

TELECOMUNICAÇÕES

Revista do Instituto Nacional de Telecomunicações

Inatel

Editorial

Tutoriais

Interactivity in Terrestrial DTV Systems: the DVB-RCT Standard
Ângelo Pinelli Martins Samia e Dayan Adionel Guimarães - INATEL 1

*Noções Sobre Decodificação Turbo: Um Estudo de Caso
para os Códigos SPC-TPC*
Dayan Adionel Guimarães - INATEL 15

Introdução à Equalização Turbo
André Fonseca dos Santos e Dayan Adionel Guimarães - INATEL 30

Revista Científica/Periódica publicada pelo Inatel - Instituto Nacional de Telecomunicações

Diretor: Wander Wilson Chaves

Vice-diretor: Carlos Roberto dos Santos

Conselho Editorial

Editor

José Marcos Câmara Brito - Inatel

Geraldo Gil Ramundo Gomes - Inatel

José Antônio Justino Ribeiro - Inatel

Maurício Silveira - Inatel

Corpo Editorial

Alfredo Goldman - USP

Anderson S. L. Gomes - UFPE

Anilton Salles Garcia - UFES

Carlos Alberto Ynoguti - Inatel

César Kyn d'Ávila - CEDET

Dayan Adionel Guimarães - Inatel

Dilson Frota de Moraes - Leucotron Equipamentos Ltda.

Edson Moschim - UNICAMP

Eduardo César Grizendi - Rail Com

Francisco José Fraga da Silva - Inatel

Franco Callegati - DEIS

Geraldo Gil R. Gomes - Inatel

Guilherme Augusto Barucke Marcondes - Inatel

Helio Waldman - UNICAMP

Ivanil S. Bonatti - UNICAMP

Jaime Portugheis - UNICAMP

João César Moura Mota - UFC

Joel J. P. C. Rodrigues - UBI (Portugal)

José Antônio Justino Ribeiro - Inatel

José de Souza Lima - LINEAR

José Edimar Barbosa Oliveira - ITA

José Ferreira de Rezende - UFRJ

Jorge Moreira de Souza - FITec

Júlio César Tibúrcio - Inatel

Luiz Geraldo Pedroso Meloni - UNICAMP

Marcelo Sampaio de Alencar - UFCG

Márcio Lourival Xavier dos Santos - UNITAU

Maria Regina Campos Caputo - Inatel

Markus Endler - PUC - RJ

Marlene Sabino Pontes - CETUC

Martin Zieher - FHTE (Alemanha)

Nelson Soares Wisnik - N. Wisnik Consultoria

Omar Carvalho Branquinho - PUC-Campinas

Paulo Gomide Cohn - Embassy Systems

Pierre Kaufmann - Mackenzie/INPE/UNICAMP

Rainer Doster - FHTE (Alemanha)

Reginaldo Palazzo Júnior - UNICAMP

Renato Baldini Filho - UNICAMP

Sandro Adriano Fasolo - Inatel

Silvio Ernesto Barbin - EPUSP

Shusaburo Motoyama - UNICAMP

Wilton Ney do Amaral Pereira - Inatel

Yuzo Iano - UNICAMP

Expediente

Assessoria de Comunicação & Marketing - ASCOM
e-mail: ascom@inatel.br

Diagramação
Centro de Informações Científicas e Tecnológicas
Setor de Editoração Eletrônica

Tiragem: 3.000 exemplares
Distribuição Gratuita

Instituto Nacional de Telecomunicações
Av. João de Camargo, 510
Caixa Postal: 05
Santa Rita do Sapucaí - MG - BRASIL
CEP 37540-000

Tel: (35) 3471.9200 Fax: (35) 3471.9314

e-mail: inatel@inatel.br

<http://www.inatel.br>

EDITORIAL

Com este número a revista Telecomunicações entra em seu oitavo ano de edição. O projeto que se iniciou com muitas incertezas, no segundo semestre de 1998, consolidou-se ao longo do tempo e constitui hoje uma importante alternativa para que a comunidade ligada ao setor de telecomunicações possa divulgar os resultados de seus trabalhos de pesquisa e desenvolvimento.

Nos treze exemplares editados tivemos 91 artigos publicados, envolvendo mais de uma centena de autores de mais de vinte instituições de ensino e pesquisa do Brasil e do exterior. Setenta e seis por cento dos artigos publicados envolveram autores externos ao Inatel, o que mostra que a revista Telecomunicações foi amplamente aceita pela comunidade acadêmica como veículo de divulgação científica, orientando-nos no sentido de continuar empreendendo esforços para o fortalecimento e a expansão do projeto. A penetração da revista no meio acadêmico está refletida, também, na extensão de seu corpo editorial, formado por 43 pesquisadores de 24 instituições.

Após atuar como editor da revista desde seu primeiro número, creio que cumpri a minha missão e, portanto, é chegada a hora de transferir a responsabilidade de condução deste projeto para outra pessoa. Gostaria de dar as boas-vindas e desejar boa sorte ao novo editor, Professor Carlos Alberto Ynoguti, que, com sua competência e dedicação, tenho certeza, irá conduzir o projeto de forma brilhante, implementando novas idéias e fazendo as renovações necessárias, sem no entanto perder de vista nossos objetivos iniciais.

Gostaria de terminar esta breve despedida agradecendo a todos os que contribuíram de alguma forma para o sucesso da revista Telecomunicações. Deixo meus mais sinceros agradecimentos aos colegas membros do conselho editorial, aos colegas membros do corpo editorial, que trabalharam arduamente e brilhantemente na revisão dos artigos submetidos, aos responsáveis pelo expediente da revista, às empresas que nos apoiaram financeira e institucionalmente ao longo destes anos e, como não poderia deixar de ser, aos autores que submeteram seus artigos para publicação na revista Telecomunicações.

Este número da revista está inteiramente dedicado a tutoriais: o primeiro analisa a interatividade em sistemas de TV digital terrestres, particularmente o padrão DVB-RCT; o segundo apresenta os principais conceitos sobre o processo de decodificação turbo, com enfoque na decodificação turbo de códigos de bloco; finalmente, o terceiro aborda os principais conceitos associados ao processo conjunto de equalização e decodificação iterativa, denominado equalização turbo. Os temas abordados são de grande atualidade e, portanto, acredito de grande interesse da comunidade de telecomunicações.

Prof. Dr. José Marcos Câmara Brito

Interactivity in Terrestrial DTV Systems: the DVB-RCT Standard[♦]

Ângelo Pinelli Martins Samia e Dayan Adionel Guimarães

National Institute of Telecommunications – Inatel
P.O. Box 05 – 37540-000 Santa Rita do Sapucaí – MG – Brazil

Abstract— The use of terrestrial broadcast digital TV platform to increase the percentage of the population with access to the Internet is conditioned to the implementation of a bi-directional link between the user and the broadcaster. One of the alternatives for the return channel is the DVB-RCT (Digital Video Broadcasting Return Channel – Terrestrial) standard, which suggests the use of a wireless return path in the VHF/UHF bands. This standard has been designed to operate with a DVB-T broadcast channel. This work is a tutorial about the DVB-RCT standard, incorporating the main features of the physical layer and issues like capacity and suitability for implementation in Brazil.

Index Terms—interactivity, digital TV systems, DVB-RCT, terrestrial broadcast, OFDMA multiple access.

I. INTRODUCTION

The digital TV (DTV) signal allows processing techniques that substantially improve the quality of the received signal and greatly increase the number of services that can be offered to users. The degree of content manipulation provided by DTV systems is practically impossible to be achieved using analog technology. Once digitized, the video and audio signals are compressed and can be multiplexed with data from other video, audio or from an application that establishes, for instance, an IP session between the broadcaster and the set-top box. Depending on the resolution of the picture, up to five programs¹ can be conveyed in a portion of the spectrum previously assigned to only one analog channel. This convergence is a benefit that comes along with technology and tends to change the way people watch TV in the near future. The degree of interactivity experienced by the user depends on what kind of service is being offered by the broadcaster. In some cases there is only *local* interactivity between the user and the set-top. It takes place, for example, in applications like Electronic Program Guides (EPG) and some games, where there is no need for the user to transmit information towards the broadcasting site. When a link is established from the user to the broadcaster, the so-called *return channel*, a higher degree of interactivity is possible and the user is able to participate in polls, to purchase products using the remote

control, to buy pay-per-view events or to surf the internet.

Currently, the main focus of discussion about digital TV in Brazil is the transmission standard for terrestrial broadcasting, and the Government has been supporting the idea of developing its own standard. Decree number 4,901 was issued in November, 2003 and instituted the SBTVD (acronym for Brazilian Digital Television System), whose objectives, among others, are to promote social inclusion, to stimulate in-country research and development (R&D), and to motivate the production of interactive content by current analog TV broadcasters [1]. SBTVD is composed by: a Committee of Development, associated to the Presidency, an Advisory Committee, and a Management Group. In a letter sent to the President of Brazil in November, 2003 [2], in which the approval of decree N° 4,901 is requested, former Minister of Communications Miro Teixeira sustained the idea that the system to be adopted should “provide interactivity and allow for the development of new applications that deliver entertainment to the population, promote the culture and improve education and citizenship”. In the same document, Teixeira complements: “The research and development of a Brazilian system intends to find solutions that are suitable to our social and economical realities, without necessarily excluding, *a priori*, the possibility of *choosing any of the foreign systems currently available, provided that they are in line with the Brazil’s social and financial interests*”.

During the opening of Telexpo 2004, Minister of Communications Eunício de Oliveira pointed out the attention the Government has been dedicating to implementing DTV systems in Brazil. Several institutions bid for the contracts to construct solutions for the physical, transport, middleware and application layers of the system; some of these institutions had started working on their projects at the time of this publication.

In order for the DTV platform to reach the purposes of the so-called digital inclusion, it is desirable that the system is two-way capable. In this context, the main objective of this work is to outline the recommendation for wireless return channel in terrestrial broadcasting networks, developed by the technical subgroup Digital Video Broadcasting Return Channel - Terrestrial (DVB-RCT) [3] and standardized in Europe by the European Telecommunications Standards Institute (ETSI) in March, 2002. Given this moment of critical definitions about which DTV system will be adopted in Brazil, another objective of this work, not less important than the first one, is to foment the discussions about a basic

[♦]This work has been partially financed by convention n° 22.02.0431.00 between Inatel, Linear Equipamentos Eletrônicos S/A and FINEP (Financiadora de Estudos e Projetos).

¹ The term *program* has the same meaning as *channel* for the analog PAL-M or NTSC systems (i.e., Globo, CNN, HBO etc.).

requirement to offer interactive services in DTV networks: the *return channel*.

A set-top box equipped with a DVB-RCT return module is able to transmit information back to the base-station using the same antenna of the broadcast signal reception. Although, DVB-RCT was originally designed to work with DVB-T (Digital Video Broadcasting–Terrestrial)[4], the operation with Advanced Television Systems Committee (ATSC) is suggested in [5]. However, the Radio-frequency Standardization Section of International Telecommunications Union (ITU-R) recommends the use of “the DVB-RCT system as the preferred wireless return path for the DVB-T Digital Terrestrial TV Systems”.

The set-top box will play an important role during the transition from the analog to the digital technology. The hardware architecture of these devices is similar to the personal computers (PCs), and likewise, the more sophisticated the set-top is the more it will cost. The hardware must comply with the needs of compatibility and processing required by each application. For example, some models are equipped with hard disks with up to 160 GB [6] of capacity for applications like personal video recording (PVR). Like the PCs, the set-tops also need an operating system (OS). Because the hardware resources are limited in the set-top box, the architecture of the OS is different from the ones designed for the PCs. Power TV OS, Microsoft Windows CE and some versions of Linux are examples of operating systems specifically designed to work with the set-top box. The smart-card technology can provide flexibility on the distribution of purchasable content and the return-channel module may be optional. It is possible to realize, through these examples, the advantages of a platform that is modular and standardized, in such a way that several manufacturers can compete against each other with the consumer getting the pricing benefits that arise from competition.

The president of ANATEL (the Brazilian National Telecommunications Agency), Pedro Jaime Ziller de Araújo, asserted [7] that there is an estimated market of US\$ 1.95 billion only to adapt the existing 65 million TV sets to the digital technology. To arrive at this number, Ziller de Araújo assumed that at least 60% of the users would buy the set-top box at an approximated price of US\$ 50. The president of the agency added that, at broadcasters’ side, the investment in digital equipment would be around US\$ 1.5 billion. The major challenge, still, is to use the technology to envisage new ways to increase revenue, keeping in mind the objective of democratizing the access to information technology.

This remainder of this article is organized as follows: Section II covers the building blocks of a conventional digital TV broadcasting site. Section III deals with possible DTV architectures with interactivity features. Section IV presents the wireless return channel based on DVB-RCT standard, with focus on the two types of channel coding supported by this standard and on the multiple access technique OFDMA (Orthogonal Frequency Division Multiple Access). In section V, the

cellular structure of the wireless interactive system is shortly described, and in section VI a case study is presented. Section VII concludes with some final remarks and in Section VIII a glossary of abbreviations aims at facilitating the reading.

II. DIGITAL TV SYSTEMS

DTV systems will allow the user to improve their viewing experience through cinema-quality pictures, CD-quality sound, more channels and improved access to a range of new entertainment services. Figure 1 depicts an architecture model for a DTV system.

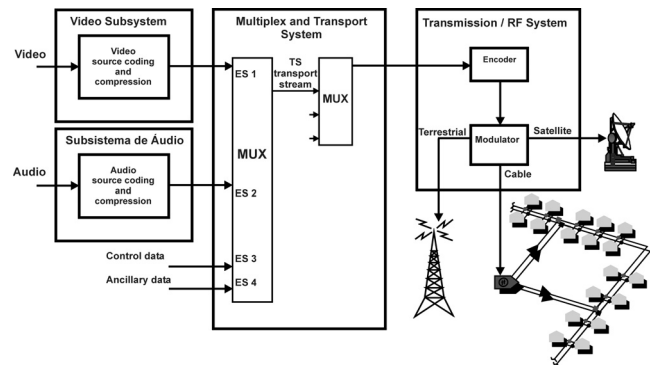


Figure 1 - Basic architecture of a DTV system

According to the model shown in Figure 1, a digital TV system can be split into three subsystems: (i) coding and compression subsystem, which reduces the bandwidth occupied by the digitized audio and video signals by means of source coding [8 pp. 568-580]; (ii) multiplex and transport subsystem, which refer to the means of dividing the digital data stream into packets of information, the means of uniquely identifying each packet or packet type, the appropriate methods of multiplexing video data-stream packets, audio data stream packets, and ancillary data-stream packets into a single transport stream (TS) [9] [10], and the mechanisms for one more level of multiplexing, the system level, where individual program transport streams are combined; (iii) the Transmission / RF stage, where the techniques used for channel coding and modulation vary according to the way the signal is broadcast (i.e., terrestrial broadcast, cable or satellite). The return channel is not represented in Figure 1 and will be covered in Section III.

II.1 SOURCE CODING AND COMPRESSION SUBSYSTEM

The source coding and compression subsystem use the appropriate methods to process the video and audio signals. These methods must take into consideration the particularities of each type of signal, and the kind of redundancy they may have. Therefore, compacting a signal consists of removing as much as possible its redundancy without losing any information. In cases like the video signal, it is acceptable that *there is* loss of information, in which situation the process is called compression. In order to have an idea on how important video compression is, suppose that frames with 1,080 lines and 1,920 pixels per line are transmitted at a rate of

30 frames per second. If each pixel is represented by three component colors with 8 bits each, the required transmission rate will be $R = 1.080 \times 1.920 \times 30 \times 3 \times 8 \cong 1.5$ Gbit/s. It is not practical to convey such a rate through a 6 MHz channel with the existing modulation and channel coding methods.

The redundancy in a video signal can be spatial, as happens when adjacent pixels have little or no variation; or temporal, when the picture practically does not change over time.

The Moving Pictures Experts Group (MPEG) was created by ISO (International Standards Organization) to develop compression standards. The MPEG-2 standard for video compression, specified by ISO/IEC 13818-2 [11] and equivalent to ITU-T H.262, was created by the MPEG group and turned out to be the *de facto* video compression standard. MPEG-2 can be used in a variety of applications, with different requirements of performance and complexity. The MPEG-2 is divided into profiles, and each profile is subdivided into levels, in a total of 24 options (not all of them are defined). Among the profiles of interest for terrestrial TV broadcast are the MP@ML (Main Profile at Main Level) and the MP@HL (Main Profile at High Level). These profiles support, respectively, the levels of resolution known as SDTV (Standard Definition Tele-Vision) – standard definition – that offers the best picture quality for transmission of conventional video, and HDTV (High Definition Tele-Vision) – high definition – used to transmit high definition picture content [14]. Although the audio compression is also defined by MPEG through the recommendation ISO/IEC 13818-3 [12], the Grand Alliance (originally formed in 1993 by AT&T, GI, MIT, Phillips, Sarnoff, Thomson and Zenith) preferred the audio compression system known as AC-3 [13]. The DVB-T uses MPEG-2 for audio compression [12].

The output of the source coding and compression subsystem contains audio or video elementary streams (ES). For practical reasons, the elementary stream is divided into packets that are called packetized elementary stream (PES). Each PES carry a unique audio or video elementary stream and contain a header with information about the identity (ID) of the sequence, size of the packet and some optional data, such as the indication of cryptography, copyright information and time stamps that are used by the receiver to learn the exact point in the time axis the packet is to be presented [15].

ITU-T H.264 [40] is among the newer video compression standards. It is equivalent to ISO/IEC 14496-10, nominally MPEG-4 Part 10. The intent of H.264/AVC project has been to create a standard that would be capable of providing good video quality at bit rates that are substantially lower (e.g., half or less) than what previous standards would need (e.g., relative to MPEG-2, H.263, or MPEG-4 Part 2). An additional goal was to do this in a flexible way that would allow the standard to be applied to a very wide variety of applications (e.g., for both low and high bit rates, and low and high resolution video) and to work well on a very wide variety of

networks and systems (e.g., for broadcast, DVD storage, RTP/IP packet networks, and ITU-T multimedia telephony systems). With higher performance of the compression stage, more spectrum will be made available for additional services like, for example, return channel services.

II.2 MULTIPLEXING AND TRANSPORT SUBSYSTEM

Multiplexing elementary streams, with or without PES packetization, creates a program transport stream² that shares a common time base. As the elementary streams are multiplexed, they are formed into transport packets, each one with 188 bytes, identified by a header field called Program ID (PID). A program transport stream contains several types of data and is identified by means of a program-mapping table (PMT), which describes what are the PIDs for audio, video and data packets corresponding to a given program. The recommendation ISO/IEC 13818-1 [10], also known as MPEG-2/SI (System Information), defines this level of multiplexing, the *system* level, that allows for the combination of several program transport streams into a unique transport stream. A new table is generated, the program association table (PAT), that contains the PMTs for all programs in the transport stream. Thus, the identification process of a program consists in two stages:

- The receiver looks for the PAT (with PID = 0), which identifies the PMT of the program chosen.
- The PIDs of the elementary streams that form the program are obtained from the PMT.

The MPEG-2 transport-stream syntax was developed for applications where channel bandwidth or recording media capacity is limited, and the requirement for an efficient transport mechanism is paramount. The MPEG-2 transport stream also was designed to facilitate interoperability with the Asynchronous Transfer Mode (ATM) transport stream [9].

II.3 TRANSMISSION / RF SUBSYSTEM

According to Figure 1, the MPEG-2 transport stream feeds the Transmission / RF subsystem. The modulation and coding used in this stage will depend on the characteristics of the medium the signal will be submitted to. TV broadcast satellite systems normally use Quaternary Phase-Shift Keying (QPSK) modulation [16]. The ITU-R J.83 [17] recommendation defines through its Annexes A, B, C e D, the transmission parameters adopted in Europe, North America, Japan and by the ATSC, respectively, for distribution via Hybrid Fiber-Coax (HFC)³ networks. Due to the controlled characteristics of these kind of networks, which do not

² A clarification on the nomenclature may be necessary at this point: the term *program stream* refers to a data stream used in applications such as DVD or in-studio connections. The program stream is not used in broadcasting systems.

³ The HFC networks are an evolution of the CATV architecture, where TV signals were distributed uniquely by means of coaxial cable. HFC networks employ a mix of optical fiber and coaxial cable for transmission.

suffer from the effects of multi-path propagation, Quadrature Amplitude Modulation (QAM) is used and covered in Annexes A, B e C. Annex D is regarded to the use of 16-VSB (Vestigial Side Band) modulation.

For terrestrial broadcast there are three standards already consolidated worldwide: the DVB-T [4] and the Integrated Services Digital Broadcasting (ISDB-T), from Europe and Japan, respectively, use the Orthogonal Frequency Division Multiplexing (OFDM) transmission technique. The modulation in the ATSC system is performed using single-carrier 8-VSB technique [18].

III. INTERACTIVITY IN DIGITAL TV SYSTEMS

The “one-way” system described so far is sufficient to provide broadcasting services. However, in order to be able to support applications like pay-per-view and Internet, a two-way link is necessary between the broadcaster and the user’s set-top box.

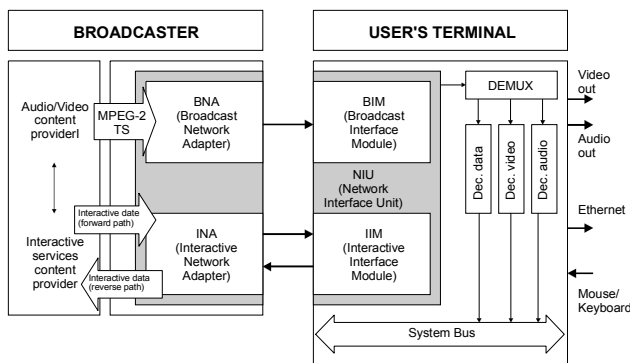


Figure 2 - A generic reference model for interactive DTV systems [1]

Figure 2 illustrates the generic model for an interactive digital TV system. Two channels are established between the broadcaster and the set-top box: the broadcast one-way channel and the interactive two-way channel. These channels can be identified in Figure 2 and are described as follows:

- **Broadcast Channel:** the audio and video content provider is equipped with coding and compression subsystems, as well as the transport service. The generated bit stream with MPEG-2 transport packets [10] feeds the Broadcast Network Adapter (BNA). This module implements the functions of the “Transmission / RF subsystem”, described in Section II. The set-top must have a Broadcast Interface Module (BIM) compatible with the BNA, so that demodulation and decoding are properly performed. At the BIM, the demultiplexer can be an Application-Specific Integrated Circuit (ASIC) that examines all PIDs of the transport stream, selects the packets corresponding to the program chosen, decrypts (if necessary) its content and sends this information to the corresponding decoders.
- **Forward interactive channel:** provides an always-on link from the broadcaster to the set-top. The

Interactive Network Adapter (INA) receives the data from the content provider and formats this data for transmission. This channel is also called Out-Of-Band (OOB) channel and is used during the authentication process with the base-station.

- **Return interactive channel:** it is the channel that allows the user to transmit information back to the broadcaster. It can be implemented using regular phone lines or by means of wireless communication.

III.1 INTERACTIVITY IN SATELLITE TV SYSTEMS

The degree of interactivity of the services offered by current Direct-To-Home (DTH) systems in Brazil is basically local. The exception to this rule is the pay-per-view service, which use the telephone line to send purchase information back to the operator. As examples, most *Sky* and *DirectTV* set-tops are now equipped with 2.4 kbit/s modems, but this rate is still very low for more advanced applications. Other systems use an uplink transmitter per user.

An advantage of satellite systems is that there is practically no restriction on geographic location for installation. The acronym DVB-RCS (Digital Video Broadcasting – Return Channel Satellite) characterizes the return channel via satellite. The physical and Medium Access Control (MAC) layers for the forward and reverse links are described in [16] and [19], respectively.

III.2 INTERACTIVITY IN CABLE TV SYSTEMS

HFC-based interactive systems normally follow the model presented in Figure 2. A portion of the spectrum is dedicated to the forward interactive channel. The return interactive channel requires that the HFC network is capable of handling bi-directional traffic. In order to do it, the available spectrum is split into two frequency bands. Typically the frequency band dedicated to the return channel is located in the lower part of the spectrum and has a bandwidth of approximately 40 MHz⁴. The forward channel goes up to 450, 550, 750 or 870 MHz, depending on the equipment used. The set-tops use the forward interactive channel for ranging procedures, which involve synchronization of the set-top with the base-station, power level adjustments and fine frequency tuning. Once on-line, these terminals can run applications whose forward data stream can also be transmitted through the broadcast channel, in a process known as *data carousel* [20]. As interactive TV applications slowly emerges on cable, subscribers from the main cities in Brazil have broadband communications available through cable modems, with rates ranging from 128 kbit/s to 512 kbit/s for domestic users.

Since CATV operator is responsible for spectrum allocation and signal content in its network, the use of a given transmission standard is not mandatory. However,

⁴ In order to cope with the demand for bandwidth, many systems currently installed deploy diplex filters with passband from 5 to 65 MHz in the return channel.

the adoption by the industry of a specified standard (of transmission, for instance) can turn out to be economically advantageous for mass production equipment, like set-tops and cable modems. An example is the Data Over Cable Service Interface Specification (DOCSIS) standard [21]. DOCSIS is largely deployed to provide broadband Internet access to cable TV subscribers. The DOCSIS project began in the U.S. in 1995 as a consortium formed at that time by the Multi-Systems Operators (MSO) Comcast, Cox, TCI and Time Warner. In 1996, version 1.0 of the radio frequency interface (RFI) was released. Version 1.1 of this RFI was issued in 1999, after CableLabs took over the project. It supports 64-QAM or 256-QAM in the forward channel, with transmission rates of 30.343 and 42.884 Mbit/s, respectively, and QPSK or 16-QAM in the return channel (with rates ranging from 0.32 to 10.24 Mbit/s).

There are two other options for standardization of the interactive channel in cable distribution systems, the ANSI/SCTE 55-1 (former DVS-178) [22] and the ANSI-SCTE 55-2 (former DVS-167) [23]. The first was developed by General Instruments (now acquired by Motorola) and issued by the Society for Cable Television Engineers (SCTE), Digital Video Systems (DVS) subcommittee. The latter encapsulates the forward data stream into MPEG-2 transport packets and transmits this stream at a rate of 2.048 Mbit/s. Messages sent through the return channel have the shape of ATM cells [24], with a 256 kbit/s transmission rate. The interactive channel employs QPSK modulation in both the direct and the return channels. The ANSI-SCTE 55-2 was submitted to SCTE by Scientific Atlanta and was extracted⁵ *verbatim* from the Digital Audio Video Council (DAVIC) standard [25]. This standard also makes use of QPSK modulation for both the downstream and the upstream. The data stream is organized in ATM cells for both the forward and the return channels.

III.3 INTERACTIVITY IN TERRESTRIAL BROADCAST TV SYSTEMS: THE DVB-RCT

One solution for implementing the return path in terrestrial TV broadcast systems is presented in the ETSI EN 301 958 standard [3], also known as DVB-RCT. The proposed interactive system consists of a forward channel compatible with DVB-T standard [4] and of a return interactive channel, which operate in different parts of the Very High Frequency / Ultra High Frequency (VHF/UHF) spectrum.

As shown in Figure 3, the forward interactive channel is embedded to the broadcast channel. The forward interactive data stream is encapsulated into MPEG-2 transport packets with specific header, multiplexed with other programs and transmitted in the same 6 MHz channel used for conventional broadcasting (analog TV systems). The transport stream sequence corresponding to

the forward interactive channel carries MAC information.

The MAC messages control user's access to the shared medium [26] and transit in both directions along the network. The difference between MAC messages in the downstream and MAC messages in the upstream is that the first ones are encapsulated into MPEG-2 transport packets, whereas the latter ones are encapsulated into ATM cells. The transmission over the physical medium of these ATM cells is carried out by means of bursts of modulated symbols, as will be seen in Section IV. The payload of each burst depends on the type of modulation and channel coding rate assigned to the terminal. Most of the MAC messages were designed to fit in the minimum payload defined for a burst. In case of contention access, when the size of the burst is smaller than an ATM cell, the MAC messages are sent directly. When the size of the burst is larger than an ATM cell, the MAC messages are encapsulated into ATM cells and then sent [3]. Data messages are always encapsulated into ATM cells before transmission.

