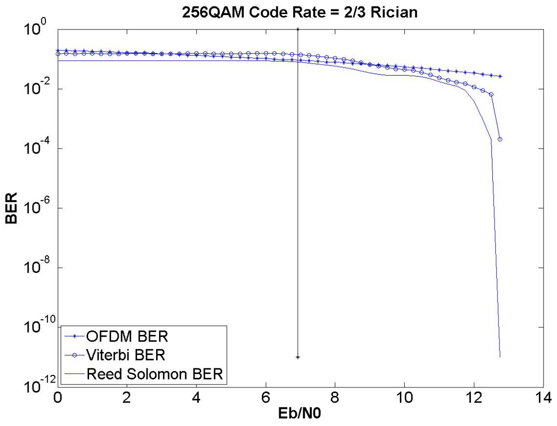
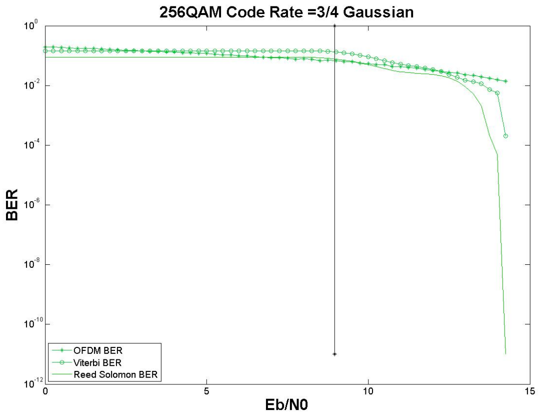


Fig. A1.1.33 Gaussian 2/3 Code rate. **Fig. A1.1.34** Rician 2/3 Code rate.

A1.1.4.3. 3/4 Code Rate



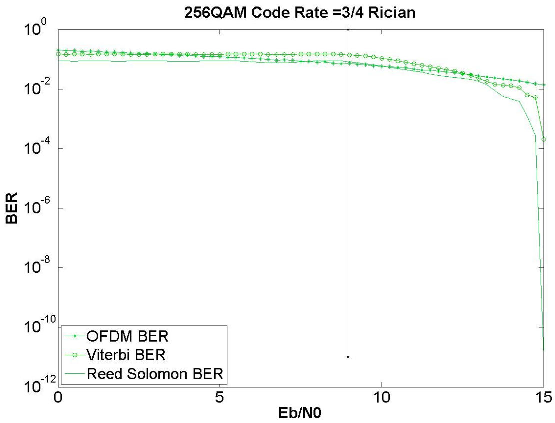
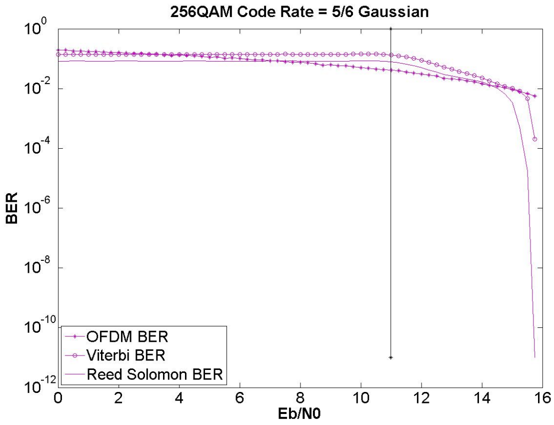


Fig. A1.1.35 Gaussian 3/4 Code rate. **Fig. A1.1.36** Rician 3/4 Code rate.

A1.1.4.4. 5/6 Code Rate



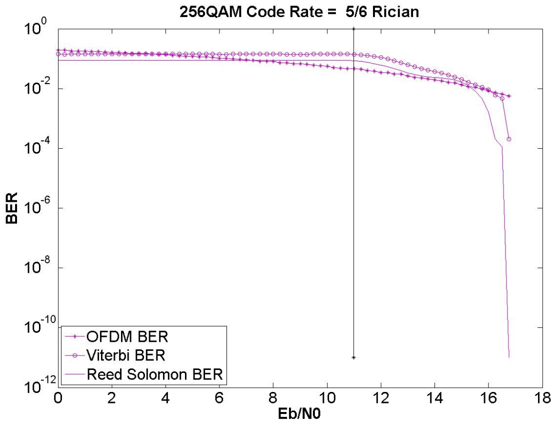
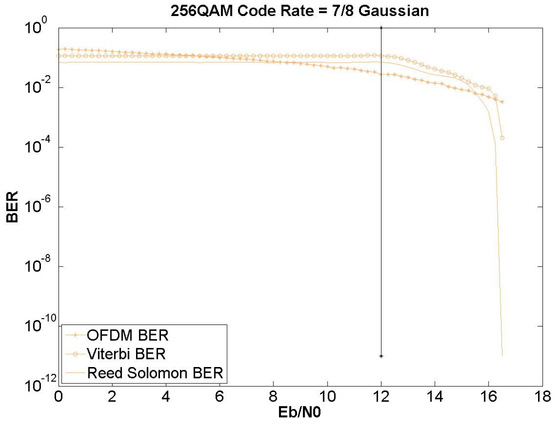