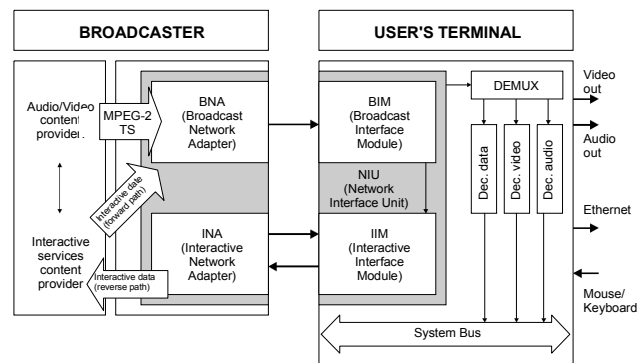


Figure 3 - A reference model for the terrestrial interactive DTV system

ATM connections allow for Quality-of-Service (QoS) provisioning, which is necessary for services employing the Voice-over-IP (VoIP) technology. The reference model for ATM protocol is divided into three layers: the ATM Adaptation Layer (AAL), ATM layer and physical layer, as shown in Figure 4.

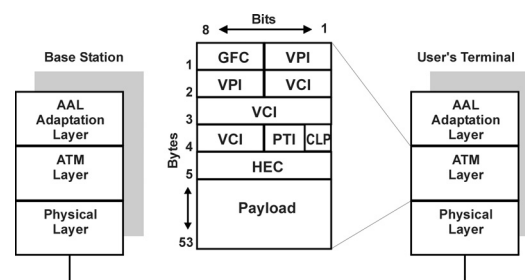


Figure 4 - ATM protocol structure [27]

The AAL layer is divided into two sub-layers: Segmentation and Reassembly (SAR) sub-layer and Convergence Sub-layer (CR). SAR sub-layer is responsible for breaking the data stream into fragments that can be accommodated in the payload of an ATM cell and for reassembling the stream from the received cells. The CS sub-layer is responsible for tasks like services

⁵ DVS-167 corresponds to Part 8 (Lower Layer Protocols and Physical Interfaces), Section 7.8 (Passband Bi-directional PHY on Coax) of DAVIC 1.2 standard [25].

multiplexing, cell loss detection and timing recovering at the destination. Four types of adaptation layers were defined, each one being capable of handling different types of services. For example, networks based on the Transfer Control Protocol / Internetworking Protocol (TCP/IP) are classified as class D and can be implemented by using ATM adaptation layer AAL-5 [27].

The ATM layer is the interface between the AAL and the physical layer. This layer is responsible for relaying cells from the AAL to the physical layer for transmission and, conversely, from the physical layer to the AAL. The header fields in an ATM cell define the functionalities of the ATM layer. The header dedicates four bits to a function called generic flow control (GFC), which was originally designed to control the amount of traffic entering the network. This allows, for example, the base-station to limit the traffic that enters the network during periods of congestion.

In order for a *switching node* to perform its function of re-transmitting the input cells to the corresponding output, it is necessary to feed the node with information about the route of the cells. The Virtual Channel Identifier (VCI), which is a single-channel identifier, together with Virtual Path Identifier (VPI) form the routing field, which associates each cell with a particular channel or circuit. The VPI allows grouping of VCs with different VCIs and allows the group to be switched together as an entity [27]. This can be imagined as if there were two layers of switching: a *lower layer*, where only the VPI is examined, and an *upper layer*, where the virtual channels from that virtual path will also be analyzed by the node when switching the input cells to the output. The Header Error Check (HEC) field is used for header error correction, whilst the Packet Type Identifier (PTI) field is used to indicate whether the cell is dedicated to transport user's data or network management information. Finally, the Cell Loss Priority (CLP) field indicates the priority in the case of a need for discarding cells.

IV. DVB-RCT PHYSICAL LAYER

IV.1 OFDM TRANSMISSION

The basic principle of OFDM multiplexing consists in transmitting data by dividing the stream into several parallel bit streams, each of which with a much lower bit rate, and by using these sub-streams to modulate several sub-carriers. Since the resulting sub-symbol has a longer duration than that of the original symbol, there will be less inter-symbol interference (ISI) due to the channel delay spread [28]. However, even with the longer duration of the transmitted symbols, some ISI can still remain; that can be resolved by using a guard interval, preferably with a duration that is longer than the *rms* delay spread caused by the channel, added to each OFDM symbol. Inter-Channel Interference (ICI) occurs when orthogonality between sub-carriers is lost at the receiver side. Practical oscillators normally have some phase modulation at their outputs, which is called *jitter*. Frequency shifts due to

phase jitter may cause ICI at the reception. The guard time also helps to combat ICI, provided that during this interval a cyclic extension of the original symbol is performed [28].

The origins of OFDM are from the mid 60's, when Chang obtained a patent [29] for a transmission structure that superimposes orthogonal modulated carriers for data transmission. The physical implementation of this technique was greatly facilitated after 1971, when Weinstein [30] introduced the idea of using the Discrete Fourier Transform (DFT) to generate OFDM signals, therefore eliminating the need for banks of analog oscillators. The construction of OFDM devices became feasible by means of algorithms that implement the Fast Fourier Transform (FFT) together with the boost experienced by the technology of Digital Signal Processors (DSPs) and programmable logic devices.

IV.2 OFDMA MULTIPLE ACCESS

The method for organizing the sharing process of the DVB-RCT channel is inspired by the DVB-T standard [4]. But, while in downstream the same MPEG-2 data packets are transmitted to all users, in the upstream a number of users need to utilize the same spectrum to transmit data back to the broadcaster or to the base-station. This demands a multiple access technique to be associated to OFDM transmission.

Since in interactive TV systems, the traffic in the return path has typically short duration, the permanent allocation of sub-carriers to a given terminal would not be efficient. DVB-RCT employs what is called Demand-Assigned Multiple Access (DAMA), which, associated to OFDM, generated the Orthogonal Frequency Division Multiple Access (OFDMA) technique. In this access rule, sub-channels are allocated to users through Medium Access Protocol (MAP) messages, which by their turn arrive to the terminal multiplexed in the program transport stream present in the broadcast channel.

A sub-channel is a set of sub-carriers out of the total available sub-carriers. In order to mitigate the frequency selective fading, the carriers of one sub-channel are spread along the channel spectrum [31]. Figure 5a shows how the frequency spectrum is organized when the total number of sub-carriers is $N = 1,024$ (1k mode). Only 841 sub-carriers are usable, because of the guard bands that are reserved at the boundaries in order to avoid adjacent channel interferences. The 841 sub-carriers are divided into $N_G = 29$ groups, each group with $N_E = 29$ sub-carriers. Some pilot carriers are also transmitted and are used for the purposes of synchronization and carriage of Channel State Information (CSI), as required by some Forward Error Correction (FEC) codes. Picking up, pseudo-randomly, one sub-carrier from each group forms one sub-channel. Therefore, in 1k mode, each sub-channel contains 29 sub-carriers. Figure 5b represents the situation for $N = 2,048$ sub-carriers (2k mode). It is noticeable from this figure that, by pseudo-randomly selecting one carrier from each subgroup, the total number of sub-

carriers per sub-channel, in this case, is 59. The user terminal receives authorization by the base-station to transmit data in one or more sub-channels, depending on the required traffic demand.

An important aspect of sub-channel utilization is related to coverage. A DVB-RCT system typically consists of a high-power transmitter installed in the base-station and a multitude of set-tops equipped with low-power transmitters. For the OFDMA option with $N = 1,024$ sub-carriers, the transmitter at the subscriber side concentrates its power into a sub-channel that has 1/29 of the channel bandwidth. For equivalent modulation and coding, this capability is directly related to the maximum coverage area.

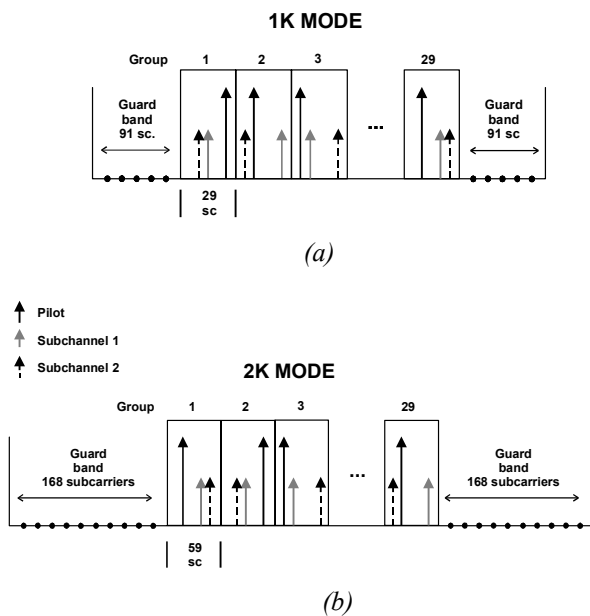


Figure 5 - Sub-channels of a system (a) DVB-RCT 1.024 and (b) 2.048 sub-carriers (DC carrier not used).

Sub-channels constitute a form of frequency hopping spread spectrum (FHSS). In every group, each set-top transmits using one pseudo-randomly selected carrier out of N_E possible ones. A terminal in an interfering cell does the same random-type selection, but in a statistically independent way. Then, the probability of collision is $1/N_E$. It is important to notice that there is no interference within the cell since its sub-channels are orthogonal: each group element is used by only one sub-channel [31].

When the input symbol stream is split in N sub-carriers, the symbol duration is augmented N times, and this tends to diminish the effects of the channel delay spread, as mentioned before. In order to provide immunity against Inter-Carrier Interference (ICI) and ISI, two types of sub-carrier shaping functions are defined:

- *Rectangular shaping*: in this case a guard interval, T_g , equal to $T_s/4$, $T_s/8$, $T_s/16$ or $T_s/32$, is included between each symbol. The total symbol duration is $T_s = T_u + T_g$, where T_u is the useful symbol duration. If the guard interval is larger than the average (rms) delay spread, the multi-path components from a given symbol will interfere less in adjacent symbols and this reduces the inter-symbol interference.

- *Nyquist shaping*: used to produce a pulse in the time domain that is associates to a root-raised cosine frequency response, with a roll-off factor $\alpha = 0.25$. The total symbol duration at the filter output is $T_s = T_u + 0.25T_u$. This type of pulse shaping yields more immunity against ICI because the resulting frequency spectrum decays more rapidly than for rectangular shaping.

In OFDMA technique, the base-station assigns to the user a fraction of the total number of available sub-carriers. The DVB-RCT transmitter module installed in the set-top must support operation using 1024 and 2048 sub-carriers, with three options for separation between sub-carriers: CS1, CS2 and CS3 (corresponding to 837, 1674 and 3348 Hz, respectively, for 6 MHz systems). Each value of sub-carrier spacing implies a given maximum transmission cell size, and a given resistance to the Doppler shift experienced when the user terminal is in motion. Table 1 shows, among other parameters, the total bandwidth occupied by the DVB-RCT channel for operation modes 1k and 2k. Once this bandwidth is taken into account for each operation mode, one could employ, in principle, any unused or partially used portion of the VHF/UHF frequency spectrum to implement the digital broadcast system's return channel.

Table 1 - DVB-RCT Transmission Modes parameters for 6 MHz DVB-T systems

Total # of sub-carriers # of usable sub-carriers	2048 1712	1024 842
CS1 (sub-carrier spacing 1)	837 Hz	837 Hz
Useful symbol duration	1195 μ s	1195 μ s
DVB-RCT channel BW	1.433 MHz	0.705 MHz
CS2 (sub-carrier spacing 2)	1674 Hz	1674 Hz
Useful symbol duration	597 μ s	597 μ s
DVB-RCT channel BW	2.866 MHz	1.410 MHz
CS3 (sub-carrier spacing 3)	3348 Hz	3348 Hz
Useful symbol duration	299 μ s	299 μ s
DVB-RCT channel BW	5.732 MHz	5.732 MHz

Two types of transmission frames, TF1 and TF2, are defined in order to allow synchronization at the base-station and to reserve specific symbols for ranging procedures. In TF1, three categories of symbols are defined: null symbols, ranging symbols and data symbols. As depicted in Figure 6, the sub-carriers corresponding to the first OFDM symbol of TF1 are not transmitted. This time interval can be used by the base-station to detect jammers, since all terminals of the network are synchronized and do not transmit during this time interval. The following six symbols are dedicated to ranging procedures and the remaining 176 symbols are used to transmit data.

Whereas TF1 performs the division of each symbol function in the time domain, Transmission Frame 2 structures each symbol function in the frequency domain; TF2 does not use null symbols. TF1 and TF2 frames are transmitted in bursts. Three types of burst structures are

defined, each carrying 144 symbols at the output of the modulator. Therefore, depending on the modulation used (QPSK, 16-QAM or 64-QAM) and the forward error correction code rate ($\frac{1}{2}$ ou $\frac{3}{4}$), each burst can transport a certain number of information bits. Table 2 shows the number of information bytes to be coded as a function of modulation type and code rate. The channel encoder operates with input blocks ranging from 144 to 648 bits.

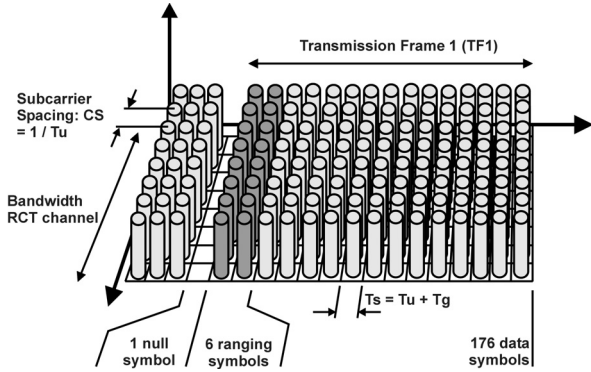


Figure 6 - DVB-RCT Transmission Frame 1.

The DVB-RCT system clock period is derived from the system clock present in the DVB-T downstream, which is defined in [4] as $7/48 \mu s$ for 6 MHz channel spacing, is defined as follows:

- Four times the DVB-T system clock period for CS1: $0.583 \mu s$;
- Twice the DVB-T system clock period for CS2: $0.292 \mu s$;
- One DVB-T system clock period for CS3: $0.146 \mu s$.

Table 2 - Number of bytes in a burst as a function of code rate and modulation type

Modulation	Code rate	Information bytes in 144 symbols
QPSK	$\frac{1}{2}$	18 bytes
	$\frac{3}{4}$	27 bytes
16-QAM	$\frac{1}{2}$	36 bytes
	$\frac{3}{4}$	54 bytes
64-QAM	$\frac{1}{2}$	54 bytes
	$\frac{3}{4}$	81 bytes

Figure 7 illustrates the block diagram of a DVB-RCT set-top box. This figure brings more details about the blocks that form the Network Interface Unit (NIU) indicated in Figure 3. The broadcast DVB-T signal goes through the duplexer filter and is routed to the Broadband Interface Module (BIM) in order for the original transport stream to be recovered.

The MAC management block obtains the information from the forward interactive channel by selecting the appropriate PIDs out of the received transport stream.

Still referring to Figure 7, the following characteristics of the DVB-RCT can be highlighted:

- Each DVB-RCT module transmits its information by using one or more modulated sub-carriers, with the

rate per sub-carrier depending on the transmission mode chosen by the base-station;

- The sub-carriers are synchronized by the base-station, i.e., the user terminals obtain the system timing reference from the DVB-T broadcast channel. Therefore, the transmission mode parameters employed by DVB-RCT are fixed and bear strict relationship with the DVB-T broadcast channel;
- At the base-station, the return signal is demodulated, so that at the output of the Interactive Network Adapter (INA), data coming from each user is available in the form of ATM cells.

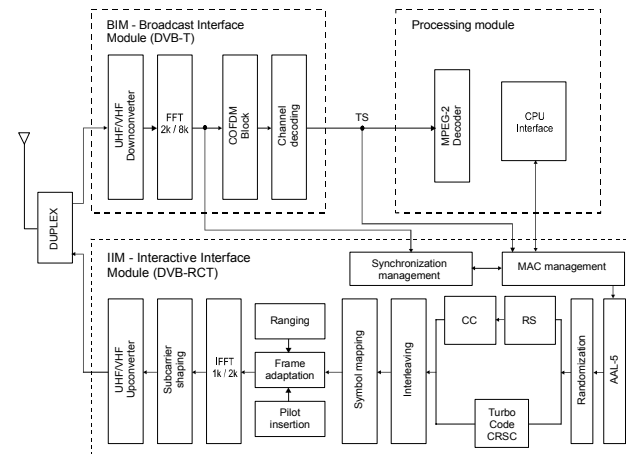


Figure 7 - Block diagram of a set-top box DVB-RCT.

The synchronization of the Interactive Interface Module (IIM) is achieved in two ways [3]: through MAC control messages received in the downstream (time-based synchronization) and through the frequency information emitted by the DVB-T modulator (frequency-based synchronization). The MAC Management and Synchronization Management blocks are responsible, respectively, for time-based and frequency-based synchronization.

The user's information is processed by the IIM, passes through the duplexer filter and is transmitted in the return channel using the procedures that will be described hereafter. At the base-station, the signals coming from the users are demodulated and sent to a MAC layer management block, which separates application messages (that will be routed to the respective application servers at the Interactive Services Provider) from MAC control messages, which are processed by the base-station and generate medium access information, which are then multiplexed with other programs to produce the definitive transport sequence that will be transmitted in the broadcast channel.

IV.3 RANDOMIZATION

Randomization is a technique where a shift register transforms the original data in a pseudo-random fashion. This procedure is used to insert transitions on long sequences of 0's or 1's of the original data, in such a way that the power spectrum density becomes more uniformly

distributed over the frequency spectrum. The receiver reorders the sequence using the inverse procedure, so that it gets the original data at the output. The length of the shift registers and the feedback connections generally identifies the circuits that implement the randomizer. DVB-RCT uses 15-bit shift registers and the connections are governed by the generator polynomial of the pseudo-random sequence $1 + X^{14} + X^{15}$ [3]. The bit emitted by the randomization block is applied to the channel encoder input.

IV.4 CHANNEL CODING

Channel coding is one of the key elements that define the performance of a communication system. For systems that operate with multiple carriers in multi-path environments, channel coding is even more important. The process of channel coding basically consists in adding controlled redundancy to the information. The receiver uses this redundancy to detect and/or correct errors. The code rate is defined as $r = k/n$, where k is the number of information bits and n is the number of bits at the coder output [8]. DVB-RCT establishes the following options for channel coding:

- Serial concatenation of Reed-Solomon with convolutional coding [32] [8];
- Circular Recursive Systematic Convolutional Turbo coding [33] [34].

IV.4.1 SERIAL CONCATENATION OF REED-SOLOMON WITH CONVOLUTIONAL CODE

The term *concatenation* was used by Forney [35] to designate the process of multilevel coding, formed by utilizing an *inner* code and an *outer* code (in case of serial concatenation). The inner code is normally configured to correct most of the channel errors, whereas the outer code reduces the probability of error to a specified level. The primary reason for using a concatenated code is to achieve a low error rate with an overall implementation complexity that is less than what would be required by a single coding operation [32]. For serially concatenated codes, it is possible to verify that if the inner and outer code rates are r_c and R_c , respectively, the rate of the concatenated code will be $R_{cc} = R_c r_c$. For the DVB-RCT, global code rates $1/2$ and $3/4$ are obtained from RS and convolutional codes with individual rates given by Table 3 [3].

Table 3 - Individual RS and convolutional code rates for global code rates $1/2$ and $3/4$.

Global Code Rate	Reed-Solomon Code Rate	Convolutional Code Rate
1/2	3/4	2/3
3/4	9/10	5/6

A. Reed-Solomon encoding

Reed-Solomon (RS) codes are cyclic linear block codes that operate with *non-binary* symbols. The input

bits are grouped in blocks of m bits. The DVB-RCT standard defines $m = 6$, and also that the code should be able to correct up to $t = 4$ symbols in one codeword. Reed-Solomon codes are particularly useful for *burst-error correction* [32], i.e., RS codes are useful in channels that have memory. The resulting $RS(n, k, t) = RS(63, 55, 4)$ with $m = 6$ follows the conventional form of RS codes:

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) = (63, 55)$$

The primitive polynomial $p(x) = 1 + x + x^6$ is used to generate the extended Galois field $GF(2^m)$. This field $GF(64)$ determines the generator polynomial $g(X) = (X + \lambda^0)(X + \lambda^1)(X + \lambda^2)...(X + \lambda^7)$, where the degree of $g(X)$ is equal to the number of parity symbols. In order to obtain the polynomial corresponding to the parity symbols during the encoding process, it is necessary (i) to shift the message vector $m(X)$ by k symbols by multiplying $m(X)$ by X^{m-k} , and (ii) to divide the result by the generator polynomial $g(X)$. The codeword at the output assumes the systematic form $RS(63, 55)$.

According to what was mentioned in Subsection IV.2, the number of information bits that are applied to the channel encoder vary in conformity with the modulation scheme and global code rate. The requirements expressed in Table 3 must be satisfied for the three types of modulation techniques and, consequently, for the different types of input block sizes. Figure 8a shows the particular case where the global code rate is $1/2$ and modulation is QPSK. From Table 2 it is seen that, in this case, 18 input bytes must be fed to the RS encoder and, since $m = 6$, this is equivalent to 24 RS symbols. Because the designed code is $RS(63, 55)$, 31 null symbols have to be appended to the 24 information symbols at the input.

The number of parity symbols remains the same 8 symbols. The resulting code rate for this $RS(36, 24)$ code is $R_c = 144/192 = 3/4$, according to what is determined by Table 3. When the required global code rate is $3/4$, DVB-RCT stipulates that only 4 out of the 8 parity symbols are sent and this is shown in Figure 8b. The codes $RS(32, 24)$ and $RS(40, 36)$ resulting from global code rates $1/2$ and $3/4$, respectively, are called *shortened codes*.

B. Convolutional encoding

The convolutional encoder responsible for generating the inner code is illustrated in Figure 9.

At each bit interval, one bit is shifted to the right in one position. The encoder memory (or *constraint length*) indicates how many shifting positions the adders operate. It is immediate to observe from Figure 9 that this code utilizes the following generator polynomials: $G_1 = 576_{oct} = 101110001_{bin}$ e $G_2 = 753_{oct} = 111101011_{bin}$. Each input bit produces two encoded bits at the output and therefore the code rate is equal to $1/2$.

Puncturing standards are defined [3] to discard certain parity bits at the transmission side, adjusting the code rate in accordance with Table 3.

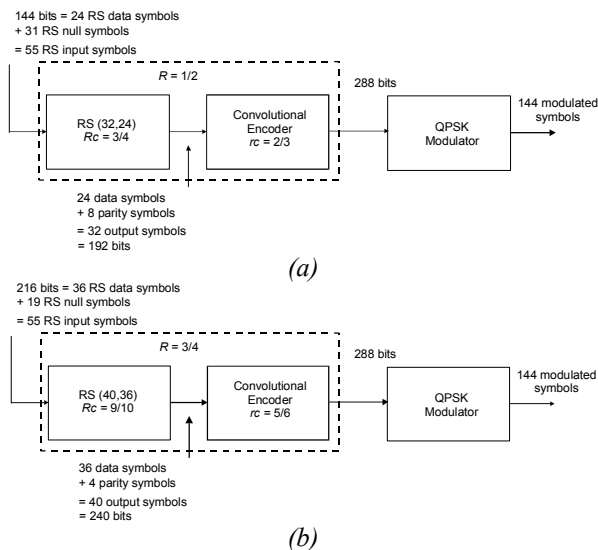


Figure 8 - Serial concatenation of Reed-Solomon with convolutional code: coding parameters for QPSK modulation with (a) code rate = 1/2; (b) code rate = 3/4.

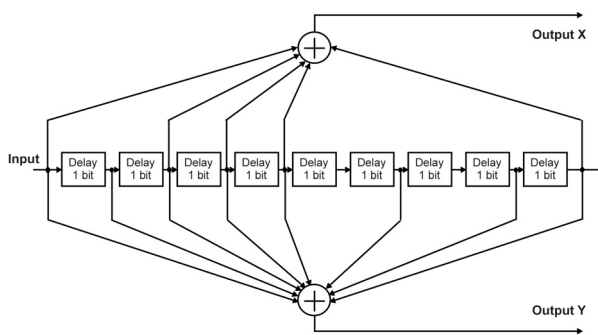


Figure 9 - Block Diagram of the convolutional encoder [3]

IV.4.2 TURBO CODE

The Circular Recursive Systematic Convolutional (CRSC) code [36], defined as one of the two alternatives for channel coding in DVB-RCT, is built from a parallel concatenation of two RSC component codes, each with two inputs. The encoder is said to be *circular* because it employs a technique in which, by the end of the encoding operation, the encoder retrieves the initial state, so that the data encoding can be represented by a circular trellis [34], dispensing the need for adding tail bits at the input for this purpose. As can be observed in Figure 10, two identical memory $v = 3$ component codes are used. The polynomial that describes the recursion is $1 + D + D^3$ and the parity bits are generated from the polynomial $1 + D^2 + D^3$. To encode the data sequence, the CRSC encoder must be fed four times, two times in normal operation mode (switch in position 1) and two times in interleaved order (switch in position 2), as shown in Figure 10. Blocks of k bits N couples ($k = 2 \times N$ bits) feed the encoder. The most significant bit of the first byte of the useful payload is assigned to input A, the next bit to input B, and so on for the remaining of the burst content. The *puncturing* patterns in order to achieve coding rates 1/2 and 3/4 are

described in [3].

An example of hardware implementation of this code is the RN-2821 modem chip [37], which has been designed specifically to operate in DVB-T / DVB-RCT systems.

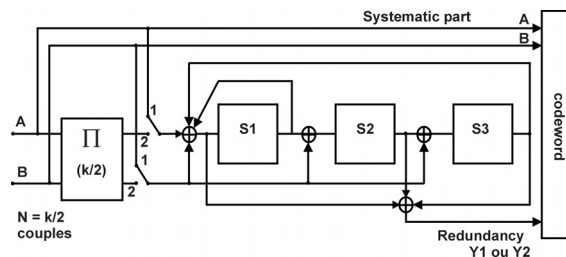


Figure 10 - Duo-binary Circular Recursive Systematic Convolutional encoder [1]

In [38], the decoding techniques *enhanced-log-APP* (EL-APP) and *enhanced-max-log-APP* (EML-APP) were proposed and compared to *log-APP* (L-APP) and *max-log-APP* (ML-APP) algorithms. The ‘max-log’ prefix indicates that those techniques use sub-optimal versions of the original algorithm and therefore are easier to implement with practical hardware.

V. DVB-RCT CELLULAR STRUCTURE

The basic configuration of a DVB-RCT cell consists of an upstream channel inside a DVB-T cell, in which the base-station typically uses an omni-directional antenna to receive the signals transmitted by the users.

There is no upstream interference within a cell since each sub-channel uses only one sub-carrier from each group at a time and the sub-carriers are orthogonal in a DVB-RCT channel. When there is re-use of frequencies in adjacent cells, the capacity supported by a cell A is determined by the interference I created by the terminals from the cells that are adjacent to cell A. To maximize this capacity the modulation schemes with the highest throughput are assigned to as many users as possible. But these schemes also have the higher C/I (signal-to-interference ratio) requirement and will transmit higher power, creating more interference. The optimal strategy is to assign the schemes with their C/I in reverse relation to the corresponding path loss, so that higher C/I schemes are assigned to terminals in location with lower path loss (generally those closer to the base-station) [31].

VI. CASE STUDY

Figure 11 depicts three possible configurations at the user’s side in a DVB-RCT system. The objective of this example is to estimate the maximum number of users that could be supported by a service provider licensed to operate a 6 MHz channel dedicated to the return path and centered at 710 MHz.

Each type of service has different traffic demands. Therefore, the first step to determine the maximum number of users in one cell is to define what kind of service will be offered. For simplicity, we will consider

that each user operates with a constant rate $R_{user} = 128$ kbit/s. Most of the interactive TV applications do not need such a rate. However, DVB-RCT could be an alternative for suburban or rural areas, where cable, ADSL or cellular coverage is not available.

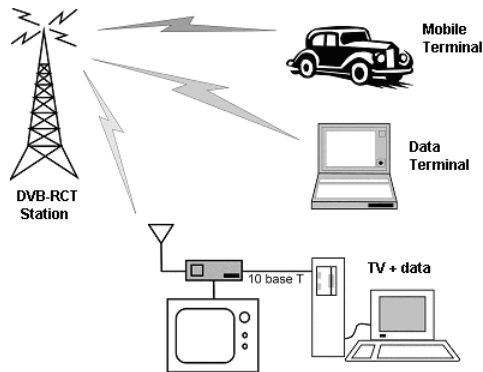


Figure 11 - Possible configurations of users in a DVB-RCT system.

In February 2002, the DVB group issued recommendation EN 301 195 [39], also known as DVB-RCG, as a ‘baseline specification for the provision of an interaction channel based on Global System for Mobile communications (GSM)’.

Another advantage of DVB-RCT over some of the aforementioned alternatives is that, once the link is established, it will be always on, with no extra costs each time the user access the medium. Therefore, besides the motivation to develop applications for interactive TV such as on-line purchase or tele-voting, the DVB-RCT could be imagined as an additional means to access the Internet. The similarities between DVB-RCT and its counterparts for satellite and cable (DVB-RCS and DVB-RCC) at the MAC and transport layers could be further studied for the sake of the design of a reference set-top box that is appropriate for the Brazilian market.

According to what was seen in Section IV.2, the maximum rate supported by a DVB-RCT channel depends on the code rate and on the type of modulation in use. The extremes are transmission modes QPSK with rate $R = 1/2$ and 64-QAM with rate $R = 3/4$. The net bit rate per sub-carrier depends also on the pulse shaping and the guard interval (for rectangular shaping). Once the net bit rate per sub-carrier is obtained, the total capacity of the DVB-RCT channel, $R_{available}$, can be computed as a function of the total number of used sub-carriers, which by its turn depends on the operation mode (see Section IV.2). Several DVB-RCT channels can be used until the whole 6 MHz of spectrum is filled. It should be clear that, if R_{user} is fixed, increasing $R_{available}$ also increases the number of subscribers that can be supported by the system.

IP traffic in the network can be highly *asymmetric*, depending on the users profile. An *Asymmetry Factor* ρ could be defined as:

$$\rho = \frac{R_{upstream}}{R_{user}} \quad (2)$$

where $R_{upstream}$ is the effective average traffic generated by the user in the upstream direction. Users that basically surf the web typically have asymmetry factor $\rho = 0.1$ (i.e., only 10% of the traffic is generated by the user). When there is a mix of e-mail, downloads and file exchanges using peer-to-peer software, the asymmetry factor goes up to 0.2 or even 0.5⁶. It is also useful to define the *Concentration Factor* γ as the relationship between N_{online} , the number of users that are on-line simultaneously, and N , the total number of users in the network.

$$\gamma = \frac{N_{online}}{N} \quad (3)$$

where $N_{online} = R_{disp} / R_{upstream}$. Without taking into consideration the IP headers, the total capacity of the system in terms of the maximum number of users can be expressed by this simple equation:

$$N_{max} = \frac{R_{available}}{\rho \cdot \gamma \cdot R_{user}} \quad (4)$$

Figure 12 shows the maximum number of users that could be supported by a cell where 30% of the users are simultaneously on-line and 25% of the traffic is generated by the user. The calculations were performed using a spreadsheet supplied by Runcom Technologies Ltd. and take into consideration the following parameters:

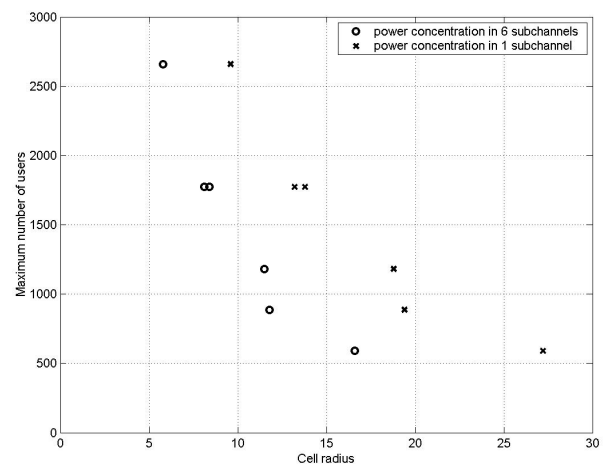


Figure 12 - Maximum number of users as a function of the cell radius (km).

- Guard interval: $T_g = 1/32 \times T_s$;
- NLOS (Non-Line-Of-Sight) propagation model;
- Transmission power at the user’s terminal: 1 W;
- Transmission power at the base-station: 100 W;
- Antenna gain at the user’s side: 18 dBi;
- Antenna gain at the base-station’s side: 18 dBi;
- Antenna height at the base-station: 50 m;
- Noise figure at the base-station receiver: 5 dB;
- Fading margin at the set-top box: 5 dB;
- Fading margin at the base-station: 5 dB.

⁶ The asymmetry factor is based on premises used by some cable operators to calculate the infrastructure needed to offer broadband connection via cable modem.

These calculations take into consideration only the traffic demand in the return channel. Since the tendency expressed by the asymmetry factor is that more throughput is required in the forward path, it is intuitive that only one 6 MHz channel may not be enough to accommodate the number of users shown in Figure 12. As an example, if 64-QAM with $\frac{3}{4}$ code rate is used in the return channel, the available rate $R_{available}$ is 25,53 Mbit/s, and this leads to a maximum number of users equal to:

$$N_{max} = \frac{25.53 \times 10^6}{0.25 \times 0.3 \times 128 \times 10^3} = 2,659$$

with a coverage radius of 9.72km if the power is concentrated in one sub-channel, as can be observed in Figure 12. A small change in Equation (4) makes it possible to calculate the rate required in the downstream to satisfy the number of users supported in the return channel.

$$N_{max} = \frac{R_{available}(downstream)}{(1-\rho) \cdot \gamma \cdot R_{user}} \quad (5)$$

Substituting the values for N_{max} , ρ and γ , yields:

$$\begin{aligned} R_{available}(downstream) &= N_{max} (1-\rho) \gamma R_{user} \\ &= 2,659 \cdot (1-0.25) \cdot 0.3 \cdot 128 \cdot 10^3 \\ &= 76.58 \text{ Mbit/s} \end{aligned}$$

Using 64-QAM and code rate $\frac{3}{4}$ in the forward DVB-T channel, the available rate is 27.14 Mbit/s and the coverage radius is 13.25 km. Therefore, if this combination is chosen at a given time, at least three 6 MHz channels would be necessary, exclusively for the forward path, to comply with the traffic demands of the 2,659 users.

VII. FINAL COMMENTS

After a short introduction on digital TV systems architectures, the physical layer of the DVB-RCT was presented. Although this system has been designed for applications with light demands for traffic, a case study investigated the capacity limits for a scenario where two 6 MHz channels were dedicated exclusively for IP-based services. The maximum number of users was calculated taking into consideration asymmetry and concentration factors. The case where the frequency is reused in adjacent cells will be covered in a forthcoming work.

The robustness of the system is important in the light of Brazilian marketplace, where more than a half of the existing TV sets receive off-air signals using internal antennas. Since 90% of the homes have at least one TV set and, up to this date, only 7% of the population have access to the internet, it seems reasonable the Government's inclination to use digital TV platform as a mechanism to stimulate what is being called digital inclusion. It might be useful, however, to broaden the

scope of the discussion about standardization to include the return channel in the list of priorities. Besides providing an overview about the possibility of implementation of a return channel, this article aims at stimulating those debates.

VIII. GLOSSARY

AAL: *ATM Adaptation Layer*
 ADSL: *Asynchronous Digital Subscriber Line*
 ANATEL: *Agência Nacional de Telecomunicações*
 APP: *A Posteriori Probability*
 ASIC: *Application-Specific Integrated Circuit*
 ATM: *Asynchronous Transfer Mode*
 ATSC: *Advanced Television Systems Committee*
 AVC: *Advanced Video Coding*
 BIM: *Broadcast Interface Module*
 BNA: *Broadcast Network Adapter*
 CLP: *Cell Loss Priority*
 CRSC: *Circular Recursive Systematic Convolutional*
 CS: *Carrier Spacing*
 CSI: *Channel State Information*
 DAMA: *Demand-Assigned Multiple Access*
 DAVIC: *Digital Audio Video Council*
 DFT: *Discrete Fourier Transform*
 DOCSIS: *Data Over Cable Interface Specification*
 DSP: *Digital Signal Processing*
 DTH: *Direct-To-Home*
 DTV: *Digital Tele-Vision*
 DVB: *Digital Video Broadcasting*
 DVB-RCC: *Digital Video Broadcasting: Return Channel-Cable*
 DVB-RCS: *Digital Video Broadcasting: Return Channel-Satellite*
 DVB-RCT: *Digital Video Broadcasting: Return Channel-Terrestrial*
 DVB-T: *Digital Video Broadcasting: Terrestrial*
 DVD: *Digital Versatile Disk*
 EL-APP: *Enhanced-Log-APP*
 EML-APP: *Enhanced-Max-Log-APP*
 EPG: *Electronic Program Guide*
 ES: *Elementary Stream*
 ETSI: *European Telecommunications Standards Institute*
 FEC: *Forward Error Correction*
 FFT: *Fast Fourier Transform*
 FHSS: *Frequency Hopping Spread Spectrum*
 GF: *Galois Field*
 GFC: *Generic Flow Control*
 GSM: *Global System for Mobile communications*
 HDTV: *High Definition Tele-Vision*
 HEC: *Header Error Check*
 HFC: *Hybrid Fiber-Coax*
 IIM: *Interactive Interface Module*
 INA: *Interactive Network Adapter*
 ISDB-T: *Integrated Services Digital Broadcast - Terrestrial*
 ICI: *Inter-Channel Interference*
 ISI: *Inter-Symbol Interference*
 ISO: *International Standards Organization*

ITU: *International Telecommunications Union*
 L-APP: *Log-APP*
 MAC: *Medium Access Control*
 MAP: *Medium Access Protocol*
 MP@HL: *Main Profile at High Level*
 MP@ML: *Main Profile at Main Level*
 MPEG: *Moving Pictures Experts Group*
 MSO: *Multi-Systems Operator*
 NIU: *Network Interface Unit*
 NLOS: *Non-Line-Of-Sight*
 OFDM: *Orthogonal Frequency Division Multiplexing*
 OFDMA: *Orthogonal Frequency Division Multiple Access*
 OOB: *Out-Of-Band*
 OS: *Operational System*
 PES: *Packetized Elementary Stream*
 PID: *Packet Identifier*
 PAT: *Program Association Table*
 PMT: *Program Map Table*
 PTI: *Packet Type Identifier*
 PVR: *Personal Video Recording*
 QAM: *Quadrature Amplitude Modulation*
 QoS: *Quality-of-Service*
 QPSK: *Quaternary Phase Shift Keying*
 RS: *Reed Solomon*
 RSC: *Recursive Systematic Convolutional*
 RTP/IP: *Real-Time Protocol / Internetworking Protocol*
 SAR: *Segmentation And Reassembly*
 SCTE: *Society for Cable Telecommunications Engineers*
 SBTVD: *Sistema Brasileiro de TV Digital*
 SDTV: *Standard Definition Tele-Vision*
 TCP/IP: *Transfer Control Protocol / Internetworking Protocol*
 TF: *Transmission Frame*
 TS: *Transport Stream*
 VC: *Virtual Channel*
 VCI: *Virtual Channel Identifier*
 VPI: *Virtual Path Identifier*
 VoIP: *Voice Over IP*
 VSB: *Vestigial Side Band*

IX. REFERENCES

- [1] BRASIL. Decreto n. 4.901, de 27 de novembro de 2003. Institui o Sistema Brasileiro de Televisão Digital – SBTVD, e dá outras providências. Atos do Poder Executivo, Brasília: Nov., 2003.
- [2] BRASIL. Exposição de Motivos do Decreto n. 4.901, de 27 de novembro de 2003. Brasília: November, 2003.
- [3] ETSI EN 301 958 V1.1.1, *Digital Video Broadcasting (DVB); Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM*, European Standard (Telecommunications series), Dec., 2001.
- [4] ETSI EN 300 744 V1.4.1, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, European Standard (Telecommunications series), Jan., 2001.
- [5] PREDERGAST, D.; WU, Y.; SALEHIAN, K.; GUILLET, M.; CARON, B. *Digital video broadcasting return channel terrestrial (DVB-RCT) and on channel repeaters (OCR) for broadband internet multimedia service to rural and remote access*. IBC 2004: International Broadcasting Convention, Amsterdam: Sep., 9 - 14, 2004, p. 121 – 131.
- [6] Datasheet, *Explorer® 8000HD™ Home Entertainment Server*, Scientific Atlanta Inc., Part Number: 4004400, Rev. C, Nov., 2003.
- [7] GONÇALVES, R. *Anatel e Minicom levantam bandeira da inclusão digital*. Tela Viva News. In <http://www.itweb.com.br/noticias/artigo.asp?id=47566> Accessed in Oct., 21st, 2004.
- [8] HAYKIN, S. *Communication Systems*. 4th. Ed. New York: John Wiley, 2001.
- [9] WHITAKER, J. *Standard Handbook of Video and Television Engineering*. 3rd Ed. New York: McGraw Hill, 2000.
- [10] ISO/IEC 13818-1 *MPEG-2: Systems*. International Standard, 1994.
- [11] ISO/IEC 13818-2 *MPEG-2: Video*. International Standard, 1994.
- [12] ISO/IEC 13818-3 *MPEG-2: Audio*. International Standard, 1994.
- [13] ATSC Advanced Television Systems Committee standard A/52, *Digital Audio Compression (AC-3)*, 10 Nov., 1994/12 Apr., 1995.
- [14] TEKTRONIX. *A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC)*, White Paper, 1997.
- [15] NALON, J. A.; IANO, Y. MPEG-2: Camada de Sistemas. *Telecomunicações*, v. 3, n. 1, p. 28-34, Apr., 2000.
- [16] ETSI EN 300 421. *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services*. *European Standard (Telecommunications series)*, 2001.
- [17] ITU-T International Telecommunication Union – Recommendation J.83: *Digital Multi-Programme Systems for Television Sound and Data Services for Cable Distribution*. Oct., 1995.
- [18] ATSC Advanced Television Systems Committee standard A/53, *Digital Television Standard*. Sep., 1995.
- [19] ETSI EN 301 790 V1.3.1. *Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems*. *European Standard (Telecommunications series)*. Mar., 2003.
- [20] FARIA, G. *DVB-RCT: The Missing Link for Digital Terrestrial TV*. Harris Broadcast Europe – ITIS products.
- [21] DOCSIS, *Data Over Cable Services Interface Specifications*, in www.cablemodem.com.
- [22] ANSI/SCTE. *Digital Broadband Delivery System: Out-of-band Transport Part 1: Mode A, 55-1*. USA Standard, 2002.
- [23] ANSI/SCTE. *Digital Broadband Delivery System: Out-of-band Transport Part 2: Mode B, 55-2*. USA Standard, 2002.

- [24] SOARES, L.F.; LEMOS, G.; COLCHER, S. *Redes de Computadores; das LAN's, MAN's e WAN's às Redes ATM* – 2ª. Ed. Rio de Janeiro: Campus, 1995.
- [25] DAVIC, *Digital Audio Video Council*, in www.davic.org.
- [26] FAZEL, K.; KAISER, S. *Multi-Carrier and Spread Spectrum Systems*. New York: John Wiley, 2003.
- [27] KAI-YEUNG, S.; RAJ, J. A Brief Overview of ATM: Protocol Layers, LAN Emulation, and Traffic Management. *ACM SIGCOMM Computer Commun. Rev.*, v.25, Issue 2, p. 6-20, 1995.
- [28] VAN NEE, R.; PRASAD, R. *OFDM for Wireless Multimedia Communications*. Norwood: Artech House, 2000.
- [29] CHANG, ROBERT W., *Orthogonal Frequency Multiplex Data Transmission System*, U.S. Patent Nr. 3,488,445. 1966.
- [30] WEINSTEIN, S. B.; EBERT, P. Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform. *IEEE Trans. Communication Tech.*, n. 5, p. 628 – 634. Oct., 1971.
- [31] KOFFMAN, I. Broadband Wireless Access Solutions Based on OFDM Access in IEEE 802.16. *IEEE Communications Mag.*, n. 4, p. 96 – 103. Apr., 2002.
- [32] SKLAR, B. *Digital Communications – Fundamentals and Applications*. 2 Ed. Upper Saddle River: Prentice Hall, 2001.
- [33] BERROU, C. et al. The Turbo Code Standard for DVB-RCS. *2nd International Symposium on Turbo Codes and Related Topics*: p. 535 – 538. Sep., 2000.
- [34] BERROU, C. Turbo Codes: Some Simple Ideas for Efficient Communications. *Seventh International Workshop on Digital Signal Processing Techniques for Space Communications*. Noordwijk, The Netherlands. Oct. 29-31, 2001.
- [35] FORNEY, G. D. *Concatenated Codes*. MIT Press, 1966.
- [36] BERROU, C.; JÉZÉQUEL, M. Non-binary convolutional codes for turbo coding. *Electronics Lett.* v.35, n. 1, p. 39-40. Jan., 1999.
- [37] Datasheet, *RN-2821 Typhoon*, Runcom Technologies Ltd., RN2821 02-0602, 2002.
- [38] KABAL, P.; GUINAND, P., OULD-CHEIKH-MOUHAMEDOU, Y. Enhanced Max-Log_APP and Enhanced Log-APP Decoding for DVB-RCS. *Proc. Int. Symposium on Turbo Codes and Related Topics*, p. 259 – 263. Brest, France. Sep., 2003.
- [39] ETSI EN 301 195. Digital Video Broadcasting (DVB): Interaction channel through the Global System for Mobile communications (GSM). *European Standard (Telecommunications series)*. Feb., 2002.
- [40] ITU-T International Telecommunication Union – Recommendation H.264: *Advanced Video Coding for Generic Audiovisual Services*, 2003.

ABOUT THE AUTHORS

Ângelo Pinelli Martins Samia was born in Olímpio Noronha, MG, Brazil, on September 17, 1971. He graduated in Electrical Engineering at Inatel in 1994 and obtained his Master degree in Telecommunications in 2004 at the same institution. He worked with Alcatel Telecomunicações S/A for two years in the Marketing Department and for two more years doing technical support in the PABX range of products. Ângelo was hired by Scientific Atlanta do Brasil in 1998, where he became responsible for structuring both Technical Support and Post-Sales activities for the customers in Latin America. He is currently with Tecsys Industrial do Brasil Ltda., collaborating with the Engineering team to develop solutions for digital TV processing and transmission over cable and satellite networks.

Dayan Adionel Guimarães was born in Carrancas, MG, Brazil, on March 01, 1969. He holds the titles: Doctor in Electrical Engineering (Unicamp, 2003); Master in Electrical Engineering (Unicamp, 1998); post-graduated (*lato-sensu*) in Data Communication Engineering (Inatel, 2003); post-graduated (*lato-sensu*) in Administration with emphasis in Human Resources Management (FAI, 1996) and Electronics Technician (ETE “FMC”, 1987). In former job functions Dr. Dayan developed sensors and equipment for instrumentation and control and also held the Production Supervisor and Product Engineering Supervisor positions at SENSE Sensores e Instrumentos, from 1988 to 1993. He was also responsible for the structure that supports practical teaching activities related to Telecommunications and Electronics at Inatel, an Institute of Telecommunications and Computer Engineering in Brazil, where, since 1995, his is a Professor. His research includes the general aspects of wireless communications, especially those related to Multicarrier CDMA systems and channel coding for fading channels, especially Block Turbo Codes.

Noções Sobre Decodificação Turbo: Um Estudo de Caso para os Códigos SPC-TPC

Dayan Adionel Guimarães

Departamento de Telecomunicações – Inatel. e-mail: dayan@inatel.br

Resumo - Neste tutorial são apresentados os principais conceitos sobre o processo de decodificação turbo, com enfoque na decodificação turbo de códigos de bloco. Os códigos produto de paridade simples foram adotados como estudo de caso. A decodificação dos códigos componentes é realizada por meio de um algoritmo MAP símbolo-a-símbolo. Alguns resultados de simulação com modulação BPSK com detecção coerente em canal AWGN e com desvanecimento plano do tipo Rayleigh são fornecidos e discutidos.

Palavras-chave - Decodificação Turbo, Códigos Produto de Paridade Simples, Códigos Turbo de Bloco.

Abstract - In this tutorial, the main concepts associated to the turbo decoding process are presented, with emphasis on turbo decoding of block codes. Single-Parity Check Turbo Product Codes (SPC-TPC) are considered as a case study. A symbol-by-symbol MAP algorithm is used for decoding the SPC-TPC component codes. Some simulation results using coherent BPSK modulation on the AWGN and flat Rayleigh fading channels are also presented and discussed.

Key words - Turbo Decoding, Single-Parity Check Product Codes, Block Turbo Codes.

I - INTRODUÇÃO

EM 1966 G. DAVID FORNEY JR. propôs uma técnica de codificação de canal implementada através da combinação de códigos componentes. Com essa técnica era possível obter códigos resultantes de maior comprimento e com maior capacidade de correção de erros que aqueles proporcionados pelos códigos componentes individualmente. Ainda assim, permitia-se que o processo de decodificação, dividido em etapas associadas a cada código componente, se tornasse menos complexo que aquele que seria necessário para decodificar um único código equivalente de comprimento igual ao do código resultante. Ao código resultante dessa implementação foi dado o nome de código concatenado [1].

Em 1989, J. Hagenauer e P. Hoeher propuseram uma variante do algoritmo de decodificação de Viterbi. Com ela eram produzidas decisões suaves e através destas uma decisão abrupta poderia ser tomada. Ou então, e principalmente, em um esquema com códigos concatenados uma métrica associada à confiabilidade dessa decisão poderia ser utilizada como entrada suave do processo de decodificação consequente. A essa variante do algoritmo de Viterbi foi dado o nome de SOVA (*Soft-Output Viterbi Algorithm*) [4].

Alguns anos depois, em 1993, os *códigos turbo* foram inventados por C. Berrou, A. Glavieux e P. Thitimajshima

[2] na *Ecole Nationale Supérieure des Télécommunications de Bretagne*, França, tendo como forte motivação as idéias de [4] e incorporando a essência da construção proposta em [1]. Este novo esquema de codificação de canal corresponde à concatenação paralela de códigos convolucionais recursivos e sistemáticos, decodificados iterativamente por um algoritmo baseado no algoritmo MAP (*maximum a posteriori*) símbolo-a-símbolo BCJR¹ [3].

Embora a invenção de 1993 se referisse apenas àquela forma de codificação e decodificação, de forma genérica, atualmente, pode-se classificar como *código turbo* todo esquema de codificação de canal que utilize: 1) processos de decodificação iterativa e 2) a concatenação de códigos componentes separados por entrelaçadores temporais. Assim, percebe-se que o termo “turbo” está diretamente associado à decodificação iterativa e não necessariamente à forma de implementação da codificação, embora os códigos componentes utilizados em [2] tenham sido também propostos pela primeira vez naquela publicação. Conforme afirmam Claude Berrou e Alain Glavieux [7]:

“Os códigos turbo são resultado de uma construção empírica e trabalhosa de um esquema completo de codificação e decodificação, utilizando blocos existentes que nunca haviam sido colocados juntos daquela forma antes”.

Os códigos turbo utilizam algoritmos do tipo SISO (*Soft-Input, Soft-Output*) no processo de decodificação iterativa. Nestes algoritmos, informações sobre a confiabilidade ou qualidade da decodificação de um dos códigos componentes (*soft output*) alimentam o processo de decodificação de outro código componente, na forma de entrada suave (*soft input*). Dessa forma, a cada iteração tem-se maior confiabilidade na estimação do bit, palavra ou sequência transmitida, dependendo da forma específica de implementação da decodificação. A obtenção dessa confiabilidade, denominada de informação extrínseca em [2], incorpora um princípio já identificado por Gallager em [5] e, em paralelo com as investigações de C. Berrou *et al.*, por Lodge *et al.* em [6]. Entretanto, todos estes trabalhos ocorreram de forma independente [7].

Hoje, pouco mais de dez anos após a invenção dos códigos turbo, as pesquisas sobre o tema se encontram em um estágio significativamente avançado e ramificaram-se, dando surgimento ao chamado *processamento turbo* no qual, de forma genérica, para a realização de um determinado processo há troca de informação entre sub-processos componentes que cooperam entre si de forma iterativa.

¹ A sigla BCJR foi adotada devido aos sobrenomes dos inventores do algoritmo: L. R. BAHL; J. COCKE; F. JELINEK e J. RAVIV.

Dentre as várias técnicas nas quais o processamento turbo pode ser aplicado destacam-se a *equalização*, a *estimação de canal*, a *codificação de fonte e canal conjunta*, a *detecção multi-usuário* e o *cancelamento de interferências*, os *sistemas MIMO (Multiple Input, Multiple Output)* e a *codificação espaço-temporal*, apenas para citar alguns exemplos.

O processamento turbo está sendo interpretado como uma das mais promissoras técnicas para a melhoria de desempenho em sistemas de comunicação. Por essa razão, a invenção dos códigos turbo está sendo considerada como o segundo grande marco do desenvolvimento científico das comunicações, desde o desenvolvimento da teoria matemática da comunicação [18] no final da década de 40. Qualquer sistema com realimentação, conforme desafiou Simon Haykin na apresentação de [19], a partir de agora deve ser interpretado não simplesmente como um sistema onde há realimentação de sinais, mas sim onde há *realimentação de informação*, como acontece no processamento turbo. De fato, durante as investigações que culminaram na invenção dos códigos turbo, C. Berrou *et al.* foram inspirados por uma idéia de amplificação de informação a partir de uma estrutura com realimentação [7]. Nesta estrutura pode-se interpretar como condição de convergência e estabilidade do processamento turbo a adequada configuração do “ponto de operação” desse amplificador.

As demais seções deste tutorial estão organizadas da seguinte maneira: na Seção II são apresentados alguns conceitos iniciais genéricos sobre os códigos turbo. Na Seção III faz-se uma revisão sobre os códigos produto de paridade simples, com destaque para as regras de implementação de um código multidimensional. A Seção IV aborda o processo de decodificação turbo de códigos produto multidimensionais e a Seção V trata especificamente da decodificação turbo de códigos produto com componentes de paridade simples. Na Seção VI é fornecido um exemplo de decodificação turbo de um código produto bidimensional com componentes de paridade simples. Na Seção VII são tecidos alguns comentários sobre a influência do aumento de dimensões e do processo de entrelaçamento temporal no desempenho de códigos produto de paridade simples. A Seção VIII apresenta e tece comentários sobre alguns resultados de desempenho de códigos produto de paridade simples nos canais AWGN e com desvanecimento Rayleigh plano. Por fim, no Apêndice trata-se de uma rotina, elaborada na plataforma Mathcad 2001i, para codificação e decodificação turbo dos códigos em questão.

Muitas das conclusões e interpretações aqui registradas devem parecer, num primeiro momento, não corresponder a um estudo introdutório simplificado sobre códigos turbo. E de fato não correspondem! Optou-se por elaborar um tutorial mais abrangente e menos básico, evitando-se repetir a estrutura de excelentes trabalhos básicos já publicados, mas complementando-os de certa forma. Por esta razão, recomenda-se que o estudo deste texto seja realizado em etapas: primeiramente deve-se buscar entender os conceitos básicos associados aos princípios da decodificação turbo, por exemplo utilizando as referências [10] e [22], principalmente os exemplos lá fornecidos. Posteriormente recomenda-se um aprofundamento gradativo deste tutorial, consultado as demais refe-

rências citadas ao seu final, sempre que necessário. A rotina computacional citada no Apêndice pode ser de particular utilidade para auxiliar o aprendizado.

II - CONCEITOS INICIAIS

BASICAMENTE TÊM-SE duas famílias de códigos turbo: uma baseada na concatenação de códigos convolucionais (CTC, *Convolutional Turbo Codes*) [2][11] e outra baseada na concatenação de códigos de bloco (BTC, *Block Turbo Codes*) [12]. Quando da invenção dos códigos turbo, C. Berrou *et al.* trataram dos códigos turbo convolucionais [2]. Mais recentemente, grande interesse tem sido demonstrado por implementações de processos de decodificação iterativa baseados em códigos de bloco [8][10][12]-[16]. Os principais objetivos dessas implementações se referem à possibilidade de redução na complexidade e aumento na velocidade de decodificação e também à possibilidade de aumento de desempenho em relação aos códigos turbo convolucionais, para códigos de taxas altas [10][12][14] e blocos relativamente curtos. Para taxas baixas, os CTCs tendem a apresentar melhor desempenho. Outra vantagem aparente dos códigos turbo de bloco se refere à possibilidade de redução do valor do “patamar de saturação da taxa de erro de bit” ou “joelho” (do Inglês *error floor*) que é percebido em curvas de taxa de erro de bit *versus* relação sinal-ruído média por bit, para esses códigos. Esse joelho corresponde a um comportamento similar a uma saturação na taxa de erro de bit, fazendo com que esta seja reduzida apenas de forma marginal, mesmo com significativos aumentos na relação E_b/N_0 . O fenômeno ocorre principalmente devido à presença de um número significativo de palavras-código de baixo peso, posto que em valores mais altos de E_b/N_0 a taxa de erro de bit passa a ser governada predominantemente por estas palavras-código [15]. O joelho pode ser reduzido através de melhorias nos processos de entrelaçamento temporal entre os códigos componentes [17].