Fig. A1.1.37 Gaussian 5/6 Code rate. **Fig. A1.1.38** Rician 5/6 Code rate.

A1.1.4.5. 7/8 Code Rate



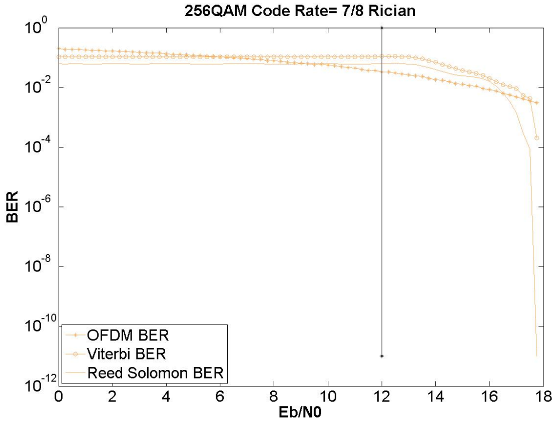


Fig. A1.1.39 Gaussian 7/8 Code rate. **Fig.A1.1.40** Rician 7/8 Code rate.

A1.2. Comparative of ETSI and simulator results

In the following graphics is represented the comparative between the ETSI standard Eb\No requirements and the results obtained by the simulation platform. In first one we can see the Gaussian Channel Comparative **(Fig. A1.2.1)** and in the second one there is represented the Rician Channel comparative **(Fig. A1.2.2)**.