Destaca-se uma interpretação sobre o processo de entrelaçamento temporal supracitado: este processo, localizado entre códigos concatenados, tem um propósito distinto daquele entrelaçamento que tipicamente é realizado após a codificação de canal. Neste último caso objetiva-se a transformação de um comportamento de memória² do canal num comportamento sem memória, do ponto de vista do decodificador de canal localizado no receptor. No que diz respeito ao entrelaçamento temporal realizado entre os códigos componentes da concatenação, o principal objetivo é fazer com que as informações trocadas no processo de decodificação iterativa sobre um determinado símbolo ou palavra-código sejam tão descorrelacionadas quanto possível, o que trará, iteração a iteração, melhorias mais significativas no desempenho do código turbo. De fato, há grande influência da forma com que é realizado o entrelaçamento temporal no desempenho do processo de decodificação iterativa, especificamente, e no desempenho do código turbo, genericamente.

² Um exemplo típico de canal com memória é o canal rádio-móvel, onde podem ocorrer erros de símbolo em rajada. O exemplo clássico de canal sem memória é o canal AWGN, no qual os erros são descorrelacionados, permitindo melhor desempenho do processo de correção de erros realizado pelo esquema de codificação de canal.

Surpreendentes resultados de desempenho de códigos turbo foram reportados na literatura: em [2]³, publicação que marca a invenção desses códigos, a 10^{-5} de taxa de erro de bit pôde-se operar a uma E_b/N_0 distante apenas 0,5 dB do limite de Shannon em canal AWGN, para modulação BPSK ($\cong 0,19$ dB). Desempenhos resultando na operação a uma E_b/N_0 distante 0,2 dB e 0,35 dB da capacidade do canal AWGN foram reportados em [8] e [7], respectivamente, também para modulação BPSK. Mais recentemente, potentes códigos Turbo e LDPC (*Low Density Parity-Check*) [23][24] chegam a operar a menos de 0,1 dB do limite de Shannon.

Apesar dos ganhos de codificação significativos que podem ser obtidos com os códigos turbo, um dos grandes obstáculos a ser ainda transposto se refere à simplificação dos algoritmos de decodificação: aqueles considerados ótimos segundo o critério MAP símbolo-a-símbolo, via de regra são complexos, demandando altas velocidades de processamento para que certas aplicações onde há comunicação em tempo real, por exemplo, possam ser viabilizadas. Várias pesquisas têm sido encaminhadas no sentido de desenvolver novas formas de algoritmos ótimos ou de desenvolver algoritmos sub-ótimos, ou ainda no sentido de modificar algoritmos ótimos, tornando-os sub-ótimos, porém com menor grau de complexidade.

Os códigos turbo permitem ainda que, a um moderado grau de complexidade, seja possível operar com valores de E_b/N_0 abaixo da taxa de corte do canal [9]. Até então se sabia que esta era uma tarefa possível, mas de alta complexidade de realização [10], pois a taxa de corte era considerada como a “capacidade prática” do canal.

III - CÓDIGOS PRODUTO DE PARIDADE SIMPLES

NESTE TEXTO foram adotados como foco de estudo os códigos turbo implementados com códigos de bloco, tendo por certo que essa escolha é suficiente à apresentação dos principais fundamentos sobre a decodificação turbo. Dentre as várias configurações que podem ser obtidas combinando-se diferentes códigos componentes de diferentes taxas, diferentes regras de entrelaçamento temporal e diferentes esquemas de decodificação iterativa, a família dos códigos produto [20] foi escolhida. Dentre os códigos produto, abordagens teóricas e exemplos são aqui apresentados para os códigos produto de paridade simples (SPC-PC, *Single-Parity Check Product Codes*), para os quais a decodificação dos códigos componentes é efetuada pelo algoritmo MAP símbolo-a-símbolo proposto em [15].

Os códigos produto podem ser interpretados como pertencentes à família dos códigos de arranjo [21], nos quais o processo de codificação pode ser associado a uma construção geométrica composta por um arranjo de códigos componentes. Além de grande simplicidade de implementação, os códigos produto apresentam ampla flexibilidade em termos de adequação da taxa do código resultante e do comprimento do bloco.

Um código produto de dimensão D pode ser definido a partir do comprimento do bloco de informação em cada

dimensão $\{k_1, k_2, \dots, k_i, \dots, k_D\}$. Podem ser utilizados códigos componentes sistemáticos ou não-sistemáticos (n_i, k_i, d_{\min}). O código produto resultante possui palavras-código de comprimento

$$v = \prod_{i=1}^D n_i \tag{1}$$

Se $r_i = k_i/n_i$ é a taxa do código componente na i -ésima dimensão, a taxa do código produto será

$$r = \prod_{i=1}^D r_i \tag{2}$$

e a distância mínima será

$$\delta_{\min} = \prod_{i=1}^D d_{\min,i} \tag{3}$$

Se os códigos componentes são de paridade simples e idênticos em todas as dimensões, ou seja, códigos caracterizados por $(n, k, d_{\min}) = (n, n - 1, 2)$, o código produto (v, κ, δ_{\min}), denotado por $(n, k, d_{\min})^D$, terá então:

$$v = n^D$$

$$\kappa = k^D = (n - 1)^D \tag{4}$$

$$\delta_{\min} = 2^D$$

A Figura 1 mostra a estrutura geométrica de um código produto bidimensional (2D). Além dos bits de paridade gerados pelo processo de codificação em cada dimensão, o código produto possui os bits de paridade gerados a partir da paridade das paridades (*check on checks*). Neste caso pode-se verificar que se trata da concatenação serial dos códigos componentes.

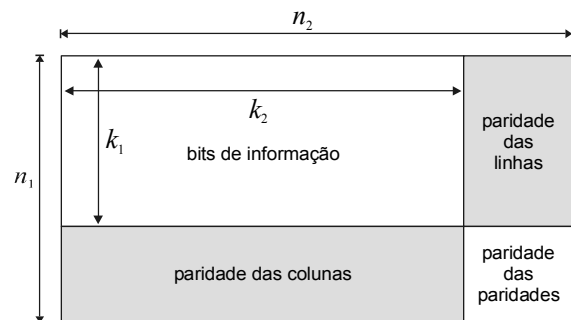


Figura 1 - Estrutura do código produto bidimensional completo.

Certos autores ainda definem uma outra categoria de códigos produto: o código produto incompleto [10][22], correspondente à concatenação paralela dos códigos componentes. Um código produto incompleto não possui a paridade das paridades e, apesar de ter implementação ligeiramente mais simples, possui desempenho inferior, assintoticamente. Isto se deve às principais razões: a distância mínima do código incompleto é menor que a do código completo [15] e o número de palavras-código de mais baixo peso é maior no código incompleto [25].

As propriedades descritas pelas expressões (1) - (4) não se aplicam aos códigos incompletos. Sendo assim, deste ponto em diante no texto o termo *código produto*

³ O resultado de simulação apresentado em [2] foi obtido com o uso de codificadores convolucionais recursivos e sistemáticos, um extenso bloco de entrelaçamento temporal pseudo-aleatório entre os codificadores (65.536 bits), 18 iterações e alguns ajustes no algoritmo BCJR.

refere-se ao *código completo*, a menos que esteja explícito o contrário.

A concatenação serial presente na estrutura D -dimensional de um código produto permite que sejam também utilizados códigos componentes não-sistemáticos [26]. Tanto a taxa de codificação final quanto a distância mínima de Hamming e também o comprimento do bloco do código serão iguais àqueles obtidos com códigos componentes na sua forma sistemática.

Cogita-se que a concatenação paralela de códigos de bloco seja mais adequada naquelas situações nas quais se deseja a operação do sistema codificado a baixos valores de E_b/N_0 , porém podendo suportar relativamente altos valores de taxa de erro de bit (BER, *Bit Error Rate*), acima do “joelho” da curva BER versus E_b/N_0 . É aparente que, na concatenação serial de códigos de bloco, têm-se maiores valores de E_b/N_0 a altos valores de taxa de erro de bit, mas o decréscimo da BER com o aumento de E_b/N_0 é consideravelmente mais abrupto, possibilitando transmissões praticamente “livres de erro” a valores de E_b/N_0 acima de um determinado limiar. Em [27, p. 925] e [28, p. 623] têm-se exemplos deste comportamento para códigos convolucionais concatenados. Uma comparação entre resultados apresentados em [15], [25] e [26, Capítulo 3] permite a identificação de exemplos similares para códigos de bloco concatenados. A Figura 2 ilustra o comportamento descrito neste parágrafo.

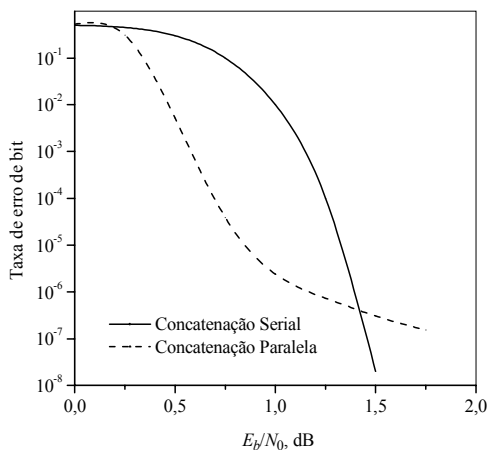


Figura 2 - Comportamento típico do desempenho de códigos de bloco concatenados em série e em paralelo.

Quanto aos processos de entrelaçamento temporal utilizados na formação do código produto, podem-se utilizar simples arranjos retangulares em bloco. O vetor contendo os bits a serem entrelaçados alimenta o bloco de entrelaçamento pelas linhas e o vetor entrelaçado é lido pelas colunas, alimentando o codificador seguinte k a k bits. O número de linhas e o número de colunas do bloco de entrelaçamento entre as dimensões (códigos concatenados) i e $i + 1$, números esses governados pela própria estrutura geométrica dos códigos produto, são dados por [26, p. 114]

$$(N_i \times N_c)^{i,i+1} = (n^i k^{D-i-1} \times n) \quad (5)$$

onde n é o comprimento da palavra-código do código componente, k é o comprimento da palavra de informação do código componente, D é o número de dimensões do código produto (número de códigos concatenados) e $i = 0, 1, \dots, D - 1$ é o índice de cada dimensão. A Figura 3 ilustra

a sequência de codificação de um código produto tridimensional (3D) com códigos componentes de paridade simples (3,2,2), denotado por (3,2,2)³. Como os códigos de paridade simples são sistemáticos, o cubo contendo os k^3 bits de informação, após a formação do código produto 3D, fica localizado na parte da frente, superior e esquerda do arranjo de n^3 bits, com os bits na mesma disposição daqueles do arranjo inicial de k^3 bits, conforme ilustra a Figura 4.

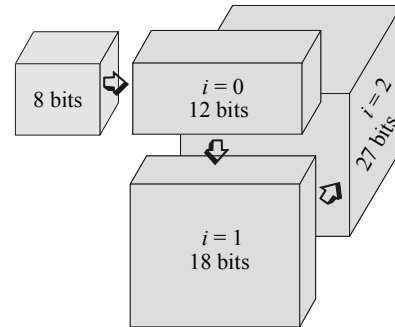


Figura 3 - Estrutura do código produto 3D, para códigos componentes (3,2,2).

Tomando novamente como referência o exemplo ilustrado pela Figura 3, as etapas de formação do arranjo correspondente a uma palavra-código do código produto de paridade simples 3D com componentes (3,2,2) são apresentadas no Algoritmo 1.

Uma propriedade relevante dos códigos produto, principalmente do ponto de vista da decodificação, corresponde ao fato de que *todas as palavras em um determinado sentido da estrutura de dimensão D são palavras-código dos códigos componentes* [26]. Por exemplo, na estrutura 3D mostrada na Figura 3, o bloco de $n^D = 27$ bits é formado por $n^{D-1} = 9$ palavras-código de $n = 3$ bits dispostas no sentido de suas três faces, totalizando $Dn^{D-1} = 27$ palavras-código do código componente (3,2,2). Objetivando melhor ilustrar este processo, admita que \mathbf{R} corresponda ao arranjo tridimensional de bits recebidos, conforme mostra a Figura 5. Em todos os sentidos indicados pelas setas da Figura 5 têm-se $n^{D-1} = 9$ palavras-código de n bits dos códigos componentes. Mais adiante essa propriedade será explorada no contexto da decodificação turbo de códigos produto.

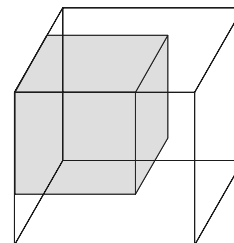


Figura 4 - Disposição dos bits de informação (bloco com preenchimento sólido) na estrutura do código produto 3D, para códigos componentes sistemáticos quaisquer.

IV - DECODIFICAÇÃO TURBO DE CÓDIGOS PRODUTO

PARA A CONCATENAÇÃO SERIAL de dois ou mais códigos de bloco separados por entrelaçadores temporais, formando a estrutura de um código produto, o

processo de decodificação iterativa pode valer-se de algoritmos derivados de algoritmos de decodificação de códigos produto, desde que estes algoritmos possam operar com estradas suaves e forneçam decisões suaves como saída. A Figura 6 apresenta o diagrama de um decodificador com entrada e saída suaves (SISO) usado na composição de um decodificador turbo, para códigos produto com componentes sistemáticos ou não-sistemáticos.

Algoritmo 1 - Formação de códigos produto de paridade simples: exemplo para o código (3,2,2)³.

1. Forme o vetor de entrada composto de $k^D = 2^3 = 8$ bits de informação;
2. na dimensão $i = 0$, codifique k a $k = 2$ a 2 , os $k^D = 2^3 = 8$ bits de entrada, gerando um vetor com $n^{i+1}k^{D-i-1} = 3^{0+1}2^{3-0-1} = 12$ bits;
3. faça o entrelaçamento temporal nesse vetor de 12 bits, entre as dimensões 0 e 1, alimentando pelas linhas um arranjo do tipo linha-coluna com $(N_l \times N_c)^{i, i+1} = (n^i k^{D-i-1} \times n) = (3^0 2^{3-0-1} \times 3) = (4 \times 3)$ bits; a leitura dos bits entrelaçados é feita pelas colunas desse arranjo;
4. na dimensão $i = 1$, codifique, 2 a 2, os 12 bits anteriormente entrelaçados, gerando um vetor com $n^{i+1}k^{D-i-1} = 3^{1+1}2^{3-1-1} = 18$ bits;
5. faça o entrelaçamento temporal no vetor de 18 bits, entre as dimensões 1 e 2, alimentando pelas linhas um arranjo do tipo linha-coluna com $(N_l \times N_c)^{i, i+1} = (n^i k^{D-i-1} \times n) = (3^1 2^{3-1-1} \times 3) = (6 \times 3)$ bits; a leitura dos bits entrelaçados é feita pelas colunas desse arranjo;
6. na dimensão $i = 2$, finalmente codifique, 2 a 2, os 18 bits anteriormente entrelaçados, gerando um vetor com $n^{i+1}k^{D-i-1} = 3^{2+1}2^{3-2-1} = 27$ bits.

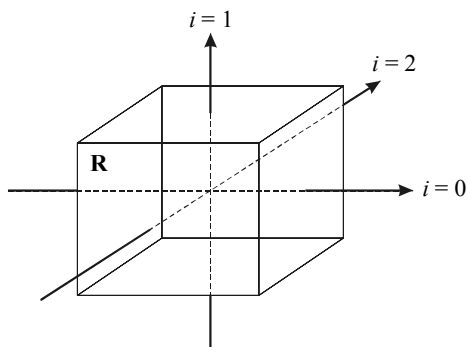


Figura 5 - Ilustração dos sentidos de decodificação elementar em cada dimensão de um código produto 3D.

A entrada suave destacada na Figura 6 está diretamente associada à saída do detector no receptor. Esta saída contém a chamada *informação de estado de canal*⁴ obtida para todos os bits codificados, mais a informação *a priori* somente dos bits de informação (para códigos componentes sistemáticos) ou para todos os bits codificados (para

⁴ No presente contexto, por *informação de estado de canal* entende-se uma métrica associada ao valor real obtido na saída de um filtro casado ou correlator (ou banco de filtros casados ou de correlatores), no momento ótimo que seria utilizado para decisão abrupta sobre o correspondente símbolo recebido.

códigos componentes sistemáticos ou não-sistemáticos). Tais valores são operados tipicamente no domínio logarítmico. As variáveis apresentadas na Figura 6 serão definidas e utilizadas mais adiante.

De forma a melhor ilustrar a composição da entrada suave mostrada na Figura 6, admita um canal de comunicação com desvanecimento plano do tipo Rayleigh. Após o sinal transmitido passar por este canal pode-se determinar a entrada suave do decodificador SISO através da *razão de verossimilhança* - que em escala logarítmica pode ser chamada de *razão de verossimilhança logarítmica* [10, p. 430] [28, p. 32] do símbolo $d \in \{\pm\sqrt{E}\}$, condicionada à saída x do filtro casado de recepção e à amplitude g do desvanecimento (*ganho do canal*):

$$L(d | x, g) = \ln \left[\frac{\Pr(d = +\sqrt{E} | x, g)}{\Pr(d = -\sqrt{E} | x, g)} \right] \quad (6)$$

Pelo *Teorema de Bayes* pode-se reescrever (6) como:

$$L(d | x, g) = \ln \left[\frac{p(x, g | d = +\sqrt{E}) \Pr(d = +\sqrt{E})}{p(x, g | d = -\sqrt{E}) \Pr(d = -\sqrt{E})} \right] \quad (7)$$

Como o canal com desvanecimento Rayleigh plano pode ser considerado como um canal condicionalmente gaussiano (condicionado à magnitude do desvanecimento), tem-se:

$$L(d | x, g) = \ln \left\{ \frac{\exp \left[-\frac{(x - g\sqrt{E})^2}{N_0} \right]}{\exp \left[-\frac{(x + g\sqrt{E})^2}{N_0} \right]} \right\} + \ln \left[\frac{\Pr(d = +\sqrt{E})}{\Pr(d = -\sqrt{E})} \right] \quad (8)$$

$$= L_c x + L(d) = 4g \frac{\sqrt{E}}{N_0} x + L(d) = 2g \frac{\sqrt{E}}{\sigma^2} x + L(d)$$

onde $\sigma^2 = N_0/2$ é a variância da componente de ruído gaussiano na variável de decisão, $N_0/2$ é densidade espectral de potência bilateral desse ruído e a variável $L_c = 4g\sqrt{E}/N_0$ pode ser interpretada como a *confiabilidade do símbolo* [10, p. 430] [28, p. 62]. Ainda com relação à expressão (8), E é a energia média por símbolo e $L(d)$ corresponde aos valores *a priori* do símbolo d . Para o canal sem desvanecimento (AWGN puro) basta fazer $g = 1$ em (8).

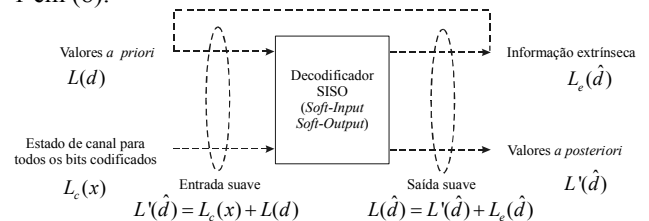


Figura 6 - Ilustração do processo de decodificação turbo (adaptado de [10] e [22])

No processo de decodificação iterativa, os valores *a priori* não são conhecidos antes da primeira iteração completa e normalmente são considerados nulos nessa etapa (probabilidades *a priori* iguais a 1/2). A *saída suave* do

decodificador SISO é composta pelos valores *a posteriori* dos bits de informação (para códigos componentes sistemáticos) ou de todos os bits codificados (para códigos componentes sistemáticos ou não-sistemáticos), mais a *informação extrínseca* obtida pelo processo de decodificação. Esse valor de informação extrínseca é realimentado à entrada do decodificador SISO como o valor da verossimilhança logarítmica *a priori* para a próxima iteração. Este procedimento faz com que os novos valores das métricas de entrada possam produzir valores de métricas de saída mais confiáveis, iteração a iteração.

A informação extrínseca pode ser interpretada como a quantidade de informação adicionada à entrada suave do decodificador para formar o valor correspondente à saída suave [30]. Ela pode também ser interpretada como a quantidade de informação obtida pelo processo de decodificação [22], porém independente dos valores antes deste processo (na saída do detector) [11]. A informação extrínseca pode ainda ser definida como a diferença entre a métrica calculada na saída do estágio de decodificação (saída suave) e a informação intrínseca representada por uma métrica realimentada à entrada do estágio de decodificação (entrada suave); *é a informação adicional obtida através da exploração das dependências que existem entre os bits de informação e os bits codificados* [31], *segundo cada regra específica de codificação*.

A razão de verossimilhança logarítmica para um determinado símbolo na entrada do decodificador SISO, deste ponto em diante representada por LLR (*Log-Likelihood Ratio*), pode ser expressa de forma simplificada por

$$L(d | x, g) = L(x | d, g) + L(d) \quad (9)$$

A métrica $L(x | d, g)$ é a LLR das medidas do canal x , dadas as condições de transmissão de um 0 ou de um 1 ($d = \pm \sqrt{E}$) e dado o ganho do canal g . $L(d)$ é a LLR *a priori* do símbolo d . Para o canal AWGN pode-se simplificar ainda mais a notação usada em (9), tal que se obtenha [22]:

$$L'(\hat{d}) = L_c(x) + L(d) \quad (10)$$

onde $L'(\hat{d})$ é a LLR estimada de um determinado símbolo d , apresentada à entrada do decodificador SISO, e $L_c(x)$ é a LLR da medida do estado de canal, correspondente ao símbolo d e obtida a partir da saída do detector.

Pode-se mostrar [11] que a LLR na saída do decodificador SISO (saída suave) pode ser expressa por:

$$L(\hat{d}) = L'(\hat{d}) + L_e(\hat{d}) \quad (11)$$

onde $L'(\hat{d})$ é a LLR de um símbolo (valor *a posteriori*) e $L_e(\hat{d})$ é definida como a LLR extrínseca ou *informação extrínseca* desse símbolo.

Combinando (10) e (11) pode-se escrever a LLR obtida através do processo de decodificação SISO como:

$$L(\hat{d}) = L_c(x) + L(d) + L_e(\hat{d}) \quad (12)$$

Esta LLR contém a informação necessária para decodificação abrupta do símbolo d , dada pela polaridade de

$L(\hat{d})$, bem como a informação de confiabilidade dessa decisão, dada pela magnitude de $L(\hat{d})$. Relembrando, $L_c(x)$ representa a LLR obtida a partir do valor de saída do detector, $L(d)$ é a LLR *a priori* do símbolo d e $L_e(\hat{d})$ é a informação extrínseca gerada pelo, e dependente do, processo de decodificação.

Na Figura 6 apresentou-se o diagrama de um decodificador SISO, onde agora pode-se observar como as variáveis supramencionadas são operadas. A saída do detector provê a LLR que contém a informação de canal obtida para todos os símbolos codificados, mais a informação *a priori* para: 1) os símbolos de informação [10] [22] ou 2) todos os símbolos codificados [15][26]. Como mencionado anteriormente, no processo de decodificação iterativa estes valores *a priori* não são conhecidos antes da primeira iteração completa e normalmente são considerados nulos (probabilidades *a priori* iguais).

A saída do decodificador SISO é composta pelos valores *a posteriori* dos símbolos de informação, mais a informação extrínseca obtida pelo processo de decodificação. Esse valor de informação extrínseca é realimentado à entrada do decodificador SISO como o valor da LLR *a priori* do símbolo em questão para a próxima iteração. Este procedimento faz com que os novos valores de LLR de entrada possam produzir valores de LLR de saída mais confiáveis, iteração a iteração.

Como mostrado mais adiante com o auxílio de um exemplo, no processo de decodificação turbo de códigos produto de paridade simples a magnitude da informação extrínseca de um símbolo em particular é *aproximadamente* igual à menor das magnitudes das LLR's dos outros símbolos em uma palavra-código, e o sinal dessa informação extrínseca é igual ao próprio sinal dessa LLR, se a decisão abrupta levar a uma palavra-código válida (se for verificada a paridade), e tem seu sinal invertido se esta paridade não é verificada. Esta operação faz com que, a cada iteração, os valores das LLR's de saída do decodificador SISO sejam reduzidos ou fortalecidos, dependendo da verificação ou não da paridade. A quantidade de redução ou fortalecimento dependerá do menor valor de LLR dos demais bits envolvidos [30][25]. Esse menor valor pode ser interpretado como sendo a menor confiabilidade obtida no resultado de verificação da paridade.

Uma outra observação a ser mencionada se refere à convergência do processo iterativo na decodificação de códigos produto de paridade simples. Com um número de iterações igual à dimensão do código, praticamente tem-se LLR's de saída já adequadas à decisão abrupta sobre os símbolos transmitidos. A partir daí percebe-se, a cada iteração, um "reforço" nas LLR's de saída do decodificador SISO, numa tendência média de confirmar a decisão que já poderia ser tomada logo após a iteração de número D . Nenhuma melhoria estaria sendo obtida em termos da decisão final a partir desse ponto. Esta observação pode ser constatada no exemplo de decodificação turbo apresentado na Seção VI.

Entretanto, há alguma chance da convergência acontecer antes do final das D iterações. Este fato indica a necessidade de se definir um critério de parada no processo iterativo de tal sorte que seja reduzido o tempo total de decodificação e se possa aumentar a vazão média de dados em um sistema de comunicação real. Em [10] são

abordados alguns desses critérios e em [30, p. 43, item 4.7] é sugerido um método bastante simples de verificação de convergência no processo iterativo e de interrupção desse processo caso a convergência seja verificada. A idéia em [30] é adequada à decodificação iterativa de códigos produto com paridade simples.

Vários algoritmos utilizados para decodificação turbo de códigos de bloco podem ser encontrados na literatura, e também outros algoritmos não específicos para decodificação turbo, mas que podem ser utilizados como base na implementação da decodificação turbo. Esta última opção é possível somente se tais algoritmos operarem com entradas suaves e permitirem que decisões suaves possam ser obtidas em suas saídas. Do rol de algoritmos propostos até o momento, praticamente todos são baseados em variações e/ou derivações dos algoritmos Chase [32], BCJR [3], Kaneko [33] ou SOVA [4]. Dentre os mais citados como referência tem-se o algoritmo de Pyndiah [12] e os algoritmos sub-ótimos classificados como Log-MAP [34].

V - DECODIFICAÇÃO TURBO DE CÓDIGOS PRODUTO DE PARIDADE SIMPLES

OS ALGORITMOS DE DECODIFICAÇÃO turbo ótimos e sub-ótimos, no domínio logarítmico das verossimilhanças, são conhecidos como algoritmos Log-MAP [34]. Em [10] e [22], uma versão sub-ótima de um algoritmo Log-MAP é utilizada em exemplos simples de decodificação turbo de códigos de paridade simples concatenados em paralelo (aqui denominados códigos produto incompletos). Na seção seguinte, o algoritmo Log-MAP apresentado em [15]⁵ é utilizado num exemplo de decodificação turbo de um código produto 2D completo, na decodificação dos códigos componentes de paridade simples. Tal algoritmo foi implementado em Mathcad e a rotina resultante pode ser acessada via *web*, conforme orientações constantes do Apêndice.

Como citado, a razão de verossimilhança logarítmica, LLR, obtida por meio do processo de decodificação SISO (saída suave) pode ser expressa pela equação (12), repetida aqui por conveniência:

$$L(\hat{d}) = L_c(x) + L(d) + L_e(\hat{d}) \quad (13)$$

onde, recordando: $L_c(x)$ representa a LLR obtida a partir da saída do detector; $L(d)$ é a LLR *a priori* do símbolo d e $L_e(\hat{d})$ é a informação extrínseca gerada em função da estrutura do código, pelo processo de decodificação SISO.

Como ilustração desse processo de decodificação, seja um código produto de paridade simples bidimensional. O algoritmo de decodificação opera de acordo com a seqüência mostrada no Algoritmo 2. Essa seqüência pode facilmente ser identificada na rotina mencionada no Apêndice e também no exemplo de decodificação apresentado na Seção VI. Ressalta-se que, para um código de dimensão D , a atualização da informação *a priori* a cada dimensão é feita através da soma das informações extrín-

secas calculadas para as *outras* dimensões até aquele momento [15].

Para realizar os cálculos do processo iterativo de decodificação, há que se utilizar um conjunto de ferramentas apropriadas ao problema. A esse conjunto foi dado o nome de álgebra no domínio logarítmico das verossimilhanças (do Inglês *log-likelihood algebra*) [11][10][22]. Os principais resultados dessa álgebra são resumidamente apresentados em seguida.

Pode-se mostrar que, para variáveis estatisticamente independentes e denotadas por d_j , a razão de verossimilhança logarítmica obtida da operação de soma (módulo 2) entre dois bits é definida por [10][22]⁶

$$L(d_1) \boxplus L(d_2) = L(d_1 \oplus d_2) = \ln \left[\frac{e^{L(d_1)} + e^{L(d_2)}}{1 + e^{L(d_1)} e^{L(d_2)}} \right] \quad (14)$$

$$\approx \text{sign}[L(d_1)] \times \text{sign}[L(d_2)] \times \min[|L(d_1)|, |L(d_2)|]$$

com as regras adicionais:

$$L(d) \boxplus \infty = L(d),$$

$$L(d) \boxplus -\infty = -L(d) \text{ e}$$

$$L(d) \boxplus 0 = 0 \quad (15)$$

A expressão (14) pode ser generalizada, levando a [10]

$$\sum_{j=1}^J \boxplus L(d_j) = \ln \left[\frac{1 + \prod_{j=1}^J \tanh(L(d_j)/2)}{1 - \prod_{j=1}^J \tanh(L(d_j)/2)} \right] \quad (16)$$

$$= 2 \text{arctanh} \left(\prod_{j=1}^J \tanh(L(d_j)/2) \right)$$

Esta expressão também pode sofrer uma aproximação, assim como em (14), levando a

$$\sum_{j=1}^J \boxplus L(d_j) = L \left(\sum_{j=1}^J \oplus d_j \right) \quad (17)$$

$$\approx \left(\prod_{j=1}^J \text{sign}[L(d_j)] \right) \min_{j=1, \dots, J} |L(d_j)|$$

O valor de $L_c(x)$, em canal AWGN, conforme (8), pode ser determinado por

$$L_c(x) = 4 \frac{\sqrt{E}}{N_0} x \quad (18)$$

onde, recordando, $N_0/2$ é densidade espectral de potência bilateral do ruído gaussiano presente nas amostras da saída do filtro casado do receptor no instante de decisão, \sqrt{E} é o valor médio dessas amostras e x é o valor real da amostra obtida.

⁵ Embora o algoritmo de [15] seja ótimo do ponto de vista da decodificação dos códigos componentes do código produto, a decodificação turbo do código produto resultante é sub-ótima. Entretanto, para valores elevados de relação sinal-ruído, o desempenho tende a ser muito próximo da decodificação ótima [15, p. 65].

⁶ A expressão (14) apresenta ligeiras diferenças nas publicações [10] e [22]. Se o mapeamento dos bits nos símbolos codificados for $\{0, 1\} \rightarrow \{-1, +1\}$ (elemento nulo = 0), deve ser utilizado um fator multiplicador $(-1)^{k-1}$ na expressão (14), sendo k o número de bits de informação do código componente ($k_1 = k_2 = k$). Se o mapeamento for $\{0, 1\} \rightarrow \{+1, -1\}$ (elemento nulo = 1), não é necessário o fator multiplicador.

Algoritmo 2 - Etapas do processo de decodificação turbo de códigos produto.

1. Faça todas as verossimilhanças logarítmicas (LLR's) *a priori* iguais a zero (probabilidades *a priori* iguais a 0,5), $L(d) = 0$;
2. conhecida a lógica de codificação, que estabelece a relação de dependência entre os bits de informação e a paridade, obtenha as informações extrínsecas na dimensão horizontal, $L_{eh}(\hat{d})$;
3. faça as novas LLR's *a priori* iguais às informações extrínsecas calculadas no passo anterior, $L(d) = L_{eh}(\hat{d})$;
4. obtenha as informações extrínsecas na dimensão horizontal, $L_{ev}(\hat{d})$, levando em conta as novas LLR's *a priori* atualizadas no passo 3;
5. faça as LLR's *a priori* iguais às informações extrínsecas calculadas no passo anterior, $L(d) = L_{ev}(\hat{d})$;
6. havendo mais iterações vá ao passo 2, levando em conta os últimos valores de LLR's *a priori* atualizados. Não havendo mais iterações, vá ao passo 7;
7. a decisão suave é $L(\hat{d}) = L_c(x) + L_{eh}(d) + L_{ev}(d)$.

O agrupamento dos bits codificados de uma palavra-código de um código produto em um arranjo 2D, de forma que cada dimensão contenha um conjunto de palavras-código do código componente, é tarefa bastante simples. Tal arranjo permite facilmente a identificação da posição dos bits que formam cada palavra-código. Para um código 3D essa tarefa ainda é simples, pois mais uma vez uma construção geométrica pode ser associada à formação do arranjo final de bits codificados e, dessa forma, pode auxiliar na identificação das posições dos bits codificados que formam palavras-código dos códigos componentes. Entretanto, para $D > 3$ essa identificação não mais pode ser auxiliada por uma figura geométrica. De forma a solucionar esse problema, considere como exemplo um código $(3,2,2)^3$. Se os índices dos $n^D = 27$ bits codificados forem dispostos em um arranjo com $D = 3$ linhas e $n^D = 27$ colunas, seguindo a regra apresentada no Algoritmo 3, ter-se-á o arranjo mostrado na Figura 7. Todos os conjuntos de $n = 3$ bits consecutivos agrupados conforme os índices de cada linha desse arranjo são palavras-código dos códigos componentes. Por exemplo, tomando como referência a primeira linha da matriz mostrada na Figura 7, os bits de índice 0, 1 e 2 formam uma palavra-código do código $(3,2,2)$, assim como os bits de índice 3, 4 e 5, e assim por diante. Tomando agora como referência a segunda linha dessa matriz, os bits de índice 0, 3 e 6 formam uma palavra-código do código $(3,2,2)$ assim como os bits de índice 9, 12 e 15, e assim por diante. Por fim, tomando como referência a terceira linha da matriz mostrada na Figura 7, os bits de índice 0, 9 e 18 formam uma palavra-código do código componente, bem como os bits de índice 1, 10 e 19, e assim sucessivamente.

A regra geral de formação do arranjo a ser utilizado na decodificação de códigos produto para D, k e n quaisquer também é descrita através do Algoritmo 3, admitindo-se

apenas que os códigos componentes em todas as dimensões possuem os mesmos valores de n e k . Esta última consideração, apesar de aparentemente restringir as possibilidades de escolha da taxa do código e de seu comprimento, é justificada admitindo-se que é consideravelmente mais simples implementar códigos componentes idênticos. Ainda assim tem-se grande flexibilidade na adequação de taxa e de comprimento do código, o que, caso necessário, pode sofrer ajustes adicionais através de adequadas técnicas de punçãoamento (do Inglês, *puncturing*).

Algoritmo 3 - Geração de arranjos similares ao mostrado na Figura 7.

1. Crie um vetor \mathbf{v} com n^D elementos tal que $v_j = j, j = 0, 1, \dots, n^D - 1$;
2. faça i variar de 0 a $D - 1$;
3. para cada valor de i , faça w variar de 0 a $n^D - 2$;
4. calcule os elementos matriz \mathbf{A} de ordem $D \times n^D$ através de: $A_{i,w} = v_{(i^D + w) \bmod (n^D - 1)}$;
5. os elementos restantes valem: $A_{i,n^D - 1} = n^D - 1$;
6. a matriz \mathbf{A} é o arranjo que se deseja.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23	26
0	9	18	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17	26

Figura 7 - Arranjo de bits codificados utilizado no processo de decodificação de um código produto $(3,2,2)^3$.

VI - EXEMPLO DE DECODIFICAÇÃO TURBO DE UM CÓDIGO PRODUTO SPC 2D

OBJETIVANDO PERMITIR a fixação dos conceitos sobre decodificação turbo abordados até aqui, nesta seção apresenta-se um exemplo de decodificação iterativa de um código produto bidimensional (2D) formado com componentes de paridade simples $(4,3,2)$. O código resultante é o código produto $(n, k, d_{\min})^D = (4,3,2)^2$. Este exemplo busca complementar, de certa forma, aqueles apresentados em [10] e [22]. Nestas referências foi utilizada uma estrutura correspondente à concatenação paralela de códigos componentes de paridade simples $(3,2,2)$, formando o que neste texto foi denominado de código produto incompleto. Aqui foi utilizada a concatenação serial dos códigos componentes $(4,3,2)$, formando um código produto $(4,3,2)^2$ completo.

Em [10] e [22], somente aos bits de informação foram associadas a informação extrínseca e a informação a priori. Apenas a informação de estado de canal foi levada em conta para todos os bits de uma palavra-código. Neste texto, todos estes valores foram operados para todos os bits da palavra-código, assim como em [15], o que eleva significativamente o desempenho do código turbo.

Quando utilizada a álgebra no domínio logarítmico das verossimilhanças adotou-se aqui, assim como em [10], [15] e [22], a forma simplificada dada pela expressão (17), em vez da forma exata obtida através de (16).

Deve-se ressaltar, entretanto, que o uso de (16) leva a resultados significativamente melhores, em termos de taxa de erro de bit final, em comparação com aqueles obtidos com o uso de (17). Esta constatação foi verificada com o auxílio de simulação e pode ser comprovada utilizando-se a rotina comentada no Apêndice. A modulação utilizada foi a BPSK, simulada através de sua equivalente antipodal em banda-base.

Admita que uma palavra correspondente a $k^D = 3^2 = 9$ bits de informação [0 0 1 0 1 0 1 0 0] seja organizada em um arranjo bidimensional. Depois de calculadas as paridades nos sentidos horizontal e vertical desse arranjo obtém-se a palavra-código do código produto $(4,3,2)^2$, conforme ilustrado pela Figura 8(a). Nesta figura e em outras similares que seguem, a parte mais clara corresponde ao arranjo dos bits de informação e a parte mais escura corresponde às paridades. Os índices dos bits associados a esses arranjos estão mostrados na Figura 9.

A seqüência de transmissão dos bits de uma palavra-código foi adotada como sendo: $b_0, b_3, b_6, b_{v0}, b_1, b_4, b_7, b_{v1}, b_2, b_5, b_8, b_{v2}, b_{h0}, b_{h1}, b_{h2}$ e b_{vh} , embora possa ser diferente.

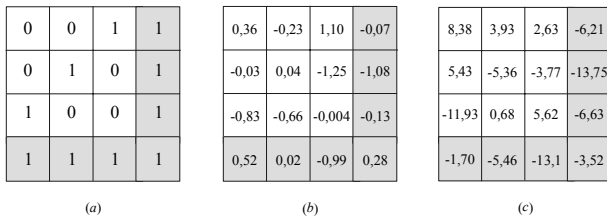


Figura 8 - Arranjos referentes ao código $(4,3,2)^2$: (a) uma palavra-código; (b) amostras de ruído na saída do detector ótimo e (c) razão de verossimilhança logarítmica (LLR) de canal para todos os bits codificados.

Na simulação realizada para compor o exemplo desta seção foi utilizado um valor de E_b/N_0 igual a 4 dB. A energia média por bit de informação, E_b , foi feita unitária e a variância das amostras do ruído na saída do correlador ou do filtro casado, aqui discriminadas por η , foi calculada através de: $N_0/2 = E_b/(2 \times E_b/N_0)$. Para o caso, $N_0/2 = 1/(2 \times 10^{4/10}) \cong 0,199$ watts. Tais amostras podem ser visualizadas na Figura 8(b).

Os símbolos transmitidos têm valor $\in \{\pm\sqrt{E}\}$, onde E é a energia média por bit codificado (que é igual à energia média por símbolo transmitido), com valor $E = r \times E_b = 9/16 \cong 0,563$ Joules. O mapeamento utilizado na sinalização antipodal foi: bit 0 $\Rightarrow +\sqrt{E}$ e bit 1 $\Rightarrow -\sqrt{E}$.

Na Figura 8(c) têm-se os valores das LLR's obtidas a partir da saída do detector, $L_c(x)$, calculados de acordo com (18). Estes valores servirão como entrada suave do decodificador turbo na etapa de inicialização, conforme (10), posto que nesse momento faz-se as LLR's *a priori* iguais a 0. Como exemplo, o valor 8,38 na Figura 8(c) (correspondente ao bit b_0) foi obtido através das operações:

$$4 \frac{\sqrt{E}}{N_0} x = 4 \frac{\sqrt{E}}{2\sigma^2} (\sqrt{E} + \eta) = 4 \frac{\sqrt{0,563}}{2 \times 0,199} (\sqrt{0,563} + 0,36) \cong 8,38.$$

Um outro exemplo pode ser dado para permitir um entendimento melhor sobre o processo: o valor $-3,52$ na

Figura 8(c), correspondente ao bit de paridade b_{vh} , foi obtido através das operações:

$$4 \frac{\sqrt{E}}{N_0} x = 4 \frac{\sqrt{E}}{2\sigma^2} (-\sqrt{E} + \eta) = 4 \frac{\sqrt{0,563}}{2 \times 0,199} (-\sqrt{0,563} + 0,28) \cong -3,52.$$

Os resultados de cálculo da informação extrínseca, obtidos da decodificação da primeira dimensão (dimensão horizontal) do código produto, na primeira iteração, são mostrados na Figura 10(a). Estes resultados foram obtidos operando-se os valores de acordo com as dependências existentes entre os bits codificados, estas governadas pela regra de codificação de paridade simples, e utilizando-se as ferramentas de álgebra das verossimilhanças apresentadas na seção anterior.

b_0	b_1	b_2	b_{h0}
b_3	b_4	b_5	b_{h1}
b_6	b_7	b_8	b_{h2}
b_{v0}	b_{v1}	b_{v2}	b_{vh}

Figura 9 - Possível distribuição dos bits numa palavra-código de um código produto $(4,3,2)^2$.

De acordo com (12), a saída suave para um determinado bit é formada pela soma da LLR obtida a partir da saída do detector, $L_c(x)$, da informação *a priori* do correspondente bit, $L(d)$ e da informação extrínseca $L_e(\hat{d})$. Esta informação extrínseca pode ser calculada através de (16) ou (17), conforme se queira um cálculo exato ou aproximado, respectivamente. Por exemplo, na decodificação na dimensão vertical, a informação extrínseca para o bit b_0 pode ser obtida operando-se as somas das LLR's $L_c(x) + L(d)$ dos bits b_3, b_6 e b_{v0} , aqui denominadas Lb_3, Lb_6 e Lb_{v0} , respectivamente. Como ainda não há estimativa de $L(d)$ para tais bits nessa etapa, admite-se seu valor igual a zero. Então, usando a aproximação dada em (17), a informação extrínseca para o bit b_0 na dimensão vertical é calculada como segue:

$$\begin{aligned} L_{ev}(b_0) &= [\text{sign}(Lb_3)] \times \text{sign}(Lb_6) \times \text{sign}(Lb_{v0}) \min[|Lb_3|, |Lb_6|, |Lb_{v0}|] \\ &= [\text{sign}(5,43)] \times \text{sign}(-11,93) \times \text{sign}(-1,70) \\ &\quad \times \min[|5,43|, |11,93|, |1,70|] \\ &= 1,70 \end{aligned}$$

Usando um procedimento análogo, a informação extrínseca para o bit b_3 na dimensão vertical é calculada da seguinte forma:

$$\begin{aligned} L_{ev}(b_3) &= [\text{sign}(Lb_0)] \times \text{sign}(Lb_6) \times \text{sign}(Lb_{v0}) \min[|Lb_0|, |Lb_6|, |Lb_{v0}|] \\ &= [\text{sign}(8,38)] \times \text{sign}(-11,93) \times \text{sign}(-1,70) \\ &\quad \times \min[|8,38|, |11,93|, |1,70|] \\ &= 1,70 \end{aligned}$$

Para o bit b_6 tem-se:

$$\begin{aligned} L_{ev}(b_6) &= [\text{sign}(Lb_0)] \times \text{sign}(Lb_3) \times \text{sign}(Lb_{v0}) \min[|Lb_0|, |Lb_3|, |Lb_{v0}|] \\ &= [\text{sign}(8,38)] \times \text{sign}(5,43) \times \text{sign}(-1,70) \\ &\quad \times \min[|8,38|, |5,43|, |1,70|] \\ &= -1,70 \end{aligned}$$

E para o bit b_{v0} tem-se:

$$L_{ev}(b_{v0}) = [\text{sign}(Lb_0) \times \text{sign}(Lb_3) \times \text{sign}(Lb_6)] \min[|Lb_0|, |Lb_3|, |Lb_6|] \\ = [\text{sign}(8,38) \times \text{sign}(5,43) \times \text{sign}(-11,93)] \\ \times \min[|8,38|, |5,43|, |-11,93|] \\ = -5,43$$

Segue-se o mesmo procedimento para os demais bits na dimensão vertical, o que permite obter todos valores mostrados na Figura 10(a). Observa-se que, se fosse tomada uma decisão abrupta a partir da informação extrínseca calculada na dimensão vertical, haveria erros nos bits b_2 e b_5 .

Agora todos os valores de informação extrínseca fornecidos na Figura 10(a) servirão como LLR's *a priori* para a decodificação na dimensão horizontal. Então, utilizando novamente a aproximação dada em (17), a informação extrínseca para o bit b_0 na dimensão horizontal é calculada da seguinte maneira:

$$L_{eh}(b_0) = [\text{sign}(Lb_1) \times \text{sign}(Lb_2)] \times \text{sign}(Lb_{h0}) \min[|Lb_1|, |Lb_2|, |Lb_{h0}|] \\ = [\text{sign}(3,93+0,68) \times \text{sign}(2,63+3,77)] \times \text{sign}(-6,21-3,52) \\ \times \min[|3,93+0,68|, |2,63+3,77|, |-6,21-3,52|] \\ = -4,61$$

Para o bit b_1 tem-se:

$$L_{eh}(b_1) = [\text{sign}(Lb_0) \times \text{sign}(Lb_2)] \times \text{sign}(Lb_{h0}) \min[|Lb_0|, |Lb_2|, |Lb_{h0}|] \\ = [\text{sign}(8,38+1,70) \times \text{sign}(2,63+3,77)] \times \text{sign}(-6,21-3,52) \\ \times \min[|8,38+1,70|, |2,63+3,77|, |-6,21-3,52|] \\ = -6,40$$

1,70	0,68	3,77	-3,52	-4,61	-6,40	-4,61	4,61	5,47	-1,79	1,79	-5,13
1,70	-0,68	-2,63	-3,52	-6,04	6,40	6,04	6,04	1,09	0,36	-0,36	-11,24
-1,70	3,93	2,63	-3,52	-4,61	8,25	4,61	-4,61	-18,24	12,86	12,86	-14,76
-5,43	-0,68	-2,63	-6,21	-6,14	-7,13	-6,14	-6,14	-13,27	-13,27	-21,88	-15,8
(a)				(b)				(c)			

Figura 10 - Resultados da primeira iteração: (a) informação extrínseca obtida da decodificação vertical; (b) informação extrínseca obtida da decodificação horizontal e (c) razão de verossimilhança logarítmica (LLR) total.

Procedendo-se de forma análoga obtêm-se os demais valores dados na Figura 10(b). Observe agora que, se fosse tomada uma decisão abrupta a partir da informação extrínseca calculada na dimensão horizontal, haveria erros nos bits b_0 , b_1 , b_{h0} , b_3 , b_4 e b_{h1} . Entretanto, somando as LLR's do canal (Figura 8(c)) com as informações extrínsecas calculadas nas dimensões vertical e horizontal (Figura 10(a) e Figura 10(b)), de acordo com o Algoritmo 2, passo 7, obtém-se o arranjo mostrado na Figura 10(c). Uma decisão abrupta a partir desse arranjo levaria a erros nos bits b_1 , b_2 , b_4 e b_5 . Observe também que praticamente todos os bits correspondentes às decisões corretas têm suas LLR's totais com magnitudes mais elevadas, o que pode ser interpretado como um aumento na confiabilidade das decisões já ao final da primeira iteração, resultado do processo de decodificação turbo.

Uma iteração está completa depois da decodificação nas dimensões vertical e horizontal. Em seguida inicia-se a segunda iteração com uma nova decodificação na dimensão vertical, utilizando os valores de informação

extrínseca calculados na dimensão horizontal da iteração anterior como LLR's *a priori*. Por exemplo, na segunda iteração a informação extrínseca para o bit b_0 na dimensão vertical é calculada através das seguintes operações:

$$L_{ev}(b_0) = [\text{sign}(Lb_3) \times \text{sign}(Lb_6)] \times \text{sign}(Lb_{v0}) \min[|Lb_3|, |Lb_6|, |Lb_{v0}|] \\ = [\text{sign}(5,43-6,04) \times \text{sign}(-11,93-4,61)] \times \text{sign}(-1,70-6,14) \\ \times \min[|5,43-6,04|, |-11,93-4,61|, |-1,70-6,14|] \\ = -0,61$$

Para o bit b_3 tem-se:

$$L_{ev}(b_3) = [\text{sign}(Lb_0) \times \text{sign}(Lb_6)] \times \text{sign}(Lb_{v0}) \min[|Lb_0|, |Lb_6|, |Lb_{v0}|] \\ = [\text{sign}(8,38-4,61) \times \text{sign}(-11,93-4,61)] \times \text{sign}(-1,70-6,14) \\ \times \min[|8,38-4,61|, |-11,93-4,61|, |-1,70-6,14|] \\ = 3,77$$

Procedendo-se de forma similar obtêm-se os demais valores apresentados na Figura 11(a). Se fosse tomada uma decisão abrupta a partir da informação extrínseca calculada na dimensão vertical da segunda iteração, haveria erros nos bits b_0 , b_1 , b_4 , b_6 e b_{v0} .

Todos os valores de informação extrínseca mostrados na Figura 11(a) agora servirão como LLR's *a priori* para a decodificação suave da dimensão horizontal na segunda iteração. Nesta etapa a informação extrínseca para o bit b_0 é calculada da seguinte forma:

$$L_{eh}(b_0) = [\text{sign}(Lb_1) \times \text{sign}(Lb_2)] \times \text{sign}(Lb_{h0}) \min[|Lb_1|, |Lb_2|, |Lb_{h0}|] \\ = [\text{sign}(3,93-1,03) \times \text{sign}(2,63-2,27)] \times \text{sign}(-6,21-7,71) \\ \times \min[|3,93-1,03|, |2,63-2,27|, |-6,21-7,71|] \\ = -0,36$$

Para o bit b_1 tem-se:

$$L_{eh}(b_1) = [\text{sign}(Lb_0) \times \text{sign}(Lb_2)] \times \text{sign}(Lb_{h0}) \min[|Lb_0|, |Lb_2|, |Lb_{h0}|] \\ = [\text{sign}(8,38-0,61) \times \text{sign}(2,63-2,27)] \times \text{sign}(-6,21-7,71) \\ \times \min[|8,38-0,61|, |2,63-2,27|, |-6,21-7,71|] \\ = -0,36$$

Operando de forma similar para todos os bits da palavra-código obtêm-se os demais valores fornecidos na Figura 11(b). Tomando-se uma decisão abrupta a partir da informação extrínseca calculada na dimensão horizontal da segunda iteração, erros nos bits b_0 , b_1 , b_{h0} , b_3 , b_4 e b_{h1} ainda permaneceriam. Somando as LLR's do canal (Figura 8(c)) com as informações extrínsecas calculadas nas dimensões vertical e horizontal da segunda iteração (Figura 11(a) e Figura 11(b)) obtêm-se o arranjo mostrado na Figura 11(c). A decisão abrupta tomada a partir desse arranjo *não mais levaria a erros* e o processo iterativo poderia ser interrompido.

-0,61	-1,03	-2,27	-7,71	-0,36	-0,36	-2,90	0,36	7,41	2,54	-2,54	-13,57
3,77	2,47	1,97	-1,61	-1,79	1,79	2,90	1,79	7,41	-1,11	1,11	-13,57
0,61	1,03	1,97	-1,61	-1,71	7,59	1,71	-1,71	-13,03	9,30	9,30	-9,94
0,61	-1,03	-1,97	-1,61	-5,13	-1,09	-1,09	-1,09	-6,22	-7,58	-16,17	-6,22
(a)				(b)				(c)			

Figura 11 - Resultados da segunda iteração: (a) informação extrínseca obtida da decodificação vertical; (b) informação extrínseca obtida da decodificação horizontal e (c) razão de verossimilhança logarítmica (LLR) total.

Observe mais uma vez, agora na Figura 11(c), que praticamente todos os bits têm suas LLR's totais com magnitudes mais elevadas em relação à Figura 10(c), demonstrando mais uma vez um aumento na confiabilidade das decisões, como consequência do processo de decodificação turbo.

Realizando-se seis iterações, os arranjos correspondentes à informação extrínseca obtida na decodificação das dimensões vertical e horizontal, bem como aquele correspondente à LLR total devem ser aqueles mostrados na Figura 12. Observe que as decisões seriam tomadas corretamente a partir de qualquer desses arranjos. Observe também que, embora a decisão não tenha sido afetada da segunda iteração para a sexta iteração, mais uma vez as magnitudes das LLR's totais tiveram seus valores aumentados em relação àqueles apresentados na Figura 11(c). Então, todas as decisões têm, agora, confiabilidades mais elevadas.

4,83	4,76	-8,07	-11,76	5,44	5,44	-8,69	-5,44	18,64	14,13	-14,13	-23,41
7,77	-3,33	7,77	-5,61	4,00	-4,00	8,69	-4,00	17,21	-12,69	12,69	-23,37
-4,83	3,33	7,77	-5,61	-4,00	12,24	4,00	-4,00	-20,77	16,24	17,39	-16,24
-4,83	-3,33	-7,77	-5,61	-8,79	-6,52	-6,52	-6,52	-15,32	-15,32	-27,40	-15,66
(a)				(b)				(c)			

Figura 12 - Resultados da sexta iteração: (a) informação extrínseca obtida da decodificação vertical; (b) informação extrínseca obtida da decodificação horizontal e (c) razão de verossimilhança logarítmica (LLR) total.

VII - COMENTÁRIOS GERAIS SOBRE CÓDIGOS PRODUTO SPC MULTIDIMENSIONAIS

OS CONCEITOS E RESULTADOS abordados neste texto, complementados com aqueles citados em [10], [15] e [22], podem ser estendidos para códigos produto de paridade simples com mais de duas dimensões, incluindo a paridade das paridades no processo de codificação/decodificação e obtendo-se um certo tipo de informação extrínseca sobre essa paridade. Assim, a informação *a priori* e a extrínseca passam a estar associadas a todos os bits, não somente aos bits de informação como em [10] e [22].

Considerando-se que um algoritmo apropriado, com desempenho ótimo ou pouco distante do ótimo esteja sendo utilizado, com o aumento da dimensão do código produto de paridade simples os retornos em termos de desempenho são sempre menores entre cada aumento de dimensão. Este fato pode ser constatado através dos resultados reportados em [15] e de outros apresentados mais adiante neste texto.

Os ganhos de desempenho em função do aumento da dimensionalidade ocorrem principalmente devido ao aumento da distância mínima do código e da aleatoriedade atribuída ao código resultante. Adicionalmente, a cada iteração em um esquema de decodificação turbo de um código produto multidimensional, percebem-se maiores retornos em desempenho em relação a esquemas bidimensionais. Este fato é devido principalmente à redução na correlação entre as probabilidades envolvidas nos cálculos das LLR's a cada iteração. Melhores desempenhos também podem ser obtidos às custas do aumento do tamanho

dos blocos para um código de dimensão D [10]. Esse fato é mais claramente percebido quando os códigos componentes possuem, individualmente, maior capacidade de correção de erros.

Como pôde ser verificado através de simulações realizadas pelo autor, para códigos produto de paridade simples bidimensionais o aumento no tamanho dos blocos não trouxe ganhos significativos em termos de desempenho. Há, inclusive, uma redução de desempenho com o uso de blocos de tamanho muito elevado, o que é justificado pela predominância da influência do número de palavras código de baixo peso nestes casos. Para baixos valores de relação sinal-ruído percebe-se também uma redução no desempenho quando é aumentado o tamanho dos blocos para esses códigos bidimensionais.