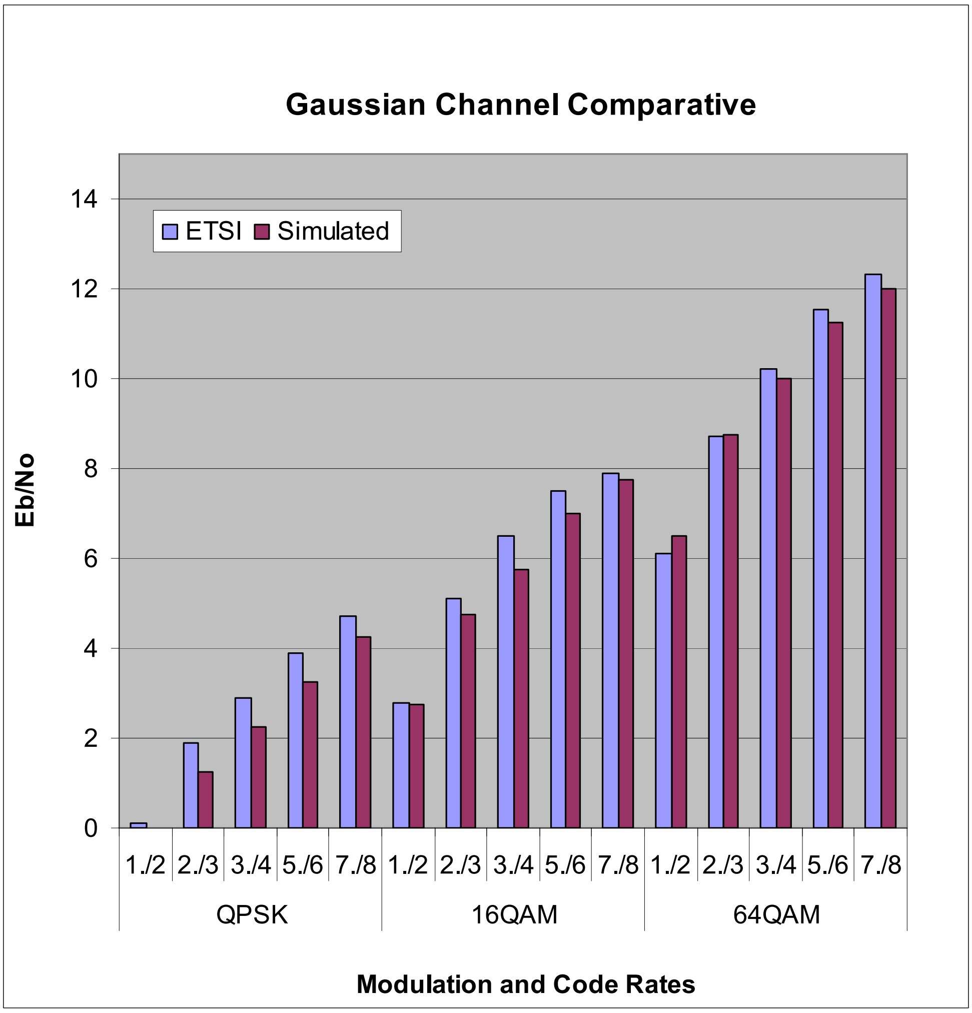


Fig. A1.2.1 Gaussian Channel Comparative for all the modulation types and code rates.

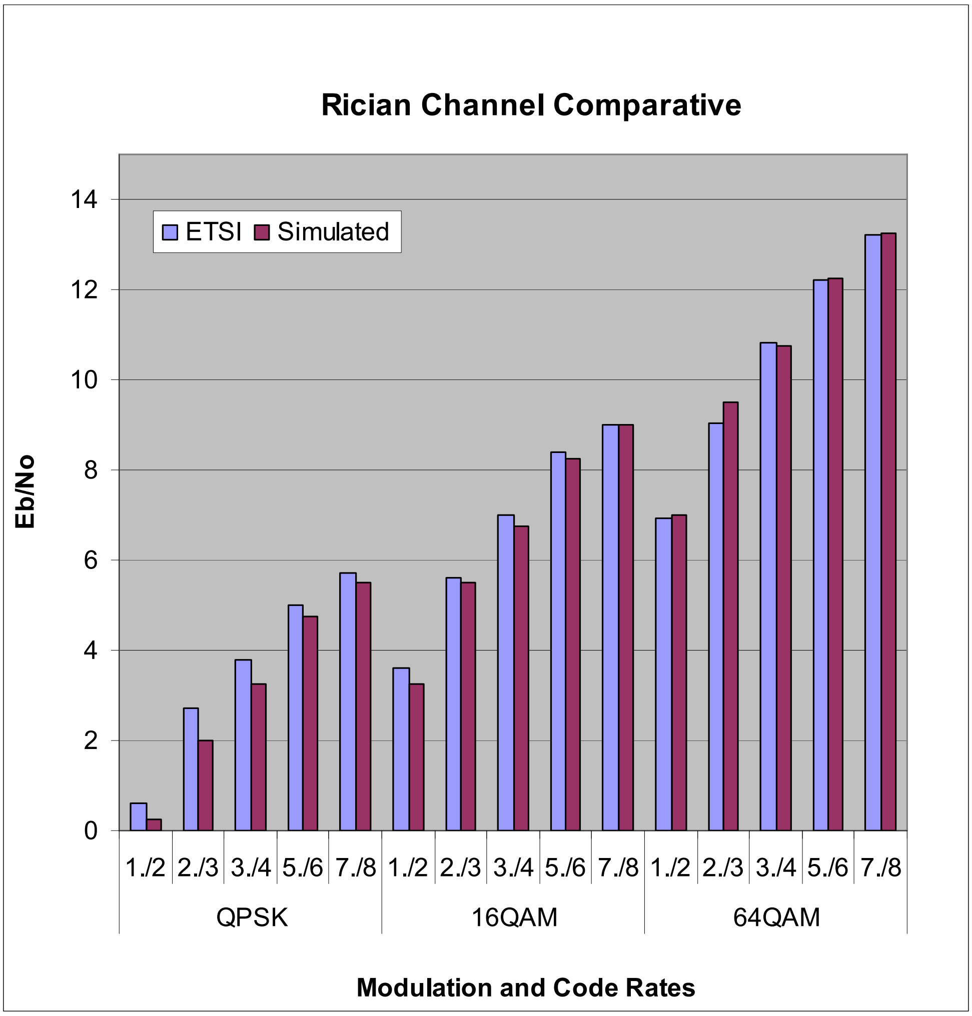


Fig. A1.2.2 Rician Channel Comparative for all the modulation types and code rates.