Uma outra constatação se refere a uma melhoria (não muito significativa, contudo) no desempenho do processo de correção de erros para códigos produto SPC completos, de dimensão maior que 2, quando é incluído um processo de entrelaçamento temporal aleatório (*random interleaving*) entre as etapas de codificação em cada dimensão⁷ [15]. Para duas dimensões é percebida uma redução nesse desempenho, em vez de uma melhoria. Um resultado adicional se refere à possibilidade de redução do patamar de saturação da taxa de erro de bit. Contudo, o entrelaçamento temporal aleatório, apesar de reduzir o número de palavras código de baixo peso reduz, também, a distância mínima do código [15].

Deve-se lembrar que, diferentemente do que acontece com os códigos de bloco turbo com concatenação paralela, o processo de entrelaçamento temporal é crucial à melhoria do desempenho dos códigos turbo convolucionais e dos códigos de bloco turbo com concatenação serial. Para estes últimos, tanto a profundidade quanto a "aleatoriedade" do processo de entrelaçamento temporal tem relação direta com o desempenho do processo de decodificação turbo [29][35][27]. Para os códigos de bloco com concatenação paralela, a profundidade do entrelaçamento temporal pode se restringir ao comprimento do bloco de mensagem ou, no pior caso, do bloco completo (informação mais paridade), sem melhorias de desempenho significativas a partir daí.

Em casos onde a dimensionalidade do código produto de paridade simples deva ser aumentada em demasia, acarretando em um excessivo aumento no tamanho dos blocos, alternativamente pode-se implementar o código com códigos componentes mais eficientes, por exemplo, códigos de Hamming. Como citado no início deste tutorial, surpreendentes resultados utilizando códigos componentes de Hamming operando com blocos de tamanho não muito elevado (1023,1013) foram reportados em [8].

Como comentário adicional acerca dos possíveis aumentos de desempenho dos códigos produto com o aumento da sua dimensão e/ou com o uso de códigos componentes de melhor capacidade de correção de erros, enfatiza-se que, embora a distância mínima seja um parâmetro relevante no dimensionamento de um esquema de codificação de canal, argumentos citados em [36] contribuem para que se conclua que o principal critério de me-

⁷ O código SPC-PC utilizado em [15] é um código completo, apresentando baixos valores de saturação na taxa de erro de bit (*error floor*) e um decréscimo mais abrupto da taxa de erro de bit (BER) com o aumento da relação E_b/N_0 .

lhoria de desempenho seja atribuído às propriedades de aleatoriedade dos códigos. Vários exemplos são citados em [36] mostrando resultados atrativos com códigos de distância mínima relativamente baixa, mas com distribuições de pesos das palavras-código que se assemelham à distribuição de um código aleatório. Em [37] são antecipados alguns comentários também nesse sentido, porém objetivando determinar qual a distância de Hamming mínima realmente necessária a um determinado desempenho alvo de um esquema de codificação de canal.

Também merecem ser comentadas algumas questões relacionadas ao tempo que o decodificador turbo precisa para fornecer sua melhor estimativa da palavra-código ou palavra de informação. Duas medidas de tempo podem ser aqui definidas: a *latência* e o *atraso*. O atraso é função da taxa de transmissão e do tamanho do bloco codificado, pois de forma independente do tipo de decodificação, há que se esperar pela recepção de todos os símbolos associados a uma palavra-código para que a estimação dessa palavra-código se inicie⁸. Já a latência está associada ao tempo de processamento necessário para que, depois de recebida uma palavra-código completa, o primeiro bit estimado válido de informação esteja disponível na saída do decodificador. Este tempo depende do número de iterações no decodificador turbo e também da complexidade do algoritmo de decodificação. Para os códigos produto, o número de dimensões também tem influência na latência de forma diretamente proporcional. Nesse sentido, um dos grandes desafios impostos ao desenvolvimento de novos algoritmos de decodificação turbo está relacionado com o objetivo de redução da latência de decodificação. Os códigos produto de paridade simples apresentam relativa vantagem nesse aspecto, posto que sua decodificação turbo pode ser significativamente mais simples que aquelas desenvolvidas para outros códigos ou mesmo para códigos produto com outros códigos componentes. Alguns exemplos neste sentido podem ser obtidos em [12] e [14].

VIII - ALGUNS RESULTADOS DE SIMULAÇÃO EM CANAL AWGN E RAYLEIGH PLANO

NESTA SEÇÃO SÃO FORNECIDOS alguns resultados de simulação para avaliação do desempenho de códigos produto de paridade simples com decodificação turbo, SPC-TPC (*Single-Parity Check Turbo Product Code*), em canal AWGN e em canal com desvanecimento plano do tipo Rayleigh. O código utilizado como teste é o código D -dimensional $(8,7,7)^D$, para D igual a 2, 3, 4 e 5. O algoritmo de decodificação dos códigos componentes é o algoritmo MAP símbolo-a-símbolo apresentado neste texto, o mesmo utilizado em [15]. Os resultados de simulação foram obtidos com a rotina comentada no Apêndice, para modulação BPSK com detecção coerente.

A Figura 13 apresenta resultados de simulação para os códigos $(8,7,7)^D$ no canal AWGN. O desempenho da sinalização BPSK sem codificação de canal também é fornecido como referência para comparação. Como espe-

rado, o desempenho do código produto de paridade simples completo é significativamente melhorado com o aumento da dimensão do código, com aumentos de ganho de codificação aproximadamente iguais de um código com dimensão D para outro com dimensão $D + 1$.

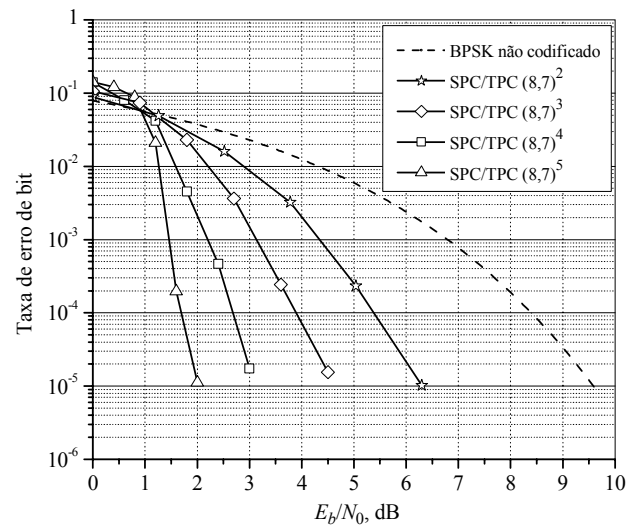


Figura 13 - Resultados de simulação para um código produto de paridade simples $(8,7,7)^D$ com decodificação turbo em canal AWGN e modulação BPSK, para $D = 2, 3, 4$ e 5 .

Observa-se ainda, na Figura 13, um comportamento similar àquele descrito através da Figura 2, no que se refere à taxa de decréscimo da probabilidade de erro de bit com o aumento da relação E_b/N_0 .

Para o código de cinco dimensões tem-se um ganho de codificação de pouco mais de 7 dB @ 10^{-5} de taxa de erro de bit. Para este código, cuja taxa é aproximadamente 0,513, o mínimo valor de E_b/N_0 para comunicação livre de erros, correspondente ao *Limite de Shannon*, é de cerca de 0,19 dB em canal AWGN com modulação BPSK [26]. Dessa forma, esse código apresenta um desempenho que dista cerca de 1,8 dB da capacidade do canal, @ 10^{-5} de taxa de erro de bit, um resultado bastante atrativo, dada a simplicidade de codificação e de decodificação.

A Figura 14 apresenta resultados de simulação para os códigos $(8,7,7)^D$, também para D igual a 2, 3, 4 e 5, em canal com desvanecimento Rayleigh plano. O desempenho da sinalização BPSK sem codificação de canal é fornecido para comparação. Tais resultados foram também obtidos utilizando-se a rotina comentada no Apêndice, com o uso do conhecimento da informação de estado de canal pelo receptor. Foi observado em outras simulações um decréscimo de desempenho de cerca de 1 dB quando a informação de estado de canal não é utilizada no processo de decodificação turbo.

Para o canal com desvanecimentos Rayleigh pode-se perceber, pelos resultados mostrados na Figura 14, que os ganhos de codificação são significativamente mais elevados que aqueles obtidos no canal AWGN. Por exemplo, para o código $(8,7,7)^5$ tem-se um ganho de codificação de cerca de 40 dB @ 10^{-5} de taxa de erro de bit. Para este código, o *Limite de Shannon* é de cerca de 1,6 dB para o canal Rayleigh com modulação BPSK [26]. Então, esse código apresenta um desempenho distante cerca de 2,5 dB do Limite de Shannon, @ 10^{-5} de taxa de erro de bit, um

⁸ Esta afirmação é válida para códigos de bloco e para outros esquemas de codificação de canal que operem com blocos de símbolos codificados, por exemplo, quando se utiliza um código convolucional com terminação na sua treliça de forma que ao final da sequência de informação de comprimento definido o estado do codificador retorne ao estado nulo.

resultado também bastante atrativo para um esquema de codificação de canal de baixa complexidade como o SPC-TPC.

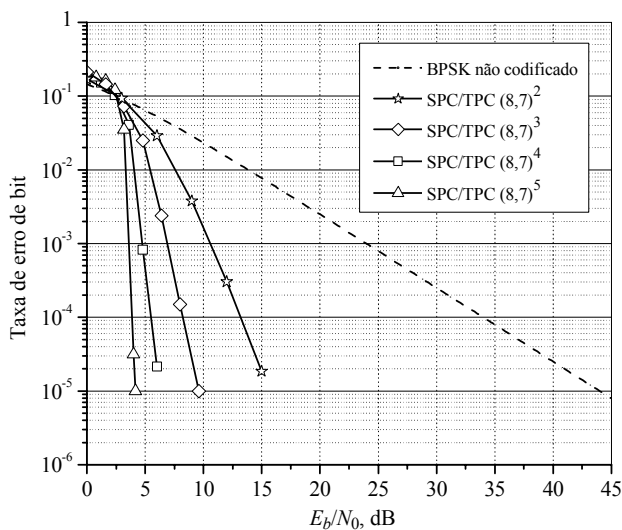


Figura 14 - Resultados de simulação para um código produto de paridade simples $(8,7,7)^D$ com decodificação turbo no canal com desvanecimentos Rayleigh planos e modulação BPSK, para $D = 2, 3, 4$ e 5 .

Tanto na Figura 13 quanto na Figura 14 não se observa o fenômeno de saturação na taxa de erro de bit, o *error floor*. De fato, como antecipado no início deste tutorial, esse fenômeno tende a ocorrer em valores mais baixos de taxa de erro de bit para o tipo de código em questão, em comparação com os códigos turbo convolucionais e com os códigos produto incompletos (concatenação paralela). Por limitações computacionais o *error floor* não foi atingindo para nenhum dos casos simulados. Para valores elevados de relação sinal-ruído, o *error floor* normalmente coincide com o Limitante de União (ou com outro limitante equivalente) para o desempenho do código analisado. Em [15] podem ser verificados exemplos que ratificam esta afirmativa: por exemplo, de acordo com o Limitante de União, o desempenho do código $(8,7,7)^5$ deverá a apresentar um modesto comportamento de saturação da taxa de erro de bit abaixo de 10^{-10} . A partir deste valor o desempenho deverá seguir o que prevê esse limitante, numa taxa de decréscimo mais suave da BER em função da relação E_b/N_0 [15, p. 65].

A empresa *Comtech AHA Corporation* fabrica soluções baseadas em códigos produto e códigos LDPC, com vazões da ordem de 30 a 300 Mbit/s para os Códigos Produto e 30 a 100 Mbit/s para os códigos LDPC. Mais detalhes sobre esses produtos podem ser obtidos em <http://www.comtechaha.com>.

AGRADECIMENTO

O AUTOR AGRADECE imensamente aos revisores pelas críticas construtivas e sugestões que em muito colaboraram para a melhoria da qualidade deste trabalho.

APÊNDICE

POR MEIO DO LINK a seguir tem-se acesso a uma rotina elaborada na plataforma *Mathcad* para análise

do desempenho de códigos produto de paridade simples com decodificação turbo:

http://cict.inatel.br/nova2/docentes/dayan/publications/Inatel_09/

Uma versão em *pdf* dessa rotina é também fornecida por meio desse link, permitindo que ela possa ser “traduzida” para um outro aplicativo, caso necessário. Nessa rotina podem ser escolhidos: os códigos componentes do código produto (que são idênticos em todas as dimensões), o número de dimensões do código resultante, a faixa de valores de E_b/N_0 , o número de pontos no gráfico de E_b/N_0 versus taxa de erro de bit, o número de erros mínimo simulado a cada valor de E_b/N_0 , o número de iterações e o canal de comunicação (AWGN ou Rayleigh plano). Pode-se ainda optar por utilizar ou não utilizar a informação de estado de canal no processo de decodificação turbo.

A modulação utilizada é a BPSK em sua forma equivalente em banda base e com detecção coerente. O desempenho teórico dessa modulação é também traçado no gráfico resultante como forma de comparação, de acordo com o canal escolhido.

Para utilizar a rotina fornecida por meio do link supramencionado, deve-se executar diretamente o arquivo com extensão *mcd* no *Mathcad*, versão 2001i ou superior. Para versões inferiores à 2001i, basta realizar uma cópia *ipsis litteris* do conteúdo do arquivo com extensão *pdf* na área de trabalho do *Mathcad*. Pode-se também traduzir o conteúdo do arquivo com extensão *pdf* para um outro aplicativo, por exemplo, para o *Matlab*. Este processo de tradução é bastante simples, dado o aspecto “amigável” da interface com o usuário do *Mathcad*.

Na rotina em questão percebe-se que o cálculo da informação extrínseca está sendo realizado de duas formas: um cálculo exato conforme a expressão (16), quando este cálculo não apresenta erro computacional, e um cálculo aproximado de acordo com a expressão (17), quando esse erro ocorre. Tal erro tem maior chance de ocorrência quando os valores das LLR’s operadas são muito elevados, por conta do cálculo numérico efetuado pelo *Mathcad* envolvendo a tangente hiperbólica presente em (16). Essa possibilidade de erro computacional também foi constatada por D. M. Rankin em [15] e a alternativa lá utilizada para evitar tais erros foi o truncamento nos valores das LLR’s operadas. Ambas as alternativas foram testadas para a composição deste trabalho e levaram aos mesmos resultados.

REFERÊNCIAS

- [1] FORNEY Jr, G. D. *Concatenated Codes*, Ph.D. Thesis, Cambridge: Massachusetts Institute of Technology, 1966.
- [2] BERROU, Claude, A. GLAVIEUX and P. THITIMAJSHIMA. Near Shannon limit error-correcting coding and decoding: Turbo-Codes. *Proceedings of the 1993 Communication Conference, ICC'93*, Geneva, Switzerland, p. 1064-1070, May 1993.
- [3] BAHL, L. R.; COCKE, J.; JELINEK, F.; RAVIV, J. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Trans. Information Theory*, p. 284-287. Mar., 1974.

- [4] HAGENAUER, J.; HOEHER, P. A Viterbi Algorithm With Soft-Decision Outputs and its Applications. *Proceedings of Globecom '89*, Dallas, Texas, p. 47.11-47.17, Nov., 1989.
- [5] GALLAGER, R. G. *Low-Density Parity-Check Codes*. Cambridge, MIT Press, 1963.
- [6] LODGE, J.; YOUNG, R.; HOEHER, P.; HAGENAUER, J. Separable MAP 'filters' for the decoding of product and concatenated codes. *Proceedings of ICC'93*, Geneva, p. 1740-1745, May, 1993.
- [7] BERROU, Claude, A. GLAVIEUX. Reflections on the Prize paper: "Near optimum error correcting coding and decoding: Turbo codes", *IEEE IT Society Newsletter*, v. 48, n. 2, Jun., 1998.
- [8] NICKL, H.; HAGENAUER, J.; BURKERT, F. Approaching Shannon's Capacity Limit by 0.2 dB Using Simple Hamming Codes. *IEEE Communications Letters*, v. 1 5, p. 130-132, Sep., 1997.
- [9] PROAKIS, J. G. *Digital Communications*. 3rd Ed. New York: McGraw Hill., 1995.
- [10] HAGENAUER, J.; OFFER, E.; PAPKE, L. Iterative Decoding of Binary Block and Convolutional Codes. *IEEE Trans. Information Theory*, p. 429-445, v. 42, n. 2, Mar., 1996.
- [11] BERROU, C.; GLAVIEUX, A. Near Optimum Error Correcting Coding And Decoding: Turbo-Codes. *IEEE Trans. Communications*, p. 1261-1271, v.. 44, n.10. Oct., 1996.
- [12] PYNDIAH, R. M. Near-Optimum Decoding of Product Codes: Block Turbo Codes. *IEEE Trans. Communication*, p. 1003-1010, v.. 46, n. 8. Aug., 1998.
- [13] HUNT, A.; CROZIER, S.; FALCONER, D.. Hyper-codes: High-performance Low-Complexity Error-Correcting Codes. *Proceedings of 19th Biennial Symposium on Communications*, p. 263-267, Kingston, Canada, Jun., 1998.
- [14] DAVE, S.; KIM, J.; KWATRA, S. C.. An Efficient Decoding Algorithm for Block Turbo Codes. *IEEE Trans. Communications*, p. 41-46, v. 49, n. 1, Jan., 2001.
- [15] RANKIN, D. M., *Single Parity Check Product Codes and Iterative Decoding*. Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, May/2001. *Veja também*: RANKIN, D. M.; GULLIVER, T. A. Single Parity Check Product Codes. *IEEE Trans. Communication*, v. 49, n. 8, p. 1354-1362, Aug., 2001.
- [16] GUIMARÃES, D. A.; PORTUGHEIS, J. A Class of Product Codes and Its Iterative (Turbo) decoding. *Proceedings of the 3rd International Symposium on Turbo Codes & Related Topics*, p. 431-434, Brest, France, Sep., 1-5, 2003.
- [17] BARBULESCU, S. A.; PIETROBON, S. S. *Turbo Codes: a Tutorial on a New Class of Powerful Error Correcting Coding Schemes. Part I: Code Structures and Interleaver Design*. Institute for Telecommunications Research, University of South Australia, Oct. 1998.
- [18] SHANNON, C. E. A Mathematical Theory of Communication. *Bell System Technical J.*, v.. 27, Jul-Oct., 1948.
- [19] HAYKIN, S.; SELLATHURAI, M.. Turbo-BLAST with Multi-loop Feedback Receiver. *Proceedings of the 3rd International Symposium on Turbo Codes & Related Topics*, p. 195-202: Brest, France, Sep. 1-5, 2003.
- [20] ELIAS, P. Error-free Coding. *IRE Trans. Information Theory*, v. PGIT-4, p. 29-37, Sep., 1954.
- [21] HONARY, B. *Trellis Decoding of Block Codes*. Norwell: Kluwer Academic, 1997.
- [22] SKLAR, B. A Primer on Turbo Code Concepts. *IEEE Communication Magazine*, p. 94-101, Dec., 1997.
- [23] GALLAGER, R. G. Low density parity check codes. *IRE Trans. Information Theory*, v. IT-8, p. 21-28, Jan., 1962. *Veja também*: GALLAGER, R. G. Low Density Parity Check Codes. *Research monograph series*, n. 21, MIT Press, Cambridge, 1963.
- [24] MACKAY, D. J. C.; NEAL, R. M. Near Shannon limit performance of low density parity check codes. *IEE Electronics Letters*, v.. 32, n. 18, p. 1645-1655, Aug., 29th, 1996.
- [25] GUIMARÃES, D. A. Decodificação Turbo de Códigos Produto de Paridade Simples. *Telecomunicações*, v. 5, n. 1, p. 11-28, Jun., 2002.
- [26] GUIMARÃES, D. A. *Uma Classe de Códigos Produto e sua Decodificação Turbo Aplicada em um Sistema CDMA Multiportadora*. Tese de Doutorado: Universidade Estadual de Campinas - Unicamp. Campinas, SP, June 2003.
- [27] BENEDETTO, S.; DIVSALAR, D.; MONTORSI, G.; POLLARA, F. Serial Concatenation of Interleaved Codes: Performance Analysis, Design and Iterative Decoding. *IEEE Trans. Information Theory*, v. 44, n. 3, p. 909-926, May, 1998.
- [28] BENEDETTO, S.; BIGLIERI, E. *Principles of Digital Transmission With Wireless Applications*. New York: Kluwer Academic / Plenum Publishers, 1999.
- [29] BARBULESCU, S. A. *Iterative Decoding of Turbo Codes and Other Concatenated Codes*. Ph.D. Thesis, Faculty of Engineering, University of South Australia, Feb., 1996.
- [30] HUNT, A. *Hyper-codes: High-performance Low-Complexity Error-Correcting Codes*. M. Sc. Thesis, Faculty of Engineering, Ottawa-Carleton Institute of Electrical Engineering, Carleton University, Ottawa, Ontario, Canada, May 1998.
- [31] HAYKIN, S. *Communication Systems*. 4th Ed. New York: John Wiley, 2001.
- [32] CHASE, D. A Class of Algorithms for Decoding Block Codes With Channel Measurement Information. *IEEE Trans. Information Theory*, v. IT-18, n. 1, p. 170-182, Jan., 1972.
- [33] KANEKO, T.; NISHIJIMA, T.; INAZUMI, H.; HIRASAWA, S. An Efficient Maximum-Likelihood-Decoding Algorithm for Linear Block Codes with Algebraic Decoder. *IEEE Trans. Information Theory*, v.40, n. 2, p. 320-327, Mar., 1994.

- [34] ROBERTSON, P.; VILLEBRUN, E.; HOEHER, P. A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain. *Proceedings of the 1995 International Conference on Communications, ICC '95*, Seattle, v. 2, p. 1009-1013, 1995.
- [35] BENEDETTO, S., MONTORSI, G. Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes. *IEEE Trans. Information Theory*, v. 42, n. 2, p. 409-428. Mar., 1996.
- [36] YUE, D. W.; YANG, E. H. Asymptotically Gaussian Weight Distribution and Performance of Multicomponent Turbo Block Codes and Product Codes. *IEEE Trans. Communications*, v. 52, n.5, p. 728-736, May, 2004.
- [37] BERROU, C.; MAURY, E.; GONZALEZ, H.. Which Minimum Hamming Distance Do We Really Need? *Proceedings of the 3rd International Symposium on Turbo Codes & Related Topics*, p. 141-148: Brest, France, Sep., 1-5, 2003.

SOBRE O AUTOR

Dayan Adionel Guimarães nasceu em Carrancas, MG, em 01 de março de 1969. Possui os títulos: *Técnico em Eletrônica* (ETE "FMC", 1987), *Engenheiro Eletricista* (Inatel, 1994), *Especialista em Engenharia de Comunicação de Dados* (Inatel, 2003), *Especialista em Administração* com ênfase em Gerência de RH (FAI, 1996), *Mestre em Engenharia Elétrica* (Unicamp, 1998) e *Doutor em Engenharia Elétrica* (Unicamp, 2003). De 1988 a 1993 desenvolveu sensores e equipamentos para instrumentação industrial e controle e também foi Supervisor de Produção e Supervisor de Engenharia de Produtos na SENSE Sensores e Instrumentos. Desde de janeiro de 1995 é Professor do Inatel onde, por oito anos, foi responsável pela estrutura que dá apoio às atividades de ensino prático nas áreas de Telecomunicações, Eletrônica e Eletrotécnica. É membro do corpo editorial da revista *Telecomunicações* do Inatel. Suas pesquisas incluem aspectos gerais sobre transmissão digital e sistemas de comunicação móvel, especificamente sistemas CDMA Multiportadora e esquemas de codificação para canais com desvanecimento, especificamente códigos turbo de bloco.

Introdução à Equalização Turbo♦

André Fonseca dos Santos e Dayan Adionel Guimarães

Departamento de Telecomunicações – Inatel - andref@inatel.br, dayan@inatel.br

Resumo – Sabe-se que a interferência intersimbólica (IIS) é uma das causas principais de degradação no desempenho de sistemas de comunicação digital que operam em canais dispersivos no tempo. Sabe-se também que a equalização é a forma usual para combater os efeitos dessa IIS. Neste contexto, este tutorial aborda os principais conceitos associados ao processo conjunto de equalização e decodificação iterativa, denominado Equalização Turbo. Algumas configurações utilizadas na implementação desse processo são apresentadas, tecendo-se conclusões sobre suas vantagens e desvantagens em termos de complexidade e capacidade de redução da IIS.

Palavras chave – equalização turbo, decodificação iterativa (turbo), interferência intersimbólica.

Abstract – It is well known that inter-symbol interference (ISI) is one of the main reasons for performance degradation in digital communication systems operating in time-dispersive channels. It is also known that equalization is the usual method for combating the effects of ISI. In this context, this tutorial aims at presenting the main concepts associated to the joint process of iterative decoding and iterative equalization, called Turbo Equalization. It presents some configurations for a turbo equalization process and derives some conclusions concerning the trade-off between complexity and ISI reduction capability for these configurations.

Key words – turbo equalization, turbo (iterative) decoding, inter-symbol interference.

I. INTRODUÇÃO

Devido à crescente demanda de serviços em sistemas de comunicação móvel terrestres, tem-se investido grande esforço na busca de sistemas de transmissão de alta capacidade que operam adequadamente em canais com múltiplos percursos de propagação.

Nos sistemas de comunicação móvel, tipicamente utilizam-se códigos corretores de erro para minimizar a influência do ruído AWGN (*Additive White Gaussian Noise*) e equalizadores para reduzir o efeito da interferência intersimbólica (IIS) causada pela dispersão temporal do canal. Essa dispersão pode ser causada, por exemplo, quando o canal é limitado em largura de faixa ou quando há propagação por múltiplos percursos [1].

No entanto, os processos de equalização e de decodificação são realizados normalmente de forma independente. O propósito da equalização turbo é realizar a equalização

e a decodificação de maneira conjunta e iterativa. Embora em [1] já se considere a equalização e a decodificação de forma conjunta, na equalização turbo o processo é iterativo.

O primeiro esquema de equalização turbo foi proposto em [2]. Nele, os símbolos transmitidos são codificados através de um codificador convolucional, que é tratado como um codificador externo, e o conjunto formado pelo modulador, pelo canal de comunicação e pelo demodulador é modelado como um codificador interno [2]. Na recepção, o processo de equalização turbo realiza sucessivas trocas de informação entre o decodificador interno (equalizador) e o decodificador externo (decodificador convolucional), assim como ocorre na decodificação turbo entre os decodificadores componentes [3].

Os resultados obtidos com esse sistema mostram que a equalização turbo pode eliminar praticamente toda a influência da IIS do canal, sem a necessidade de aumento da relação sinal-ruído (RSR). Seu desempenho chega a se assemelhar ao desempenho de um sistema com codificação de canal em um canal AWGN e sem IIS [4].

Este tutorial tem como principal objetivo a apresentação dos conceitos envolvidos na equalização turbo. A Seção II apresenta o modelo discreto (no tempo) do conjunto formado pelo modulador, pelo canal com múltiplos percursos e pelo demodulador. A partir deste modelo discreto pode-se observar que um símbolo transmitido pode se sobrepor aos símbolos vizinhos, resultando em IIS. Nota-se também que este modelo discreto se comporta de forma semelhante a um codificador convolucional [2], o que permite sua caracterização como um codificador interno do sistema. Para fins de simplificação, no texto deste tutorial este modelo constituído por modulador, canal e demodulador será citado apenas como *modelo discreto do canal*. Na Seção III é apresentado o princípio de funcionamento de um equalizador turbo, no qual as iterações são apresentadas como uma cascata de módulos [2] [5] [6]. A Seção IV trata do algoritmo MAP (*Maximum a Posteriori*) símbolo-a-símbolo e sua implementação nos dois blocos do equalizador turbo: o bloco de equalização e o bloco de decodificação [4]. Na Seção V utiliza-se outra configuração do equalizador turbo, conforme apresentado em [5] e [7], onde é usado um *Cancelador de Interferência* no bloco equalizador e o algoritmo MAP no bloco de decodificação. A Seção VI faz uma referência aos critérios de parada apresentados em [12], os quais determinam o número ótimo de iterações que devem ser realizadas para que o sistema proporcione um desempenho alvo. Finalmente, na Seção VII faz-se uma análise de resultados obtidos por simulação e reportados em publicações recentes sobre o assunto, tecendo-se comentários finais sobre quesitos como complexidade e desempenho.

♦ Este trabalho foi parcialmente financiado pelo convênio nº 22.02.0431.00, celebrado entre o Inatel, a *Linear Equipamentos Eletrônicos S/A* e a *FINEP*.

II. MODELO DISCRETO DO CANAL

Ao longo deste tutorial o sistema de transmissão utilizado como referência é aquele apresentado na Figura 1. Primeiramente, os bits de informação b_k são codificados através de um codificador convolucional em símbolos c_n . Os símbolos c_n são embaralhados por um entrelaçador temporal (*interleaver*), usado para que erros em rajada se apresentem descorrelacionados no receptor, após a passagem pelo canal e pelo bloco de desentrelaçamento temporal (*de-interleaver*) desse receptor. Os símbolos entrelaçados são mapeados por um modulador BPSK (*Binary Phase Shift Keying*) em banda base, resultando nos símbolos c_n^π , que são transmitidos através do canal.

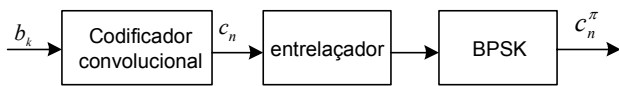


Figura 1 - Sistema de transmissão.

Em um canal onde há propagação por múltiplos percursos, o sinal recebido é resultado da soma de infinitas réplicas (ou ecos) do sinal transmitido [8, p. 532]. Os múltiplos percursos são causados principalmente por difração, reflexão e espalhamento do sinal nos obstáculos que circundam o receptor e/ou que se localizam entre transmissor e receptor, e também por influência da atmosfera. Como é inviável modelar esse canal com infinitos percursos de propagação, principalmente por conta da dificuldade de análise matemática, pode-se admitir como aproximação que ele envolva a soma dos L percursos mais significativos. Assim, o modelo discreto de um canal com múltiplos percursos pode ser representado por uma linha de atrasos com derivações (do Inglês, *tapped delay line*) [1], como descrito pela expressão:

$$r_n = w_n + \sum_{l=0}^L g_l c_{n-l}^\pi = w_n + v_n \tag{1}$$

onde os coeficientes $g_1, g_2, \dots, g_l, \dots, g_L$ representam as variações de intensidade do sinal no l -ésimo percurso de propagação, também podendo ser denominados de ganhos do canal. v_n é a n -ésima amostra de saída do canal com IIS sem a presença de ruído AWGN, w_n é a n -ésima amostra de ruído AWGN adicionado ao sinal e $\mathbf{r} = [r_1 \ r_2 \ \dots \ r_n \ \dots \ r_N]^T$ é o vetor que contém as amostras do sinal recebido. O número de amostras N , no vetor \mathbf{r} , é função do número de amostras operadas em cada bloco no receptor, como pode ser observado mais adiante.

O modelo discreto do canal conforme (1) pode ser representado por uma treliça associada a uma cadeia de Markov, da mesma maneira que o codificador convolucional da Figura 2 [2]. Como exemplo, a treliça da Figura 3 representa a transição de estados de um codificador convolucional como o da Figura 2, ou de um canal com dois múltiplos percursos (canal com IIS).

Se o canal ou o codificador é composto por M atrasadores, as correspondentes treliças serão compostas por 2^M estados. No exemplo em questão $M = 2$ para o canal e para o codificador, o que resulta em uma treliça de quatro estados. Para exemplificar a formação da treliça toma-se a

transição de S_0 para S_1 . O estado S_0 é composto pelos símbolos $\{-1,-1\}$ na saída dos registradores da linha de atraso. Ao transmitir o símbolo $+1$ pelo canal, o conteúdo na saída dos registradores passa a ser $\{+1,-1\}$, que corresponde ao estado S_1 . Da mesma maneira são formadas as demais transições da treliça.

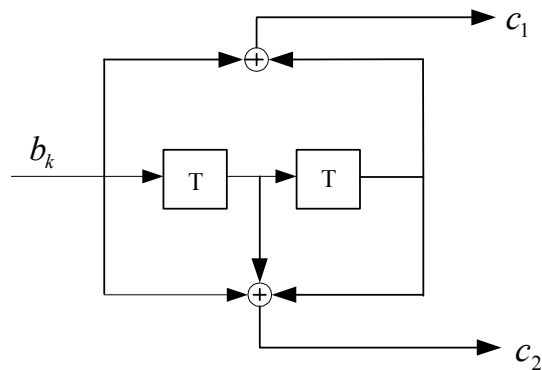


Figura 2 - Codificador convolucional de taxa $\frac{1}{2}$ com polinômios geradores $g_1(D)=1+D^2$ e $g_2(D)=1+D+D^2$.

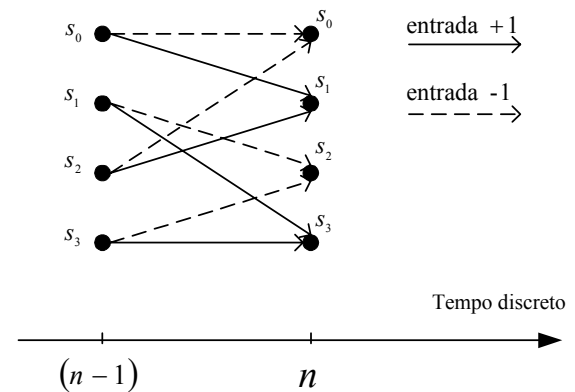


Figura 3 - Treliça que representa as transições de estados para um codificador convolucional de comprimento 3 e de um canal com dois múltiplos percursos, onde $S_0 = \{-1,-1\}$, $S_1 = \{+1,-1\}$, $S_2 = \{-1,+1\}$ e $S_3 = \{+1,+1\}$. Para o codificador o valor -1 é trocado por 0 nos valores dos estados.

Partindo desse princípio de que o canal com IIS pode ser representado como uma treliça da mesma maneira que um codificador convolucional, para o entendimento do equalizador turbo o canal será tratado como um codificador componente do sistema de transmissão, num processo de concatenação serial, o que permitirá que a equalização e a decodificação sejam realizadas de maneira iterativa.

III. PRINCÍPIOS DA EQUALIZAÇÃO TURBO

A Figura 4(a) apresenta o canal como um codificador interno do sistema de transmissão e o codificador convolucional como um codificador externo. Nos sistemas convencionais de recepção a recuperação da informação é feita de acordo com a Figura 4(b). Primeiramente, é realizada a equalização do canal através de, por exemplo, um filtro empregando o algoritmo LMS (*Least Mean Square*) ou um equalizador de treliça empregando o critério MLSE (*Maximum Likelihood Sequence Estimation*) [1]. Em

seguida, os símbolos codificados c_n^π na saída do equalizador são reordenados pelo desentrelaçador temporal que fornece os símbolos c_n para o decodificador convolucional. Finalmente, na saída do decodificador os bits de informação b_k são estimados.

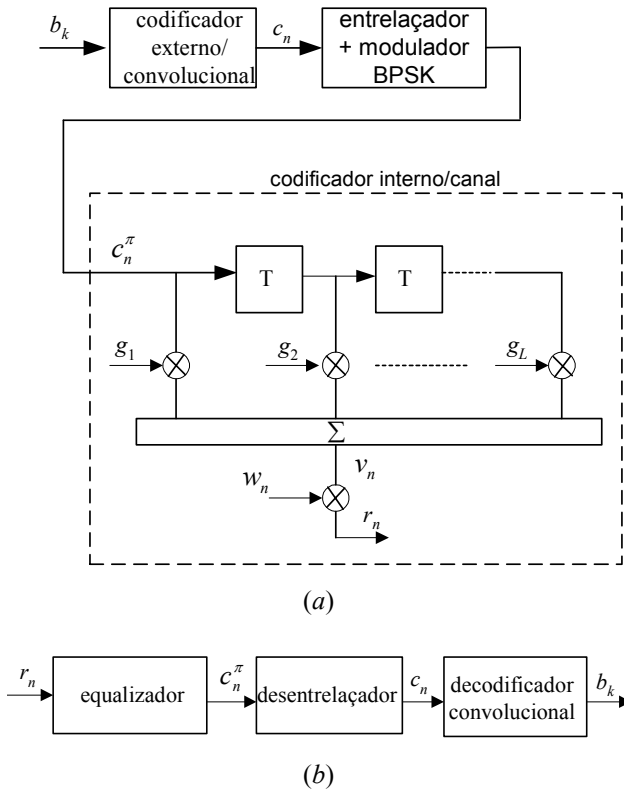


Figura 4 - (a) Sistema de transmissão onde o canal é modelado como codificador interno. (b) Sistema de recepção convencional com decisões tomadas no equalizador e no decodificador separadamente.

A equalização e a decodificação são realizadas separadamente e o equalizador não explora a estrutura do código para realizar seu processamento. Na equalização turbo realiza-se, iterativamente, uma realimentação de informação das decisões do decodificador para o equalizador, de tal maneira que este produza decisões mais confiáveis. De maneira similar, as decisões do equalizador alimentam o decodificador, permitindo que este forneça decisões mais confiáveis. Dessa forma, o processo conjunto de equalização e decodificação turbo atingirá melhor desempenho, iteração a iteração.

Para que as informações trocadas entre o bloco equalizador e o bloco decodificador carreguem dados de confiabilidade nas decisões, elas devem ser suaves. Por esta razão o equalizador turbo é constituído de dois blocos de entradas e saídas suaves (do Inglês, *Soft-Input Soft-Output*, SISO). O primeiro é o bloco equalizador apresentado na Figura 5. Este bloco é um equalizador que tem sua saída na forma de uma razão de verossimilhança logarítmica (do Inglês, *Log-Likelihood Ratio*, LLR) das probabilidades *a posteriori* dos símbolos transmitidos. Tal LLR vale:

$$L^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1 | \mathbf{r})}{P(c_n^\pi = -1 | \mathbf{r})} \quad (2)$$

onde c_n^π são os símbolos da modulação BPSK em banda-base e \mathbf{r} é o vetor de amostras recebidas do canal com IIS. A magnitude de $L^E(c_n^\pi)$ contém a informação da confiabilidade da decisão tomada pelo equalizador. Pode-se também, a partir do sinal de $L^E(c_n^\pi)$, tomar uma decisão abrupta sobre os símbolos c_n^π da seguinte maneira:

$$\begin{aligned} \text{Se } L^E(c_n^\pi) \geq 0, & \text{ então } c_n^\pi = +1 \\ \text{Se } L^E(c_n^\pi) < 0, & \text{ então } c_n^\pi = -1 \end{aligned}$$

O critério de decisão supracitado é conhecido como critério MAP (*Maximum a Posteriori*) símbolo-a-símbolo. Este critério minimiza a probabilidade de decisão errada por cada símbolo.

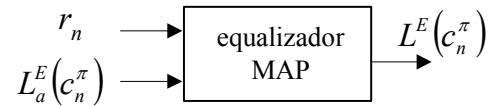


Figura 5 - Equalizador MAP.

Para o cálculo da saída suave $L^E(c_n^\pi)$, o bloco equalizador tem como entrada o vetor de amostras recebidas do canal, com IIS, \mathbf{r} , e a LLR da informação *a priori* dos símbolos c_n^π dada por:

$$L_a^E(c_n^\pi) = \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (3)$$

Aplicando a regra de Bayes em (2), a saída suave do bloco equalizador pode ser decomposta em duas parcelas:

$$\begin{aligned} L^E(c_n^\pi) &= \ln \frac{P(c_n^\pi = +1 | \mathbf{r})}{P(c_n^\pi = -1 | \mathbf{r})} \\ &= \ln \left[\frac{p(\mathbf{r} | c_n^\pi = +1)}{p(\mathbf{r} | c_n^\pi = -1)} \right] + \ln \left[\frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \right] \quad (4) \end{aligned}$$

A primeira parcela em (4) refere-se à *informação extrínseca* na saída do bloco equalizador, que será representada daqui em diante por $L_{ext}^E(c_n^\pi)$. A segunda parcela se refere à LLR das probabilidades *a priori* dos símbolos c_n^π , que será representada deste ponto em diante por $L_a^E(c_n^\pi)$. A informação extrínseca pode ser interpretada como o ganho de informação obtido sobre um símbolo no processo de equalização ou decodificação [2] ou como a LLR na saída de um decodificador SISO menos a LLR da entrada [8].

O outro bloco SISO do equalizador turbo é o decodificador mostrado na Figura 6. Este bloco é constituído de uma entrada que recebe o vetor \mathbf{Z} , que tem como elementos uma estimativa das probabilidades *a priori* dos símbolos c_n :

$$\mathbf{Z} = [P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r}) \dots P(c_N | \mathbf{r})] \quad (5)$$

A partir do vetor de entrada \mathbf{Z} , o bloco decodificador calcula as LLR's dos símbolos c_n e dos bits de informação b_k .



Figura 6 - Bloco decodificador MAP.

Da mesma maneira que no bloco equalizador, a saída $L^D(c_n)$ do bloco decodificador pode ser decomposta nas LLR's da informação extrínseca e da informação *a priori*:

$$L^D(c_n) = \ln \left(\frac{P(c_n = +1 | \mathbf{Z})}{P(c_n = -1 | \mathbf{Z})} \right) = \ln \left(\frac{p(\mathbf{Z} | c_n = +1)}{p(\mathbf{Z} | c_n = -1)} \right) + \ln \left(\frac{P(c_n = +1)}{P(c_n = -1)} \right) \quad (6)$$

onde o primeiro termo de (6) é a informação extrínseca $L_{ext}^D(c_n)$ do símbolo c_n e o segundo termo é a informação *a priori* desse símbolo, $L_a^D(c_n)$.

Na equalização turbo utiliza-se a informação extrínseca $L_{ext}^D(c_n)$ obtida pelo bloco decodificador como uma estimativa da informação *a priori* a ser operada pelo bloco equalizador. De forma análoga, a informação extrínseca obtida pelo bloco equalizador é utilizada como informação *a priori* a ser processada pelo bloco decodificador. Portanto, um equalizador turbo opera como um decodificador turbo, onde um código componente é o codificador convolucional e outro código componente pode interpretado como o canal com múltiplos percursos.

A Figura 7 ilustra o processo de equalização turbo. Através do vetor de amostras recebidas do canal, \mathbf{r} , e da informação *a priori* $L_a^E(c_n^\pi)$ o bloco equalizador calcula sua saída suave $L^E(c_n^\pi)$. De acordo com (4), subtrai-se da saída suave total do bloco equalizador $L^E(c_n^\pi)$ a informação *a priori* $L_a^E(c_n^\pi)$ para que se transmita somente a informação extrínseca $L_{ext}^E(c_n^\pi)$ para o bloco decodificador. A informação extrínseca $L_{ext}^E(c_n^\pi)$ passa pelo desentrelaçador temporal e é utilizada como uma estimativa da informação *a priori* dos símbolos, $L_a^D(c_n)$, para o bloco decodificador. O bloco decodificador calcula sua saída suave $L^D(c_n)$ e subtrai a parcela da informação *a priori* $L_a^D(c_n)$ para o cálculo da informação extrínseca $L_{ext}^D(c_n)$, de acordo com (6). A informação extrínseca $L_{ext}^D(c_n)$ passa pelo entrelaçador temporal e é usada como uma estimativa da informação *a priori* dos símbolos, $L_a^E(c_n^\pi)$, para o bloco equalizador. Note que um dos papéis do entrelaçador e do desentrelaçador no equalizador turbo é de organizar as LLR's *a priori* na entrada do bloco equalizador e do bloco decodificador respectivamente. Outra função dos processos de entrelaçamento/desentrelaçamento é a de descorrelacionar as informações trocadas pelos blocos constituintes, o que será explicado com mais detalhes no decorrer deste tutorial.

O processo descrito no parágrafo anterior se repete tantas vezes quanto necessário, até que se atinja a convergência em termos de estabilização dos valores de infor-

mação extrínseca, o que leva a uma estabilização no desempenho final da equalização turbo em termos de taxa de erro de bit ou de bloco. Pode-se também interromper o processo iterativo utilizando outros critérios, como exemplificado mais adiante, objetivando uma redução no número médio de iterações e, por conseqüência, um aumento na vazão de informação.

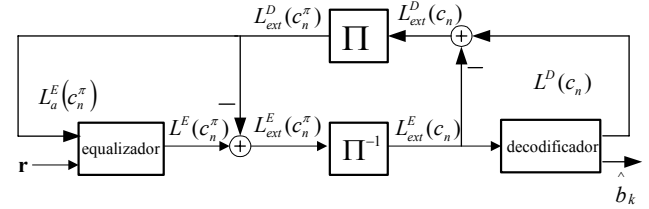


Figura 7 - Equalizador turbo.

A cada repetição (iteração) há um melhoramento da confiabilidade das informações extrínsecas dos blocos equalizador e decodificador, diminuindo a chance de erros nas decisões sobre os bits b_k ou sobre os símbolos c_n [11]. Na primeira iteração, como não há informação extrínseca disponível vinda do decodificador, a LLR *a priori* operada pelo bloco equalizador, $L_a^E(c_n^\pi)$, é feita igual a zero (probabilidades *a priori* admitidas iguais a 1/2) e a cada iteração a estimativa desta LLR é melhorada. Ao final das iterações pode-se realizar uma decisão abrupta sobre os bits de informação b_k ou sobre os símbolos c_n , a partir da saída suave do bloco decodificador.

O uso da informação extrínseca como estimativa da informação *a priori* dos blocos constituintes do processo de equalização turbo deve garantir que ela seja descorrelacionada em relação aos valores de informação extrínseca calculados pelo bloco anterior. Isto é feito para que um bloco não opere com valores que são resultado de decisões suaves geradas pelo próprio bloco constituinte em questão. Por exemplo, de acordo com (4), a saída suave $L^E(c_n^\pi)$ do equalizador é composta pela informação *a priori* $L_a^E(c_n^\pi)$ e pela informação extrínseca $L_{ext}^E(c_n^\pi)$. Portanto, a saída suave total $L^E(c_n^\pi)$ do bloco equalizador não deve ser usada como estimativa da informação *a priori* para o bloco decodificador, já que a parcela $L_a^E(c_n^\pi)$ que compõe a saída $L^E(c_n^\pi)$ é obtida a partir da saída do bloco decodificador. Dessa forma, a saída suave $L^E(c_n^\pi)$ do bloco equalizador torna-se correlacionada com as LLR's fornecidas pelo decodificador. Do mesmo modo, a saída $L^D(c_n)$ do bloco decodificador apresenta certa correlação com as LLR's de saída obtidas anteriormente pelo bloco equalizador. Para evitar esta redundância é que se utiliza apenas a informação extrínseca como estimativa da entrada *a priori* do bloco seguinte.

Para que a correlação remanescente entre os valores de informação extrínseca operados pelos blocos equalizador e decodificador seja minimizada, um adequado projeto do entrelaçador temporal utilizado faz-se necessário, pois este elemento tem como papel principal descorrelacionar tais valores, além de minimizar os efeitos dos erros em rajada causados por canais com memória. Desta maneira, quanto maior é a capacidade do entrelaçador de tornar aleatórios os valores por ele operados, maior é o ganho que se tem em termos de melhoria da taxa de erro

de bit a cada iteração do equalizador turbo [4].

Assim como qualquer processo iterativo, o processo de equalização turbo também pode ser representado como uma linha de módulos (do Inglês, *pipe-line*), como ilustrado na Figura 8, onde cada módulo representa uma iteração do equalizador. Cada módulo possui duas entradas: o vetor de amostras recebidas do canal, \mathbf{r} , e a informação *a priori* do bloco equalizador, $L_a^E(c_n)$. Na saída de um módulo tem-se a informação extrínseca do bloco decodificador, que é utilizada como estimativa da informação *a priori* para o próximo módulo (próxima iteração). A saída de cada módulo possui maior confiabilidade em relação à saída do módulo anterior [9] [10]. Ao final das iterações, a partir da saída suave do último módulo é realizada uma decisão abrupta em relação aos bits de informação b_k ou em relação aos símbolos c_n . A representação em módulos foi inserida neste texto com o objetivo de facilitar o entendimento do processo de equalização turbo.

Neste ponto merecem ser comentadas algumas questões relacionadas ao tempo que o equalizador turbo precisa para obter convergência. Duas medidas de tempo podem ser aqui consideradas: a *latência* e o *atraso*. O atraso é inversamente proporcional à taxa de transmissão e diretamente proporcional ao tamanho do bloco codificado, pois de forma independente do tipo de decodificação, há que se esperar pela recepção de todos os símbolos associados a uma palavra-código para que a estimação dessa palavra-código se inicie¹. Já a latência está associada ao tempo de processamento necessário para que, depois recebida uma palavra-código completa, o primeiro bit estimado válido de informação esteja disponível na saída do equalizador turbo. Este tempo depende do número de iterações (número de módulos) no processo de equalização turbo e também da complexidade dos algoritmos utilizados nos blocos componentes (bloco de equalização e bloco de decodificação). Nesse sentido, um dos grandes desafios apresentados ao desenvolvimento de novos processos de equalização turbo está relacionado com o objetivo de redução da latência, assim como ocorre com os códigos turbo.

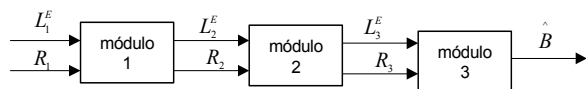


Figura 8 - Equalizador turbo composto por módulos.

Os primeiros sistemas de equalização turbo tratados em [2], [11] e [13] são constituídos de blocos equalizador e decodificador que utilizam o algoritmo ótimo de decodificação MAP, chamado BCJR [14] e que possui uma complexidade computacional exponencialmente crescente com o número de estados das treliças que representam o canal e o codificador. Outras alternativas que simplificam o algoritmo BCJR, como o algoritmo *Log-MAP* que opera no domínio logarítmico, sem perda de desempenho em relação ao BCJR, e o algoritmo sub-ótimo *Max-Log-MAP*

¹ Esta afirmação é válida para códigos de bloco e para outros esquemas de codificação de canal que operem com blocos de símbolos codificados, por exemplo, quando se utiliza um código convolucional com terminação na sua treliça de forma que ao final da seqüência de informação de comprimento definido o estado do codificador retorne ao estado nulo.

[12]. Ainda no intuito de simplificar o algoritmo BCJR, podem-se utilizar técnicas de redução de estados da treliça que representa o canal [15]. Em [5] é utilizada uma alternativa sub-ótima através de um *Cancelador de Interferências* no bloco de equalização. Esta alternativa apresenta significativa redução de complexidade de implementação em relação ao algoritmo BCJR. Outra alternativa para o bloco equalizador é apresentada em [16], onde é usado no equalizador um filtro que procura minimizar o erro médio quadrático (*MMSE - Minimum Mean Square Error*), entre o sinal transmitido e o estimado.

Nas próximas seções, optou-se por utilizar apenas o algoritmo BCJR e o Cancelador de Interferências como componentes do processo de equalização turbo. Formas alternativas podem ser consultadas em [12], [16] e [15], apenas para citar alguns exemplos.

IV. ALGORITMO BCJR

O algoritmo BCJR, que implementa o critério de decisão MAP símbolo-a-símbolo, foi proposto em 1974 por Bahl, Cocke, Jelinek e Raviv [14]. Para o entendimento do algoritmo tomam-se os símbolos² b_k que passarão por um sistema que é modelado por uma treliça como a da Figura 9. Esta treliça pode representar um modelo de Markov, como um canal com múltiplos percursos ou um codificador convolucional. O algoritmo BCJR tem como objetivo calcular a LLR

$$L(b_k) \equiv \ln \left(\frac{P(b_k = +1 | \mathbf{r})}{P(b_k = -1 | \mathbf{r})} \right) \quad (7)$$

Aplicando o teorema de Bayes $P(a, b) = P(a|b)P(b)$ em (7), obtém-se:

$$L(b_k | \mathbf{r}) = \ln \left(\frac{p(b_k = +1, \mathbf{r})}{p(b_k = -1, \mathbf{r})} \right) \quad (8)$$

Por meio da Figura 9 pode-se fazer uma interpretação sobre os valores operados na expressão (8). A probabilidade de b_k ser +1 é a probabilidade de ocorrer as transições representadas pelas linhas cheias na Figura 9, enquanto a probabilidade de b_k ser -1 é a probabilidade de ocorrer as transições representadas pelas linhas tracejadas.

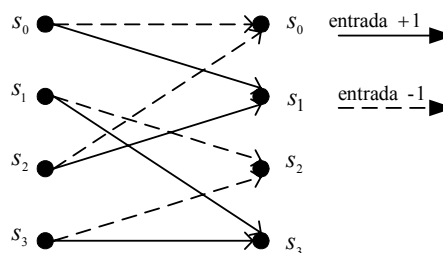


Figura 9 - Modelo de Markov representado por uma treliça.

Como o conjunto das probabilidades de transições é mutuamente exclusivo, a equação (8) pode ser expressa da seguinte maneira [17]:

² Deve-se atentar para a re-utilização da notação b_k , agora de forma mais abrangente, podendo estar associada a bits de informação ou a símbolos ao longo do sistema de transmissão e de recepção, dependendo de onde e como se aplica o algoritmo BCJR.

$$L(b_k) = \ln \frac{\left(\sum_{b_k=+1} p(S_{k-1}, S_k, \mathbf{r}) \right)}{\left(\sum_{b_k=-1} p(S_{k-1}, S_k, \mathbf{r}) \right)} \quad (9)$$

onde S_k refere-se ao estado que a treliça se encontra no instante de tempo presente e S_{k-1} se refere ao estado que a treliça se encontrava no instante de tempo anterior. Note que o numerador na parte da direita de (9) representa as probabilidades das transições causadas pelo símbolo +1 (linhas contínuas da Figura 9), enquanto que o denominador se refere aos símbolos -1.

Ainda referindo-se à expressão (9) pode-se mostrar que [17]:

$$p(S_{k-1}, S_k, R) = p(r_{j>k} | s) p(\{r_k, s\} | s') p(s', r_{j<k}) \quad (10)$$

que, numa notação alternativa, pode ser re-escrita como:

$$p(S_{k-1}, S_k, R) = \beta_k(S_k) \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \quad (11)$$

onde s se refere ao estado que a treliça se encontra no instante de tempo presente e s' ao instante de tempo anterior. O termo α_{k-1} em (11) se refere à função densidade da probabilidade que a treliça estava no estado anterior s' e recebeu uma sequência passada $r_{j<k}$. O valor de α pode ser calculado através da equação [17]:

$$\alpha_k(S_k) = \sum_{S_{k-1}} \gamma_k(S_{k-1}, S_k) \alpha_{k-1}(S_{k-1}) \quad (12)$$

O termo β_k em (11) se refere à função densidade da probabilidade de receber uma sequência futura $r_{j>k}$, dado o estado atual s da treliça. O valor de β_k pode ser calculado através de [17]:

$$\beta_k(S_k) = \sum_{S_{k+1}} \gamma_k(S_{k-1}, S_k) \beta_{k+1}(S_{k+1}) \quad (13)$$

A Figura 10 exemplifica o cálculo de α e β , onde se deseja calcular os valores de α_k e β_k do estado 0. A figura apresenta as possíveis transições que terminam e partem do estado 0 e os respectivos cálculos de α_k e β_k a partir das equações (12) e (13). Os valores de α são obtidos partindo do início para o fim da treliça de acordo com (12). Os valores de β são obtidos partindo do fim para o início da treliça através de (13).

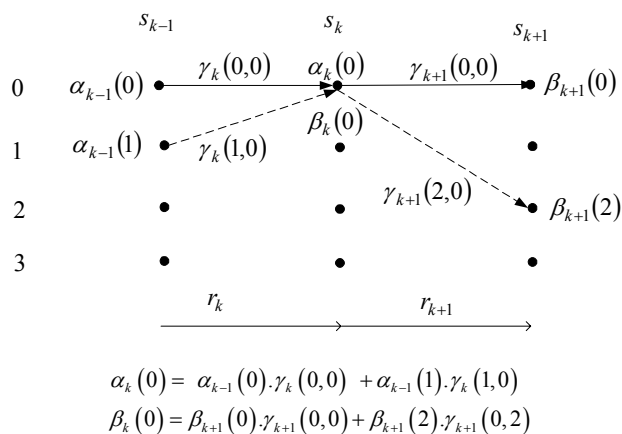


Figura 10 - Cálculo de α e β [17].

Caso seja forçado, através de transmissão de zeros, que a treliça inicie e termine no estado S_0 , α e β podem ser inicializados como as expressões [12]:

$$\begin{aligned} \alpha_{inicial}(0) &= 1 \text{ e } \alpha_{inicial}(s) = 0 \quad \forall s \neq 0 \\ \beta_{final}(0) &= 1 \text{ e } \beta_{final}(s) = 0 \quad \forall s \neq 0 \end{aligned} \quad (14)$$

Quando o estado inicial e o final não forem conhecidos, $\alpha_{inicial}$ e β_{final} são inicializados com 1 para todos os estados [4]. De posse dos valores de α e β , resta calcular os valores de γ para que se obtenha o valor estimado por (11), para todas as transições possíveis, e resta também determinar $L(b_k)$ através de (9).

Um receptor ideal seria aquele que utilizasse o algoritmo BCJR em uma supertreliça que englobaria o codificador, o entrelaçador e o canal [4]. No entanto, a modelagem dessa treliça seria extremamente complexo, assim como o uso do algoritmo BCJR. O equalizador turbo trata de maneira separada a treliça do canal e a treliça do decodificador. Portanto, o algoritmo BCJR pode ser aplicado ao equalizador e ao decodificador. Para ambos os casos, o cálculo de α e β é executado da mesma maneira. O que irá diferenciar o emprego do algoritmo BCJR nos blocos equalizador e decodificador é a maneira de calcular o valor de γ . Estas observações são mais bem exploradas adiante.

IV.1 ALGORITMO BCJR APLICADO AO BLOCO EQUALIZADOR

Nesta seção é apresentado o algoritmo BCJR aplicado ao bloco equalizador. Os valores de α e β são calculados a partir das equações (12) e (13), respectivamente. Para os cálculos envolvendo (12) e (13) é necessário calcular o valor de γ . De (10) e (11) tem-se:

$$\gamma_n(s', s) = p(\{r_n, S_n = s\} | S_{n-1} = s') \quad (15)$$

Aplicando a regra de Bayes modificada $P[(a,b)|c] = P[a|(b,c)]P(b|c)$ em (15), tem-se:

$$\gamma_n(s', s) = p(r_n | \{s, s'\}) P(s | s') \quad (16)$$

O primeiro termo do lado direito de (16), $p(r_n | \{s, s'\})$, é a função densidade de probabilidade de recepção de uma amostra r_n do canal, dado que ocorreu uma transição $\{s, s'\}$. A partir do modelo discreto do canal representado por (1), observa-se que esta função densidade de probabilidade é dada por uma distribuição Gaussiana com média v_n , que é o sinal com IIS sem adição de ruído, e com variância σ^2 devido à adição do ruído AWGN w_n . Portanto, este termo é dado por:

$$p(r_n | v_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (17)$$

O segundo termo do lado direito de (16), $P(s | s')$, é a probabilidade da treliça se encontrar no estado s , dado que ela estava no estado s' . Esta probabilidade é governada pela probabilidade *a priori* do símbolo de entrada c_n^π , já que é este símbolo que irá definir qual transição ocorrerá na treliça do canal. A partir desses resultados, a expressão (16) pode ser re-escrita como:

$$\gamma(s', s) = P(c_n^\pi) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (18)$$

onde $P(c_n^\pi)$ é a probabilidade *a priori* dos símbolos c_n^π , estimada através da informação extrínseca do decodificador, $L_{ext}^D(c_n^\pi)$. Como a informação extrínseca $L_{ext}^D(c_n^\pi)$ é usada como estimativa da informação *a priori* $P(c_n^\pi)$, ela pode ser escrita como [2]:

$$L_{ext}^D(c_n^\pi) \approx \ln \frac{P(c_n^\pi = +1)}{P(c_n^\pi = -1)} \quad (19)$$

A partir da equação (19) obtém-se a expressão da probabilidade *a priori* $P(c_n^\pi)$ em função de $L_{ext}^D(c_n^\pi)$:

$$P(c_n^\pi = c) \cong \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + \exp(L_{ext}^D(c_n^\pi))} \quad (20)$$

onde $c \in \{0, 1\}$ para o cálculo de $P(c_n^\pi = +1)$ ou $P(c_n^\pi = -1)$, respectivamente. Aplicando (20) em (18) obtém-se finalmente o valor de γ para o bloco equalizador:

$$\gamma(s', s) = \frac{\exp(c L_{ext}^D(c_n^\pi))}{1 + \exp(L_{ext}^D(c_n^\pi))} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_n - v_n)^2}{2\sigma^2}\right) \quad (21)$$

A partir de (21) pode-se notar o ganho de informação no processo iterativo do equalizador turbo. Na primeira iteração não há informação extrínseca $L_{ext}^D(c_n^\pi)$ do decodificador realimentada para o equalizador. Sendo assim, os símbolos c_n^π são admitidos equiprováveis e $P(c_n^\pi)$ é considerada igual a $1/2$. Nas próximas iterações o valor de $L_{ext}^D(c_n^\pi)$ é usado para o cálculo de γ conforme (21). A cada iteração há um aumento na confiabilidade da informação extrínseca do decodificador, $L_{ext}^D(c_n^\pi)$, e, conseqüentemente, o valor de γ é aprimorado.

A inicialização do algoritmo BCJR admite que os estados inicial e final da treliça sejam desconhecidos. Desta maneira, os valores de α e β são inicializados com o valor 1. Em uma implementação prática, os valores de α e β convergem rapidamente para zero, o que pode gerar algumas instabilidades numéricas no algoritmo. Em [18] é usada uma normalização de α e β que elimina essas instabilidades.

Para se empregar o algoritmo BCJR é necessário que os coeficientes g_i do modelo discreto do canal sejam conhecidos. Em [19] é apresentado um algoritmo MAP que, além de equalizar o canal, também estima os coeficientes g_i . Como exemplo de aplicação do algoritmo BCJR no bloco equalizador, a Figura 11 apresenta a treliça de um canal com os coeficientes [0,407 ; 0,815 ; 0,407]. Os valores associados às transições dessa treliça são os valores de v_n utilizados na equação (21).

IV.2 ALGORITMO BCJR APLICADO AO BLOCO DECODIFICADOR

Nesta secção é apresentado o algoritmo BCJR aplicado ao bloco de decodificação. Os valores de α e β são calculados a partir das equações (12) e (13), da mesma

maneira que para o bloco equalizador. Para o cálculo de γ utiliza-se agora a expressão:

$$\gamma_n(s', s) = p(z_n | \{s, s'\}) \cdot P(s | s') \quad (22)$$

onde z_n são os elementos do vetor \mathbf{Z} definido em (5). O primeiro termo do lado direito de (22) é governado pelos símbolos de saída c_n do codificador convolucional. O segundo termo é governado pelos bits de informação b_k na entrada do codificador convolucional.

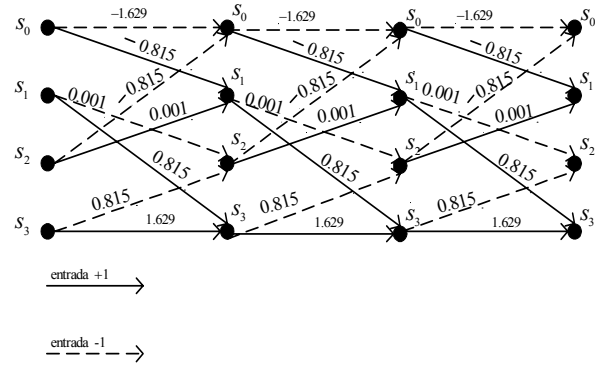


Figura 11 - Treliça para canal com os coeficientes $g_1=0,407$, $g_2=0,815$ e $g_3=0,407$. As linhas tracejadas representam transições causadas por entradas -1 no canal e as linhas contínuas por entradas $+1$. De [4].

Como apresentado na Seção III, o bloco decodificador tem como entrada o vetor \mathbf{Z} constituído das probabilidades referentes aos símbolos c_n . Portanto, o valor de γ para o bloco decodificador é dado pela expressão:

$$\gamma(s_i, s_j) = P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r}) P(b_k) \quad (23)$$

onde c_1 e c_2 são, respectivamente, o primeiro e o segundo símbolo de saída para as transições da treliça do codificador convolucional, como na Figura 2. O termo $P(b_k)$ é a probabilidade *a priori* dos bits de informação. De início esta informação não é disponível e, portanto, os bits são considerados identicamente distribuídos e o valor de $P(b_k)$ é igual a $1/2$. As probabilidades $P(c_1 | \mathbf{r}) P(c_2 | \mathbf{r})$ de (23) são retiradas diretamente do vetor \mathbf{Z} , este obtido a partir da informação extrínseca do bloco equalizador, $L_{ext}^E(c_n)$:

$$P(c_n | \mathbf{r}) \cong \frac{\exp(c L_{ext}^E(c_n))}{1 + \exp(L_{ext}^E(c_n))} \quad (24)$$

onde $c \in \{0, 1\}$ para os símbolos 0 e 1, respectivamente.

O algoritmo a seguir apresenta um resumo da equalização turbo empregando o algoritmo MAP BCJR. São usadas as seguintes convenções: $\pi^{-1}(x)$ significa que x passou pelo desentrelaçador temporal, $\pi(x)$ significa que x passou pelo entrelaçador temporal e $MAP(x)$, significa que x foi processado pelo algoritmo MAP no bloco equalizador ou no bloco decodificador:

Equalização turbo com uso do algoritmo BCJR

Entrada: vetor de amostras na saída do canal \mathbf{r} e o número de iterações p .

Inicialização: $L_a^E(c_n^\pi) = 0$

Para cada iteração p faça:

$$L^E(c_n^\pi) = \text{MAP}(L_a^E(c_n^\pi), \mathbf{r})$$

$$L_{\text{ext}}^E(c_n^\pi) = L^E(c_n^\pi) - L_a^E(c_n^\pi)$$

$$L_{\text{ext}}^E(c_n) = \pi^{-1}(L_{\text{ext}}^E(c_n^\pi))$$

$$L^D(c_n) = \text{MAP}(L_{\text{ext}}^E(c_n))$$

$$L_{\text{ext}}^D(c_n) = L^D(c_n) - L_{\text{ext}}^E(c_n)$$

$$L_{\text{ext}}^D(c_n^\pi) = \pi(L_{\text{ext}}^D(c_n))$$

$$L_a^E(c_n^\pi) = L_{\text{ext}}^D(c_n^\pi)$$

fim das iterações

cálculo de $L(b_k)$

Saída: b_k = decisão abrupta sobre $L(b_k)$

V. EQUALIZADOR TURBO COM CANCELADOR DE INTERFERÊNCIA

A abordagem usada até aqui neste tutorial utiliza o algoritmo BCJR no bloco equalizador e no bloco decodificador. Existem alguns inconvenientes neste tipo de abordagem. Para o emprego do algoritmo BCJR no bloco equalizador é necessário que haja conhecimento total do canal para que o mesmo seja modelado como uma treliça. Além disso, para canais com grande dispersão temporal e modulações de alta ordem, o número de estados que modelam a treliça do bloco equalizador pode tornar o uso de um equalizador BCJR proibitivo computacionalmente.

Em [5] é proposto um novo tipo de equalização turbo, onde o bloco equalizador de treliça é trocado por um *Cancelador de interferência* [20] [21]. O bloco equalizador deste sistema é apresentado na Figura 12. O filtro F1 é um filtro casado e, portanto, ele maximiza a relação sinal-ruído em sua saída, com pena do aumento de IIS. O filtro F2 é um filtro usado para estimar a IIS presente no canal e na saída do filtro F1. Tendo estimado a IIS, ela é subtraída da saída do filtro casado, eliminando-a sem diminuição da relação sinal-ruído [5] [6].

O *Cancelador de Interferência* da Figura 13 tem duas entradas: a entrada do filtro F1, que é o vetor de amostras recebidas do canal \mathbf{r} , e a média dos símbolos c_n^π como entrada do filtro F2. A média do símbolo c_n^π é estimada através da informação extrínseca $L_{\text{ext}}^D(c_n^\pi)$ do decodificador da seguinte maneira:

$$\begin{aligned} \overline{c_n^\pi} &= E\{c_n^\pi\} = P(c_n^\pi = +1)1 + P(c_n^\pi = -1)-1 \\ &= \frac{\exp(L_{\text{ext}}^D(c_n^\pi))}{1 + \exp(L_{\text{ext}}^D(c_n^\pi))} + \frac{-1}{1 + \exp(L_{\text{ext}}^D(c_n^\pi))} \quad (25) \\ &= \tanh\left(\frac{L_{\text{ext}}^D(c_n^\pi)}{2}\right) \end{aligned}$$

Na primeira iteração, como não há informação extrínseca do decodificador, $L_{\text{ext}}^D(c_n^\pi)$, disponível, o equalizador usado pode ser um equalizador convencional como um filtro empregando o algoritmo LMS (*Least Mean Square*).

Nas próximas iterações o *Cancelador de Interferência* utiliza os valores de $L_{\text{ext}}^D(c_n^\pi)$ para eliminar a IIS. A cada iteração o valor de $L_{\text{ext}}^D(c_n^\pi)$ se torna mais confiável e, conseqüentemente, o desempenho do *Cancelador de Interferência* se aproxima de seu desempenho ótimo. Este desempenho ótimo é alcançado quando os símbolos c_n^π são conhecidos. Nesta situação, toda IIS é eliminada sem diminuição da relação sinal-ruído [5].

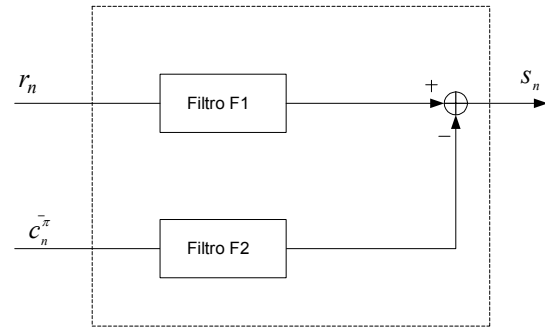


Figura 12 - Bloco equalizador utilizando cancelador de interferência.

Os coeficientes dos filtros F1 e F2 apresentados em [5] são calculados a partir do critério de minimização do erro médio quadrático (do Inglês, *Mean Square Error*, MSE) entre s_n e c_n^π :

$$MSE = E\{|s_n - c_n^\pi|^2\} \quad (26)$$

Em [5] e [6] os coeficientes dos filtros F1 e F2 são calculados de maneira adaptativa através do algoritmo LMS (*Least Mean Square*), onde no início da transmissão é usada uma sequência de treinamento e em seguida o equalizador entra no modo de decisão direta [8]. Em [6] também é apresentado um algoritmo de adaptação dos coeficientes $g(n)$, baseado no algoritmo RLS (*Recursive Least Squares*), para canais variantes no tempo.

VI. CRITÉRIOS DE PARADA

O número de iterações realizadas para que se obtenha um desempenho desejado na equalização turbo não é sempre o mesmo para todos os blocos processados. Para alguns blocos, apenas algumas iterações são suficientes para que o canal esteja praticamente livre de erros. Os chamados critérios de parada apresentados em [11] determinam quando o número de iterações em um bloco já é suficiente. Desta forma, o uso de critérios de parada otimiza o processo de equalização turbo diminuindo a latência do sistema, aumentando a vazão de dados transmitidos. Nesta seção são apresentados resumidamente dois dos critérios de parada abordados em [11]: o da *observação de decisões* abruptas e o da *função risco*.

VI.1. OBSERVANDO DECISÕES ABRUPTAS

Este critério consiste em observar, a cada iteração, a saída abrupta dos bits de informação do decodificador. Quando a saída não se alterar por duas iterações consecutivas, o equalizador pára com o processo iterativo. Este critério, apesar de alcançar bons resultados, torna necessário que se faça uma iteração a mais no bloco processado para que se observe que a saída não foi alterada.

VI.2. FUNÇÃO RISCO

Partindo do princípio que as LLR's medem a confiabilidade das decisões, este critério se baseia em observar o valor das LLR's dos bits de informação em cada iteração. Quando as LLR's estiverem com valores altos, o risco da decisão tomada se torna pequeno e o processo iterativo pode ser encerrado.

Calculando as probabilidades $P(b_k | \mathbf{r})$ a partir das LLR's dos bits b_k , a função risco é dada por:

$$R = \frac{1}{L} \left(\sum_{\substack{k=1 \\ L(u_k > 0)}}^L P(b_k = -1 | \mathbf{r}) + \sum_{\substack{k=1 \\ L(u_k < 0)}}^L P(b_k = +1 | \mathbf{r}) \right) \quad (27)$$

onde L é o tamanho do bloco de informação. A função R deve estar abaixo de um determinado limiar para se obter um taxa de erro aceitável. Em [12] é dado um valor prático de 5×10^{-5} para este limiar.

VII. COMENTÁRIOS FINAIS

Neste tutorial foi apresentado que a equalização turbo realiza a equalização e a decodificação de canal de maneira iterativa. O equalizador turbo é composto de um bloco equalizador e um bloco decodificador, ambos com entradas e saídas suaves. As saídas suaves do bloco equalizador e do bloco decodificador são dadas na forma de LLR's e possuem a informação da confiabilidade das decisões estimadas pelos respectivos blocos. Para evitar uma redundância na realimentação de informação durante as iterações, deve-se subtrair as LLR's da saída de um bloco do equalizador turbo (equalizador ou decodificador) da LLR da entrada. Desta maneira, somente a informação extrínseca de um bloco será usada como informação *a priori* do bloco seguinte.

A Figura 13 apresenta os resultados obtidos por simulação computacional de um equalizador turbo operando em um canal com coeficientes $g_{-1} = 0,407$, $g_0 = 0,815$ e $g_1 = 0,407$.

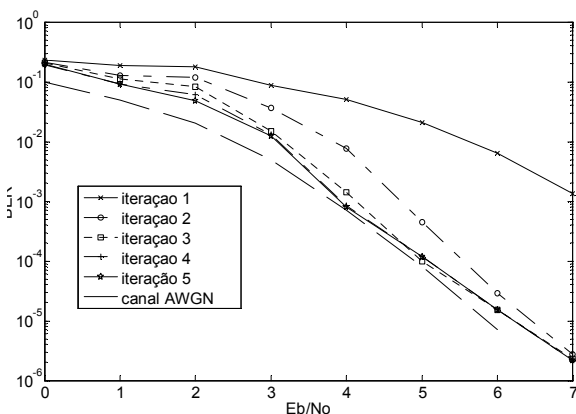


Figura 13 - Desempenho ($BER \times E_b/N_0$) de um equalizador turbo em um canal com coeficientes $g_{-1} = 0,407$, $g_0 = 0,815$ e $g_1 = 0,407$.

A Figura 14 apresenta os resultados obtidos por simulação computacional de um equalizador turbo operando em um canal com coeficientes $g_{-1} = 0,671$; $g_0 = 0,5$; $g_1 = 0,387$; $g_2 = 0,316$ e $g_3 = 0,224$. Nas simulações apresentadas aqui foi simulada a transmissão de blocos de 512 bits

que foram codificados por um codificador convolucional de taxa $1/2$, como o da Figura 3. Os 1024 símbolos na saída do codificador foram embaralhados por um entrelaçador *S-random*, com $S = 16$, e transmitidos através do canal com IIS. Na recepção foi utilizado um equalizador turbo com cinco iterações empregando o algoritmo BCJR no bloco equalizador e no bloco decodificador.

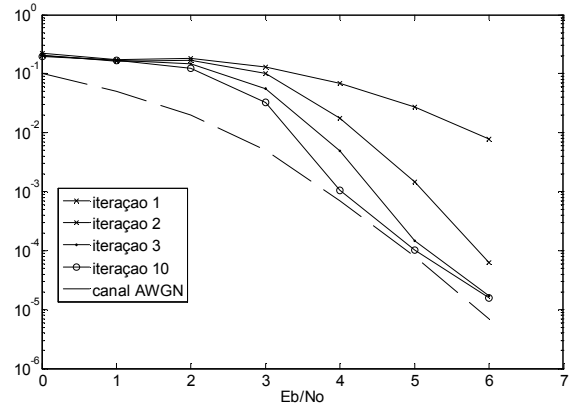


Figura 14 - Desempenho ($BER \times E_b/N_0$) de um equalizador turbo em um canal com coeficientes $g_{-1} = 0,671$; $g_0 = 0,5$; $g_1 = 0,387$; $g_2 = 0,316$ e $g_3 = 0,224$.

Por meio das figuras 13 e 14 nota-se que existe um ganho considerável da primeira iteração para a segunda iteração do processo de equalização turbo, enquanto não há decréscimos significativos de taxa de erro de bit (do Inglês, *Bit Error Rate*, BER) nas próximas iterações. Nota-se também que o desempenho do equalizador turbo possui um limitante dado por um sistema codificado em um canal com ruído AWGN, sem a presença de IIS. Este limitante vem a confirmar que o equalizador turbo pode proporcionar aproximadamente o mesmo desempenho daquele obtido em um canal AWGN com codificação, sem a presença de IIS.

Outros sistemas de equalização turbo vêm sendo propostos. Além das modificações realizadas no bloco equalizador apresentadas na Seção III, outros blocos decodificadores têm sido empregados. Em [22] é apresentado um sistema que realiza de maneira combinada a equalização e a decodificação turbo utilizando códigos turbo convolucionais [3]. Em [23] é apresentado um sistema que também realiza, conjuntamente, a equalização e a decodificação turbo utilizando códigos turbo de bloco. Resultados obtidos por equalizadores turbo utilizando os códigos convolucional, turbo convolucional e turbo de bloco podem ser encontrados em [24]. Em [25] é apresentado um novo esquema de equalização turbo, onde é realizada uma decodificação turbo parcial de códigos produto de paridade simples multidimensionais no bloco decodificador. A abordagem apresentada em [25] possui baixa complexidade e supera o desempenho das abordagens utilizadas em [24] quando se utilizam altas taxas de codificação.

VIII. REFERÊNCIAS

[1] J. G. Proakis, "Digital Communications," third edition. New York: McGraw-Hill, 1995.
 [2] C. Douillard, M. Jezequel, C. Berrou, A Picart, P. Didier, and A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo Equalization," in

- European trans. on telecomm., vol. 6, pp. 507–511, Sep-Oct 1995.
- [3] C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correcting and Decoding: Turbo-Codes,” presented at the international conference on communication (ICC), pp. 1064-1070, IEEE, May 1993.
- [4] Ralf Koetter, A. C. Singer and Michael Tüchler, “Turbo Equalization,” IEEE Signal Processing Magazine, invited paper, 2003.
- [5] A. Glavieux, C. Laot, and J. Labat, “Turbo Equalization Over a Frequency Selective Channel,” in Proc. Of the Intern. Symposium on Turbo codes, Brest, France, pp. 96–102, September 1997.
- [6] C. Laot, A. Glavieux and J. Labat, “Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized,” IEE Journal on Selected Areas in Communications, vol.19, n°9, September, 2001
- [7] M. Tüchler, R. Koetter, and A. Singer, “Turbo Equalization: Principles and New Results,” IEEE Trans. on Comm., vol. 50, pp. 754–767, May 2002.
- [8] S. Haykin, “Communication Systems,” 4th edition: John Wiley and Sons, Inc, New York, 2001.
- [9] S. T. Brink, “Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes,” IEEE Transactions on communications, vol.49, n°10, October 2001.
- [10] R. Otnes, T. Micheal, “EXIT Chart Analysis Applied To Adaptive Turbo Equalization,” presented at the Nordic Signal Processing Symposium, NORSIG, 2002, Hurlgruten Tromso-Trondheim, Norway, October 4-7 2002.
- [11] G. Bauch, H. Khorram and J. Hagenauer, “Iterative Equalization and Decoding in Mobile Communication Systems,” presented at the 2nd EPMCC’97 and 3rd ITG-Fachtagung Telecomm, Mobile Communication, Bonn, Germany, Oct. 1997.
- [12] G. Bauch and V. Franz, “A Comparison of Soft-in/Soft-out Algorithms for Turbo detection,” presented at the intern. conf. on telecomm., pp. 259–263, June 1998.
- [13] Anastasopoulos and K. Chugg, “Iterative Equalization/Decoding for TCM for Frequency-Selective Fading Channels,” Conf. Record 31th Asilomar Conf. On Signals, Systems & Comp., vol. 1, pp. 177–181, Nov 1997.
- [14] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” IEEE Transactions on Information Theory, vol. IT-20, pp. 284-287, March, 1974.
- [15] M. D. Cosac, “Técnicas de Redução de Estados Aliadas à Equalização Turbo,” Dissertação de Mestrado: Universidade Estadual de Campinas - UNICAMP, Campinas, SP, 2004.
- [16] M. Tüchler, R. Koetter, and A. Singer, “Iterative Correction of ISI via Equalization and Decoding using Priors,” Proc. Intern. Symp. on Information Theory, Sorrento, Italy, pp. 194, July 2000.
- [17] L. Hanzo, T. H. Liew, and B. L. Yeap, “Turbo Coding, Turbo equalization and Space-Time Coding for Transmission Over Fading Channels,” John Wiley and Sons, Inc 2002.
- [18] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proceedings of the IEE, vol.77, n°2, February 1989.
- [19] M. D. Linda, B. C. Iain, and P. Hoeher, “Joint Map Equalization and Channel Estimation for Frequency-Selective and Frequency-Flat Fast Fading Channels,” IEEE Transactions on Communications, vol. 49, n°12, December 2001.
- [20] A. Gersho and T. Lim, “Adaptive cancellation of intersymbol interference for data transmission,” Bell Syst. Tech. J., vol. 60, n° 11, pp.1997-2021, Nov.1981.
- [21] M. S. Muller and J. Salz, “A unified theory of data-aided equalization,” Bell Syst. Tech. J., vol. 60, n° 9, pp.2023-2038, Nov.1981.
- [22] D. Raphaeli, Y. Zarai, “Combined Turbo Equalization And Turbo Decoding,” Proc. Global Telecommunications Conf., Phoenix , AZ, Nov. 3-8, pp. 639-641, 1997.
- [23] M. Noorbakhsh and K. Mohamed-Pour, “Combined Turbo Equalization and Block Turbo Decoding,” IEE proceedings. Communications, vol. 150, n°3, pp.149-152, 2003.
- [24] B. L. Yeap, T. H. Liew, J. Hamorsky, and L. Hanzo, “Comparative Study of Turbo Equalization Schemes Using Convolutional, Convolutional Turbo, and Block-Turbo Codes,” IEE Transactions on Wireless Communications, vol. 1, N°2, April 2002.
- [25] A. F. Santos, “Um Esquema de Equalização Turbo Aplicando Decodificação Turbo de Códigos Produto de Paridade Simples Multidimensionais,” Dissertação de Mestrado: Instituto Nacional de Telecomunicações-INATEL, Santa Rita do Sapucaí, MG, Março de 2005.

SOBRE OS AUTORES

André Fonseca dos Santos nasceu em Pouso Alegre, em 05 de abril de 1980. Possui os títulos: *Engenheiro Eletricista* (Inatel, 2003) e *Mestre em Telecomunicações* (Inatel, 2005). Em 2003 estagiou na área de calibração de equipamentos de telecomunicações no CPqD. *Telecom & IT Solutions*. Desde janeiro de 2005 faz parte do corpo de engenheiros do INATEL, onde trabalha em um projeto financiado pela FAPEMIG que visa o desenvolvimento de uma Unidade Receptora e Conversora (URC) para sinais de televisão digital. Em 2003 ministrou aulas no INATEL na área de controle e automação e atualmente é professor na área de comunicação digital. Suas pesquisas incluem aspectos sobre televisão digital, sistemas OFDM, sincronismo, equalização turbo, decodificação turbo, especificamente equalização turbo conjunta com decodificação turbo.

Dayan Adionel Guimarães nasceu em Carrancas, MG, em 01 de março de 1969. Possui os títulos: *Técnico em Eletrônica* (ETE “FMC”, 1987), *Engenheiro Eletricista* (Inatel, 1994), *Especialista em Engenharia de Comunicação de Dados* (Inatel, 2003), *Especialista em Administração* com ênfase em Gerência de RH (FAI, 1996), *Mestre em Engenharia Elétrica* (Unicamp, 1998) e *Doutor em Engenharia Elétrica* (Unicamp, 2003). De 1988 a 1993 desenvolveu sensores e equipamentos para instrumentação industrial e controle e também foi Supervisor de Produção e Supervisor de Engenharia de Produtos na SENSE Sensores e Instrumentos. Desde de janeiro de 1995 é Professor do Inatel onde, por oito anos, foi responsável pela estrutura que dá apoio às atividades de ensino prático nas áreas de Telecomunicações, Eletrônica e Eletrotécnica. É membro do corpo editorial da revista *Telecomunicações* do Inatel. Suas pesquisas incluem aspectos gerais sobre transmissão digital e sistemas de comunicação móvel, especificamente sistemas CDMA Multiplicadora e esquemas de codificação para canais com desvanecimento, especificamente códigos turbo de bloco.